ECE 380: Control Systems

Course Notes: Winter 2014

Prof. Shreyas Sundaram

Department of Electrical and Computer Engineering
University of Waterloo

Acknowledgments

Parts of these course notes are loosely based on lecture notes by Professors Daniel Liberzon, Sean Meyn, and Mark Spong (University of Illinois), on notes by Professors Daniel Davison and Daniel Miller (University of Waterloo), and on parts of the textbook *Feedback Control of Dynamic Systems* (5th edition) by Franklin, Powell and Emami-Naeini. I claim credit for all typos and mistakes in the notes.

The LaTeX template for *The Not So Short Introduction to LaTeX 2$_\varepsilon$* by T. Oetiker et al. was used to typeset portions of these notes.

Shreyas Sundaram
University of Waterloo

Contents

Chapter 1

Introduction

1.1 Dynamical Systems

For the purposes of this course, a *system* is an abstract object that accepts *inputs* and produces *outputs* in response. Systems are often composed of smaller components that are interconnected together, leading to behavior that is more than just the sum of its parts. In the control literature, systems are also commonly referred to as *plants* or *processes.*

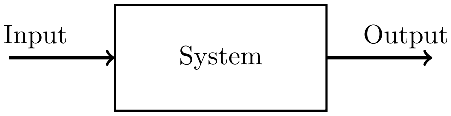


Figure 1.1: An abstract representation of a system.

The term *dynamical system* loosely refers to any system that has an internal state and some dynamics (i.e., a rule specifying how the state evolves in time). This description applies to a very large class of systems, from automobiles and aviation to industrial manufacturing plants and the electrical power grid. The presence of dynamics implies that the behavior of the system cannot be entirely arbitrary; the temporal behavior of the system's state and outputs can be predicted to some extent by an appropriate *model* of the system.

Example 1. Consider a simple model of a car in motion. Let the speed of the car at any time t be given by $v(t)$. One of the inputs to the system is the acceleration $a(t)$, applied by the throttle. From basic physics, the evolution of the speed is given by

$$\frac{dv}{dt} = a(t). \tag{1.1}$$

The quantity $v(t)$ is the state of the system, and equation (1.1) specifies the dynamics. There is a speedometer on the car, which is a sensor that measures the speed. The value provided by the sensor is denoted by $s(t) = v(t)$, and this is taken to be the output of the system. □

As shown by the above example, the inputs to physical systems are applied via *actuators*, and the outputs are measurements of the system state provided by *sensors*.

Other examples of systems: Electronic circuits, DC Motor, Economic Systems, ...

1.2 What is Control Theory?

The field of *control systems* deals with applying or choosing the inputs to a given system to make it behave in a certain way (i.e., make the state or output of the system follow a certain trajectory). A key way to achieve this is via the use of *feedback*, where the input depends on the output in some way. This is also called *closed loop control*.

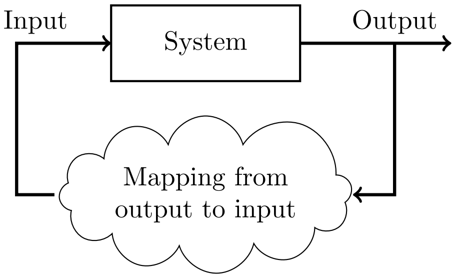


Figure 1.2: Feedback Control.

Typically, the mapping from outputs to inputs in the feedback loop is performed via a computational element known as a *controller*, which processes the sensor measurements and converts it to an appropriate actuator signal. The basic architecture is shown below. Note that the feedback loop typically contains *disturbances* that we cannot control.

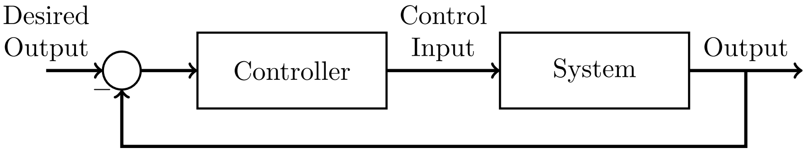


Figure 1.3: Block Diagram of a feedback control system.

Example 2 (Cruise Control)**.** Consider again the simple model of a car from Example 1. A *cruise control system* for the car would work as follows.

- The speedometer in the car measures the current speed and produces $s(t) = v(t)$.

- The controller in the car uses these measurements to produce control signals: if the current measurement $s(t)$ is less than the desired cruising speed, the controller sends a signal to the throttle to accelerate, and if $s(t)$ is greater than the desired speed, the throttle is asked to allow the car to slow down.

- The throttle performs the action specified by the controller.

The motion of the car might also be affected by disturbances such as wind gusts, or slippery road conditions. A properly designed cruise control system will maintain the speed at (or near) the desired value despite these external conditions. □

Example 3 (Inverted Pendulum)**.** Suppose we try to balance a stick vertically in the palm of our hand. The sensor, controller and actuator in this example are our eyes, our brain, and our hand, respectively. This is an example of a feedback control system. Now what happens if we try to balance the stick with our eyes closed? The stick inevitably falls. This illustrates another type of control, known as *feedforward* or *open loop* control, where the input to the system does not depend on the output. As this example illustrates, feedforward control is not *robust* to disturbances – if the stick is not perfectly balanced to start, or if our hand moves very slightly, the stick will fall. This illustrates the benefit of feedback control. □

As we will see later, feedback control has many strengths, and is used to achieve the following objectives.

- **Good tracking**. Loosely speaking, feedback control allows us to make the output of the system follow the desired reference input (i.e., make the system behave as it should).

- **Disturbance rejection**. Feedback control allows the system to maintain good behavior even when there are external inputs that we cannot control.

- **Robustness**. Feedback control can work well even when the actual model of the plant is not known precisely; sufficiently small errors in modeling can be counteracted by the feedback input.

Feedback control is everywhere; it appears not only in engineered systems (such as automobiles and aircraft), but also in economic systems (e.g., choosing the interest rates to maintain a desired rate of inflation, growth, etc.), ecological systems (predator/prey populations, global climate) and biological systems (e.g., physiology in animals and plants).

Example 4 (Teary Eyes on a Cold Day)**.** Blinking and tears are a feedback mechanism used by the body to warm the surface of the eyeball on cold days – the insides of the lids warm the eyes. In very cold situations, tears come from inside the body (where they are warmed), and contain some proteins and salts that help to prevent front of eyes from freezing. □

1.3 Outline of the Course

Since control systems appear in a large variety of applications, we will not attempt to discuss each specific application in this course. Instead, we will deal with the underlying mathematical theory, analysis, and design of control systems. In this sense, it will be more mathematical than other engineering courses, but will be different from other math courses in that it will pull together various branches of mathematics for a particular purpose (i.e., to design systems that behave in desired ways).

The trajectory of the course will be as follows.

- **Modeling:** Before we can control a system and make it behave in a desired manner, we need to represent the input-output behavior of the system in a form that is suitable for mathematical analysis.
- **Analysis:** Once we understand how to model systems, we need to have a basic understanding of what the model tells us about the system's response to input signals. We will also need to formulate how exactly we want the output to get to its desired value (e.g., how quickly should it get there, do we care what the output does on the way there, can we be sure that the output will get there, etc.)
- **Design:** Finally, once we have analyzed the mathematical model of the system, we will study ways to design controllers to supply appropriate control (input) signals to the system so that the output behaves as we want it to.

We will be analyzing systems both in the time-domain (e.g., with differential equations) and in the frequency domain (e.g., using Laplace transforms). We will start by reviewing some relevant mathematical concepts.

Chapter 2

Review of Complex Numbers

Consider the polynomial $f(x) = x^2 + 1$. The roots of the polynomial are the values of x for which $f(x) = 0$, or equivalently $x^2 = -1$. Clearly, there are no *real* numbers that satisfy this equation. To address this problem, let us define a new "number" j, such that $j^2 = -1$. Since this number does not belong to the set of real numbers, we will call it an *imaginary* or *complex* number. With this number in hand, we can actually generate an infinite set of other complex numbers.[1]

Definition 1 (Complex Numbers)**.** A complex number s is of the form $s = \sigma + j\omega$, where σ and ω are real numbers. The number σ is called the *real part of* s, and is denoted by $\sigma = \text{Re}(s)$. The number ω is called the *imaginary part of* s and is denoted by $\omega = \text{Im}(s)$. The set of all complex numbers is denoted by $\mathbb{C}$.

A complex number $\sigma + j\omega$ can alternatively be viewed simply as a pair of real numbers (σ, ω). Thus, we can plot complex numbers in a two-dimensional plane called the *complex plane*. The horizontal axis is called the *real axis* and the vertical axis is called the *imaginary axis*. The real axis represents all complex numbers s such that $\text{Im}(s) = 0$; in other words, it contains all real numbers, and thus real numbers are a subset of the complex numbers. The imaginary axis represents all complex numbers s such that $\text{Re}(s) = 0$. The following regions of the plane will be useful to our discussions.

[1] An excellent (and intuitive) perspective on complex numbers can be found in a New York Times essay written by Steven Strogatz; it is available at http://opinionator.blogs.nytimes.com/2010/03/07/finding-your-roots/.

- All complex numbers s satisfying $\mathrm{Re}(s) < 0$ are said to lie in the **Open Left Half Plane** (OLHP).
- All complex numbers s satisfying $\mathrm{Re}(s) \leq 0$ are said to lie in the **Closed Left Half Plane** (CLHP).
- All complex numbers s satisfying $\mathrm{Re}(s) > 0$ are said to lie in the **Open Right Half Plane** (ORHP).
- All complex numbers s satisfying $\mathrm{Re}(s) \geq 0$ are said to lie in the **Closed Right Half Plane** (CRHP).

Figure 2.1: A complex number $s = \sigma + j\omega$ in the complex plane.

An alternative representation of complex numbers is the *polar form.* Specifically, given a complex number $s = \sigma + j\omega$, note that s is also specified uniquely by its distance r from the origin, and the angle θ that s (as a vector from the origin) makes with the positive real axis:

$$r = \sqrt{\sigma^2 + \omega^2}, \quad \theta = \arctan\frac{\omega}{\sigma} \ .$$

The number r is called the *magnitude* of s, and is denoted by $r = |s|$. The number θ is called the *phase* of s (in radians), and is denoted by $\theta = \angle s$. Conversely, we have $\sigma = r\cos\theta$ and $\omega = r\sin\theta$.

Example. What is the polar form representation of $s = 3 - j4$?

Note from Euler's equation that $e^{j\theta} = \cos\theta + j\sin\theta$, and so the complex number s can be denoted by $s = re^{j\theta}$.

Given the complex number $s = \sigma + j\omega$, its *complex conjugate* is defined as the complex number $s^* = \sigma - j\omega$. Note that

$$ss^* = (\sigma + j\omega)(\sigma - j\omega) = \sigma^2 + \omega^2 = |s|^2 = |s^*|^2 \ .$$

In the geometric representation, the complex conjugate of a complex number is obtained by reflecting the vector about the real axis.

As we will see throughout the course, many interesting properties of control systems are related to the roots of certain polynomials. The following result explains why complex numbers are so useful.

Theorem 1 (Fundamental Theorem of Algebra)**.** *Consider any polynomial $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, where the coefficients $a_n, a_{n-1}, \ldots, a_0$ are complex numbers. Then the polynomial has n complex roots. Furthermore, if all coefficients are real, then all non-real roots of $f(x)$ appear in complex conjugate pairs.*

Decomposing a Real Polynomial into Factors

Let $f(x)$ be a polynomial of degree n with all real coefficients. Suppose that $f(x)$ has q roots at $x = 0$, r nonzero real roots at locations $z_1, z_2, \ldots, z_r$, and $2p$ complex conjugate roots at $c_1, c_1^*, c_2, c_2^*, \ldots, c_p, c_p^*$ (we know that these complex roots appear in complex conjugate pairs). Now, note that

$$(x - c_i)(x - c_i^*) = x^2 - (c_i + c_i^*)x + c_i c_i^* = x^2 - 2\mathrm{Re}(c_i)x + |c_i|^2.$$

If we define $\omega_i = |c_i|$ and

$$\zeta_i = -\frac{\mathrm{Re}(c_i)}{\omega_i}$$

we can write

$$(x - c_i)(x - c_i^*) = x^2 + 2\zeta_i\omega_i x + \omega_i^2.$$

We will be using this notation frequently throughout the course. The polynomial $f(x)$ can therefore always be decomposed as

$$f(x) = K_0 x^q (x - z_1)(x - z_2)\cdots(x - z_r)(x^2 + 2\zeta_1\omega_1 x + \omega_1^2)\cdots(x^2 + 2\zeta_p\omega_p x + \omega_p^2)$$

for some constant K_0.

Chapter 3

Review of Laplace Transforms

3.1 The Laplace Transform

Suppose $f(t)$ is a function of time. In this course, unless otherwise noted, we will only deal with functions that satisfy $f(t) = 0$ for $t < 0$.

The **Laplace Transform** of the function $f(t)$ is defined as

$$\mathcal{L}\{f(t)\} = \int_0^\infty f(t)e^{-st}dt \ \ .$$

This is a function of the complex variable s, so we can write $\mathcal{L}\{f(t)\} = F(s)$.

Note: There are various conditions that $f(t)$ must satisfy in order to have a Laplace Transform. For example, it must not grow faster than e^{st} for some s. In this course, we will only deal with functions that have (unique) Laplace Transforms.[1]

Example. Find the Laplace Transform of $f(t) = e^{-at}, t \geq 0$, where $a \in \mathbb{R}$.
Solution.

$$\begin{aligned} F(s) = \mathcal{L}\{f(t)\} &= \int_0^\infty e^{-at}e^{-st}dt \\ &= \int_0^\infty e^{-(s+a)t}dt \end{aligned}$$

[1] Uniqueness may be lost at points of discontinuity. More specifically, if $f(t)$ and $g(t)$ are piecewise continuous functions, then $F(s) = G(s)$ for all s implies that $f(t) = g(t)$ everywhere except at the points of discontinuity.

$$= -\frac{1}{s+a}\, e^{-(s+a)t}\Big|_0^\infty = \frac{1}{s+a} \quad (\text{if } \mathrm{Re}(s+a) > 0) \ .$$

Example. Find the Laplace Transform of the *unit step function* $\mathbf{1}(t)$, defined as

$$\mathbf{1}(t) = \begin{cases} 1 & \text{if } t \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Figure 3.1: The unit step function $\mathbf{1}(t)$.

Solution.

Example: The Dirac delta function (or *impulse* function). Consider the function $\delta_\epsilon(t)$ defined as

$$\delta_\epsilon(t) = \begin{cases} \frac{1}{\epsilon} & \text{if } 0 \leq t \leq \epsilon, \\ 0 & \text{otherwise,} \end{cases}$$

where ϵ is a small positive number.

Figure 3.2: (a) The function $\delta_\epsilon(t)$. (b) The impulse function $\delta(t)$.

Note that $\int_{-\infty}^{\infty} \delta_\epsilon(t)dt = 1$. Now suppose that we let $\epsilon \to 0$; we still have

$$\lim_{\epsilon \to 0} \int_{-\infty}^{\infty} \delta_\epsilon(t)dt = 1 \ .$$

Note that as ϵ gets smaller, the function gets narrower, but taller. Define the *impulse function*

$$\delta(t) = \lim_{\epsilon \to 0} \delta_\epsilon(t) = \begin{cases} \infty & \text{if } t = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Some properties of $\delta(t)$:

- $\int_{-\infty}^{\infty} \delta(t)dt = 1$.
- Let $f(t)$ be any function. Then $\int_{-\infty}^{\infty} \delta(t-\tau)f(t)dt = f(\tau)$. This is called the *sifting* property.

What is the Laplace Transform of $\delta(t)$?
Solution. $\mathcal{L}\{\delta(t)\} = \int_0^{\infty} \delta(t)e^{-st}dt = e^0 = 1$.

Transforms of other common functions can be obtained from Laplace Transform tables:

Table 3.1: Sample Table of Laplace Transforms.

$f(t), t \geq 0$	$F(s)$
$\delta(t)$	1
$\mathbf{1}(t)$	$\frac{1}{s}$
e^{-at}	$\frac{1}{s+a}$
t	$\frac{1}{s^2}$
$\sin \omega t$	$\frac{\omega}{s^2+\omega^2}$
$\cos \omega t$	$\frac{s}{s^2+\omega^2}$
$\vdots$	$\vdots$

Note: The functions $f(t)$ in this table are only defined for $t \geq 0$ (i.e., we are assuming that $f(t) = 0$ for $t < 0$).

Some properties of the Laplace Transform

1. **Linearity.** $\mathcal{L}\{\alpha f_1(t) + \beta f_2(t)\} = \alpha F_1(s) + \beta F_2(s)$.
   Example. What is $\mathcal{L}\{5e^{-3t} - 2\cos 4t\}$?
   Solution.

2. **Time Delay.** Consider a *delayed* signal $f(t-\lambda)$, $\lambda \geq 0$ (i.e., a shifted version of $f(t)$).

$$\begin{aligned}\mathcal{L}\{f(t-\lambda)\} = \int_0^{\infty} f(t-\lambda)e^{-st}dt &= \int_{\lambda}^{\infty} f(t-\lambda)e^{-st}dt \\ &= \int_0^{\infty} f(\tau)e^{-s(\tau+\lambda)}d\tau \quad \text{(letting } \tau = t-\lambda) \\ &= e^{-s\lambda}F(s)\ .\end{aligned}$$

Figure 3.3: The time-delayed signal $f(t-\lambda)$.

Example. What is $\mathcal{L}\{\mathbf{1}(t-5)\}$?
Solution.

3. **Differentiation.** $\mathcal{L}\{\frac{df}{dt}\} = sF(s) - f(0)$. More generally,

$$\mathcal{L}\{\frac{d^m f}{dt^m}\} = s^m F(s) - s^{m-1} f(0) - s^{m-2}\frac{df}{dt}(0) - \cdots - \frac{d^{m-1}f}{dt^{m-1}}(0) \ .$$

Example. What is $\mathcal{L}\{\frac{d}{dt}e^{-at}\}$?
Solution.

4. **Integration.** $\mathcal{L}\{\int_0^t f(\tau)d\tau\} = \frac{1}{s}F(s)$.
Example. What is $\mathcal{L}\{\int_0^t \cos\tau d\tau\}$?
Solution.

5. **Convolution.** The convolution of two signals $f_1(t)$ and $f_2(t)$ is denoted by

$$f_1(t) * f_2(t) = \int_0^t f_1(\tau)f_2(t-\tau)d\tau \ .$$

The Laplace Transform of the convolution of two signals is given by

$$\mathcal{L}\{f_1(t) * f_2(t)\} = F_1(s)F_2(s) \ .$$

This is a very important property of Laplace Transforms! Convolution of two signals in the time-domain (which could be hard to do) is equivalent to the multiplication of their Laplace Transforms in the s domain (which is easy to do).
Example. What is $\mathcal{L}\{\mathbf{1}(t) * \mathbf{1}(t)\}$?
Solution.

3.2 The Inverse Laplace Transform

Given the Laplace Transform $F(s)$, we can obtain the corresponding time-domain function $f(t)$ via the Inverse Laplace Transform:

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \frac{1}{2\pi j}\int_{\sigma - j\infty}^{\sigma + j\infty} F(s)e^{st}ds \ .$$

This is quite tedious to apply in practice, and so we will not be using it in this class. Instead, we can simply use the Laplace Transform tables to obtain the corresponding functions.

Example. What is $\mathcal{L}^{-1}\{\frac{1}{s(s+1)}\}$?
Solution.

$$\begin{aligned}
\mathcal{L}^{-1}\{\frac{1}{s(s+1)}\} &= \mathcal{L}^{-1}\{\frac{1}{s} - \frac{1}{s+1}\} \\
&= \mathcal{L}^{-1}\{\frac{1}{s}\} - \mathcal{L}^{-1}\{\frac{1}{s+1}\} \\
&\qquad \text{(by linearity of the (inverse) Laplace Transform)} \\
&= \mathbf{1}(t) - e^{-t}, \ \ t \geq 0 \ .
\end{aligned}$$

In the above example, we "broke up" the function $\frac{1}{s(s+1)}$ into a sum of simpler functions, and then applied the inverse Laplace Transform (by consulting the Laplace Transform table) to each of them. This is a general technique for inverting Laplace Transforms.

3.2.1 Partial Fraction Expansion

Suppose we have a *rational* function

$$F(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} = \frac{N(s)}{D(s)} \ ,$$

where the a_i's and b_i's are constant real numbers.

Definition 2. If $m \leq n$, the rational function is called *proper*. If $m < n$, it is *strictly proper*.

By factoring $N(s)$ and $D(s)$, we can write

$$F(s) = K\frac{(s+z_1)(s+z_2)\cdots(s+z_m)}{(s+p_1)(s+p_2)\cdots(s+p_n)} \quad .$$

Definition 3 (Zeros and Poles)**.** The complex numbers $-z_1, -z_2, \ldots, -z_m$ are the roots of $N(s)$ and are called the **zeros** of $F(s)$. The complex numbers $-p_1, -p_2, \ldots, -p_n$ are the roots of $D(s)$ and are called the **poles** of $F(s)$.

Note: Remember these terms, as the poles and zeros of a system will play a very important role in our ability to control it.

First, suppose each of the poles are distinct and that $F(s)$ is strictly proper. We would like to write

$$F(s) = \frac{k_1}{s+p_1} + \frac{k_2}{s+p_2} + \cdots + \frac{k_n}{s+p_n} \quad ,$$

for some constants $k_1, k_2, \ldots, k_n$, since the inverse Laplace Transform of $F(s)$ is easy in this form. How do we find $k_1, k_2, \ldots, k_n$?

Heaviside's Cover-up Method. To find the constant k_i, multiply both sides of the expansion of $F(s)$ by $(s + p_i)$:

$$(s+p_i)F(s) = \frac{k_1(s+p_i)}{s+p_1} + \frac{k_2(s+p_i)}{s+p_2} + \cdots + k_i + \cdots + \frac{k_n(s+p_i)}{s+p_n} \quad .$$

Now if we let $s = -p_i$, then all terms on the right hand side will be equal to zero, except for the term k_i. Thus, we obtain

$$k_i = (s+p_i)F(s)\big|_{s=-p_i} \quad .$$

Example. What is the partial fraction expansion of $F(s) = \frac{s+5}{s^3+3s^2-6s-8}$?

Solution.

The partial fraction expansion when some of the poles are repeated is obtained by following a similar procedure, but it is a little more complicated. We will not worry too much about this scenario here. One can also do a partial fraction expansion of nonstrictly proper functions by first dividing the denominator into the numerator to obtain a constant and a strictly proper function, and then applying the above partial fraction expansion. The details will be covered in a homework problem.

3.3 The Final Value Theorem

Let $F(s)$ be the Laplace Transform of a function $f(t)$. Often, we will be interested in how $f(t)$ behaves as $t \to \infty$ (this is referred to as the *asymptotic* or *steady state* behavior of $f(t)$). Can we obtain this information directly from $F(s)$?

Theorem 2 (Final Value Theorem)**.** *If all poles of $sF(s)$ are in the open left half plane, then*

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s) \ .$$

Example. $F(s) = \frac{1}{s(s+1)}$. What is $\lim_{t\to\infty} f(t)$?
Solution.

Why do we need the poles of $sF(s)$ to be in the OLHP in order to apply the Final Value Theorem? First, note from the partial fraction expansion of $F(s)$ that

$$\begin{aligned} F(s) &= \frac{k_1}{s} + \frac{k_2}{s+p_2} + \frac{k_3}{s+p_3} + \cdots + \frac{k_n}{s+p_n} \\ \Leftrightarrow f(t) &= k_1\mathbf{1}(t) + k_2 e^{-p_2 t} + k_3 e^{-p_3 t} + \cdots + k_n e^{-p_n t} \quad . \end{aligned}$$

Note that if $F(s)$ does not have a pole at $s = 0$, then the constant k_1 will simply be zero in the above expansion. Based on the above expansion, if one of the poles has a positive real part, the corresponding exponential term will explode, and thus $f(t)$ will have no final value! On the other hand, if all poles have negative real parts, all of the exponential terms will go to zero, leaving only the term $k_1\mathbf{1}(t)$ (corresponding to $\frac{k_1}{s}$ in the partial fraction expansion). Thus, the asymptotic behavior of $f(t)$ is simply k_1, and this is obtained by calculating $\lim_{s\to 0} sF(s)$.

One must be careful about applying the Final Value Theorem; if the signal $f(t)$ does not settle down to some constant steady state value, the theorem might yield nonsensical results. For example, consider the function $f(t) = \sin t$, with Laplace transform

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

The function $sF(s)$ does not have all poles in the OLHP (it has two poles on the imaginary axis). However, if we forgot to check this before applying the Final Value Theorem, we would get

$$\lim_{s\to 0} sF(s) = 0,$$

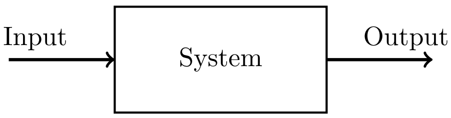
and would mistakenly assume that $f(t) \to 0$. Clearly $f(t)$ has no steady state value (it constantly oscillates between -1 and 1).

Chapter 4

Linear Time-Invariant Systems

4.1 Linearity, Time-Invariance and Causality

Consider the system:



The system is said to be **linear** if the *Principle of Superposition* holds.

Definition 4 (Principle of Superposition)**.** Suppose that the output of the system is $y_1(t)$ in response to input $u_1(t)$ and $y_2(t)$ in response to input $u_2(t)$. Then the output of the system in response to the input $\alpha u_1(t) + \beta u_2(t)$ is $\alpha y_1(t) + \beta y_2(t)$, where α and β are arbitrary real numbers. Note that this must hold for *any* inputs $u_1(t)$ and $u_2(t)$.

The system is said to be **time-invariant** if the output of the system is $y(t-\tau)$ when the input is $u(t-\tau)$ (i.e., a time-shifted version of the input produces an equivalent time-shift in the output).

The system is said to be **causal** if the output at time t depends only on the input up to time t. In particular, this means that if $u(t) = 0$ for $t < \tau$, then $y(t) = 0$ for $t < \tau$.

Figure 4.1: The Principle of Superposition.

Figure 4.2: The Time-Invariance Property

In this class, we will primarily be dealing with the analysis of linear time-invariant causal systems (if the system is nonlinear, we will *linearize* it). We will be interested in how the system responds to certain types of inputs. The **impulse response** of a system is the output of the system when the input to the system is $\delta(t)$, and is denoted by the signal $h(t)$. The **step response** is the output of the system when the input to the system is $\mathbf{1}(t)$.

Note: The impulse response for causal systems satisfies $h(t) = 0$ for $t < 0$.

4.2 Transfer Functions

Recall from the *sifting property* of the delta function that $u(t) = \int_0^\infty u(\tau)\delta(t-\tau)d\tau$. Note that this can be interpreted in the following way: the signal $u(t)$ is the *sum* of an infinite series of weighted and shifted delta functions. The delta function at $t = \tau$ is weighted by the value $u(\tau)$. Now, the time-invariance of

Figure 4.3: (a) A causal system. (b) A non-causal system.

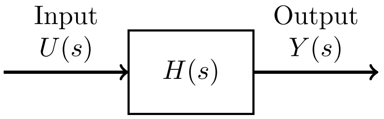
the system says that the output of the system due to the input $\delta(t-\tau)$ will be $h(t-\tau)$. Using the Principle of Superposition, we see that the output of the system when $u(t)$ is the input will be the infinite sum (or integral) of weighted and shifted impulse responses (when all initial conditions are equal to zero):

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

Note that the output of the system is just the *convolution* of the signals $u(t)$ and $h(t)$! This is a well known relationship for linear time-invariant systems. Applying Laplace Transforms to both sides of the above equation, we obtain $Y(s) = H(s)U(s)$, or equivalently,

$$H(s) = \frac{Y(s)}{U(s)} \ .$$

The function $H(s)$ is the ratio of the Laplace Transform of the output to the Laplace Transform of the input (when all initial conditions are zero), and it is called the **transfer function** of the system. Note that this transfer function is *independent* of the actual values of the inputs and outputs – it tells us how any input gets transformed into the output. It is a property of the system itself. We will frequently represent systems in block diagrams via their transfer functions:



Note: The transfer function is the Laplace Transform of the impulse response of the system.

4.2.1 Obtaining the transfer function of a differential equation model

We will be dealing with systems whose input and output are related via a differential equation of the form

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_1\dot{y} + a_0 y = b_m u^{(m)} + b_{m-1}u^{(m-1)} + \cdots + b_1\dot{u} + b_0 u \ .$$

Taking Laplace Transforms of both sides (assuming that all initial conditions are zero), we get

$$(s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0)Y(s) = (b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0)U(s) \ ,$$

from which we obtain the transfer function

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \ .$$

The impulse response of this system is given by $h(t) = \mathcal{L}^{-1}\{H(s)\}$.

Example. Consider the system given by the differential equation $\dot{y}+3y = 2u(t)$. What is the transfer function of this system? What is the impulse response? What is the step response?
Solution.

4.3 Frequency Response

Suppose that the input to the system is $u(t) = e^{s_0 t}, t \geq 0$, for some $s_0 = \sigma + j\omega$. Recall that

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

Since both $u(t)$ and $h(t)$ are zero for $t < 0$, the above expression becomes

$$y(t) = \int_0^t u(t-\tau)h(\tau)d\tau = \int_0^t e^{s_0(t-\tau)}h(\tau)d\tau = e^{s_0 t}\int_0^t e^{-s_0\tau}h(\tau)d\tau \ .$$

The quantity $\int_0^t e^{-s_0\tau}h(\tau)d\tau$ looks a lot like the Laplace Transform of the signal $h(\tau)$ evaluated at $s = s_0$, except that the upper limit on the integration is t instead of ∞. Suppose that we examine what happens when t becomes very large (i.e., as $t \to \infty$). In this case, if the integral exists, we can write

$$y(t) = H(s_0)e^{s_0 t} \ .$$

Thus, the asymptotic response (if $H(s_0)$ exists) to a complex exponential input is that same complex exponential, scaled by the transfer function evaluated at s_0. This gives us one potential way to identify the transfer function of a given "black box" system: apply the input $e^{s_0 t}$ for several different values of s_0, and use that to infer $H(s)$.

Problem: If $\text{Re}(s_0) > 0$, then $e^{s_0 t}$ blows up very quickly, and if $\text{Re}(s_0) < 0$, it decays very quickly. What about $s_0 = j\omega$? This would solve the problem. How do we apply $e^{j\omega t}$ in practice?

Solution: Sinusoids. Recall the identity $\cos\omega t = \frac{e^{j\omega t}+e^{-j\omega t}}{2}$. Using the Principle of Superposition and the property derived above:

- The response of the system to the input $u_1(t) = \frac{1}{2}e^{j\omega t}$ is

$$y_1(t) = \frac{1}{2}H(j\omega)e^{j\omega t}.$$

- The response of the system to the input $u_2(t) = \frac{1}{2}e^{-j\omega t}$ is

$$y_2(t) = \frac{1}{2}H(-j\omega)e^{-j\omega t}.$$

Note that $H(j\omega)$ is just a complex number, and so we can write it in the polar form $H(j\omega) = |H(j\omega)|e^{j\angle H(j\omega)}$, where $|H(j\omega)|$ is the magnitude of $H(j\omega)$ and $\angle H(j\omega)$ is the phase of $H(j\omega)$ (they will both depend on the choice of ω). Similarly, $H(-j\omega)$ is just the complex conjugate of $H(j\omega)$,[1] and so we can

[1] You should be able to prove this by using the fact that the numerator and denominator of $H(s)$ are polynomials with real coefficients.

write $H(-j\omega) = |H(j\omega)|e^{-j\angle H(j\omega)}$. Using these identities, and the Principle of Superposition, the output of the system when the input is $u(t) = u_1(t)+u_2(t) = \cos\omega t$ is

$$\begin{aligned} y(t) = y_1(t) + y_2(t) &= \frac{1}{2}H(j\omega)e^{j\omega t} + \frac{1}{2}H(-j\omega)e^{-j\omega t} \\ &= \frac{1}{2}|H(j\omega)|e^{j\angle H(j\omega)}e^{j\omega t} + \frac{1}{2}|H(j\omega)|e^{-j\angle H(j\omega)}e^{-j\omega t} \\ &= \frac{1}{2}|H(j\omega)|\left(e^{j(\omega t+\angle H(j\omega))} + e^{-j(\omega t+\angle H(j\omega))}\right) \\ &= |H(j\omega)|\cos(\omega t + \angle H(j\omega)) \ . \end{aligned}$$

In other words, the (steady-state) response to the sinusoid $\cos\omega t$ is a scaled and phase-shifted version of the sinusoid! This is called the **frequency response** of the system, and will be a useful fact to identify and analyze linear systems. Later, we will be plotting the magnitude and phase of the system as we sweep ω from 0 to ∞; this is called the *Bode plot* of the system.

As an example, consider a linear system with transfer function

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \ ,$$

where ζ and ω_n are some real numbers. We will study systems of this form in more detail later in the course. Since the denominator of this transfer function has degree 2, it is called a *second order* system. The magnitude of this function at $s = j\omega$ is given by

$$\begin{aligned} |H(j\omega)| &= \frac{\omega_n^2}{|-\omega^2 + 2\zeta\omega_n\omega j + \omega_n^2|} \\ &= \frac{\omega_n^2}{\sqrt{(\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2}} = \frac{1}{\sqrt{(1-(\frac{\omega}{\omega_n})^2)^2 + 4\zeta^2(\frac{\omega}{\omega_n})^2}} \ , \end{aligned}$$

and the phase is given by

$$\begin{aligned} \angle H(j\omega) &= \angle\frac{\omega_n^2}{-\omega^2 + 2\zeta\omega_n\omega j + \omega_n^2} \\ &= \angle\frac{1}{-(\frac{\omega}{\omega_n})^2 + 2\zeta(\frac{\omega}{\omega_n})j + 1} = -\tan^{-1}\left(\frac{2\zeta\frac{\omega}{\omega_n}}{1-\left(\frac{\omega}{\omega_n}\right)^2}\right) \ . \end{aligned}$$

Since these quantities are a function of $\frac{\omega}{\omega_n}$, we can plot them vs $\frac{\omega}{\omega_n}$ for various values of ζ. Note that in the following plots, we used a logarithmic scale for the frequency. This is commonly done in order to include a wider range of frequencies in our plots. The intervals on logarithmic scales are known as

decades.

We can label the following important characteristics of the magnitude plot:

- The magnitude of the transfer function for low frequencies (i.e., near $\omega = 0$) is called the **low frequency gain** or the **DC gain**.
- The **bandwidth** of the system is the frequency at which the magnitude drops to $\frac{1}{\sqrt{2}}$ times the DC gain, and is denoted by ω_{BW}. For the second order system considered above, the plot shows that the bandwidth is approximately equal to ω_n.
- The **resonant peak** is the difference between maximum value of the frequency response magnitude and the DC gain, and is denoted by M_r.
- The **resonant frequency** is the frequency at which the resonant peak occurs, and is denoted by ω_r.

The concepts of bandwidth and DC gain will play an important role in this course, and you should be comfortable manipulating and deriving these quantities. While we identified these metrics using the magnitude plot of a second order system, these quantities can be used to discuss the frequency response of any transfer function. For simple first order systems, we can calculate these quantities explicitly, as shown in the following example.

Example. Find the bandwidth and DC gain of the system with transfer function $H(s) = \frac{b}{s+a}$.
Solution. The DC gain is the magnitude of the transfer function when $s = j\omega$, with $\omega = 0$. In this case, the DC gain is

$$|H(0)| = \frac{b}{a}.$$

The bandwidth is the frequency ω_{BW} at which the magnitude $|H(j\omega_{BW})|$ is equal to $\frac{1}{\sqrt{2}}$ of the DC gain. In this case, we have

$$\frac{1}{\sqrt{2}}\frac{b}{a} = |H(j\omega_{BW})| = \frac{b}{|j\omega_{BW} + a|} = \frac{b}{\sqrt{a^2 + \omega_{BW}^2}}.$$

Cancelling out the b and squaring both sides, we obtain

$$2a^2 = a^2 + \omega_{BW}^2,$$

which means that $\omega_{BW} = a$. □

Next, we will develop some ways to systematically draw the magnitude and phase for general transfer functions.

Chapter 5

Bode Plots

A Bode plot is a plot of the magnitude and phase of a linear system, where the magnitude is plotted on a logarithmic scale, and the phase is plotted on a linear scale. Specifically, consider the linear system with transfer function $H(s) = \frac{N(s)}{D(s)}$. For the moment, assume that all poles and zeros of the transfer function are real (to avoid cumbersome notation), and write

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

When working with Bode plots, we will find it more convenient to write this system as:

$$H(s) = K_o\frac{(\frac{s}{z_1}+1)(\frac{s}{z_2}+1)\cdots(\frac{s}{z_m}+1)}{(\frac{s}{p_1}+1)(\frac{s}{p_2}+1)\cdots(\frac{s}{p_n}+1)} \ ,$$

where $K_o = K\frac{z_1 z_2 \ldots z_m}{p_1 p_2 \ldots p_n}$. This is called the **Bode form**, and the reason for doing this is that the DC gain of the above transfer function can now immediately be obtained as K_o, which will be useful when drawing Bode plots. We will handle more general transfer functions after the following discussion.

The magnitude of $H(j\omega)$ is given by

$$|H(j\omega)| = |K_o|\frac{|j\frac{\omega}{z_1}+1||j\frac{\omega}{z_2}+1|\cdots|j\frac{\omega}{z_m}+1|}{|j\frac{\omega}{p_1}+1||j\frac{\omega}{p_2}+1|\cdots|j\frac{\omega}{p_n}+1|} \ .$$

Now if we take the logarithm of both sides (any base is acceptable, but base 10 is conventional), we get

$$\log|H(j\omega)| = \log|K_o| + \sum_{i=1}^{m}\log|j\frac{\omega}{z_i}+1| - \sum_{i=1}^{n}\log|j\frac{\omega}{p_i}+1| \ .$$

In other words, when viewed on a logarithmic scale, the magnitudes of each of the individual terms in the transfer function *add* together to produce the magnitude of the overall transfer function. This is quite useful, and the reason for introducing the logarithm. In keeping with convention, we will multiply both sides of the above equation by 20, and work with the units in *decibels*; this only scales the magnitudes, but does not change the additivity due to the logarithm.

Note that the phase of $H(j\omega)$ already satisfies the additivity property:

$$\angle H(j\omega) = \angle K_o + \sum_{i=1}^{m} \angle (j\frac{\omega}{z_i} + 1) - \sum_{i=1}^{n} \angle (j\frac{\omega}{p_i} + 1) \ ,$$

and thus it suffices to consider the phase on a linear scale.

Note that we can always draw Bode plots for any transfer function by simply evaluating the magnitude and phase for each value of ω and then plotting these values. However, will want to come up with some quick rules to sketch these plots.

5.1 Rules for Drawing Bode Plots

Now that we understand the general motivation behind drawing the log-magnitude and phase of Bode plots, we will study how to draw the Bode plots of general transfer functions (potentially involving complex conjugate terms as well). We will assume for now that the transfer function $H(s)$ has all zeros and poles in the CLHP (we will discuss the more general case later). Recall from Chapter 2 that a polynomial with real coefficients can always be decomposed into a product of a constant, first order terms (corresponding to real roots) and second order terms (corresponding to complex conjugate roots). Thus, we can assume (without loss of generality) that $H(s)$ is composed of four different types of factors:

- K_o (a constant)
- s^q (corresponding to zeros at the origin if q is a positive integer, or poles at the origin if q is a negative integer)
- $(\frac{s}{p} + 1)^{-1}$ and $(\frac{s}{z} + 1)$ (corresponding to real poles and zeros)
- $\Big((\frac{s}{\omega_n})^2 + 2\zeta(\frac{s}{\omega_n}) + 1\Big)^{\pm 1}$ (corresponding to complex conjugate zeros if the exponent is 1, and complex conjugate poles if the exponent is -1).

Example. Consider $H(s) = 3\frac{(s+2)(s^2+2s+4)}{s^3(s+1)(s^2+3s+4)}$. Write this in Bode form, and write the logarithm of the magnitude of $H(j\omega)$ in terms of the logarithm of the magnitudes of each of the factors. Also write the phase of $H(j\omega)$ in terms of the phases of each of the factors.

Since the log-magnitude and phase are obtained by simply adding together the log-magnitudes and phases of the individual factors, we can draw the Bode plot for the overall system by drawing the Bode plots for each of the individual factors, and then adding the plots together.

5.1.1 Bode Plot for K_o

The Bode plot of constant terms is very easy to draw. The log-magnitude is simply $20 \log |K_o|$, and the phase is

$$\angle K_o = \begin{cases} 0 & \text{if } K_o > 0 \\ \pi & \text{if } K_o < 0. \end{cases}$$

Both the magnitude and phase are just horizontal lines.

Example. Draw the Bode plot of $\frac{1}{10}$.
Solution.

Example. Draw the Bode plot of -10.
Solution.

5.1.2 Bode Plot for s^q

The log-magnitude of the factor s^q for $s = j\omega$ is given by

$$20 \log |(j\omega)^q| = 20q \log |j\omega| = 20q \log |\omega| \ .$$

On a log scale, this simply a straight line with slope $20q$, going through the point 0 when $\omega = 1$.

The phase of s^q at $s = j\omega$ is

$$\angle(j\omega)^q = q\angle j\omega = q\frac{\pi}{2} \ ,$$

which is just a horizontal line at $q\frac{\pi}{2}$.

Example. Draw the Bode plot of s^2.
Solution.

Example. Draw the Bode plot of $\frac{1}{s^3}$.
Solution.

5.1.3 Bode Plot for $(\frac{s}{p}+1)^{-1}$ and $(\frac{s}{z}+1)$

A system zero at $s = -z$ will correspond to the factor $(\frac{s}{z}+1)$ in the Bode form of the transfer function, and a system pole at $s = -p$ will correspond to the factor $(\frac{s}{p}+1)^{-1}$, where we are assuming that $z > 0$ and $p > 0$. Let's deal with the zero first. The log-magnitude of the factor $(\frac{s}{z}+1)$ at $s = j\omega$ is given by

$$20\log|j\frac{\omega}{z}+1| = 20\log\sqrt{1+(\frac{\omega}{z})^2}\ .$$

If $\omega \ll z$, we have $20\log|j\frac{\omega}{z}+1| \approx 0$, and thus for values of ω less than z, the magnitude is simply a horizontal line at 0. If $\omega \gg z$, we have $20\log|j\frac{\omega}{z}+1| \approx 20\log(\frac{1}{z}) + 20\log(\omega)$. On a log scale, this is a line of slope 20, going through the point 0 when $\omega = z$. These two rules together produce a magnitude plot that looks like this:

The point $\omega = z$ is called the **breakpoint**. These straight lines were derived based on values of ω that were much smaller or much larger than the breakpoint, and thus they are called **asymptotes**. Note that they are only approximations to the shape of the actual magnitude plot – for example, the actual value of $20\log(j\frac{\omega}{z}+1)$ at $\omega = z$ is equal to 3 dB. However, the approximations will suffice for us to obtain some general intuition about Bode plots.

The phase of $(j\frac{\omega}{z}+1)$ is given by

$$\angle(j\frac{\omega}{z}+1) = \tan^{-1}\frac{\omega}{z} \quad .$$

For $\omega \ll z$, this is approximately equal to 0, and for $\omega \gg z$, this is approximately equal to $\frac{\pi}{2}$. At the breakpoint $\omega = z$, the phase is $\tan^{-1}(1) = \frac{\pi}{4}$. The phase curve transitions smoothly from 0 to $\frac{\pi}{2}$ through $\frac{\pi}{4}$; for convenience, we will draw the transition as a line starting at $\omega = 0.1z$ and ending at $\omega = 10z$ (i.e., one decade before and one decade after z):

The magnitude and phase plots for a factor corresponding to a pole follow the same rules as the plot for the zero, except that everything is negated. Specifically, for a factor of the form $(j\frac{\omega}{p}+1)^{-1}$, we have

$$\begin{aligned}20\log|(j\frac{\omega}{p}+1)^{-1}| &= 20\log(\sqrt{1+(\frac{\omega}{z})^2})^{-1} = -20\log(\sqrt{1+(\frac{\omega}{p})^2}) \ , \\ \angle(j\frac{\omega}{p}+1)^{-1} &= -\tan^{-1}\frac{\omega}{p} \ .\end{aligned}$$

The Bode plot for the factor $(j\frac{\omega}{p}+1)^{-1}$ looks like this:

Example. Draw the Bode plot for $H(s) = 10\frac{s+10}{(s+1)(s+100)}$.
Solution.

5.1.4 Bode Plot for $\left((\frac{s}{\omega_n})^2 + 2\zeta(\frac{s}{\omega_n}) + 1\right)^{\pm 1}$

We have already seen what the magnitude and phase plots look like for a second order system of this form. To derive general rules for drawing this, note that the magnitude of this function at $s = j\omega$ is given by

$$20 \log \left|(\frac{j\omega}{\omega_n})^2 + 2\zeta(\frac{\omega}{\omega_n})j + 1\right| = 20 \log \sqrt{(1 - (\frac{\omega}{\omega_n})^2)^2 + 4\zeta^2(\frac{\omega}{\omega_n})^2} \ .$$

For $\omega \ll \omega_n$, we have $\frac{\omega}{\omega_n} \approx 0$, and so $20 \log \left|(\frac{j\omega}{\omega_n})^2 + 2\zeta(\frac{\omega}{\omega_n})j + 1\right| \approx 0$. For $\omega \gg \omega_n$, we have

$$\sqrt{(1 - (\frac{\omega}{\omega_n})^2)^2 + 4\zeta^2(\frac{\omega}{\omega_n})^2} \approx \sqrt{\left(\frac{\omega}{\omega_n}\right)^4} = \left(\frac{\omega}{\omega_n}\right)^2 \ ,$$

and so $20 \log \left|(\frac{j\omega}{\omega_n})^2 + 2\zeta(\frac{\omega}{\omega_n})j + 1\right| \approx 40 \log \omega - 40 \log \omega_n$. This is a line of slope 40 passing through the point 0 when $\omega = \omega_n$. The general magnitude curve for $20 \log \left|(\frac{j\omega}{\omega_n})^2 + 2\zeta(\frac{\omega}{\omega_n})j + 1\right|$ thus looks like:

The phase of $\left((\frac{j\omega}{\omega_n})^2 + 2\zeta(\frac{\omega}{\omega_n})j + 1\right)$ is given by

$$\angle\left((\frac{j\omega}{\omega_n})^2 + 2\zeta(\frac{\omega}{\omega_n})j + 1\right) = \tan^{-1}\left(\frac{2\zeta\frac{\omega}{\omega_n}}{1 - \left(\frac{\omega}{\omega_n}\right)^2}\right) \ .$$

For $\omega \ll \omega_n$, the argument of the arctan function is almost 0, and so the phase curve starts at 0 for small ω. For $\omega = \omega_n$, the argument is ∞, and so the phase curve passes through $\frac{\pi}{2}$ when $\omega = \omega_n$. For $\omega \gg \omega_n$, the argument of the arctan function approaches 0 from the negative side, and so the phase curve approaches π for large values of ω. Just as in the first order case, we will take the phase curve transitions to occur one decade before and after ω_n. This produces a phase curve of the form:

The Bode plot for the factor $\left((\frac{j\omega}{\omega_n})^2 + 2\zeta(\frac{\omega}{\omega_n})j + 1\right)^{-1}$ looks just like the Bode plot for the factor $\left((\frac{j\omega}{\omega_n})^2 + 2\zeta(\frac{\omega}{\omega_n})j + 1\right)$, except that everything is flipped:

Example. Draw the Bode Plot of $H(s) = \frac{(s+1)(s^2+3s+100)}{s^2(s+10)(s+100)}$

Solution.

5.1.5 Nonminimum Phase Systems

So far we have been looking at the case where all zeros and poles are in the CLHP. Bode plots can also be drawn for systems that have zeros or poles in the RHP – however, note that for systems that have RHP poles, the steady state response to a sinusoidal input will not be a sinusoid (there won't even be a steady state response, as the output will blow up). This does not change the fact that the transfer function will have a magnitude and phase at every frequency (since the transfer function is simply a complex number at every frequency ω). Transfer functions with zeros in the right half plane are called **nonminimum phase systems**, and those with all zeros and poles in the CLHP are called **minimum phase systems**.

To gain intuition about how the Bode plot of a nonminimum phase system compares to that of a minimum phase system, let us see how the Bode plots of the terms $H_1(s) = s + 1$ and $H_2(s) = s - 1$ compare. First, note that

$$|H_1(j\omega)| = |j\omega + 1| = \sqrt{1 + \omega^2} = |H_2(j\omega)|,$$

and thus the magnitude plots of the two terms are identical. To compare the phase contribution, it is useful to examine the complex number representation of $H_1(j\omega)$ and $H_2(j\omega)$:

From this, we see that $\angle H_2(j\omega) = \angle(j\omega - 1) = \pi - \angle H_1(j\omega)$. Thus, the phase plots of the two terms look like this:

An alternative method to draw the phase plot of $H_2(s) = s - 1$ is to first convert it to Bode form to obtain $H_2(s) = -1(-s+1)$, where we now have a gain of -1 in front. This gain contributes nothing to the log-magnitude, but it contributes a phase of π. The phase of $-s+1$ is the negative of the phase of $s+1$ (since they are complex conjugates), and once again, we obtain that the phase of $H_2(j\omega)$ is $\pi - \angle H_1(j\omega)$.

Example. Draw the Bode plots for the systems

$$H_1(s) = 10\frac{s+1}{s+10}, \qquad H_2(s) = 10\frac{s-1}{s+10} \quad .$$

Solution.

As we can see from the above example, the magnitudes of the two transfer functions do not depend on whether the zero is in RHP or the LHP. However, the phase plots are quite different. Based on the above analysis, we see that the phase contribution of a zero in the right half plane is always at least as large (in absolute terms) as the phase contribution of a zero in the left half plane – this is the reason for calling systems with such zeros (or poles) nonminimum phase. Note that for minimum phase systems, the magnitude plot uniquely determines the transfer function, but for nonminimum phase systems, we need both the magnitude plot and the phase plot in order to determine the transfer function.

Chapter 6

Modeling and Block Diagram Manipulation

With the mathematical foundations from the previous chapters in hand, we are now ready to move on to the modeling of control systems.

6.1 Mathematical Models of Physical Systems

The first task of control system design is to obtain an appropriate *mathematical model* of the plant. In many applications, this can be difficult to do, as experimental data is often noisy, and real-world systems are often quite complex. Thus, we must frequently come up with an *approximate* model, maintaining a tradeoff between complexity and how accurately it captures the actual physical plant. In general, we try to follow Einstein's rule: "Make things as simple as possible, but no simpler."

If the system is known (or approximated) to be linear, then one way to identify a model is to apply sinusoidal inputs of appropriate frequencies to the system, and then try and fit an appropriate transfer function to the resulting Bode plot. An alternative (and complementary) method is to use physical laws to obtain the model. We will now study some examples of this approach.

6.1.1 Mechanical Systems

The key equation governing the model of many mechanical systems is *Newton's Law*: $\mathbf{F} = \mathbf{ma}$. In this equation, $\mathbf{F}$ represents the vector sum of all the forces acting on a body, $\mathbf{m}$ represents the mass of the body, and $\mathbf{a}$ represents the acceleration of the body. The forces acting on the body can be generated by an outside entity (as an input to the system), or by *springs* and *dampers* attached to the body.

Example: Mass-Spring-Damper System.

6.1.2 Electrical Systems

Standard components in electrical systems include **resistors**, **capacitors** and **inductors**, connected together in various ways. The quantities of interest in electrical systems are **voltages** and **currents**.

Figure 6.1: (a) Components of Electrical Systems. (b) An Electrical System.

The main modeling technique for electrical systems is to use **Kirchoff's Laws**.

- **Kirchoff's Voltage Law (KVL):** The algebraic sum of the voltages around a closed loop is zero.

- **Kirchoff's Current Law (KCL):** The algebraic sum of the currents coming into a node is zero.

We will see an example of an electric circuit model in the context of a DC motor below.

6.1.3 Rotational Systems

When the system involves rotation about a point, the system dynamics are governed by a modified form of Newton's Law: $\tau = \mathbf{J}\ddot{\theta}$. Here, τ is the sum of all external torques about the center of mass, $\mathbf{J}$ is the moment of inertia of the body, and θ is the angular position of the body (so that $\ddot{\theta}$ is the angular acceleration).

Example: A DC motor consists of an electrical component and a rotational component. The input voltage to the electrical component induces a current, which then provides a torque to the motor shaft via a magnetic field. This torque causes the shaft to rotate. In turn, this torque also induces a voltage drop (called the *back emf*) in the electrical circuit. Derive the overall system equations. and find the transfer function from the input voltage to the angular velocity ($\omega = \dot{\theta}$) of the DC motor.

All of the above examples considered above yielded models that involved linear differential equations of the form

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_1\dot{y} + a_0 y = b_m u^{(m)} + b_{m-1}u^{(m-1)} + \cdots + b_1\dot{u} + b_0 u \ .$$

As we saw in Section 4.2, the transfer function of such systems is given by

$$H(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \ .$$

In practice, many systems are actually *nonlinear*, and there is a whole set of tools devoted to controlling such systems. One technique is to *linearize* the system around an operating point, where the nonlinearities are *approximated* by linear functions. In the rest of the course, we restrict our attention to linear systems, and assume that these linearization techniques have been applied to any nonlinearities in the system. We will now study how to manipulate interconnections of linear systems.

6.2 Block Diagram Manipulation

Control systems typically involve several smaller systems (or components) that are interconnected together in various ways – the output of one system will be the input to other systems. For example, remember from the first lecture that the basic block diagram of a feedback control system looks like this:

We will frequently want to manipulate block diagram representations of systems in order to find the overall transfer function of the system in terms of the transfer functions of the individual components or subsystems. There are three types of interconnections that we will be studying.

Series Connection. In this case, the output of one system feeds directly into the input of another system.

Figure 6.2: Two Systems Connected in Series.

The overall transfer function from $U(s)$ to $Y(s)$ can be obtained as follows:

$$\left.\begin{array}{rcl} Y(s) & = & H_2(s)Y_1(s) \\ Y_1(s) & = & H_1(s)U(s) \end{array}\right\} \Rightarrow Y(s) = H_2(s)H_1(s)U(s) \ ,$$

and thus the overall transfer function is $H(s) = \frac{Y(s)}{U(s)} = H_2(s)H_1(s)$.

Parallel Connection. In this case, two (or more systems) obtain the same input $U(s)$, and their outputs are summed together to produce the output of the overall system.

Figure 6.3: Two Systems Connected in Parallel.

The overall transfer function from $U(s)$ to $Y(s)$ can be obtained as follows:

$$\left.\begin{array}{rcl} Y_1(s) & = & H_1(s)U(s) \\ Y_2(s) & = & H_2(s)U(s) \\ Y(s) & = & Y_1(s) + Y_2(s) \end{array}\right\} \Rightarrow Y(s) = \left(H_1(s) + H_2(s)\right)U(s) \ ,$$

and thus the overall transfer function is $H(s) = \frac{Y(s)}{U(s)} = H_1(s) + H_2(s)$.

Feedback Connection. In this case, the output of one system feeds into the input of a second system, and the output of this second system feeds back into the input of the first system (perhaps in conjunction with another signal).

The overall transfer function from $R(s)$ to $Y(s)$ can be obtained by noting that

$$\begin{aligned} Y(s) &= H_1(s)E(s) \\ Y_2(s) &= H_2(s)Y(s) \\ E(s) &= R(s) - Y_2(s), \end{aligned}$$

which yields

$$Y(s) = H_1(s)(R(s) - Y_2(s)) = H_1(s)R(s) - H_1(s)H_2(s)Y(s).$$

Figure 6.4: Feedback Connection of Two Systems.

Thus the overall transfer function is

$$H(s) = \frac{Y(s)}{R(s)} = \frac{H_1(s)}{1 + H_1(s)H_2(s)} \quad .$$

This is an important transfer function; the numerator is called the "forward gain", and the denominator is described as "1 + the loop gain".

Note that in the above feedback configuration, the output of the system $H_2(s)$ is *subtracted* from the reference signal $R(s)$ to produce the error $E(s)$. This configuration is thus called a **negative feedback** configuration. If the signal $Y_2(s)$ is instead *added* to the reference signal, the configuration is called a **positive feedback** configuration. In this case, the transfer function would be given by

$$H(s) = \frac{Y(s)}{R(s)} = \frac{H_1(s)}{1 - H_1(s)H_2(s)} \quad .$$

Note: The basic feedback control system shown at beginning of this section is a special case of the negative feedback configuration shown above, with $H_1(s) = P(s)C(s)$, and $H_2(s) = 1$. The basic feedback control system is thus said to be in "unity feedback" configuration.

Based on the above configurations, we can derive the following rules to modify block diagrams.

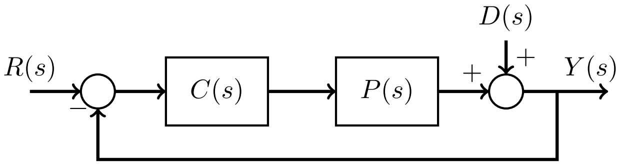
Example. Compute the transfer function from $U(s)$ to $Y(s)$.

6.2.1 Systems with Multiple Inputs and Outputs

We will frequently encounter interconnected systems that have multiple inputs (e.g., control signals, disturbances, etc.) and outputs (e.g., different sensor measurements, etc.). For linear systems, we can simply use the Principle of Superposition to find the transfer function from the inputs to the outputs. More specifically, if we have a set of inputs $U_1(s), U_2(s), \ldots, U_m(s)$ and a set of outputs $Y_1(s), Y_2(s), \ldots, Y_p(s)$, then we find the transfer function $H_{ij}(s)$ from input $U_i(s)$ to output $Y_j(s)$ by simply removing all of the other inputs and outputs from the system and calculating the resulting transfer function as above. By linearity, each output can then be written as

$$Y_i(s) = H_{1i}(s)U_1(s) + H_{2i}(s)U_2(s) + \cdots + H_{mi}(s)U_m(s), \ \ i \in \{1, 2, \ldots, p\}.$$

Example. Write the output $Y(s)$ of the following system in terms of the inputs $R(s)$ and $D(s)$.



Chapter 7

Step Responses of Linear Systems

Consider a system with transfer function of the form

$$H(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \ .$$

The degree (n) of the denominator polynomial is called the **order** of the system. As we have already seen, systems of order one and two arise frequently in practice; even when the order of a system is higher than two, one can sometimes approximate the system by a first or second order system in order to obtain intuition about the system's behavior. It is therefore important for us to study the responses of first and second order systems to various types of inputs (and in particular, to step inputs). Recall that for step inputs $u(t) = \mathbf{1}(t)$, we have $U(s) = \frac{1}{s}$.

Note that if all poles of $H(s)$ are in the OLHP, the Final Value Theorem states that the steady-state value of the step response is given by

$$\lim_{s\to 0} sY(s) = \lim_{s\to 0} sH(s)U(s) = \lim_{s\to 0} sH(s)\frac{1}{s} = \lim_{s\to 0} H(s) = H(0) \ .$$

The quantity $H(0)$ is called the **DC gain** of the system. This is true regardless of the order of the system.

7.1 Step Response of First Order Systems

Consider a first order system with transfer function

$$H(s) = \frac{b_0}{s + a_0}, \quad a_0 \neq 0.$$

This system has a single pole at $s = -a_0$. The step response of this system is obtained by calculating

$$\begin{aligned} Y(s) = H(s)U(s) &= \frac{b_0}{s + a_0}\frac{1}{s} \\ &= \frac{b_0}{a_0}\left(\frac{1}{s} - \frac{1}{s + a_0}\right) \ , \end{aligned}$$

which produces

$$y(t) = \frac{b_0}{a_0}\mathbf{1}(t) - \frac{b_0}{a_0}e^{-a_0 t}, \ \ t \geq 0. \tag{7.1}$$

Consider two cases:

- If $a_0 > 0$, then the pole of the transfer function is in the OLHP, and $e^{-a_0 t} \to 0$ as $t \to \infty$. The step response thus reaches a steady-state value of $\frac{b_0}{a_0}$ (which is the DC gain of the system). The response is said to be **stable**.
- If $a_0 < 0$, the pole of the transfer function is in the ORHP, and $e^{-a_0 t} \to \infty$ as $t \to \infty$. The step response therefore goes to ∞, and there is no steady state value. The response is said to be **unstable**.

Figure 7.1: Step Response of First Order System. (a) Pole in OLHP. (b) Pole in ORHP.

We will now study two measures of *performance* that can be used to evaluate the step-response.

7.1.1 Rise time

The *rise time* t_r of a step response is defined as the amount of time required for the response to go from 10% of its final value to 90% of its final value.

The rise time of a first order system with transfer function $H(s) = \frac{b_0}{s+a_0}$ is

$$t_r = \frac{\ln 9}{a_0}$$

(you should be able to easily prove this). Note that the larger a_0 is, the smaller the rise time becomes. This can also be seen from the actual step-response (7.1): a larger value of a_0 means that the term $e^{-a_0 t}$ dies out faster, leading the response to get to its steady state quicker.

7.1.2 Settling time

Another measure of performance is the *settling time* of the response, defined as the time required for the response to get to within a certain percentage of its final value and stay there. We will take this percentage to be 2%, but sometimes the settling time is defined in terms of 1% or 5% as well. By equating (7.1) to $0.98\frac{b_0}{a_0}$ and solving for t, we find that the 2% settling time for the first order step response is

$$t_s \approx \frac{3.91}{a_0}.$$

Note: The quantity $\tau = \frac{1}{a_0}$ is called the *time-constant* of the system, and (7.1) is commonly written as

$$y(t) = \frac{b_0}{a_0}\mathbf{1}(t) - \frac{b_0}{a_0}e^{-\frac{t}{\tau}}.$$

A larger time-constant means that the system takes longer to settle to its steady state. To find the time-constant in practice, note that at $t = \tau$ we have

$$y(\tau) = \frac{b_0}{a_0}(1 - e^{-1}) \approx 0.63\frac{b_0}{a_0}.$$

Since the steady state value of $y(t)$ is $\frac{b_0}{a_0}$ (the DC gain of the system), we see that the time-constant is the point in time where the output of the system reaches 63% of its final value. □

One can also readily verify that the bandwidth of the system $H(s)$ occurs at $s = a_0$ (i.e., $|H(ja_0)| = \frac{1}{\sqrt{2}}|H(0)|$). Thus, we get the following rule of thumb.

Fast response ⇔ Small τ ⇔ Large a_0 ⇔ Large bandwidth

7.2 Step Response of Second Order Systems

Consider a second order system of the form

$$H(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} \ .$$

In order to obtain intuition about these systems, we will be focusing on a particular form of second order system:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \quad \omega_n > 0 \ .$$

The poles of this transfer function are obtained from the quadratic formula as $s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$. The location of these poles in the complex plane will vary based on the value of ζ. We analyze three different cases:

- $0 \le \zeta < 1$: The system has two complex poles in the CLHP (they will be in the OLHP if $\zeta > 0$). The system is said to be **underdamped**.
- $\zeta = 1$: The system has two repeated poles at $s = -\omega_n$. The system is said to be **critically damped**.
- $\zeta > 1$: The system has two poles on the negative real axis. The system is said to be **overdamped**.

Figure 7.2: Location of Poles in Complex Plane for Different Ranges of ζ. (a) $0 \le \zeta < 1$. (b) $\zeta = 1$. (c) $\zeta > 1$.

7.2.1 Underdamped and Critically Damped Systems ($0 \le \zeta \le 1$)

The poles of the transfer function in this case are given by $s = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$. To simplify the notation, define

$$\sigma = \zeta\omega_n, \quad \omega_d = \omega_n\sqrt{1 - \zeta^2} \ ,$$

in which case the poles are at $s = -\sigma \pm j\omega_d$. The transfer function can be written as

$$H(s) = \frac{\omega_n^2}{(s+\sigma+j\omega_d)(s+\sigma-j\omega_d)} = \frac{\omega_n^2}{(s+\sigma)^2+\omega_d^2} \ .$$

The Laplace Transform of the step response is given by $Y(s) = H(s)\frac{1}{s}$, and using a Laplace Transform table, we see that

$$y(t) = 1 - e^{-\sigma t}\left(\cos\omega_d t + \frac{\sigma}{\omega_d}\sin\omega_d t\right) \ . \tag{7.2}$$

- When $\zeta = 0$, we have $\sigma = 0$ and $\omega_d = \omega_n$, and the response becomes $y(t) = 1 - \cos\omega_n t$, which oscillates between 0 and 2 for all t.
- When $\zeta = 1$, we have $\sigma = \omega_n$ and $\omega_d = 0$, and the response becomes $y(t) = 1 - e^{-\omega_n t}(1+\omega_n t)$; one can verify that this does not have any oscillations at all, and asymptotically reaches the steady state value of 1 (this is the DC gain $H(0)$).

The response for intermediate values of ζ falls in between these two extremes. The behavior of $y(t)$ for different values of ζ (with a fixed value of ω_n) is shown in Fig. 7.3.

Figure 7.3: Step response as a function of ζ, for $0 \leq \zeta \leq 1$.

As we can see from the responses for various values of ζ, a larger value of ζ corresponds to fewer oscillations and less **overshoot**, and thus ζ is said to represent the amount of **damping** in the response (a higher value of ζ corresponds to more damping). ζ is called the **damping ratio**. Examining Fig. 7.2, we see that the angle between the imaginary axis and the vector connecting the pole to the origin is given by $\sin^{-1}\zeta$. The magnitude of the vector is given by ω_n. Thus, as ζ increases with a fixed ω_n, the poles move on the perimeter of a circle toward the real axis.

The quantity ω_n is called the **undamped natural frequency**, since it represents the location of the pole on the imaginary axis when there is no damping (i.e., when $\zeta = 0$). The quantity ω_d is called the **damped natural frequency**, since it represents the imaginary part of the pole when there is damping. When $\zeta = 0$, we have $\omega_d = \omega_n$. If $\zeta > 0$ and we increase ω_n, the poles move further to the left in the OLHP. This causes the term $\sigma = \zeta\omega_n$ to increase, which causes the term $e^{-\sigma t}$ to die out faster in the step response (equation (7.2)). Thus, increasing ω_n has the effect of making the oscillations die out faster (and, in general, making the system respond faster). Recall from the magnitude plot of the second order frequency response from Section 4.3 that the bandwidth of the system is approximately equal to ω_n. Thus, we obtain the following rule of thumb for the underdamped system.

$$\boxed{\text{Larger } \omega_n \Leftrightarrow \text{Larger bandwidth} \Leftrightarrow \text{Faster response}}$$

We will see this more formally in the next chapter.

7.2.2 Overdamped System ($\zeta > 1$)

The poles of the transfer function in this case are given by $s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$, which are both on the negative real axis (as $\zeta \to \infty$, one pole approaches $s = 0$, and the other pole approaches $s = -\infty$). The transfer function can be written as

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

The Laplace Transform of the step response is given by $Y(s) = H(s)\frac{1}{s}$, and using partial fraction expansion, we see that the step response will have the form

$$y(t) = 1 - k_1 e^{(-\zeta\omega_n - \omega_n\sqrt{\zeta^2-1})t} - k_2 e^{(-\zeta\omega_n + \omega_n\sqrt{\zeta^2-1})t} ,$$

for some constants k_1 and k_2. This response has no oscillations; a sample response is shown in Fig. 7.4.

7.2.3 Discussion

Based on the above analysis, we can come to the following general conclusions about how the poles of second order systems affect their step responses.

Figure 7.4: Step response of an overdamped system ($\zeta > 1$).

- If the poles are complex, the step response will have oscillations and overshoot.
- As the poles move toward the real axis while maintaining a fixed distance from the origin, the amount of oscillation decreases (leading to less overshoot) – this corresponds to the damping ratio ζ increasing (with a fixed ω_n),
- If ω_n increases, the poles move further left in the OLHP, and the oscillations die out faster.
- If all poles are on the negative real axis, there will be no oscillation and no overshoot.

Note that if the transfer function has one or more poles in the open right half plane, the step response will contain a term that goes to ∞ as $t \to \infty$, and thus there is no steady state response.

Based on these general rules, the step responses for various pole locations are shown below:

Chapter 8

Performance of Second Order Step Responses

We have seen that the step response of a second order system with transfer function

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

will have different characteristics, depending on the location of the poles (which are a function of the values ζ and ω_n). We will now introduce some measures of performance for these step responses.

8.1 Performance Measures

We will be studying the following metrics in order to evaluate the performance of second order step responses.

- **Rise time (t_r):** The time taken for the response to first get close to its final value. This is typically measured as the time taken for the response to go from 10% of its final value to 90% of its final value.

- **Settling time (t_s):** The time taken for the response to *stay* close to its final value. We will take this to be the time after which the response stays within 2% of its final value. Other measures of "closeness" can also be used (e.g., 1% instead of 2%, etc.).

- **Peak value (M_p) and overshoot OS:** This is largest value of the step response. One can also calculate the overshoot as the maximum amount that the response overshoots its final value, divided by the final value (often expressed as a percentage).

- **Peak time (t_p):** This is the time at which the response hits its maximum value (this is the peak of the overshoot).

Figure 8.1: Step response of second order system, with t_r, M_p, t_p and t_s shown.

8.1.1 Rise time (t_r)

An explicit formula for the rise time is somewhat hard to calculate. However, we notice that rise time increases with ζ and decreases with ω_n. The best linear fit to the curve gives us the approximation

$$t_r \approx \frac{2.16\zeta + 0.6}{\omega_n} \ ,$$

which is reasonably accurate for $0.3 < \zeta < 0.8$. A cruder approximation is obtained by finding a best fit curve with $\zeta = 0.5$, yielding

$$t_r \approx \frac{1.8}{\omega_n} \ .$$

Keep in mind that these are only approximations, and iteration may be required if design specifications are not met.

8.1.2 Peak value (M_p), Peak time (t_p) and Overshoot OS

Recall that the step response will have overshoot only if the poles are complex – this corresponds to the case $0 \leq \zeta < 1$, and the corresponding system is called underdamped. The overshoot is defined as

$$OS = \frac{M_p - y(\infty)}{y(\infty)} \ ,$$

where M_p is the peak value of the step response, and $y(\infty)$ is the final (steady state) value of the step response. Note that if we want to express the overshoot as a *percentage*, we simply multiply OS by 100, and denote this by $\%OS$.

We can explicitly calculate the peak value of the step response (and the corresponding time) as follows. First, note that at the peak value, the derivative of the step response is zero. Recalling that the step response is given by

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

calculate $\frac{dy}{dt}$ and set it equal to zero to obtain

$$\begin{aligned}0 = \frac{dy}{dt} &= \sigma e^{-\sigma t}(\cos\omega_d t + \frac{\sigma}{\omega_d}\sin\omega_d t) - e^{-\sigma t}(-\omega_d\sin\omega_d t + \sigma\cos\omega_d t)\\ &= \frac{\sigma^2}{\omega_d}e^{-\sigma t}\sin\omega_d t + \omega_d e^{-\sigma t}\sin\omega_d t\\ &= (\frac{\sigma^2}{\omega_d} + \omega_d)e^{-\sigma t}\sin\omega_d t \ .\end{aligned}$$

From this, we obtain $\sin\omega_d t = 0$, which occurs for $t = 0, \frac{\pi}{\omega_d}, \frac{2\pi}{\omega_d}, \cdots$. The first peak therefore occurs at the peak time

$$t_p = \frac{\pi}{\omega_d} \ .$$

Substituting this into the expression for $y(t)$, we obtain the value of the step response at the peak to be

$$\begin{aligned}M_p = y(\frac{\pi}{\omega_d}) &= 1 - e^{-\sigma\frac{\pi}{\omega_d}}(\cos\pi + \frac{\sigma}{\omega_d}\sin\pi)\\ &= 1 + e^{-\sigma\frac{\pi}{\omega_d}} \ .\end{aligned}$$

Substituting $y(\infty) = 1$, $\sigma = \zeta\omega_n$, $\omega_d = \omega_n\sqrt{1-\zeta^2}$, and the above expression for M_p into the definition of overshoot, we obtain

$$\begin{aligned}OS = \frac{1 + e^{-\sigma\frac{\pi}{\omega_d}} - 1}{1} &= e^{-\zeta\omega_n\frac{\pi}{\omega_n\sqrt{1-\zeta^2}}}\\ = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \ .\end{aligned}$$

8.1.3 Settling Time (t_s)

The settling time t_s is the time taken for the step response to stay within a certain percentage of its final value. This percentage will vary depending on the application. In this course, we will take 2% as a good measure. To calculate t_s, we note once again that the step response is

$$y(t) = 1 - e^{-\sigma t}\left(\cos\omega_d t + \frac{\sigma}{\omega_d}\sin\omega_d t\right) \ .$$

We can obtain a rough approximation of t_s by calculating the time at which the exponential term $e^{-\sigma t}$ becomes equal to 0.02 (since the cosine and sine terms oscillate forever between two fixed values):

$$e^{-\sigma t_s} \approx 0.02 \Rightarrow -\sigma t_s \approx \ln 0.02 \Rightarrow t_s \approx \frac{4}{\sigma} = \frac{4}{\zeta\omega_n} \ .$$

Summary. The expressions for the various performance metrics are:

$$\begin{aligned}
t_r &\approx \frac{2.16\zeta + 0.6}{\omega_n} \text{ or } t_r \approx \frac{1.8}{\omega_n} \\
M_p &= 1 + e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \text{ (for systems with DC gain of 1)} \\
OS &= e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \\
t_p &= \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} = \frac{\pi}{\omega_d} \\
t_s &\approx \frac{4}{\zeta\omega_n} \ .
\end{aligned}$$

Note that the expressions for t_r and t_s are only approximations; more detailed expressions for these quantities can be obtained by performing a more careful analysis. For our purposes, the above relationships will be sufficient to obtain intuition about the performance of second order systems. In particular, we note the following trends:

- As ω_n (bandwidth) increases (with ζ fixed), t_r decreases, OS stays the same, t_s decreases and t_p decreases.
- As ζ increases (with ω_n fixed), t_r stays approximately constant, OS decreases, t_s decreases and t_p increases.

We would typically like a small OS, small t_r and small t_s.

8.2 Choosing Pole Locations to Meet Performance Specifications

As we will see later in the course, we can design a controller for a given system so that the overall closed loop transfer function has poles at any desired locations in the complex plane. We will need to choose these pole locations so that the step response of the system achieves certain performance specifications. For example, suppose that we want the step response to have overshoot less than some value $\bar{OS}$, rise time less than some value $\bar{t}_r$, settling time less than some value $\bar{t}_s$, and peak time less than some $\bar{t}_p$. From the expressions given at the end of the previous section, these constraints on the step response can be translated into constraints on ζ and ω_n:

$$\begin{aligned}
e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} &\leq \bar{OS} \Rightarrow \text{ Find upper bound for } \zeta \text{ from this.}\\
\frac{1.8}{\omega_n} &\leq \bar{t}_r \Rightarrow \omega_n \geq \frac{1.8}{\bar{t}_r}\\
\frac{4}{\sigma} &\leq \bar{t}_s \Rightarrow \sigma \geq \frac{4}{\bar{t}_s}\\
\frac{\pi}{\omega_d} &\leq \bar{t}_p \Rightarrow \omega_d \geq \frac{\pi}{\bar{t}_p} \quad .
\end{aligned}$$

Noting that σ is the real part of the poles of the transfer function, ω_n is the distance of the complex poles from the origin, ω_d is imaginary part of the poles, and $\sin^{-1}\zeta$ is the angle between the imaginary axis and the vector joining the origin to the pole, we can formulate appropriate regions in the complex plane for the pole locations in order to satisfy the given specifications.

Figure 8.2: Pole Locations in the Complex Plane. (a) Overshoot. (b) Rise time. (c) Settling time. (d) Peak time. (e) Combined specifications.

Example. We would like our second order system to have a step response with overshoot $OS \leq 0.1$ and settling time $t_s \leq 2$. What is the region in the complex plane where the poles can be located?
Solution.

8.3 Effects of Poles and Zeros on the Step Response

So far, we have been studying second order systems with transfer functions of the form

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad .$$

This transfer function produced a step response $y(t)$, with Laplace Transform $Y(s) = H(s)\frac{1}{s}$.

Note: All of the analysis so far also holds if we consider the transfer function:

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

for some constant K. In this case, the step response is simply scaled by K, but the time characteristics (such as rise time, settling time, peak time and overshoot) of the response are not affected.

We will now consider what happens when we add zeros or additional poles to the system.

8.3.1 Effect of a Zero on the Step Response

Suppose that we modify the second order transfer function given above by adding a zero at $s = -z$, for some z. The modified transfer function is given by

$$H_z(s) = \frac{(\frac{1}{z}s + 1)\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \ .$$

Note that the reason for writing the zero term as $\frac{1}{z}s + 1$ instead of $s + z$ is to maintain a DC gain of 1 for the transfer function (just so that we can compare it to the original transfer function). We can split the above transfer function into the sum of two terms:

$$H_z(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} + \frac{1}{z}s\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = H(s) + \frac{1}{z}sH(s) \ ,$$

where $H(s)$ is the transfer function of the original system (without the zero). Denote the Laplace Transform of the step response for this system by $Y_z(s) = H_z(s)\frac{1}{s}$. Using the above decomposition of $H_z(s)$, we obtain

$$Y_z(s) = \left(H(s) + \frac{1}{z}sH(s)\right)\frac{1}{s} = H(s)\frac{1}{s} + \frac{1}{z}sH(s)\frac{1}{s} = Y(s) + \frac{1}{z}sY(s) \ .$$

Noting that the inverse Laplace Transform of $sY(s)$ is $\dot{y}$, we obtain

$$y_z(t) = y(t) + \frac{1}{z}\dot{y}(t) \ .$$

Thus the step response of the second order system with a zero at $s = -z$ is given by the step response of the original system plus a scaled version of the derivative of the step response of the original system. A sample plot for $z > 0$ (i.e., corresponding to the zero being in the OLHP) is shown in Fig. 8.3.

Figure 8.3: Step response of second order system with transfer function $H_z(s) = \frac{(\frac{1}{z}s+1)\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$, $z > 0$.

Note that as z increases (i.e., as the zero moves further into the left half plane), the term $\frac{1}{z}$ becomes smaller, and thus the contribution of the term $\dot{y}(t)$ decreases (i.e., the step response of this system starts to resemble the step response of the original system). From the above figure, the effect of a LHP zero is to increase the overshoot, decrease the peak time, and decrease the rise time; the settling time is not affected too much. In other words, a LHP zero makes the step response *faster*. One can also see this by thinking about the effect of the zero on the bandwidth of the system. Since the presence of the term $(\frac{1}{z}s + 1)$ in the numerator of the transfer function will only *increase* the magnitude of the Bode plot at frequencies above $\omega = z$, we see that adding a zero will generally increase the bandwidth of the system. This fits with our rule of thumb that a larger bandwidth corresponds to a faster response.

Now consider what happens if z is negative (which corresponds to the zero being in the ORHP). In this case, the derivative $\dot{y}(t)$ is actually subtracted from $y(t)$ to produce $y_z(t)$. A sample plot is shown in Fig. 8.4. Note that the response can actually go in the *opposite* direction before rising to its steady state value. This phenomenon is called **undershoot**.

Figure 8.4: Step response of second order system with transfer function $H_z(s) = \frac{(\frac{1}{z}s+1)\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$, $z < 0$.

Recall from Section 5.1.5 that zeros in the right half plane are called **nonminimum phase** – this is due to the fact the phase of the system has a large swing from its maximum and minimum values (as compared to the phase plot of the system with the same magnitude plot, but with all zeros in the OLHP). The effect of a RHP zero is to slow down the system, and perhaps introduce undershoot. However, the magnitude plot of the system is affected in the same way as with a LHP zero: the bandwidth increases. Thus, for a nonminimum phase zero, our rule of thumb about a larger bandwidth implying a faster response no longer holds.

It is also worth noting that if $z = 0$, the above analysis no longer holds directly. Adding a zero at $s = 0$ produces the transfer function $H_z(s) = sH(s)$, and the step response of this system is purely the derivative of the step response of the original system. However, the steady state value of this step response is zero, not 1 (note that this agrees with the DC gain of the new system).

8.3.2 Effect of Poles on the Step Response

A general n–th order system will have n poles in the complex plane. As we have already discussed, if any of these poles are in the RHP, the step response will be unstable (i.e., it will go to infinity). Consider the case of all poles being in the OLHP; a sample distribution is shown in Fig. 8.5.

Figure 8.5: Sample distribution of poles in OLHP, with dominant poles encircled.

From the partial fraction expansion, we know that a pole $-p$ contributes a term of the form e^{-pt} to the step response. If p is large, this exponential term dies out quickly. Suppose that the set of poles for a system can be divided into a cluster of poles that are closest to the origin, and another cluster that are very far away in comparison (e.g., 5 times further away). The poles that are closest to the origin are called the **dominant** poles of the system. The exponential terms in the step response corresponding to the far away poles will die out very quickly in relation to the exponential terms corresponding to the dominant poles. Thus, the system effectively behaves as a lower order system with only the dominant poles. This is one way to approximate a high order system by a lower order system (such as a first or second order system).

Since each additional pole contributes an additional exponential term that must die out before the system reaches its final value, each additional pole increases the rise time of the system. In other words, *adding a pole to the system makes* *the step response more sluggish.* Again, one can see this by looking at the bandwidth of the system: including an additional term of the form $(\frac{1}{p}s + 1)$ in the denominator of the transfer function causes the Bode plot to decrease faster after $\omega = p$, which generally decreases the bandwidth of the system.

Example. Plot the step response of the system with transfer function

$$H(s) = \frac{4}{(\frac{1}{p}s + 1)(s^2 + 2s + 4)} \ ,$$

for $p = 0.1, 2, 10, 100$.
Solution.

Summary. Based on our discussions, we can make the following observations:

- Adding a LHP zero to the transfer function makes the step response faster (decreases the rise time and the peak time) and increases the overshoot. The bandwidth of the system is increased.

- Adding a RHP zero to the transfer function makes the step response slower, and can make the response undershoot.[1] The bandwidth of the system is increased.

- Adding a LHP pole to the transfer function makes the step response slower. The bandwidth of the system is decreased.

- If the system has a cluster of poles and zeros that are much closer (5 times or more) to the origin than the other poles and zeros, the system can be approximated by a lower order system with only those dominant poles and zeros.

[1] Actually, one can show that the step response for a given (stable) linear time-invariant system will have undershoot if and only if the transfer function has an *odd* number of zeros in the ORHP. See the paper "On Undershoot and Nonminimum Phase Zeros" by M. Vidyasagar (*IEEE Transactions on Automatic Control*, vol. 31, no. 5, May 1986, p. 440) for the derivation of this result.

Chapter 9

Stability of Linear Time-Invariant Systems

Recall from our discussion on step responses that if the transfer function contains poles in the open right half plane, the response will go to infinity. However, if all poles of the transfer function are in the open left half plane, the response will settle down to the DC gain of the transfer function. To describe these characteristics of systems, we define the following terms:

- A signal $f(t)$ is **bounded** if there exists some constant M such that $|f(t)| \leq M$ for all t. It is called **unbounded** if it is not bounded.
- A system is **bounded-input bounded-output (BIBO) stable** if every bounded input leads to a bounded output. We will also refer to such systems simply as **stable**.
- A system is **unstable** if there is at least one bounded input that produces an unbounded output.

A linear time-invariant system is stable if all the poles of the transfer function are in the OLHP, and it is unstable if at least one pole is in the CRHP.

9.1 Pole-zero cancellations and stability

Consider the linear time-invariant system given by the transfer function

$$H(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} = \frac{N(s)}{D(s)} \ .$$

Recall that this system is stable if all of the poles are in the OLHP, and these poles are the roots of the polynomial $D(s)$. It is important to note that **one should not cancel any common poles and zeros of the transfer function** before checking the roots of $D(s)$. Specifically, suppose that both of the polynomials $N(s)$ and $D(s)$ have a root at $s = a$, for some complex (or real) number a. One must *not* cancel out this common zero and pole in the transfer function before testing for stability. The reason for this is that, even though the pole will not show up in the response to the input, it will still appear as a result of any initial conditions in the system, or due to additional inputs entering the system (such as disturbances). If the pole and zero are in the CRHP, the system response might blow up due to these initial conditions or disturbances, even though the input to the system is bounded, and this would violate BIBO stability.

To see this a little more clearly, consider the following example. Suppose the transfer function of a linear system is given by

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

Noting that $s^2 + 2s - 3 = (s+3)(s-1)$, suppose we decided to cancel out the common pole and zero at $s = 1$ to obtain

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

Based on this transfer function, we might (erroneously) conclude that the system is stable, since it only has a pole in the OLHP. What we should actually do is look at the original denominator $D(s)$, and correctly conclude that the system is unstable because one of the poles is in the CRHP. To see why the pole-zero cancellation hides instability of the system, first write out the differential equation corresponding to the transfer function to obtain

$$\ddot{y} + 2\dot{y} - 3y = \dot{u} - u \ .$$

Take the Laplace Transform of both sides, taking initial conditions into account:

$$s^2Y(s) - sy(0) - \dot{y}(0) + 2sY(s) - 2y(0) - 3Y(s) = sU(s) - u(0) - U(s) \ .$$

Rearrange this equation to obtain

$$Y(s) = \underbrace{\frac{s-1}{s^2+2s-3}}_{H(s)}U(s) + \frac{s+2}{s^2+2s-3}y(0) + \frac{1}{s^2+2s-3}\dot{y}(0) - \frac{1}{s^2+2s-3}u(0) \ .$$

Note that the denominator polynomial in each of the terms on the right hand sides is equal to $D(s)$ (the denominator of the transfer function). For simplicity, suppose that $y(0) = y_0$ (for some real number y_0), $\dot{y}(0) = 0$ and $u(0) = 0$. The partial fraction expansion of the term $\frac{s+2}{s^2+2s-3}y_0$ is given by

$$\frac{s+2}{s^2+2s-3}y_0 = \frac{y_0}{4}\left(\frac{1}{s+3} + \frac{1}{s-1}\right) \ ,$$

and this contributes the term $\frac{y_0}{4}\left(e^{-3t}+e^{t}\right), t \geq 0$, to the response of the system. Note that the e^t term blows up, and thus the output of the system blows up if y_0 is not zero, even if the input to the system is bounded.

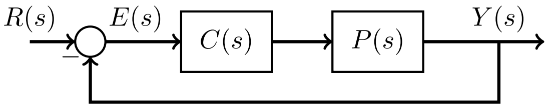
If all poles of the transfer function are in the OLHP (before any pole-zero cancellations), all initial conditions will decay to zero, and not cause the output of the system to go unbounded.

The above example demonstrates the following important fact:

Stability of a transfer function must be checked **without** canceling any common poles and zeros from the transfer function. In particular, systems with unstable pole-zero cancellations are **unstable**.

9.2 Stability of the Unity Feedback Loop

Consider the standard feedback structure shown below.



Suppose that we write $P(s) = \frac{n_p(s)}{d_p(s)}$ and $C(s) = \frac{n_c(s)}{d_c(s)}$ for some polynomials $n_p(s), d_p(s), n_c(s), d_c(s)$. The transfer function from r to y is given by

$$H(s) = \frac{P(s)C(s)}{1+P(s)C(s)} = \frac{\frac{n_p(s)n_c(s)}{d_p(s)d_c(s)}}{1+\frac{n_p(s)n_c(s)}{d_p(s)d_c(s)}} = \frac{n_p(s)n_c(s)}{d_p(s)d_c(s)+n_p(s)n_c(s)}.$$

The denominator of the above transfer function is called the *characteristic polynomial* of the closed loop system, and we have the following result.

The unity feedback system is stable if and only if all roots of the characteristic polynomial $d_p(s)d_c(s) + n_p(s)n_c(s)$ are in the OLHP.

Note that the above test captures unstable pole/zero cancellations: if there is an unstable pole $s = a$ in $d_p(s)$ or $d_c(s)$, and that same root appears in either $n_c(s)$ or $n_d(s)$, then $s = a$ would be a root of $d_p(s)d_c(s)+n_p(s)n_c(s)$ and would thus cause the characteristic polynomial to fail the test for stability.

9.3 Tests for Stability

Consider a general transfer function of the form

$$H(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} = \frac{N(s)}{D(s)} \ .$$

We would like to determine whether all poles of this transfer function are in the OLHP. One way to do this would be to actually find the poles of the system (by finding the roots of $D(s)$). However, finding the roots of a high-degree polynomial can be complicated, especially if some of the coefficients are symbols rather than numbers (this will be the case when we are designing controllers, as we will see later). Furthermore, note that we do not need the actual values of the poles in order to determine stability – we only need to know if all poles are in the OLHP. Can we determine this from the coefficients of the polynomial?

9.3.1 A Necessary Condition for Stability

Suppose that the polynomial $D(s)$ has roots $-p_1, -p_2, \ldots, -p_n$ (for simplicity, assume that these are all real, but the following analysis also holds if some of the roots are complex). The polynomial can then be written as

$$D(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0 = (s+p_1)(s+p_2)\ldots(s+p_n) \ .$$

How do the coefficients $a_0, a_1, \ldots, a_{n-1}$ relate to $p_1, p_2, \ldots, p_n$? To see the pattern, consider the following cases:

$$\begin{aligned}
(s+p_1)(s+p_2) &= s^2 + (p_1+p_2)s + p_1p_2 \\
(s+p_1)(s+p_2)(s+p_3) &= s^3 + (p_1+p_2+p_3)s^2 + (p_1p_2+p_1p_3+p_2p_3)s \\
&\quad + p_1p_2p_3 \\
(s+p_1)(s+p_2)(s+p_3)(s+p_4) &= s^4 + (p_1+p_2+p_3+p_4)s^3 \\
&\quad + (p_1p_2+p_1p_3+p_1p_4+p_2p_3+p_2p_4+p_3p_4)s^2 \\
&\quad + (p_1p_2p_3+p_1p_2p_4+p_1p_3p_4+p_2p_3p_4)s \\
&\quad + p_1p_2p_3p_4 \\
&\vdots
\end{aligned}$$

Based on the above examples, we see that:

- a_{n-1} is the sum of all the p_i's.
- a_{n-2} is the sum of all products of the p_i's taken two at a time.
- a_{n-3} is the sum of all products of the p_i's taken three at a time.

$\vdots$

- a_0 is the product of all the p_i's.

Now, suppose that all roots are in the OLHP (i.e., $p_1 > 0, p_2 > 0, \ldots, p_n > 0$). This means that $a_0 > 0, a_1 > 0, \ldots, a_{n-1} > 0$ as well. This leads us to the following conclusion.

> The polynomial $D(s) = s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0$ has all roots in the OLHP *only if* all of the coefficients $a_0, a_1, \ldots, a_{n-1}$ are positive.

Note that the above condition is *necessary*, but it is not *sufficient* in general. We will see this in the following examples.

Examples.

- $D(s) = s^3 - 2s^2 + s + 1$:

- $D(s) = s^4 + s^2 + s + 1$:

- $D(s) = s^3 + 2s^2 + 2s + 1$:

- $D(s) = s^3 + 2s^2 + s + 12$:

In the last example above, the polynomial can be factored as $D(s) = (s + 3)(s^2 - s + 4)$, and the second factor has roots in the CRHP. This shows that a polynomial having all coefficients positive does not mean that all of the roots are in the OLHP. Only the converse is true: if all of the roots are in the OLHP, then all of the coefficients are positive.

Although the test for positive coefficients allows us to immediately determine whether the roots of a polynomial are *not* stable, we need a way to conclusively determine whether all of the roots *are* stable.

9.3.2 A Necessary and Sufficient Condition: Routh-Hurwitz Test

The problem of determining how many roots of the polynomial $D(s) = a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0$ are in the ORHP was studied by E. J. Routh in the late 1800's. He devised the following technique for solving this problem (we will assume that the polynomial $D(s)$ has no roots on the imaginary axis, so it is sufficient to check for poles in the ORHP, as opposed to CRHP).

First, construct a **Routh array**:

$$\begin{array}{c|cccc}
s^n & a_n & a_{n-2} & a_{n-4} & \cdots \\
s^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & \cdots \\
\hline
s^{n-2} & b_1 & b_2 & b_3 & \cdots \\
s^{n-3} & c_1 & c_2 & c_3 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots \\
s^2 & * & * & & \\
s^1 & * & & & \\
s^0 & * & & &
\end{array}$$

The first two rows of this array contain the coefficients of the polynomial. The numbers $b_1, b_2, b_3, \ldots$ on the third row are defined as:

$$\begin{aligned}
b_1 &= -\frac{1}{a_{n-1}}\begin{vmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{vmatrix} = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}} \quad , \\
b_2 &= -\frac{1}{a_{n-1}}\begin{vmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{vmatrix} = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}} \quad , \\
b_3 &= -\frac{1}{a_{n-1}}\begin{vmatrix} a_n & a_{n-6} \\ a_{n-1} & a_{n-7} \end{vmatrix} = \frac{a_{n-1}a_{n-6} - a_n a_{n-7}}{a_{n-1}} \quad , \\
&\vdots
\end{aligned}$$

Notice the pattern: the i–th element on the third row is obtained by taking the negative determinant of the matrix consisting of the first column and the $(i+1)$–th column in the first two rows, divided by the first element in the second row. The third row will have one less element than the first two rows.

Similarly, the numbers $c_1, c_2, c_3, \ldots$ on the fourth row are defined as:

$$\begin{aligned}
c_1 &= -\frac{1}{b_1}\begin{vmatrix} a_{n-1} & a_{n-3} \\ b_1 & b_2 \end{vmatrix} = \frac{b_1 a_{n-3} - a_{n-1}b_2}{b_1} \quad , \\
c_2 &= -\frac{1}{b_1}\begin{vmatrix} a_{n-1} & a_{n-5} \\ b_1 & b_3 \end{vmatrix} = \frac{b_1 a_{n-5} - a_{n-1}b_3}{b_1} \quad , \\
c_3 &= -\frac{1}{b_1}\begin{vmatrix} a_{n-1} & a_{n-7} \\ b_1 & b_4 \end{vmatrix} = \frac{b_1 a_{n-7} - a_{n-1}b_4}{b_1} \quad , \\
&\vdots
\end{aligned}$$

Again, **notice the pattern**: the i–th element on the fourth row is obtained by taking the negative determinant of the matrix consisting of the first column and the $(i+1)$–th columns of the preceding two rows, divided by the first element in the immediately preceding row.

We continue this process until the $(n+1)$–th row (corresponding to s^0 in the Routh array), which will have only one entry. After this process is complete, we can use the following result[1] to check stability of $D(s)$.

The number of sign changes in the first column of the Routh array indicates the number of roots that are in the ORHP. All roots are in the OLHP *if and only if* there are no sign changes in the first column (i.e., either all entries are positive, or all are negative).

Note: The above algorithm requires us to divide certain determinants by the first entry in each row of the Routh array. If this entry is zero, but all other entries in the row are nonzero, we can replace the zero by a small positive value ϵ, and continue as before (the above result still holds in this case). On the other hand, if an entire row of the Routh array is zero, this signifies the potential existence of poles on the imaginary axis, and a slightly more complex procedure will have to be followed. Be aware that these special cases exist, but we will not deal with them in this course.

Example. Determine whether the polynomial $D(s) = s^4 + 3s^3 + 2s^2 + 2s + 9$ has all roots in the OLHP.
Solution.

Example. Determine whether the polynomial $D(s) = s^4 + 5s^3 + 9s^2 + 5s + 2$

[1] A relatively simple proof of this result can be found in the paper "Elementary Proof of the Routh-Hurwitz Test" by G. Meinsma (*Systems and Control Letters*, vol. 25, no. 4, 1995, pp. 237-242).

has all roots in the OLHP.
Solution.

9.3.3 Testing For Degree of Stability

There may be times when we would like to test a polynomial to see if the real parts of all of its roots are less than a certain value; so far, we have been considering this value to be zero (i.e., we have been testing to see if all roots lie in the OLHP). Suppose that we would like to see if all roots of a polynomial $D(s)$ have real part less than λ. Consider the polynomial $\bar{D}(s) = D(s + \lambda)$. It is easy to see that the roots of $\bar{D}(s)$ are the roots of $D(s)$ shifted by λ: if $s = a$ is a root of $D(s)$, then $s = a - \lambda$ is a root of $\bar{D}(s)$. Thus, all roots of $D(s)$ have real parts less than λ if and only if all roots of $\bar{D}(s)$ are in the OLHP, and we can use the Routh-Hurwitz test on $\bar{D}(s)$ to see whether all roots of $D(s)$ lie to the left of λ.

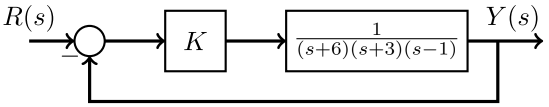
Example. Determine whether the polynomial $D(s) = s^3 + 3s^2 + 3s + 9$ has all roots with real parts less than -1.

Solution.

9.3.4 Testing Parametric Stability with Routh-Hurwitz

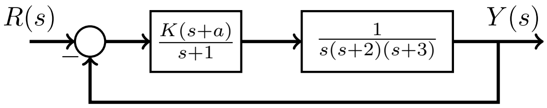
In control system design, we will frequently run across cases where some of the coefficients of the polynomial are parameters for us to design or analyze (such as control gains, or unknown values for system components). We can use the Routh-Hurwitz test to determine ranges for these parameters so that the system will be stable.

Example. In the feedback control loop shown below, determine the range of values for K for which the closed loop transfer function (from $R(s)$ to $Y(s)$) will be stable.



Solution.

Example. In the feedback loop shown below, determine the values of K and a for which the closed loop system is stable.



Solution.

Chapter 10

Properties of Feedback

With the tools to analyze linear systems in hand, we now turn our attention to feedback control. Recall the unity feedback control loop that we looked at earlier:

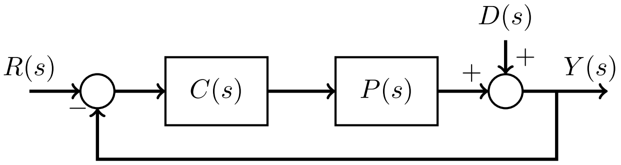


Figure 10.1: Feedback Control.

In the above figure, $P(s)$ is the plant, $C(s)$ is the controller, r is the reference signal, y is the output of the system, and d is a **disturbance** affecting the control loop (e.g., wind on an airplane, noise in a sensor, faults in an electrical grid, etc.).

Recall from Chapter 1 that there are three main properties that a good control system should have:

- **Tracking.** The output of the system should behave like the reference signal. This property is studied by examining the transfer function from the reference input to the output.
- **Disturbance Rejection.** The disturbances should affect the output as little as possible. This property is studied by examining the transfer function from the disturbance to the output.
- **Robustness.** The output should track the reference signal even if the plant model is not exactly known, or changes slightly. This property is studied by examining the sensitivity of the transfer function to perturbations in the plant, as we show below.

Sensitivity. Let $T_{ry}(s)$ denote the transfer function of the control system from the reference r to the output y. Now suppose that we allow the plant $P(s)$ to change by a small amount $\delta P(s)$ to become the new plant $\bar{P}(s) = P(s) + \delta P(s)$. This will cause the transfer function $T_{ry}(s)$ to also change by a small amount $\delta T_{ry}(s)$, to become $\bar{T}_{ry}(s) = T_{ry}(s) + \delta T_{ry}(s)$. The question is: how does $\delta T_{ry}(s)$ compare to $\delta P(s)$? More specifically, the **sensitivity** is defined as the *fractional* (or percentage) change in the transfer function as related to the fractional change in the plant model:

$$\boxed{S(s) = \frac{\frac{\delta T_{ry}(s)}{T_{ry}(s)}}{\frac{\delta P(s)}{P(s)}} = \frac{\delta T_{ry}(s)}{\delta P(s)} \frac{P(s)}{T_{ry}(s)} \ .}$$

Note that for small perturbations $\delta P(s)$ and $\delta T_{ry}(s)$, the expression $\frac{\delta T_{ry}(s)}{\delta P(s)}$ is the derivative of $T_{ry}(s)$ with respect to $P(s)$. In order to have good robustness, **we want the sensitivity to be as small as possible**.

We will analyze the tracking, disturbance rejection and robustness properties for both a *feedforward* control configuration and a *feedback* control configuration, and see why feedback is an important concept for control system design.

10.1 Feedforward Control

Recall from the first lecture that feedforward control does not make use of measurements of the system output in order to supply the input. The block diagram for feedforward control is shown in Fig. 10.2.

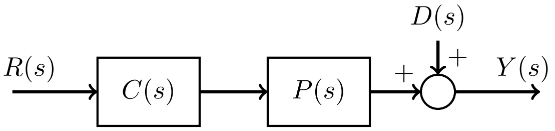


Figure 10.2: Feedforward Control.

We can examine how well feedforward control satisfies the properties listed above.

- **Tracking.** The transfer function from the reference to the output is obtained by assuming that the disturbance is not present (i.e., take $d = 0$), and is given by

$$T_{ry}(s) = \frac{Y(s)}{R(s)} = P(s)C(s) \ .$$

We can potentially get perfect tracking by choosing $C(s) = P(s)^{-1}$, in which case we have $Y(s) = R(s)$. Note, however, that "inverting" the plant in this way may not always be possible in practice. Setting the implementation details aside for now, we note that **feedforward control can potentially provide perfect tracking.**

- **Disturbance Rejection.** The transfer function from the disturbance to the output is obtained by assuming that the reference signal is not present (i.e., $r = 0$), and is given by

  $$T_{dy}(s) = \frac{Y(s)}{D(s)} = 1 \ .$$

  Thus, the output is completely affected by the disturbance. Furthermore, the feedforward control scheme provides us with no way to reduce this influence. Thus, **feedforward control is bad at rejecting disturbances.**

- **Robustness.** To examine how robust feedforward control is to variations in the plant, we examine the sensitivity

  $$S(s) = \frac{\delta T_{ry}(s)}{\delta P(s)} \frac{P(s)}{T_{ry}(s)} \ .$$

  Since $T_{ry}(s) = P(s)C(s)$, we have

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

  This shows that the transfer function is 100% sensitive to changes in the plant, which is **bad for robustness**.

We can summarize the above findings as follows.

Feedforward control is **potentially good for tracking**, **bad for disturbance rejection** and **bad for robustness**.

10.2 Feedback Control

Now consider the feedback control scheme shown in Fig. 10.1. We will study how well this control scheme satisfies the properties described earlier.

- **Tracking.** The transfer function from the reference to the output is obtained by assuming that the disturbance is not present (i.e., take $d = 0$), and is given by

  $$T_{ry}(s) = \frac{Y(s)}{R(s)} = \frac{P(s)C(s)}{1 + P(s)C(s)} \ .$$

We would like to make this transfer function as close to 1 as possible, so that we can get $Y(s) = R(s)$. One way to accomplish this would be to choose $C(s) = K$ for some very large constant K (this is called "high gain control"). In this case, the magnitude of $P(s)C(s)$ will be large for all s, and thus $T_{ry}(s)$ is approximately equal to 1. Thus, **feedback control can provide good tracking.**

- **Disturbance Rejection.** The transfer function from the disturbance to the output is obtained by assuming that the reference signal is not present (i.e., $r = 0$), and is given by

$$T_{dy}(s) = \frac{Y(s)}{D(s)} = \frac{1}{1 + P(s)C(s)} \quad .$$

We would like to make this transfer function as small as possible (so that the disturbance does not affect the output). Once again, suppose we choose $C(s) = K$ for some very large constant K. In this case, the denominator of $T_{dy}(s)$ becomes very large, and so $T_{dy}(s)$ becomes small. Thus, **feedback control can be good at rejecting disturbances.**

- **Robustness.** To examine how robust feedback control is to variations in the plant, we examine the sensitivity

$$S(s) = \frac{\delta T_{ry}(s)}{\delta P(s)} \frac{P(s)}{T_{ry}(s)} \quad .$$

Since $T_{ry}(s) = \frac{P(s)C(s)}{1+P(s)C(s)}$, we have

$$S(s) = \frac{C(s)}{(1 + P(s)C(s))^2} \frac{P(s)}{\frac{P(s)C(s)}{1+P(s)C(s)}} = \frac{1}{1 + P(s)C(s)} \quad .$$

Once again, choosing $C(s) = K$, for a large constant K, makes $S(s)$ very small. Thus, **feedback control is robust to variations in the plant**.

We can summarize the above findings as follows.

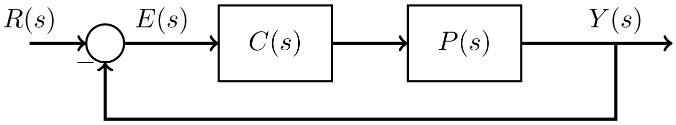
Feedback control is **good for tracking**, **good for disturbance rejection** and **good for robustness**.

In addition to the above benefits, we will see later that feedback control can also stabilize an unstable plant (whereas feedforward control cannot). Also, it is worth noting that although we used high gain control to show that feedback provides good tracking, disturbance rejection and robustness, it is not the only option (and sometimes not even the best option). In practice, high gains are hard to implement in hardware (due to saturation effects and physical limitations). Furthermore, in some cases, high gain can even destabilize a stable system (we will see this when we study root-locus methods). We will be studying ways to design more sophisticated feedback controllers to get around these problems.

Chapter 11

Tracking of Reference Signals

Recall the basic feedback control loop that we looked at earlier:



In the above figure, $P(s)$ is the plant, $C(s)$ is the controller, r is the reference signal, y is the output of the system, and e is the error (i.e., $e = r - y$). We will neglect disturbances for now. Note that the signal y is directly subtracted from the reference signal in this feedback loop, so this configuration is called **unity feedback**. The transfer function from r to y is

$$T_{ry}(s) = \frac{Y(s)}{R(s)} = \frac{P(s)C(s)}{1 + P(s)C(s)} \quad .$$

Recall that the product $P(s)C(s)$ is called the **forward gain** of the feedback loop. We can always write $P(s)C(s)$ as

$$P(s)C(s) = \frac{a(s)}{s^q b(s)} \quad ,$$

for some polynomials $a(s)$ and $b(s)$, and some nonnegative integer q (this is the number of poles at the origin in the product $P(s)C(s)$). The polynomial $b(s)$ has no roots at $s = 0$ (otherwise this root can be grouped into the term s^q); this means that $b(0) \neq 0$. The transfer function then becomes

$$T_{ry}(s) = \frac{\frac{a(s)}{s^q b(s)}}{1 + \frac{a(s)}{s^q b(s)}} = \frac{a(s)}{s^q b(s) + a(s)} \quad .$$

Example. Suppose $P(s) = \frac{s+1}{s^2}$ and $C(s) = \frac{12(s+2)}{s(s+4)}$. What are $a(s), b(s)$ and q? What is the transfer function?
Solution.

We'll assume that $C(s)$ is chosen so that the closed loop system is stable (otherwise, we do not have any hope of tracking any signal). This means that all roots of $s^q b(s) + a(s)$ are in the OLHP. We will now examine how well this feedback loop tracks certain types of reference signals.

11.1 Tracking and Steady State Error

We will start by examining the **steady state error** of the above feedback loop to reference signals of the form $r(t) = t^m, t \geq 0$, where m is some nonnegative integer. The reference $r(t)$ is a **step** input when $m = 0$, a **ramp** input when $m = 1$, and a **parabolic** input when $m = 2$. Note from the Laplace Transform table that the Laplace transform of $r(t) = t^m, t \geq 0$ is $R(s) = \frac{m!}{s^{m+1}}$.

Figure 11.1: The reference input $r(t) = t^m, t \geq 0$, for $m = 0, 1, 2$.

To determine how well the system tracks (or follows) the reference inputs, we will examine the error signal $e = r - y$. Specifically, the Laplace Transform of $e(t)$ is given by:

$$E(s) = R(s) - Y(s) = R(s) - P(s)C(s)E(s) \ ,$$

from which we obtain

$$E(s) = \frac{1}{1 + P(s)C(s)}R(s) = m!\frac{1}{(1 + P(s)C(s))s^{m+1}} \ .$$

Recall that the Final Value Theorem states that if all poles of $sE(s)$ are in the OLHP, then the signal $e(t)$ settles down to some finite steady state value. From the above expression for $E(s)$, we have

$$sE(s) = m!\frac{1}{(1 + P(s)C(s))s^{m}} \ .$$

Note that, as written, $sE(s)$ seems to have m poles at the origin. Note from our earlier discussion, however, that $P(s)C(s)$ has q poles at the origin, so that we can write $P(s)C(s) = \frac{a(s)}{s^q b(s)}$, for some polynomials $a(s)$ and $b(s)$. The expression for $sE(s)$ can then be written as

$$sE(s) = m!\frac{1}{(1 + \frac{a(s)}{s^q b(s)})s^m} = m!\frac{b(s)}{(s^q b(s) + a(s))s^{m-q}} \ .$$

Recall that the polynomial $s^q b(s) + a(s)$ has all roots in the OLHP (by our assumption of stability). We thus only have to check for poles at the origin (given by the quantity s^{m-q}). We consider three different cases:

- If $q > m$, the function $sE(s)$ will have all poles in the OLHP, and $q - m$ zeros at the origin. The steady state error is obtained from the Final Value Theorem as

$$e_{ss} = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) = m!\frac{0^{q-m}b(0)}{0^q b(0) + a(0)} = 0 \ .$$

  Thus, if $q > m$, we have **perfect steady state tracking** (i.e., the steady state error is zero).

- If $q = m$, the function $sE(s)$ has all poles in the OLHP, and no zeros at the origin. From the Final Value Theorem, we have

$$\begin{aligned} e_{ss} = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) &= \lim_{s\to 0} m!\frac{b(s)}{s^q b(s) + a(s)} \\ &= \lim_{s\to 0} m!\frac{1}{s^q + \frac{a(s)}{b(s)}} \\ &= \lim_{s\to 0} m!\frac{1}{s^q + s^q P(s)C(s)} \ , \end{aligned}$$

which is some nonzero constant.[1] Note that if $q = 0$, the steady state error is $e_{ss} = \lim_{s \to 0} m! \frac{1}{1+P(s)C(s)}$, and if $q > 0$, the steady state error is $e_{ss} = \lim_{s \to 0} m! \frac{1}{s^q P(s)C(s)}$. Thus, we have **steady state tracking to within a constant finite error.**

- If $q < m$, the function $sE(s)$ will have $m - q$ poles at the origin. Thus the signal $e(t)$ blows up as $t \to \infty$, and there is no steady state value. In other words, the system output **does not track the reference input at all.**

Summary. Suppose $P(s)C(s)$ has q poles at the origin (q is a nonnegative integer), and suppose the reference input to the unity feedback loop is $r(t) = t^m$, $t \geq 0$, where m is a nonnegative integer.

- If $q > m$, the output $y(t)$ will track the reference $r(t)$ perfectly in steady state (i.e., the steady state error will be zero).
- If $q = m$, the output $y(t)$ will track the reference $r(t)$ to within a constant (finite) steady state error.
- If $q < m$, the output $y(t)$ will not track the reference $r(t)$ at all.

Note that if a linear system can track a signal $t^m, t \geq 0$, then it can also track any polynomial of degree m or less (by linearity).

System type. The above results indicate that the number of poles at the origin in $P(s)C(s)$ determines the type of reference inputs that the closed loop system can track. Thus, the integer q is called the **system type**. Specifically, a system of type q can track reference signals that are polynomials of degree q or less to within a constant finite steady state error.

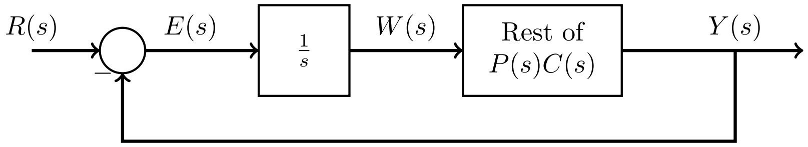
Note: It does not matter whether the poles in $P(s)C(s)$ come from the plant $P(s)$ or from the controller $C(s)$. The only thing that matters is how many poles their *product* has. We can therefore use this fact to construct controllers with a certain number of poles in order to track certain types of reference signals, even if the plant does not have the required number of poles. We will see this in the next lecture.

[1] Do not try to memorize this. Instead, always just derive the tracking error using first principles, first calculating $E(s)$ in terms of $T_{ry}(s)$ and $R(s)$, and then applying the final value theorem.

Example. Consider the unity feedback loop with $C(s) = \frac{12(s+2)}{s(s+4)}$ and $P(s) = \frac{s+1}{s^2}$. What is the system type? What is the steady state tracking error for the signals $r(t) = \mathbf{1}(t)$, $r(t) = t\mathbf{1}(t)$, $r(t) = t^2\mathbf{1}(t)$, $r(t) = t^3\mathbf{1}(t)$, and $r(t) = t^4\mathbf{1}(t)$?
Solution.

Example. Suppose that $C(s) = K$ (for some positive constant K), and $P(s) = \frac{1}{s+1}$. What is the system type? What is the steady state tracking error for the signals $r(t) = \mathbf{1}(t)$, $r(t) = t\mathbf{1}(t)$ and $r(t) = t^2\mathbf{1}(t)$?
Solution.

To see why having an integrator guarantees perfect tracking for a step input (if the closed loop system is stable), rearrange the closed loop system as follows:



In the above diagram, we have simply pulled the integrator out of the product $P(s)C(s)$, and denoted the signal at the output of the integrator by $W(s)$ (or $w(t)$ in the time-domain). Now suppose that the closed loop system is stable (note that this is a necessary assumption); in this case, all of the signals in the system will settle down to some steady state values when $r(t)$ is a unit step input. This includes the signal $w(t)$, which is related to $e(t)$ as

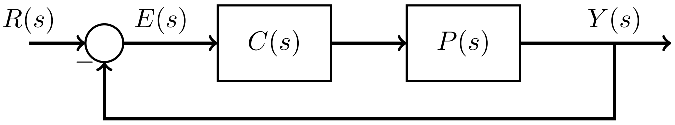
$$W(s) = \frac{1}{s}E(s) \Leftrightarrow w(t) = \int_0^t e(\tau)d\tau.$$

If $e(t)$ settles down to some nonzero value, the above integral will become unbounded, and thus $w(t)$ will not settle down to some steady state value, contradicting the fact that the system is stable. Thus the only way for all signals to have settled to a steady state value (which is guaranteed by stability) is if $e(t) \rightarrow 0$. An alternative way to see this is to note that if $w(t)$ settles down to a steady state value, then $\dot{w}(t) = e(t) = 0$.

This is an example of what is known as the **internal model principle**: if we wish to perfectly track a signal of a certain form, we should include a model of that signal inside our feedback loop.

Chapter 12

PID Control



So far, we have examined the benefits of feedback control, and studied how the poles of $P(s)C(s)$ affect the ability of the control system to track reference inputs. We will now study a type of controller $C(s)$ that is commonly used in practice, called a **proportional-integral-derivative (PID) controller**. To develop this controller, we will assume that the plant is a second order system of the form

$$P(s) = \frac{b_0}{s^2 + a_1 s + a_0} \quad .$$

Note that the transfer function from r to y for the above feedback loop is given by

$$T_{ry}(s) = \frac{P(s)C(s)}{1 + P(s)C(s)} \quad .$$

12.1 Proportional (P) Control

We start with a simple controller of the form $C(s) = K_P$, where K_P is a constant that we will choose. In this case, the input to the plant is simply $u(t) = K_P e(t)$, which is proportional to the error. Thus, this type of controller is called a **proportional controller**.

With this controller, the transfer function from r to y in the feedback control system becomes

$$T_{ry}(s) = \frac{K_P b_0}{s^2 + a_1 s + (a_0 + K_P b_0)} \quad .$$

Recall that the poles of this transfer function dictate how the system behaves to inputs. In particular, we would like to ensure that the system is stable (i.e., all poles are in the OLHP). Since the gain K_P affects one of the coefficients in the denominator polynomial, it can potentially be used to obtain stability.

Example. Suppose $P(s) = \frac{1}{s^2+3s-1}$. Can we stabilize this plant with proportional control?
Solution.

Example. Suppose $P(s) = \frac{1}{s^2-3s-1}$. Can we stabilize this plant with proportional control?
Solution.

The above examples demonstrate that simple proportional control can stabilize some plants, but not others.

Another benefit of proportional control is that it can potentially be used to speed up the response of the system. Recall the standard second order system had a denominator of the form $s^2 + 2\zeta\omega_n s + \omega_n^2$, and the larger ω_n is, the faster the system responds. In the closed loop transfer function $T_{ry}(s)$ above, the term ω_n^2 is given by $a_0 + K_P b_0$, and thus we can potentially make ω_n very large by choosing K_P to be very large, thereby speeding up the system.

Now let's consider tracking. Recall from the previous lecture that in order to track a step input perfectly (i.e., with zero steady state error), the system must be of type 1 (a type 0 system would track a step within a finite steady state error). If $C(s) = K_P$ and $P(s) = \frac{b_0}{s^2+a_1 s+a_0}$, the system would only be of type 0 (if a_0 is not zero), and thus we will not be able to track a step perfectly. To rectify this, we will have to add an integrator to the controller in order to make the system type 1.

12.2 Proportional-Integral (PI) Control

To obtain perfect tracking for step inputs, we will introduce an integrator into the controller (i.e., we will add a pole at the origin) in order to ensure that the system will be of type 1. The controller thus becomes

$$C(s) = K_P + \frac{K_I}{s} \quad .$$

In the time-domain, this corresponds to the input to the plant being chosen as

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau \quad ,$$

and thus this is called a **proportional-integral controller**. With this controller, the transfer function from r to y is

$$T_{ry}(s) = \frac{\frac{b_0}{s^2+a_1 s+a_0} \frac{K_P s+K_I}{s}}{1 + \frac{b_0}{s^2+a_1 s+a_0} \frac{K_P s+K_I}{s}} = \frac{b_0(K_P s + K_I)}{s^3 + a_1 s^2 + (a_0 + K_P b_0)s + K_I b_0} \quad .$$

Note that we now have a third order system. Two of the coefficients of the denominator polynomial can be arbitrarily set by choosing K_P and K_I appropriately. Unfortunately, we still have no way to stabilize the system if $a_1 < 0$ (recall that for stability, all coefficients must be positive). Even if the system is stable with the given value of a_1, we might want to be able to choose better pole locations for the transfer function in order to obtain better performance. To do this, we add one final term to the controller.

12.3 Proportional-Integral-Derivative (PID) Control

Consider the PI controller from the last section, and add a term that corresponds to the *derivative* of the error. The controller with this additional term has the form

$$C(s) = K_P + \frac{K_I}{s} + K_D s \quad .$$

In the time-domain, the input to the plant due to this controller is given by

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \dot{e}(t) \ ,$$

and so this controller is called a **proportional-integral-derivative controller**. The transfer function from r to y is

$$\begin{aligned} T_{ry}(s) &= \frac{\frac{b_0}{s^2+a_1 s+a_0} \frac{K_P s+K_I+K_D s^2}{s}}{1 + \frac{b_0}{s^2+a_1 s+a_0} \frac{K_P s+K_I+K_D s^2}{s}} \\ &= \frac{b_0(K_P s + K_I + K_D s^2)}{s^3 + (a_1 + K_D b_0)s^2 + (a_0 + K_P b_0)s + K_I b_0} \ . \end{aligned}$$

Note that we are now able to arbitrarily set all coefficients of the denominator polynomial, via appropriate choices of K_P, K_I and K_D. Thus we can now guarantee stability (only for a second order plant, though), good transient behavior, and perfect tracking!

Example. Consider the plant $P(s) = \frac{1}{s^2-3s-1}$. Design a PID controller so that the closed loop system has perfect tracking for a step input, and has poles at $s = -5, -6, -7$.

Solution.

12.4 Implementation Issues

PID control is extremely common in industry, as it is easy to design – we just have to "tune the knobs" corresponding to the gains K_P, K_I and K_D appropriately. However, it is worth noting that it may not be desirable to implement the controller as given above; in practice, all signals will contain high frequency noise, and differentiating noise will once again create signals with large magnitudes. To avoid this, the derivative term $K_D s$ is usually implemented in conjunction with a low pass filter of the form $\frac{1}{\tau s+1}$, for some small τ. This has the effect of attenuating the high frequency noise entering the differentiator, and produces the controller

$$C(s) + K_P + \frac{K_I}{s} + \frac{K_D s}{\tau s + 1} \ .$$

We will be seeing controllers of this form frequently in the second half of the course.

Chapter 13

Root Locus

Based on our discussion so far, we know that the response of a linear system to an input is dictated by the location of the poles of the transfer function. For example, the response will be unstable if there are any poles in the CRHP, or may contain oscillations if the poles appear in complex conjugates. We have also seen that feedback control can be used to move the poles of a closed loop system: by choosing the controller gain appropriately, one can potentially stabilize unstable systems (and perhaps even destabilize stable systems). In this section of the course, we will examine in more detail how the poles of a transfer function vary in the complex plane in response to changes in a certain parameter. We will begin with some examples.

Example. Consider the unity feedback loop with $C(s) = K$ and $P(s) = \frac{1}{s-3}$. How does the pole of the closed loop system vary with K (for $K \geq 0$)?
Solution.

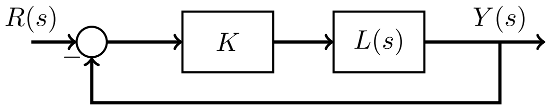
Example. Consider the unity feedback loop with $C(s) = K$ and $P(s) = \frac{1}{s^2+2s}$. How do the poles of the closed loop system vary with K (for $K \geq 0$)?
Solution.

Example. Consider the closed loop transfer function $T_{ry}(s) = \frac{1}{s^2+bs+1}$. How do the poles of the system vary with b (for $b \geq 0$)?
Solution.

While we are able to easily draw the locations of the poles for first and second order systems, we cannot do the same for higher order systems (because it becomes difficult to explicitly calculate the roots). Furthermore, we would like to obtain some intuition about how the poles of the system will be affected by our choice of controller. We would thus like to come up with a way to sketch how the poles of a system behave in response to a change in a parameter. Since the poles are given by the roots of the denominator polynomial, such a sketch is called a **root locus**. The trajectory of each root in the plane is called a **branch** of the root locus.

13.1 The Root Locus Equations

Consider the unity feedback loop:



The transfer function $L(s)$ could represent the plant, or it could represent some composite system (such as the combination of a controller and plant). We will write

$$L(s) = \frac{N(s)}{D(s)} = \frac{s^m + b_{m-1}s^{m-1} + b_{m-2}s^{m-2} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_1 s + a_0},$$

where $N(s)$ and $D(s)$ are polynomials in s. As usual, the degree of $N(s)$ is m, the degree of $D(s)$ is n, and we assume that $n \geq m$ (i.e., $L(s)$ is proper). The transfer function from r to y is given by

$$T_{ry}(s) = \frac{KL(s)}{1 + KL(s)} = \frac{KN(s)}{D(s) + KN(s)} = \frac{KN(s)}{\Delta(s)} \enspace .$$

The polynomial $\Delta(s) = D(s) + KN(s)$ is called the **characteristic polynomial** of the system. Note that the roots of $D(s) + KN(s)$ are the **closed loop poles**, the roots of $D(s)$ are the **open loop poles**, and the roots of $N(s)$ are the **open loop zeros**. When we plot these elements graphically, we will use $\times$ to denote poles, and $\circ$ to denote zeros.

The root locus is a graph of how the roots of $D(s) + KN(s)$ vary with K. Equivalently, the root locus is the set of all solutions s to the equation $L(s) = -\frac{1}{K}$.

Note that the root locus can actually be used to find how the roots of *any* polynomial vary with a single parameter, and thus it is a very general tool. For example, aside from analyzing how the poles of a unity feedback loop vary with a controller gain K, the root locus can also be used to analyze how the poles vary in response to a change in one of the system parameters. We saw this in the third example above, and we will see more of it later.

To start, we will only focus on the case where K varies from 0 to ∞ – this is called the **positive root locus**. We will deal with the negative root locus later. We will also assume that both $N(s)$ and $D(s)$ are **monic** (i.e., the coefficient corresponding to the highest power in both polynomials is equal to 1). This is not a strict assumption, because we can always divide the entire polynomial $D(s) + KN(s)$ by the leading coefficient of $D(s)$, and then absorb the leading coefficient of $N(s)$ into K to define a new gain $\bar{K}$. After plotting the root locus, we can then map the gain $\bar{K}$ back to K.

13.1.1 Phase Condition

To obtain an alternative (but equivalent) characterization of the root locus, consider the equation

$$L(s) = -\frac{1}{K} \ .$$

Since we are focusing on the case where $K \geq 0$, we see that $L(s)$ must be a negative real number, and the root locus consists of all points satisfying this condition. In particular, since the phase of a negative real number is 180 degrees (or π radians), we obtain the following characterization of the root locus.

The positive root locus is the set of all points s in the complex plane for which $\angle L(s) = (2l + 1)\pi$ radians (where l is any integer).

We can obtain a geometric interpretation of the above condition as follows. Let

$$L(s) = \frac{(s + z_1)(s + z_2)\cdots(s + z_m)}{(s + p_1)(s + p_2)\cdots(s + p_n)} \ ,$$

where $-z_1, -z_2, \ldots, -z_m$ are the open loop zeros, and $-p_1, -p_2, \ldots, -p_n$ are the open loop poles. The phase of $L(\bar{s})$, for some point $\bar{s}$ in the complex plane, is given by

$$\begin{aligned}\angle L(\bar{s}) = \angle(\bar{s} + z_1) + \angle(\bar{s} + z_2) + \cdots + \angle(\bar{s} + z_m) \\ - \angle(\bar{s} + p_1) - \angle(\bar{s} + p_2) - \cdots - \angle(\bar{s} + p_n) \ .\end{aligned}$$

Note that the phase of the point $\bar{s}+z_i$ is given by the angle between the positive real axis and the vector from $-z_i$ to $\bar{s}$:

The same holds for the phase of the point $\bar{s}+p_i$. The phase of $L(\bar{s})$ can therefore be obtained by summing these angles, and this will allow us to determine if $\bar{s}$ is on the root locus.

Example. Consider $L(s) = \frac{s+4}{s((s+1)^2+1)}$. Is the point $s = -3$ on the root locus? Is the point $s = -4 + j$ on the root locus?
Solution.

While we can use the above method to test if specific points are on the root locus, it is quite cumbersome to determine *all* points in the complex plane that are on the root locus in this way. What we need are some general rules for sketching the root locus for a given $L(s)$ (or equivalently, for the equation $D(s)+KN(s)$).

13.2 Rules for Plotting the Positive Root Locus

We will now develop some rules for plotting the positive root locus for a given polynomial $D(s) + KN(s)$.

13.2.1 Start Points and (Some) End Points of the Root Locus

Consider the equation $D(s) + KN(s) = 0$. Since this is an n–th degree polynomial, it will have n roots, and thus the root locus will have n branches. Let's start by considering what happens for the extreme values of K:

- When $K = 0$, the roots of this equation are simply the roots of $D(s)$, which are the open loop poles.
- When $K \to \infty$, we consider the equivalent root locus equation $\frac{N(s)}{D(s)} = -\frac{1}{K}$. As $K \to \infty$, the right side goes to zero, and thus the root locus consists of the points that cause $\frac{N(s)}{D(s)}$ to be zero. This will be true for the m roots of $N(s)$ (i.e., the open loop zeros), and thus m of the branches end at the open loop zeros. We will see what happens to the other $n - m$ branches later.

This brings us to the following rule:

Rule 1. The n branches of the root locus begin at the open loop poles (when $K = 0$). Of the n branches, m branches end at the open loop zeros (when $K = \infty$).

13.2.2 Points on the Real Axis

Recall that the positive root locus contains all points $\bar{s}$ such that

$$\angle(\bar{s} + z_1)+\angle(\bar{s} + z_2)+\cdots+\angle(\bar{s} + z_m)-\angle(\bar{s} + p_1)-\angle(\bar{s} + p_2)-\cdots-\angle(\bar{s} + p_n) \\ = (2l + 1)\pi \text{ radians} \ .$$

Consider a point $s = \bar{s}$ on the real axis. Each real pole or zero to the right of $\bar{s}$ contributes $-\pi$ radians or π radians to the angle. Each pair of complex conjugate poles or zeros contributes nothing to the angle of a $\bar{s}$ (since the angles of the complex conjugate poles or zeros will sum to zero). Each pole or zero to the left of $\bar{s}$ will also contribute nothing (i.e., 0 radians) to the angle.

Thus, in order for $\bar{s}$ to satisfy the above condition, there must be an **odd** number of zeros or poles to the right of $\bar{s}$.

Rule 2. The positive root locus contains all points on the real axis that are to the left of an odd number of zeros or poles.

Example. Consider $L(s) = \frac{(s+3)(s+7)}{s^2((s+1)^2+1)(s+5)}$. Determine the portions of the real axis that are on the root locus.
Solution.

13.2.3 Asymptotic Behavior of the Root Locus

We have already seen that m out of the n branches end up at the open loop zeros as $K \to \infty$. To see what happens to the other $n - m$ branches, consider the root locus equation $L(s) = -\frac{1}{K}$. As $K \to \infty$, the right side goes to zero, and thus the root locus consists of the points that cause $L(s)$ to be zero. This will be true for the m open loop zeros – are there any other choices of s that will make the left hand side equal to zero? To answer this, write $L(s)$ as

$$\begin{aligned} L(s) &= \frac{s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \\ &= \frac{\frac{1}{s^{n-m}} + b_{m-1}\frac{1}{s^{n-m+1}} + \cdots + b_1\frac{1}{s^{n-1}} + b_0\frac{1}{s^n}}{1 + a_{n-1}\frac{1}{s} + \cdots + a_1\frac{1}{s^{n-1}} + a_0\frac{1}{s^n}} \ . \end{aligned}$$

If $n > m$, we see that the numerator goes to zero if $|s| \to \infty$. Thus, the system $L(s) = \frac{N(s)}{D(s)}$ is said to have $n-m$ **zeros at infinity**, in addition to the m finite zeros (i.e., the roots of $N(s)$). We can thus conclude that m of the n branches go to the open loop zeros, and the remaining $n-m$ branches go off to infinity as $K \to \infty$. The question is, *how* do these branches approach infinity? The following rule characterizes the behavior of these branches.

Rule 3. Of the n branches in the root locus, $n-m$ of the branches go to infinity, and asymptotically approach lines coming out of the point $s = \alpha$ with angles Φ_l, where

$$\alpha = \frac{\Sigma \text{ open loop poles} - \Sigma \text{ open loop zeros}}{n-m}, \quad \Phi_l = \frac{(2l+1)\pi}{n-m}, \quad (13.1)$$

for $l = 0, 1, 2, \ldots, n-m-1$.

The asymptotes for a few values of $n-m$ are shown below:

We will now go over a sketch of the proof of this result.[1] First, note that the

[1] This proof is borrowed from Prof. Daniel Liberzon at the University of Illinois.

transfer function $L(s)$ has n poles and m zeros at certain locations. Now, if we were to consider some point s with a very large magnitude (i.e., very far away from the other poles and zeros), the poles and zeros would essentially look like they were clustered at one point; let's call this point α. So, for large $|s|$, we would like to find a good value of α so that we can approximate the transfer function as

$$\begin{aligned} L(s) &= \frac{s^m + b_{m-1}s^{m-1} + b_{m-2}s^{m-2} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_1 s + a_0} \approx \frac{1}{(s-\alpha)^{n-m}} \\ \Leftrightarrow &\ \frac{s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_1 s + a_0}{s^m + b_{m-1}s^{m-1} + b_{m-2}s^{m-2} + \cdots + b_1 s + b_0} \approx (s-\alpha)^{n-m}. \end{aligned}$$

Note that we are taking the exponent to be $n - m$, because to the point s, the m zeros look like they are directly on top of n poles, and thus they 'cancel' each other out. To see what value of α to choose, let's start dividing the denominator on the left hand side into the numerator to obtain the first couple of terms of the quotient:

Thus, we can write

$$(s-\alpha)^{n-m} \approx s^{n-m} + (a_{n-1} - b_{m-1})s^{n-m-1} + \cdots ,$$

and since $(s-\alpha)^{n-m} = s^{n-m} - \alpha(n-m)s^{n-m-1} + \cdots$, if we compare the second term in the two expansions, we obtain

$$\alpha = \frac{b_{m-1} - a_{n-1}}{n-m}. \tag{13.2}$$

Next, suppose that

$$\begin{aligned} N(s) &= (s+z_1)(s+z_2)\cdots(s+z_m) = s^m + \sum_{i=1}^{m} z_i s^{m-1} + \cdots \\ D(s) &= (s+p_1)(s+p_2)\cdots(s+p_n) = s^n + \sum_{i=1}^{n} p_i s^{n-1} + \cdots . \end{aligned}$$

Thus, we have $b_{m-1} = \sum_{i=1}^{m} z_i$ and $a_{n-1} = \sum_{i=1}^{n} p_i$, which we substitute into (13.2) to produce α in (13.1).

To derive the asymptote angles in (13.1), consider the root locus of $(s-\alpha)^{-(n-m)}$. Recall that the root locus of a function is the set of all s such that

$$\sum \angle s + z_i - \sum \angle s + p_i = (2l+1)\pi \qquad (13.3)$$

for some integer l, where z_i are the zeros and p_i are the poles. In this case, the function $(s-\alpha)^{-(n-m)}$ has no zeros, and all $n-m$ poles at α. Thus, we have $\sum \angle s + p_i \approx (n-m)\angle s - \alpha$, and the above expression becomes

$$\angle s - \alpha = \frac{(2l+1)\pi}{n-m}.$$

Note that the negative sign in front of the pole angles in (13.3) contributes an angle of $-\pi$, which can just be absorbed into the term $(2l+1)\pi$. There are $n-m$ different possibilities for the angle on the right hand side of the above equation, corresponding to $l = 0, 1, \ldots, n-m-1$ (after this, the angles start repeating), and thus there are $n-m$ different asymptotes leading out from the poles at α with the angles specified by (13.1).

Example. Consider $L(s) = \frac{1}{s(s+2)}$. Draw the portions of the real axis that are on the positive root locus, and determine the asymptotes.
Solution.

Example. Consider $L(s) = \frac{1}{s((s+1)^2+1)}$. Draw the portions of the real axis that are on the positive root locus, and determine the asymptotes.

Solution.

Example. Consider $L(s) = \frac{s+6}{s((s+1)^2+1)}$. Draw the portions of the real axis that are on the positive root locus, and determine the asymptotes.
Solution.

Since we are only interested in polynomials $D(s)+KN(s)$ that have purely real coefficients, the roots of the polynomial will either be real or appear as complex conjugate pairs. This produces the following important fact.

> The root locus is always **symmetric about the real axis**.

13.2.4 Breakaway Points

We have already seen some examples of root loci where two or more points come together and then bounce off each other in opposite directions. Such points are called **breakaway** points. Note that at such points, the characteristic polynomial $\Delta(s) = D(s) + KN(s)$ will have multiple roots at the breakaway point. Let the breakaway point be $s = \bar{s}$. Then we can write

$$\Delta(s) = (s - \bar{s})^q \bar{D}(s) \ ,$$

where $q \geq 2$ is the multiplicity of the root $\bar{s}$, and $\bar{D}(s)$ is some polynomial. This means that $\Delta(\bar{s}) = 0$ and $\frac{d\Delta}{ds}(\bar{s}) = 0$. Substituting $\Delta(s) = D(s) + KN(s)$, we have

$$\begin{aligned} D(\bar{s}) + KN(\bar{s}) &= 0 \\ \frac{dD}{ds}(\bar{s}) + K\frac{dN}{ds}(\bar{s}) &= 0 \ . \end{aligned}$$

Solving the first equation, we get $K = -\frac{D(\bar{s})}{N(\bar{s})}$, and substituting this into the second equation, we come to the following rule.

Rule 4. The root locus will have multiple roots at the points $\bar{s}$ for which both of the following conditions are satisfied.

- $N(\bar{s})\frac{dD}{ds}(\bar{s}) - D(\bar{s})\frac{dN}{ds}(\bar{s}) = 0$.
- $-\frac{D(\bar{s})}{N(\bar{s})} = K$ is a positive real number.

Example. Draw the positive root locus for $L(s) = \frac{s+6}{s(s+2)}$.
Solution.

Example. Verify that the branches in the positive root locus for $L(s) = \frac{1}{s((s+1)^2+1)}$ never intersect.

Solution.

There are various other rules that we could derive to draw root locus plots, but they tend to be cumbersome to apply by hand. The above rules will be sufficient for us to get intuition about many systems.

13.2.5 Some Root Locus Plots

Example. Consider a control system in unity feedback with $P(s) = \frac{1}{s^2}$ and $C(s) = K\frac{s+1}{s+12}$. Draw the positive root locus.
Solution.

Example. Consider a control system in unity feedback with $P(s) = \frac{1}{s^2}$ and $C(s) = K\frac{s+1}{s+4}$. Draw the positive root locus.
Solution.

Example. Consider a control system in unity feedback with $P(s) = \frac{1}{s^2}$ and $C(s) = K\frac{s+1}{s+9}$. Draw the positive root locus.
Solution.

The above examples show that as the pole of the controller moves in closer to the root locus, it tends to push the branches of the locus to the right. From our discussion so far, we can state the following rules of thumb: **poles repel, and zeros attract**.

13.2.6 Choosing the Gain from the Root Locus

The positive root locus tells us how the poles of the closed loop system vary as we increase K from 0 to ∞. Once we've plotted the root locus, we may wish to select points on the branches so that the closed loop system will have certain desired properties (such as rise time, overshoot, peak time, etc). We then need to find the gain K so that the closed loop poles will be at the desired locations. To find the gain, note that

$$L(s) = -\frac{1}{K} \Leftrightarrow K = -\frac{1}{L(s)} \quad .$$

Since K is a positive real number, we can examine the magnitude of both sides of the above equation to obtain

$$\boxed{K = \frac{1}{|L(s)|} = \frac{|s+p_1||s+p_2|\cdots|s+p_n|}{|s+z_1||s+z_2|\cdots|s+z_m|}\ .}$$

Specifically, if we have a desired point $\bar{s}$ on the root locus, we can find the gain K that produces a pole at $\bar{s}$ by multiplying and dividing the lengths of the vectors from each of the poles and zeros to $\bar{s}$, according to the above equation.

Example. Suppose $L(s) = \frac{1}{s^2+2s}$. Find the gain K that results in the closed loop system having a peak time of at most 2π seconds.
Solution.

Note that MATLAB is an extremely useful tool for doing this in practical controller design. Once one has plotted the root locus for a given system in MATLAB (using the `rlocus` command), one can simply click on the root locus branch at any desired location to find the value of the gain at that point.

13.3 Rules for Plotting the Negative Root Locus

In the last section, we saw how to plot the root locus for positive values of the gain K. We can now complete the root locus by considering negative values of K – this is called the **negative root locus**. The only difference in this case stems from the phase condition. Recall that the root locus consists of all points s satisfying $L(s) = -\frac{1}{K}$. If K is a negative real number, $L(s)$ is a positive real number, and thus we can state the following.

The negative root locus is the set of all points s in the complex plane for which $\angle L(s) = 2l\pi$ radians (where l is any integer).

All of the rules for plotting the positive root locus translate directly once we consider this new phase condition:

- **Rule 1.** The n branches of the root locus begin at the open loop poles (when $K = 0$). Of the n branches, m branches end at the open loop zeros (when $K = -\infty$).
- **Rule 2.** The negative root locus contains all points on the real axis that are to the left of an **even** number of zeros or poles.
- **Rule 3.** Of the n branches in the root locus, $n - m$ of the branches go to infinity, and asymptotically approach lines coming out of the point $s = \alpha$ with angles Φ_l, where

  $$\alpha = \frac{\Sigma \text{ open loop poles} - \Sigma \text{ open loop zeros}}{n - m}, \quad \Phi_l = \frac{2l\pi}{n - m},$$

  for $l = 0, 1, 2, \ldots, n - m - 1$.
- **Rule 4.** The root locus will have multiple roots at the points $\bar{s}$ for which both of the following conditions are satisfied.

  $N(\bar{s})\frac{dD}{ds}(\bar{s}) - D(\bar{s})\frac{dN}{ds}(\bar{s}) = 0.$

  $-\frac{D(\bar{s})}{N(\bar{s})} = K$ is a **negative** real number.

Note that the gain for a particular point s on the negative root locus is given by

$$K = -\frac{1}{|L(s)|} = -\frac{|s + p_1||s + p_2| \cdots |s + p_n|}{|s + z_1||s + z_2| \cdots |s + z_m|} .$$

The asymptotes for the negative root locus look like this:

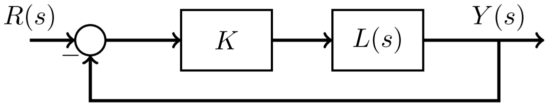
Example. Determine the negative root locus for $L(s) = \frac{1}{s((s+1)^2+1)}$, and then draw the complete root locus (both positive and negative).
Solution.

Chapter 14

Stability Margins from Bode Plots

The last chapter showed how to analyze and understand the closed loop system from a root locus perspective. We will now study the use of Bode plots to analyze closed loop systems, complementing the root locus techniques. In the next chapter, we will use these ideas to design controllers (building on our study of PID controllers).

Suppose we're given the Bode plot for the transfer function $L(s)$, and we would like to study properties of the following feedback loop:



In other words, we would like to infer some things about the **closed loop system** based on the **open loop Bode plot**. Remember that we also did this when we studied root locus plots: we studied the locations of the closed loop poles by starting with the open loop poles and zeros.

Recall the root locus equation $1 + KL(s) = 0$ (the closed loop poles are the values s that satisfy this equation). When K is a positive real number, this means that $|KL(s)| = 1$ and $\angle KL(s) \equiv \pi$ (modulo 2π). A point $s = j\omega$ on the imaginary axis (for some ω) will be on the positive root locus if $|KL(j\omega)| = 1$ and $\angle KL(j\omega) \equiv \pi$. Since we have access to $|KL(j\omega)|$ and $\angle KL(j\omega)$ from the Bode plot, we should be able to determine the imaginary axis crossings by finding the frequencies ω (if any) on the plot that satisfy the conditions $|KL(j\omega)| = 1$ (or $20 \log |KL(j\omega)| = 0$) and $\angle KL(j\omega) \equiv \pi$.

To develop this further, suppose that $L(s) = \frac{1}{s(s+1)(\frac{s}{100}+1)}$. The Bode plot of

$KL(s)$ (with $K = 1$) is given below:

We'll be using the following terminology.

- **Gain crossover frequency:** This is the frequency ω_{cg} such that $|KL(j\omega_{cg})| = 1$ (or equivalently, $20 \log |KL(j\omega_{cg})| = 0$).
- **Phase crossover frequency:** This is the frequency ω_{cp} such that $\angle KL(j\omega_{cp}) \equiv \pi$.

For the above example, we have $\omega_{cg} \approx 1$ and $\omega_{cp} \approx 10$. The phase at ω_{cg} is approximately $-\frac{3\pi}{4}$, and so the feedback configuration with $K = 1$ does not have any closed loop poles on the imaginary axis. Is there another value of K for which the closed loop system will have poles on the imaginary axis (and thus cross the boundary from stability to instability)? To determine this from the Bode plot, note that $20 \log |KL(j\omega)| = 20 \log K + 20 \log |L(j\omega)|$ and $\angle KL(j\omega) = \angle L(j\omega)$ (for $K > 0$). Thus, K has no effect on the phase, and it affects the magnitude plot by shifting it up or down by $20 \log K$. For example, when $K = 10$, the entire magnitude gets shifted up by $20 \log 10 = 20$ dB (when the vertical axis denotes $20 \log |KL(s)|$); this is shown on the above Bode plot by the dashed lines. Based on this plot, we see that changing K has the effect of changing the gain crossover frequency (but not the phase crossover frequency).

In order to find the value of K that causes some closed loop poles to lie on the imaginary axis in the above example, we need to find out how to make the gain crossover frequency and the phase crossover frequency coincide. Examining the magnitude plot, we see that $20 \log |L(j10)| \approx -40$, and thus the magnitude curve needs to be shifted up by approximately 40 dB in order to set $\omega_{cg} = \omega_{cp}$, which can be accomplished by setting $20 \log K \approx 40$, or $K \approx 100$. Thus, we can conclude that the closed loop system will have an imaginary axis crossing when $K \approx 100$. One can easily see from the Bode plot that this is the only positive value of K for which this will happen.

We can verify this result by examining the positive root locus of $L(s)$:

As expected, the branches cross the imaginary axis only once (other than the trivial case where $K = 0$). To find the locations where the branches cross the imaginary axis, we note that

$$\begin{aligned} & 1 + KL(s) = 0 \\ \Leftrightarrow \quad & 1 + K\frac{1}{s(s+1)(\frac{s}{100}+1)} = 0 \\ \Leftrightarrow \quad & s^3 + 101s^2 + 100s + 100K = 0 \ . \end{aligned}$$

We use the Routh-Hurwitz test to determine the region of stability as $0 < K < 101$. Thus, we have a potential imaginary axis crossing at $K = 101$. To find the points on the imaginary axis where this happens, we set $s = j\omega$ and $K = 101$ and solve the equation

$$\begin{aligned} & (j\omega)^3 + 101(j\omega)^2 + 100(j\omega) + 100(101) = 0 \\ \Leftrightarrow \quad & -j\omega^3 - 101\omega^2 + 100\omega j + 100(101) = 0 \ . \end{aligned}$$

Setting the imaginary and real parts to zero, we find that $\omega = 10$. Thus, we have an imaginary axis crossing at $s = \pm 10j$ when $K = 101$. Note that this agrees with the analysis from the Bode plot (the Bode plot actually told us $K \approx 100$, since we approximated the Bode plot with straight lines).

While we could determine imaginary axis crossings by looking at the Bode plot, we didn't necessarily know which direction the branches were going – are we going from stability to instability, or instability to stability? We could determine this information by looking at the root locus, but we will later develop a completely frequency domain approach to characterizing the stability of the closed loop system. For now, we will assume that the closed loop system is stable with a given value of K, and investigate ways to design controllers using a frequency domain analysis in order to **improve** the stability of the closed loop system.

Stability Margins

We will define some terminology based on the discussion so far. Consider again the Bode plot of $KL(s)$ with $K = 1$:

Assuming that the closed loop system is stable, we can ask the question: How far from instability is the system? There are two metrics to evaluate this:

- **Gain margin:** This is the amount by which K can be multiplied before $|KL(j\omega_{cp})| = 1$ (i.e., the gain crossover frequency and phase crossover frequencies coincide).
- **Phase margin:** This is the amount by which the phase at ω_{cg} exceeds $-\pi$; more specifically, it is defined as

$$PM = \angle L(j\omega_{cg}) + \pi \ .$$

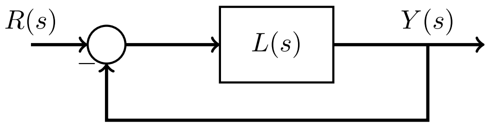
In general, we would like to have large gain and phase margins in order to improve the stability of the system. In the above example with $K = 1$, the gain margin is approximately 100, and the phase margin is approximately $\frac{\pi}{4}$. Let us consider some more examples, just to be clear on the concept of gain and phase margins.

Example. What is the gain margin and phase margin for $KL(s) = \frac{1}{s(s+1)^2}$?
Solution.

Example. What is the gain margin and phase margin for $KL(s) = \frac{s+1}{s^2(\frac{s}{10}+1)}$?
Solution.

In the above example, we noticed that the gain margin is ∞, since the phase only hits $-\pi$ at $\omega = \infty$. However, note that as we increase K, the gain crossover frequency starts moving to the right, and the phase margin decreases. If we examine the root locus for the system $L(s) = \frac{s+1}{s^2(\frac{s}{10}+1)}$, we see that we have a set of poles that move vertically in the plane as $K \to \infty$, and thus the damping ratio ζ for these poles decreases as $K \to \infty$. This seems to indicate that there might be some relationship between the phase margin and the damping ratio ζ. We will now derive an explicit relationship between these two quantities.

Consider the system $L(s) = \frac{\omega_n^2}{s(s+2\zeta\omega_n)}$, which is placed in the feedback configuration



The closed loop transfer function is $T_{ry}(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$, which is the standard second order system with damping ratio ζ. The phase plot of $L(j\omega)$ is given by

This shows that the gain margin is ∞ (and this is easily verified by looking at the root locus). Next, let's examine the phase margin. By setting the magnitude of $L(j\omega)$ equal to 1, one can verify that the Bode plot of $L(s)$ has gain crossover frequency equal to

$$\omega_{cg} = \omega_n\sqrt{\sqrt{1+4\zeta^4}-2\zeta^2}\ .$$

Using the fact that $PM = \angle L(j\omega_{cg}) + \pi$, we obtain (after some algebra)

$$PM = \tan^{-1}\left(\frac{2\zeta}{\sqrt{\sqrt{1+4\zeta^4}-2\zeta^2}}\right)\ .$$

Notice that the phase margin is a function of ζ and not ω_n. Interestingly, this seemingly complicated expression can be approximated fairly well by a straight line for small values of ζ:

For $0 \leq \zeta \leq 0.7$, the phase margin (in degrees) and damping ratio are related by

$$PM \approx 100\zeta \ .$$

While we derived the above expression for the standard second order system, we can also use it as a general rule of thumb for higher order systems. Specifically, as the phase margin decreases, the system becomes less stable, and might exhibit oscillatory behavior. We can use the above relationship to design control systems in the frequency domain in order to obtain certain time-domain characteristics (such as meeting overshoot specifications).

Another useful rule-of-thumb that is generally adopted for the closed loop bandwidth ω_{BW} is as follows:

In general, we have $\omega_{cg} \leq \omega_{BW} \leq 2\omega_{cg}$, and $\omega_{BW} \approx \omega_n$, which leads to the approximation

$$\omega_{BW} \approx \omega_{cg} \approx \omega_n \ .$$

Note that our discussions here have assumed a typical Bode plot that has large magnitude at low frequencies, and low magnitude at high frequencies, with a single gain crossover frequency. We can deal with more complicated Bode plots by generalizing our discussions, but we'll focus on these typical Bode plots for now.

In order to obtain fast transient behavior, we typically want a large gain crossover frequency, but this would come at the cost of decreasing the phase margin. Furthermore, in order to obtain better steady state tracking, we would typically want to increase the gain K in order to boost the low frequency behavior, but this would again move the gain crossover frequency to the right and decrease the phase margin. Therefore, we must consider more complicated controllers (other than just a simple proportional controller K) in order to obtain a good phase margin, a good gain crossover frequency, and good steady state tracking. This will be the focus of the next part of the course.

Chapter 15

Compensator Design Using Bode Plots

We now turn our attention to designing dynamic controllers (also called compensators), building on our earlier study of PID controllers. We have seen so far that the phase margin of a given system is representative of the system's stability, and is directly related to the damping of the system – a larger phase margin makes the system more stable, and increases the damping. Given a system, we thus want to design a controller that improves the phase margin. In certain systems, one way to do this would be to decrease the gain of the system, so that the gain crossover frequency moves to the left (in the direction of increasing phase). However, we have seen that the low frequency gain of the system is related to how well the system tracks reference inputs – a larger low frequency gain corresponds to better tracking. Another metric is the gain crossover frequency: since the gain crossover frequency is approximately equal to the bandwidth and the natural frequency of the system, a larger gain crossover frequency corresponds to faster response, but also leads to smaller phase margins. Therefore, we would like to design more sophisticated controllers in order to keep the low frequency gain large (in order to meet tracking specifications), or to increase the gain crossover frequency (in order to obtain faster transients), and also to increase the phase at the gain crossover frequency (in order to boost the phase margin).

In this chapter, we will study the design of *lead* and *lag* compensators using Bode plots. We will start by introducing the form of these controllers.

15.1 Lead and Lag Compensators

In our discussion on root locus plots, we came up with the following rule of thumb: poles repel and zeros attract. In particular, we saw that adding a zero to the system via the controller can produce stability, when a simple proportional controller could not. Recall that this was the same conclusion that we reached when we were studying PID control.

Example. For the unity feedback loop with $P(s) = \frac{1}{s^2}$, draw the positive root locus when $C(s) = K$ and $C(s) = K(s+1)$.
Solution.

However, we also discussed the fact that the PD controller $C(s) = K_P + K_D s$ is not physically implementable, since it is not proper, and it would differentiate high frequency noise, thereby producing large swings in output. In order to avoid this, we replaced the PD controller with a controller of the form $C(s) = K_P + K_D \frac{sp}{s+p}$; the larger the value of p, the better the controller approximates PD control, but the more susceptible it is to high frequency noise. The modified PD controller can be written as

$$\begin{aligned} C(s) = K_P + K_D \frac{ps}{s+p} &= \frac{K_P(s+p) + K_D ps}{s+p} \\ &= \frac{(K_P + K_D p)s + K_P p}{s+p} \\ &= (K_P + K_D p)\frac{s + \frac{K_P p}{K_P + K_D p}}{s+p} \quad . \end{aligned}$$

If we let $K = K_P + K_D p$ and $z = \frac{K_P p}{K_P + K_D p}$, we obtain the dynamic controller

$$C(s) = K\frac{s+z}{s+p} \quad .$$

This controller is called a **lead controller** (or lead compensator) if $z < p$ and a **lag controller** (or lag compensator) if $z > p$. To see where this terminology comes from, recall that if we applied a sinusoidal input $\cos(\omega t)$ to the controller $C(s)$, the output would be $|C(j\omega)|\cos(\omega t + \angle C(j\omega))$ in steady state. The phase of $C(j\omega)$ is given by $\angle C(j\omega) = \angle(j\omega + z) - \angle(j\omega + p)$, and if $z < p$, we have $\angle C(j\omega) > 0$ (i.e., the output *leads* the input). On the other hand, if $z > p$, we have $\angle C(j\omega) < 0$ (i.e., the output *lags* the input).

As described above, the approximation to the PD controller has a large pole p, and thus is a lead compensator. As one might expect, a lag compensator is then an approximation to a PI controller. Specifically, consider a PI controller of the form

$$C(s) = K_P + \frac{K_I}{s} = \frac{K_P s + K_I}{s} = K_P \frac{s + \frac{K_I}{K_P}}{s} \ .$$

Recall that the reason for considering PI control was that it guaranteed perfect tracking for a step (if the closed loop system is stable). It did this by introducing a pole at the origin in $P(s)C(s)$, which is the same as saying that it increased the DC gain of $P(s)C(s)$ to infinity. In general, the higher we can make the DC gain of $P(s)C(s)$, the better it can track reference inputs. One way to do this would be to add a PI controller to the loop, but in some circumstances, we might be interested in using a *stable* controller (whereas the PI controller by itself is unstable). Thus, suppose we approximate the PI controller by

$$C(s) = K\frac{s + z}{s + p} \ ,$$

where $K = K_P$, $z = \frac{K_I}{K_P}$, and p is a small positive number. The smaller we make p, the better this controller approximates the PI controller. In particular, we are interested in using this controller to boost the DC gain of $P(s)C(s)$, and so we would like to make $z > p$. As we discussed earlier, this is a **lag controller**.

Lag controllers are frequently used to improve the steady state error after we have already satisfied the transient specs (potentially with some other controller). The objective is typically to boost the DC gain of the controller, while trying to keep the poles of the closed loop system from changing too much. One example of this is **lead-lag** compensator design: use a lead compensator to stabilize the system, and then use a lag compensator to boost the tracking accuracy. We will discuss this later.

To summarize:

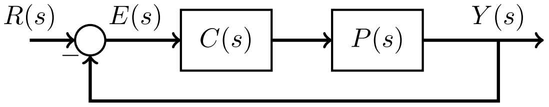
- Lead compensation approximates PD control, and is used to stabilize the system and improve the transient characteristics (by moving the locus to the left and improving the phase margin).

- Lag compensation approximates PI control, and is used to boost DC gain.

We are now ready to design lead and lag compensators.

15.2 Lead Compensator Design

First, consider the standard unity feedback loop:



A lead controller will have the form $C(s) = K_c\frac{s+z}{s+p}$, where $p > z$. Since $p > z$, we can write $z = \alpha p$ for some $0 < \alpha < 1$. The Bode form of the above controller is then given by

$$C(s) = K_c\frac{s+\alpha p}{s+p} = K_c\frac{\alpha p(\frac{s}{\alpha p}+1)}{p(\frac{s}{p}+1)} = \underbrace{K_c\alpha}_{K}\underbrace{\frac{\frac{s}{\alpha p}+1}{\frac{s}{p}+1}}_{C_l(s)} \ .$$

The phase margin of the closed loop system can be obtained by examining the Bode plot of $C(s)P(s)$, which is obtained by simply adding together the Bode plots of $C_l(s)$ and $KP(s)$ (since the magnitude is on a log scale, and the phases inherently add). The gain K of the compensator can be first be chosen to meet steady state error specifications, or to obtain a certain crossover frequency. Once that is done, let's see what the lead compensator contributes to the system by examining the Bode plot of $C_l(s)$:

We see that the phase plot of $C_l(s)$ has a bump, and we can use this positive phase contribution to increase the phase of $KP(s)$. Specifically, we would like to choose α and p so that the bump occurs near the crossover frequency of $KC_l(s)P(s)$, thereby increasing the phase margin of the system. To see how to choose the pole and zero, note that the phase of $C_l(j\omega)$ is given by

$$\angle C_l(j\omega) = \tan^{-1}(\frac{\omega}{\alpha p}) - \tan^{-1}(\frac{\omega}{p}) \ .$$

From the phase plot, we note that the maximum phase occurs halfway between the zero and the pole (on a logarithmic scale). If we denote the frequency where the maximum phase occurs as $\omega_{\max}$, we have

$$\log \omega_{\max} = \frac{1}{2}\left(\log(\alpha p) + \log(p)\right) = \log \sqrt{\alpha p^2}\ ,$$

from which we obtain $\omega_{\max} = \sqrt{\alpha} p$. If we denote $\phi_{\max} = \angle C_l(j\omega_{\max})$, we obtain (after some algebra)

$$\sin \phi_{\max} = \frac{1-\alpha}{1+\alpha}\ .$$

These expressions are important, so let's restate them:

The maximum phase of the lead compensator with zero at αp and pole at p is denoted by $\phi_{\max}$ and occurs at the frequency $\omega_{\max} = \sqrt{\alpha} p$. The maximum phase satisfies the equation

$$\sin \phi_{\max} = \frac{1-\alpha}{1+\alpha} \quad \text{or equivalently,} \quad \alpha = \frac{1-\sin\phi_{\max}}{1+\sin\phi_{\max}}\ .$$

The idea will be to choose the pole and zero of the compensator such that $\omega_{\max}$ lies on the crossover frequency of $KP(s)$, with the hope of contributing an extra $\phi_{\max}$ degrees of phase margin. Let's try an example to see how this works.

Example. Consider $KP(s) = \frac{1}{s(s+1)}$. Draw an approximate Bode plot for $C_l(s)KP(s)$ when the pole and zero of the compensator are such that the maximum compensator phase occurs at the gain crossover frequency of $KP(s)$.

Solution.

From the above example, we see that although the compensator does contribute $\phi_{\max}$ to the phase at the gain crossover frequency of $KP(s)$, the gain crossover frequency of $C_l(s)KP(s)$ actually shifts to the right due to the positive magnitude contribution of $C_l(s)$. Thus, the phase margin of $KC_l(s)P(s)$ is actually a little less than the phase margin of $KP(s)$ plus $\phi_{\max}$. In order to still get our desired phase margin, we should therefore make $\phi_{\max}$ a little larger than we need (usually about 10° extra is enough), so that the phase margin of $KC_l(s)P(s)$ will meet the specification.

The complete design procedure for lead compensators is as follows.

Lead Compensator Design

1. Choose K to meet a steady state error specification, or to meet a gain crossover frequency specification (in the latter case, one can choose K so that the crossover frequency of $KP(s)$ is a little less than desired, since the lead compensator will shift the frequency to the right a little bit).
2. Find the phase margin of $KP(s)$ (from the Bode plot).
3. Find how much extra phase is required in order to meet the phase margin spec. Set $\phi_{\max}$ to be this extra phase plus 10°.
4. Find $\alpha = \frac{1-\sin\phi_{\max}}{1+\sin\phi_{\max}}$.
5. Set $\omega_{\max}$ to be the gain crossover frequency of $KP(s)$. From this, we can calculate $p = \frac{\omega_{\max}}{\sqrt{\alpha}}$ and $z = \sqrt{\alpha}\omega_{\max}$. The compensator is given by
$$C(s) = K\frac{\frac{s}{\alpha p}+1}{\frac{s}{p}+1}.$$
6. Check if the compensator achieves the specifications. If not, iterate or add another lead compensator.

Note: The lead compensator also sometimes appears as $C_{lead}(s) = Gain_1\frac{1+aTs}{1+Ts}$. Comparing this to the lead compensator given above, we have $Gain_1 = K$, $T = \frac{1}{p}$ and $a = \frac{1}{\alpha}$.

Example. Consider $P(s) = \frac{1}{s(s+1)}$. Design a lead compensator so that the closed loop system has a steady state tracking error of 0.1 to a ramp input, and overshoot less than 25%.

Solution.

15.3 Lag Compensator Design

Let us now consider lag controller design using the Bode plot method. Recall that lag controllers are approximations to PI controllers, and are used to boost the DC gain (in order to improve the steady state tracking error). Here, we will see how to design lag compensators using the Bode plot method to improve the phase margin and satisfy steady state tracking specs. The Bode plot analysis will give us a perspective on why the pole and zero of the lag compensator are usually chosen to be very small.

A lag controller will have the form $C(s) = K_c\frac{s+z}{s+p}$, where $z > p$. Since $z > p$, we can write $z = \beta p$ for some $\beta > 1$. The Bode form of the above controller is then given by

$$C(s) = K_c\frac{s+\beta p}{s+p} = K_c\frac{\beta p(\frac{s}{\beta p}+1)}{p(\frac{s}{p}+1)} = K_c\underbrace{\beta\frac{\frac{s}{\beta p}+1}{\frac{s}{p}+1}}_{C_g(s)} .$$

Note that we are interested in the gain-boosting properties of the lag controller, and so we will group the DC gain β with the dynamics of the controller in the term $C_g(s)$ (this is in contrast to the lead controller, where we grouped the DC gain α with the gain K_c). In this case, we will be using the gain K_c to obtain a desired phase margin, and the controller $C_g(s)$ to boost the DC gain. Since the Bode plot of $C(s)P(s)$ is obtained simply by adding together the Bode plots of $C_g(s)$ and $K_cP(s)$, let us examine the Bode plot of $C_g(s)$:

Note from the magnitude plot that $C_g(s)$ will add to the magnitude of $K_cP(s)$ at low frequencies, thereby reducing the steady state tracking error:

Furthermore, we see that the phase plot of $C_g(s)$ has a dip between $\omega = p$ and $\omega = z$, which will reduce the phase of $K_cP(s)$ in that frequency range. This is generally bad, because a lower phase might lead to a reduced phase margin. The idea will be to choose the pole and zero very small, so that the dip in phase will occur at very low frequencies (far away from the gain crossover frequency).

The design procedure for lag compensators can be summarized as follows.

Lag Compensator Design

1 Choose K_c to meet the phase margin specification (with about 10° of buffer to accommodate the phase lag induced by the lag controller) by moving the gain crossover frequency to the left.

2 Find the low frequency gain of $K_cP(s)$, and determine how much extra gain β should be contributed by $C_g(s)$ in order to meet the tracking specification.

3 Choose the zero z of the compensator to be about one decade below the gain crossover frequency of $K_cP(s)$.

4 Choose the pole p of the compensator as $p = \frac{z}{\beta}$. The compensator is given by

$$C(s) = K_c\beta\frac{\frac{s}{\beta p}+1}{\frac{s}{p}+1}.$$

5 Check if the compensator achieves the specifications. If not, iterate or add another compensator.

Example. Consider $P(s) = \frac{1}{s(s+1)}$. Design a lag compensator so that the closed loop system has a steady state tracking error of 0.1 to a ramp input, and overshoot less than 10%.
Solution.

Note that we can use either a lead compensator or a lag compensator to satisfy the specs here. The difference is that the lag compensator increases the phase margin by *reducing* the gain crossover frequency, whereas the lead compensator increases the phase margin by adding more phase to the system. Therefore, the response of the system with the lead compensator will generally be faster than that of the same system with a lag compensator. However, the lag compensator is capable of boosting the DC gain of the system without substantially moving the gain crossover frequency or reducing the phase margin. Thus, a lag compensator is often used in order to improve the tracking characteristics of an existing controller, without affecting the other performance metrics too much. The choice of controller will generally depend on the application requirements and constraints.

Chapter 16

Nyquist Plots

So far, we have studied root locus methods and Bode plot methods for analyzing the behavior of closed loop systems from the open loop transfer functions. The root locus allows us to see how the poles of the transfer function change when we vary a certain parameter, and allows us to visualize the effect of adding additional poles and zeros. However, the root locus is not capable of handling delays in the feedback loop (because a delay of τ contributes a term $e^{-s\tau}$ to the transfer function, which does not have a nice zero/pole interpretation). Furthermore, the root locus cannot handle general uncertainties in the model (it can, however, tell us something about the locations of the poles when a single parameter is allowed to change slightly).

Bode plots are able to capture uncertainties and delay, and we have seen how to use them to design controllers and analyze properties of the closed loop system. However, up to this point, we have been assuming that the closed loop system is stable when we put $P(s)$ in the unity feedback loop with a certain controller gain K. Under this condition, we have seen how to use the Bode plot of the open loop system $KP(s)$ to determine how much we can boost K before the closed loop poles cross the imaginary axis. We have also seen how to use Bode plots to design lead and lag controllers in order to meet certain performance specifications. We will now study **Nyquist plots**, which complement Bode plots to provide us with frequency response techniques to determine the stability of the closed loop system (i.e., we will not have to assume initial stability, as we did in the Bode plot analysis). Furthermore, the Nyquist plots will provide us with an alternative mechanism to evaluate the robustness of the system (via the gain margin and phase margin). To develop these concepts, we will need the notion of **contours** in the complex plane.

Contour: A contour is a piecewise smooth path in the complex plane. The contour is **closed** if it starts and ends at the same point. A contour is **simple**

if it doesn't cross itself at any point.

Suppose we have a system with transfer function

$$H(s) = \frac{(s+z_1)(s+z_2)\cdots(s+z_m)}{(s+p_1)(s+p_2)\cdots(s+p_n)} \ ,$$

and we consider a simple closed clockwise contour C in the complex plane. At each point $\bar{s}$ on the contour C, $H(\bar{s})$ is simply some complex number. If we evaluate $H(s)$ at all points on C, we get a closed (but not necessarily simple) contour which we will denote by $H(C)$:

Let's focus on a particular point $\bar{s}$ on the contour C. The complex number $H(\bar{s})$ has a magnitude and a phase; the latter is given by

$$\angle H(\bar{s}) = \sum_{i=1}^{m} \angle(\bar{s}+z_i) - \sum_{i=1}^{n} \angle(\bar{s}+p_i) \ .$$

Note that $H(\bar{s})$ can be represented as a vector from the origin with magnitude $|H(\bar{s})|$ and angle $\angle H(\bar{s})$, as shown in the above figure. We will be interested in seeing how the phase of $H(\bar{s})$ changes as the point $\bar{s}$ moves around the contour C. To do this, we see from the above expression for $\angle H(\bar{s})$ that we can examine how each of the quantities $\angle(\bar{s}+z_i)$ and $\angle(\bar{s}+p_i)$ vary as $\bar{s}$ moves around the contour C. Suppose the contour C and the distribution of poles and zeros looks like this:

The quantity $\angle(\bar{s} + z_i)$ is given by the angle between the vector from $-z_i$ to $\bar{s}$ and the positive real axis, and the quantity $\angle(\bar{s} + p_i)$ is given by the angle between the vector from $-p_i$ to $\bar{s}$ and the positive real axis. Now, consider a zero $-z_i$ that is outside the contour C. As $\bar{s}$ moves around the contour C and comes back to its starting point, the vector $\bar{s} + z_i$ swings up and down, but it does not swing all the way around. As a result, the net change in $\angle(\bar{s} + z_i)$ is $0°$. The same analysis holds for a pole outside the contour C.

Now consider a zero $-z_j$ inside the contour C. As $\bar{s}$ moves around C, the vector $\bar{s} + z_j$ turns all the way around, and the net change in $\angle(\bar{s} + z_j)$ is therefore $-360°$. Similarly, if we consider a pole $-p_j$ inside C, the net change in $\angle(\bar{s} + p_j)$ is also $-360°$.

If we put this all together, we see that every zero and pole inside the contour C induces a net phase change of $-360°$ as $\bar{s}$ moves around C, and every zero and pole outside the contour C induces a net phase change of $0°$. Let Z denote the number of zeros of $H(s)$ inside the contour C, and let P denote the number of poles of $H(s)$ inside the contour C. From the earlier expression for $\angle H(\bar{s})$, we see that $\angle H(\bar{s})$ undergoes a net change of $-(Z - P)360°$ as $\bar{s}$ moves around the contour C. Since each net change of $-360°$ means that the vector from the origin to $H(\bar{s})$ swings clockwise around the origin for one full rotation, a net change of $-(Z - P)360°$ means that the contour $H(C)$ must encircle the origin in the clockwise direction $Z - P$ times. This leads us to the following principle.

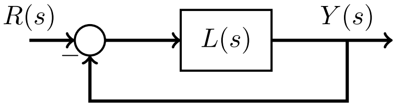
The Principle of the Argument: Let C be a simple closed clockwise contour, and consider the contour $H(C)$ which is generated by evaluating a function $H(s)$ on C. The contour $H(C)$ will encircle the origin in a clockwise direction $Z - P$ times, where Z is the number of zeros and P is the number of poles of $H(s)$ that are contained inside C.

Note: The reason for calling this the Principle of the Argument is that the phase of a complex number is also sometimes called the *argument* of the number, and the above principle is derived by considering how the phase of $H(\bar{s})$ changes as $\bar{s}$ moves around the contour C.

Also note that we assume that the contour C does not pass through any of the poles or zeros of the transfer function (because the phase contribution of a zero or pole is undefined if we evaluate the contour at that point). Similarly, the above argument only applies if the contour $H(C)$ does not pass through the origin; the number of encirclements of the origin is undefined otherwise.

16.1 Nyquist Plots

We will now use the Principle of the Argument to study the stability of a closed loop system. Consider a system $L(s)$ placed in the following feedback loop:



The transfer function from r to y is given by

$$T_{ry}(s) = \frac{L(s)}{1 + L(s)} \ ,$$

and the closed loop poles of this system are the set of all values s such that $1 + L(s) = 0$. Define $H(s) = 1 + L(s)$ (note that $H(s)$ is a transfer function). Specifically, if we denote $L(s) = \frac{N(s)}{D(s)}$, we obtain

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

Note that the poles of $H(s)$ are actually the poles of $L(s)$ (which are the open-loop poles). The zeros of $H(s)$ are the poles of the transfer function $T_{ry}(s)$, and thus they are the closed loop poles of the system. To determine how many closed loop poles lie in the CRHP, we use the Principle of the Argument. First, define the contour C as follows:

This contour encloses the entire right half plane. Part C_1 contains points of the form $s = j\omega$ as ω ranges from 0 to ∞, and part C_2 contains points of the form $s = -j\omega$, as ω ranges from ∞ to 0. Parts C_1 and C_2 form the left boundary of the right half plane. Part C_3 is a semicircle of infinite radius, and can be described by the points $s = \epsilon e^{j\theta}$, where $\epsilon \to \infty$, and θ starts at 90 degrees and ends at -90 degrees. Suppose we evaluate $H(s)$ on C to produce the contour $H(C)$; for this choice of C, $H(C)$ is called the **Nyquist Plot** of H. If we let N denote the number of times $H(C)$ encircles the origin clockwise, the Principle of the Argument tells us that

$$N = Z - P \ ,$$

where

- Z is number of zeros of $H(s)$ enclosed by the contour C (which corresponds to the number of closed loop poles in the right half plane),
- P is the number of poles of $H(s)$ enclosed by the contour C (corresponding to the number of open loop poles in the right half plane).

Since P is known (for example, by knowing $L(s)$, or by assuming that $L(s)$ is stable), and since N can be determined by looking at the contour $H(C)$, we can figure out Z, and thus determine the stability of the closed loop system. We will see an example of a Nyquist plot in the next example, and then we will discuss how to draw these plots by hand.

Example. Suppose $L(s) = \frac{10}{(s+1)^3}$. Determine the stability of the closed loop system by examining the Nyquist plot of H, where $H(s) = 1 + L(s)$, and C is the contour containing the right half plane.
Solution.

In order to apply the above technique, we needed to draw the Nyquist plot of H (which is the contour $H(C)$). We can relate the Nyquist plot of H to the Nyquist plot of the open loop system $L(s)$ by noting that $H(s) = 1 + L(s)$. The contour $H(C)$ is thus obtained by shifting the contour $L(C)$ one unit to the right in the complex plane:

Encirclements of the origin by $H(C)$ thus correspond to encirclements of the point -1 on the real axis by $L(C)$. This means that we can simply focus on drawing the Nyquist plot of L, and seeing how many times it encircles the -1 point. Next, we will see how to draw the Nyquist plot of L.

16.2 Drawing Nyquist Plots

Consider again the contour C:

The Nyquist plot of $L(s)$ is obtained by combining the contours $L(C_1), L(C_2)$ and $L(C_3)$, where C_1, C_2 and C_3 are the three portions of the contour C. We will now examine how to draw each of these contours.

Contour C_1

Note that the contour C_1 is made up of points of the form $s = j\omega$, as ω ranges from 0 to ∞. Each point on the contour $L(C_1)$ is then of the form $L(j\omega)$, which is just a complex number with magnitude $|L(j\omega)|$ and phase $\angle L(j\omega)$. We have access to these quantities from the Bode plot of $L(s)$, and so we can draw the contour $L(C_1)$ by drawing the magnitude and phase plots from the Bode plot together in the complex plane.

Example. Consider $L(s) = \frac{10}{(s+1)^2}$. Draw the contour $L(C_1)$.
Solution.

Example. Consider $L(s) = \frac{10}{(s+1)^3}$. Draw the contour $L(C_1)$.
Solution.

Contour C_2

Now that we have drawn the contour $L(C_1)$, let us turn our attention to the contour $L(C_2)$. Note that the points on C_2 are of the form $s = -j\omega$, as ω ranges from ∞ to 0. The points on the contour $L(C_2)$ are thus of the form $L(-j\omega)$, which is the complex conjugate of $L(j\omega)$. The magnitude of $L(j\omega)$ and $L(-j\omega)$ are the same, but the phases are negatives of each other. This means that the contour $L(C_2)$ is simply a mirrored version of $L(C_1)$ about the real axis. The contour $L(C_2)$ can now be added to the plots in the examples above.

Contour C_3

The contour C_3 is described by points of the form $s = \epsilon e^{j\theta}$, where $\epsilon \to \infty$, and θ ranges from 90° to −90°. The contour $L(C_3)$ is made up of points of the form $L(\epsilon e^{j\theta})$, and each of these points can be evaluated by substituting $\epsilon e^{j\theta}$ into $L(s)$. Specifically, note that since ϵ is taken to be very large (infinite, in fact), this term will dominate every factor that it appears in. Thus, if $L(s)$ is strictly proper, $L(\epsilon e^{j\theta})$ will simply evaluate to zero (and thus $L(C_3)$ is a single point at the origin). If $L(s)$ is nonstrictly proper, then $L(\epsilon e^{j\theta})$ will be some constant. This will become clearer by evaluating $L(C_3)$ for the previous two examples, and also from the following additional example.

Example. Draw the Nyquist plot of $L(s) = \frac{s+1}{s+10}$.
Solution.

16.2.1 Nyquist Plots For Systems With Poles/Zeros On The Imaginary Axis

The Principle of the Argument assumes that the contour C does not pass through any of the zeros or poles of the function $H(s)$ (because the angle contribution of a pole or zero is undefined when we evaluate the contour at that point). However, there may be cases when we want to draw the Nyquist plot of a system $L(s)$ that has poles or zeros on the imaginary axis. In this case, we modify the standard contour C to take a small detour around the poles or zeros:

For example, the new portion C_4 on the contour is described by points of the form $s = \epsilon e^{j\theta}$ where $\epsilon \to 0$, and θ ranges from -90° to 90°. We can evaluate $L(C_4)$ by substituting $s = \epsilon e^{j\theta}$ into $L(s)$, and examining what happens as $\epsilon \to 0$ (similarly to what was done for the portion C_3). A few examples will make this clear.

Example. Consider $L(s) = \frac{1}{s(s+1)}$. Draw the Nyquist Plot of $L(s)$.
Solution.

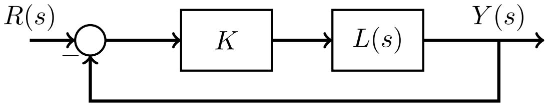
Example. Consider $L(s) = \frac{1}{s^2(s+1)}$. Draw the Nyquist Plot of $L(s)$.
Solution.

Example. Consider $L(s) = \frac{s(s+1)}{(s+10)^2}$. Draw the Nyquist Plot of $L(s)$.
Solution.

16.3 Stability Margins from Nyquist Plots

Now that we have seen how to establish stability of a closed loop system by drawing the Nyquist plot of the open loop system $L(s)$, we can return to the topic of stability margins. Specifically, when we were looking at Bode plots, we considered gain and phase margins as indicators of how stable the closed loop system was. Since the Nyquist plot tells us about the stability of the system, it can also provide us with the gain and phase margins.

To see this, consider the closed loop system:



The gain K simply scales the Nyquist plot (since the magnitude on every point of the contour $KL(C)$ simply gets multiplied by K). In other words, increasing K serves to push all of the points on the Nyquist plot further away from the origin. We will see this from the following example.

Example. Draw the Nyquist plot of $KL(s)$ for $K = 1$ and $K = 200$, where $L(s) = \frac{1}{s(s+1)(s+10)}$.
Solution.

Note that the system is stable for small K, but unstable for large K (this can be confirmed from the root locus of $L(s)$):

Recall that the gain margin is defined as the factor by which K can be increased before the closed loop system becomes unstable (we will define the gain margin only for systems that are closed loop stable for an initial value of K). In the Nyquist plot, this corresponds to scaling the plot so that the number of encirclements of the -1 point changes. Note that this is in complete accordance with the Bode plot analysis. Specifically, the point -1 in the complex plane is a complex number with magnitude 1 and phase $-180°$. When we looked at Bode plots, we showed that imaginary axis crossings occur when the gain crossover frequency and phase crossover frequency coincide (which corresponds to the case where $KL(j\omega) = -1$).

Similarly, the phase margin is defined as the amount by which the angle of $L(j\omega_{cg})$ exceeds $-180°$, where ω_{cg} is the point where $|L(j\omega_{cg})| = 1$. In the Nyquist plot, this can be obtained in the following way. First, draw a line from the origin to the point where the Nyquist plot crosses a circle of radius 1 centered at the origin. This crossing point corresponds to $|L(j\omega)| = 1$. The phase margin is then the angle between this line and the negative real axis:

Example. Identify the gain margin and phase margin for the example given above (with $L(s) = \frac{1}{s(s+1)(s+10)}$ and $K = 1$).

Example. Investigate the stability (and associated margins) for the system $KL(s) = \frac{10(s+1)}{s(s-1)}$.
Solution.

Nyquist Plots With Uncertainty and Delay

One of the benefits of Nyquist plots and Bode plots is that they provide a means to deal with uncertainty in the system. Most systems are susceptible to high frequency noise, and it is usually hard to get a good indication of the magnitude and phase of $L(j\omega)$ for large values of ω. In this cases, we usually have a region of uncertainty in the Nyquist plots and Bode plots for high frequencies:

The gain margin and phase margin are clearly useful concepts for dealing with this issue. Typically, we would like to design the system to have sufficiently high margins so that the closed loop system is stable even with the **worst case** uncertainty.

Another benefit of Nyquist and Bode plots is that they can readily handle delays in the system. For example, consider the system $\ddot{y}(t) + 2\dot{y}(t) + y(t) = u(t - T)$. In this system, the output at time t is a function of the input at time $t - T$. Since the Laplace transform of a delayed signal $u(t-T)$ is $e^{-sT}U(s)$, the transfer function for this system is given by

$$L(s) = \frac{Y(s)}{U(s)} = \frac{e^{-sT}}{(s+1)^2} \quad .$$

When $s = j\omega$, the term $e^{-j\omega T}$ is a complex number with magnitude 1 and phase $-\omega T$. This term has the effect of subtracting ωT radians from the phase at each frequency ω on Bode plot of $\frac{1}{(s+1)^2}$:

The Nyquist plot of $L(s)$ thus looks like this:

Once again, the phase margin and gain margin come in handy, as they give us an indication of the maximum delay that the system can tolerate before going unstable. More specifically, note that a large delay can cause the Nyquist plot to rotate enough that the number of encirclements of -1 changes, indicating that the closed loop system becomes unstable. In other words, a large delay can destabilize a feedback loop that is otherwise stable!

Chapter 17

Modern Control Theory: State Space Models

Up to this point, we have been analyzing and designing control systems by using a transfer-function approach (which allowed us to conveniently model the system, and use techniques such as root-locus, Bode plots and Nyquist plots). These techniques were developed and studied during the first half of the twentieth century in an effort to deal with issues such as noise and bandwidth issues in communication systems. Transfer function methods have various drawbacks however, since they cannot deal with nonlinear systems, are not very convenient when considering systems with multiple inputs and outputs, and are difficult to use for formulating 'optimal' control strategies. Starting in the 1950's (around the time of the space race), control engineers and scientists started turning to **state-space** models of control systems in order to address some of these issues. These are purely time-domain ordinary differential equation models of systems, and are able to effectively represent concepts such as the internal state of the system, and also present a method to introduce optimality conditions into the controller design procedure. This chapter will provide an introduction to the state-space approach to control design (sometimes referred to as "modern control").

17.1 State-Space Models

State-space models are simply a set of differential equations defining a system, where the highest derivative in each equation is of order 1. To derive these models, it is easiest to start with an example. Consider the system that is given by the differential equation

$$\ddot{y} + 3\dot{y} + 2y = 4u \ .$$

The transfer function for this system can be readily found to be $H(s) = \frac{Y(s)}{U(s)} = \frac{4}{s^2+3s+2}$. To represent this model in state-space form, we first draw an **all-integrator block diagram** for this system. Specifically, the all-integrator block diagram is simply a set of integrator blocks that are chained together according to the constraints imposed by the system. To obtain the diagram for this system, we first solve for the highest derivative:

$$\ddot{y} = -2y - 3\dot{y} + 4u \ .$$

Starting from the highest derivative ($\ddot{y}$), we need to somehow obtain the lower derivatives y and $\dot{y}$. This can be done by integrating $\ddot{y}$ twice, so we chain together two integrator blocks. From this, we can easily obtain the all-integrator block diagram as:

Each integrator block in this diagram can be viewed as representing one of the internal **states** of the system. Let us assign a **state variable** to the output of each integrator in order to represent the states. In this case, we will use the state variables x_1 and x_2 defined as

$$x_1 = y, \quad x_2 = \dot{y} \ .$$

Using the all integrator block diagram, we can differentiate each of the state variables to obtain

$$\begin{aligned} \dot{x}_1 &= \dot{y} = x_2 \\ \dot{x}_2 &= \ddot{y} = -2y - 3\dot{y} + 4u = -2x_1 - 3x_2 + 4u \\ y &= x_1 \ . \end{aligned}$$

The above **first-order differential equations** form the state-space model of the system. We can represent this model more concisely in **matrix-vector** form as

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 0 \\ 4 \end{bmatrix}}_{\mathbf{B}} u$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\mathbf{x}} \ .$$

General linear state-space models are thus given by

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}\\ \mathbf{y} &= \mathbf{C}\mathbf{x}\ .\end{aligned} \tag{17.1}$$

- The vector $\mathbf{x}$ is called the **state vector** of the system. We will denote the number of states in the system by n, so that $\mathbf{x} \in \mathbb{R}^n$. The quantity n is often called the **order** of the system. In the above example, we have $n = 2$.
- In general, we might have multiple inputs $u_1, u_2, \ldots, u_m$ to the system. In this case, we can define an **input vector** $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix}'$ (the notation $\mathbf{M}'$ indicates the transpose of matrix $\mathbf{M}$). In the above example, $m = 1$.
- In general, we might have multiple outputs $y_1, y_2, \ldots, y_p$. In this case, we can define the **output vector** $\mathbf{y} = \begin{bmatrix} y_1 & y_2 \cdots & y_p \end{bmatrix}'$. Note that each of these outputs represents a sensor measurement of some of the states of the system. In the above example, we have $p = 1$.
- The **system matrix A** is an $n \times n$ matrix representing how the states of the system affect each other.
- The **input matrix B** is an $n \times m$ matrix representing how the inputs to the system affect the states. The inputs are applied by m actuators.
- The **output matrix C** is a $p \times n$ matrix representing the portions of the states that are measured by the outputs. The outputs are provided by p sensors.

Let us consider another example with multiple inputs and outputs.

Example. Derive the state-space model for the following mass-spring-damper system. The inputs to the system are the forces F_1 and F_2, and we would like to measure the positions of each of the masses.
Solution.

The states in state-space models often represent physical quantities in the system. For example, one common model[1] for an F-8 aircraft contains four states:

- V: the horizontal-velocity deviation in feet/second.
- γ: the flight-path angle in radians.
- α: the angle of attack in radians.
- q: the pitch rate in radians/second.

Furthermore, the input to the system is applied via a deflection in the elevator angle, and is denoted by δ_e. These quantities are shown visually below:

The state-space model for this system is given by

$$\underbrace{\begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{\alpha} \\ \dot{q} \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} -1.357 \times 10^{-2} & -32.2 & -46.3 & 0 \\ 1.2 \times 10^{-4} & 0 & 1.214 & 0 \\ -1.212 \times 10^{-4} & 0 & -1.214 & 1 \\ 5.7 \times 10^{-4} & 0 & -9.01 & -6.696 \times 10^{-1} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} V \\ \gamma \\ \alpha \\ q \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} -0.433 \\ 0.1394 \\ -0.1394 \\ -0.1577 \end{bmatrix}}_{\mathbf{B}} \delta_e$$

$$\mathbf{y} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{C}} \mathbf{x}.$$

Note that real aircraft dynamics are more complicated than this, and are nonlinear; however they can be approximated by choosing the dominant states of the system, and linearizing the dynamics. We will look at this in more detail next.

[1] See the paper *Linear Regulator Design for Stochastic Systems by a Multiple Time-Scales Method* by Teneketzis and Sandell, IEEE Transactions in Automatic Control, vol 22, no. 4, Aug 1977, pp. 615-621, for more details.

17.2 Nonlinear State-Space Models and Linearization

Although we have been focusing on linear systems so far in the course, many practical systems are **nonlinear**. Since state-space models are time-domain representations of systems, they can readily capture nonlinear dynamics.

Example: Pendulum.

While we can represent nonlinear systems using state-space models, we have much better analysis techniques and tools for linear systems. Thus, nonlinear systems are frequently approximated by linear systems through a process known as **linearization**. For example, suppose that we are interested in the model of the pendulum when it is close to vertical (i.e., when θ is close to zero). In this case, note that $\sin\theta \approx \theta$ for $\theta \approx 0$:

The second state equation becomes $\dot{x}_2 = -\frac{g}{l}x_1 + \frac{1}{ml^2}T_e$, and thus the nonlinear pendulum model can be approximated by the linear model

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}T_e \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{x}\ .\end{aligned}$$

Next, we will see how to do this linearization more systematically.

17.2.1 Linearization via Taylor Series

Consider a general nonlinear differential equation of the form

$$\dot{x} = f(x),$$

where x is a real-valued signal and $f(\cdot)$ is some nonlinear function. In order to linearize general nonlinear systems of this form, we will use the **Taylor Series** expansion of functions. Suppose that $\bar{x}$ is a point such that $f(\bar{x}) = 0$. In this case, the point $\bar{x}$ is called an **equilibrium point** of the above system, since we have $\dot{x} = 0$ when $x = \bar{x}$ (i.e., the system reaches an equilibrium at $\bar{x}$). Recall that the Taylor Series expansion of $f(x)$ around the point $\bar{x}$ is given by

$$f(x) = f(\bar{x}) + \left.\frac{df}{dx}\right|_{x=\bar{x}} (x-\bar{x}) + \frac{1}{2}\left.\frac{d^2 f}{dx^2}\right|_{x=\bar{x}} (x-\bar{x})^2 + \frac{1}{6}\left.\frac{d^3 f}{dx^3}\right|_{x=\bar{x}} (x-\bar{x})^3 + \cdots \ .$$

This can be written as

$$f(x) = f(\bar{x}) + \underbrace{\left.\frac{df}{dx}\right|_{x=\bar{x}}}_{a}(x - \bar{x}) + \text{ higher order terms.}$$

For x sufficiently close to $\bar{x}$, these higher order terms will be very close to zero, and so we can drop them to obtain the approximation

$$f(x) \approx f(\bar{x}) + a(x - \bar{x}) \ .$$

Since $f(\bar{x}) = 0$, the nonlinear differential equation $\dot{x} = f(x)$ can be approximated near the equilibrium point by

$$\dot{x} = a(x - \bar{x}) \ .$$

To complete the linearization, we define the **perturbation state** (also known as **delta state**) $\delta x = x - \bar{x}$, and using the fact that $\delta\dot{x} = \dot{x}$, we obtain the linearized model

$$\delta\dot{x} = a\delta x \ .$$

Note that this linear model is valid **only** near the equilibrium point (how “near” depends on how nonlinear the function is).

Extension To Functions of Multiple States and Inputs

The extension to functions of multiple states and inputs is very similar to the above procedure. Suppose the evolution of state x_i is given by

$$\dot{x}_i = f_i(x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_m) \ ,$$

for some general function f_i. Suppose that the equilibrium points are given by $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m$, so that

$$f_i(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m) = 0 \quad \forall i \in \{1, 2, \ldots, n\} \ .$$

Note that the equilibrium point should make *all* of the functions f_i equal to zero, so that all states in the system stop moving when they reach equilibrium. The linearization of f_i about the equilibrium point is then given by

$$f_i(x_1, \ldots, x_n, u_1, \ldots, u_m) \approx \sum_{j=1}^{n} \left.\frac{\partial f_i}{\partial x_j}\right|_{x_j = \bar{x}_j} (x_j - \bar{x}_j) + \sum_{j=1}^{m} \left.\frac{\partial f_i}{\partial u_j}\right|_{u_j = \bar{u}_j} (u_j - \bar{u}_j) \ .$$

If we define the delta states and inputs $\delta x_j = x_j - \bar{x}_j$ (for $1 \le j \le n$) and $\delta u_j = u_j - \bar{u}_j$ (for $1 \le j \le m$), the linearized dynamics of state x_i are given by

$$\delta \dot{x}_i = \sum_{j=1}^{n} \left.\frac{\partial f_i}{\partial x_j}\right|_{x_j = \bar{x}_j} \delta x_j + \sum_{j=1}^{m} \left.\frac{\partial f_i}{\partial u_j}\right|_{u_j = \bar{u}_j} \delta u_j \ .$$

Note: Sometimes the “δ” notation is dropped in the linearized equation, with the implicit understanding that we are working with a linearized system.

Example. Linearize the nonlinear state-space model

$$\begin{aligned} \dot{x}_1 &= x_1^2 + \sin x_2 - 1 \\ \dot{x}_2 &= -x_2^3 + u \\ y &= x_1 + x_2 \end{aligned}$$

around the equilibrium point $\bar{x}_1 = 1, \bar{x}_2 = 0, \bar{u} = 0$.
Solution.

17.3 The Transfer Function of a Linear State-Space Model

Since we can generally convert nonlinear models to a linear model (in a small region around the equilibrium point), we will focus on linear state-space models of the form

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, \mathbf{y} \in \mathbb{R}^p \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \ .\end{aligned}$$

for the rest of the chapter. Since this model represents a linear system, we can ask how the matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ relate to the transfer function of the system. To see this, take the Laplace Transform of the above state space equations:

$$\begin{aligned}s\mathbf{X}(s) - \mathbf{x}(0) &= \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s) \\ \mathbf{Y}(s) &= \mathbf{C}\mathbf{X}(s) \ .\end{aligned}$$

Note that this includes the initial conditions of all the states. The first equation can be rearranged to solve for $X(s)$ as follows:

$$(s\mathbf{I} - \mathbf{A})\mathbf{X}(s) = \mathbf{x}(0) + \mathbf{B}\mathbf{U}(s) \Leftrightarrow \mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}(0) + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) \ .$$

The term $\mathbf{I}$ represents the $n \times n$ identity matrix. Substituting this into the equation for $\mathbf{Y}(s)$, we obtain

$$\mathbf{Y}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}(0) + \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{U}(s) \ .$$

The transfer function of the state-space model $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, $\mathbf{y} = \mathbf{C}\mathbf{x}$ (when $\mathbf{x}(0) = 0$) is

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

Note that $\mathbf{H}(s)$ is a $p \times m$ matrix, and thus it is a generalization of the transfer function for standard single-input single-output systems. In fact, it is a matrix where entry i, j is a transfer function describing how the j–th input affects the i–th output. When $p = 1$ and $m = 1$, we get the transfer function that we studied in the first part of the course.

Example. Calculate the transfer function for the state space model

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}}_{\mathbf{A}} \mathbf{x} + \underbrace{\begin{bmatrix} 0 \\ 4 \end{bmatrix}}_{\mathbf{B}} u, \quad y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{C}} \mathbf{x} \ .$$

Solution.

Note that the above solution agrees with the transfer function at the beginning of the section.

17.4 Obtaining the Poles from the State-Space Model

In the last section, we saw that the transfer function of the linear system with state-space model

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

is given by $\mathbf{H}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$. Now, note that inverse of any matrix M is given by $M^{-1} = \frac{1}{\det(M)}\text{adj}(M)$, where $\det(M)$ is the **determinant** of M and adj(M) is the **adjoint** matrix corresponding to M (this is just a matrix whose entries are polynomials obtained from the entries of M). This means that the denominator of every entry in matrix $(s\mathbf{I}-\mathbf{A})^{-1}$ is $\det(s\mathbf{I}-\mathbf{A})$, and so the poles of the transfer function $\mathbf{H}(s)$ are the values of s for which $\det(s\mathbf{I}-\mathbf{A}) = 0$.

Next, recall the definition of the *eigenvalues* and *eigenvectors* of an $n \times n$ square matrix $\mathbf{A}$. Specifically, a complex number λ and a $n \times 1$ vector $\mathbf{v}$ are said to be an eigenvalue and eigenvector, respectively, of $\mathbf{A}$ if they satisfy the equation

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}.$$

In other words, multiplying $\mathbf{A}$ by $\mathbf{v}$ produces a scaled version of $\mathbf{v}$, and the scaling factor is λ. Rearranging this equation, we get $(\lambda\mathbf{I}-\mathbf{A})\mathbf{v} = \mathbf{0}$. Recall that this equation has a nonzero solution for $\mathbf{v}$ if and only if the matrix $\lambda\mathbf{I}-\mathbf{A}$ has linearly dependent columns, which is equivalent to saying that $\det(\lambda\mathbf{I}-\mathbf{A}) = 0$. Thus every possible λ that causes the determinant to be zero is an eigenvalue of $\mathbf{A}$, and as noted above, these are also the poles of the transfer function of the linear system.

The poles of the transfer function $\mathbf{H}(s) = \mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}$ are exactly the eigenvalues of matrix $\mathbf{A}$. In other words, the poles are the values s that satisfy $\det(s\mathbf{I}-\mathbf{A}) = 0$.

Example. Find the poles of the system with $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -20 & -9 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 2 & 3 \end{bmatrix}$.

Solution.

Example. Find the poles of the system with $\mathbf{A} = \begin{bmatrix} 1 & 1 & -2 \\ 0 & -9 & 3 \\ 0 & 0 & 2 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$.

Solution.

17.5 An Overview of Design Approaches for State-Space Models

There are various properties of systems that one can analyze in the state-space domain. For the plant given by the state-space equations in (17.1), the following definitions are commonly used.

- The system (17.1) is said to be **controllable** if there exists an input signal $u(t)$ such that the state $\mathbf{x}(T)$ achieves any desired value at some time-step T.
- The system (17.1) is said to be **observable** if it is possible to recover the entire state $\mathbf{x}(t)$ by looking at the outputs $\mathbf{y}(t)$ of the system over some period of time.

Controllability is about making the system state (not just the output) behave how we want it to by applying proper inputs. Observability is about whether we can determine what the internal state of the system is doing by looking at the outputs. Both of these concepts play central roles in modern control systems design.

When it comes to stabilization of systems in the state-space domain, one commonly considers the use of **linear state feedback**. Specifically, suppose that we have access to the entire state $\mathbf{x}(t)$ of the system for all time; this is unrealistic, since we only have access to $\mathbf{y}(t)$, which measures only a few of the states, but let's just assume it for now. Linear state feedback control applies an input of the form

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t),$$

for some matrix $\mathbf{K}$ that we will choose. The closed loop system is then

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}\mathbf{x} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x}.$$

The dynamics of the closed loop system are given by the matrix $\mathbf{A} - \mathbf{B}\mathbf{K}$; specifically, as discussed in the previous section, the poles of this system are given by the eigenvalues of this matrix. Thus, in order to obtain a stable system, we have to choose $\mathbf{K}$ so that all eigenvalues of $\mathbf{A} - \mathbf{B}\mathbf{K}$ are stable (i.e., in the OLHP). It turns out that it is possible to do this if the system is controllable.

The above feedback mechanism assumed that we have access to the entire state. Since we only have access to the measurements of a few state variables (provided by the output $\mathbf{y}(t)$), one strategy would be to try to reconstruct the entire state, based on the measurements available. This is possible if the system is observable, in which case one can construct a **state-estimator** that provides an estimate of $\mathbf{x}(t)$ to be used with the linear state feedback input described above. The architecture of state feedback control with a state estimator looks like this:

The details of these topics, along with issues such as choosing the inputs optimally, dealing with noise, etc., are treated in more advanced undergraduate and graduate courses. Hopefully this course has piqued your interest in control systems, and motivated you to learn more about this subject in future courses!

The End.