

Notes for ECE-320

Winter 2011-2012

by
R. Throne

Contents

1	Table of Laplace Transforms	5
2	Laplace Transform Review	6
2.1	Poles and Zeros	6
2.2	Proper and Strictly Proper Transfer Functions	6
2.3	Impulse Response and Transfer Functions	6
2.4	Partial Fractions with Distinct Poles	7
2.5	Partial Fractions with Distinct and Repeated Poles	10
2.6	Complex Conjugate Poles: Completing the Square	15
2.7	Common Denominator/Cross Multiplying	19
2.8	Complex Conjugate Poles-Again	21
3	Final Value Theorem and the Static Gain of a System	22
4	Step Response, Ramp Response, and Steady State Errors	24
4.1	Step Response and Steady State Error	24
4.2	Ramp Response and Steady State Error	28
4.3	Summary	31
5	Response of a Ideal Second Order System	33
5.1	Step Response of an Ideal Second Order System	33
5.2	Time to Peak, T_p	34
5.3	Percent Overshoot, PO	35
5.4	Settling Time, T_s	36
5.5	Constraint Regions in the s -Plane	38
5.6	Summary	45
6	Characteristic Polynomial, Modes, and Stability	47
6.1	Characteristic Polynomial, Equation, and Modes	47
6.2	Characteristic Mode Reminders	48
6.3	Stability	49
6.4	Settling Time and Dominant Poles	49
7	Time Domain Response and System Bandwidth	51
8	Block Diagrams	59
8.1	Basic Feedback Configuration	59
8.2	Mason's Gain Formula	60
9	Model Matching	67
9.1	ITAE Optimal Systems	69
9.2	Deadbeat Systems	70
9.3	Summary and Caveates	71

10 System Type and Steady State Errors	73
10.1 Review	73
10.2 System Type For a Unity Feedback Configuration	73
10.3 Steady State Errors for Step and Ramp Inputs	74
10.4 Examples	75
11 Controller Design Using the Root Locus	79
11.1 Standard Root Locus Form	79
11.2 Examples	81
11.3 Loci Branches	82
11.4 Real Axis Segments	83
11.5 Asymptotic Angles and Centroid of the Asymptotes	88
11.6 Common Industrial Controller Types	98
11.7 Controller and Design Constraint Examples	100
11.8 Seemingly Odd Root Locus Behavior	124
12 System Sensitivity	127
12.1 Sensitivity to Parameter Variations	127
12.2 Sensitivity to External Disturbances	132
12.3 Summary	133
13 State Variables and State Variable Feedback	134
13.1 State Variable to Transfer Function Model	136
13.2 State Variable Feedback	139
13.3 Controllability for State Variable Systems	144
13.4 Summary	145
16 z-Transforms	146
16.1 Special Functions	146
16.2 Impulse Response and Convolution	146
16.3 A Useful Summation	148
16.4 z -Transforms	151
16.5 z -Transform Properties	153
16.6 Inverse z -Transforms	156
16.7 Second Order Transfer Functions with Complex Conjugate Poles	161
16.8 Solving Difference Equations	165
16.9 Asymptotic Stability	168
16.10 Mapping Poles and Settling Time	168
16.11 Sampling Plants with Zero Order Holds	170
16.12 Final Notes	174
14 Transfer Function Based Discrete-Time Control	175
14.1 Implementing Discrete-Time Transfer Functions	175
14.2 Not Quite Right	175
14.3 PID and Constant Prefilters	176

15 Discrete-Time State Equations	180
15.1 The Continuous-Time State Transition Matrix	180
15.2 Solution of the Continuous-Time State Equations	182
15.3 Computing the State Transition Matrix, e^{At}	183
15.3.1 Truncating the Infinite Sum	183
15.3.2 Laplace Transform Method	184
15.4 Discretization with Delays in the Input	186
15.5 State Variable to Transfer Function	189
15.6 Poles and Eigenvalues	190
16 State Variable Feedback	191
16.1 Pole Placement by Transfer Functions	192
16.2 State Feedback Examples	194
16.3 General Guidelines for State Feedback Pole Locations	196
17 Frequency Domain Analysis	202
17.1 Phase and Gain Margins	202
18 Lead Controllers for Increasing Phase Margin	209
18.1 Algorithm for Controller Design Using Bode Plots	211
18.2 Examples	212
A Matlab Commands	i
A.1 Figures	i
A.2 Transfer Functions	i
A.3 Feedback Systems	ii
A.4 System Response to Arbitrary Inputs	ii
A.5 Changing the Line Thickness	iii
A.6 Poles and Zeros	iv
A.7 Roots and Polynomials	iv
A.8 Root Locus Plots	v
A.9 Bode Plots, Gain and Phase Margins	v

1 Table of Laplace Transforms

$f(t)$	$F(s)$
$\delta(t)$	1
$u(t)$	$\frac{1}{s}$
$tu(t)$	$\frac{1}{s^2}$
$\frac{t^{n-1}}{(n-1)!}u(t)$ ($n = 1, 2, 3, \dots$)	$\frac{1}{s^n}$
$t^n u(t)$ ($n = 1, 2, 3, \dots$)	$\frac{n!}{s^{n+1}}$
$e^{-at}u(t)$	$\frac{1}{s+a}$
$te^{-at}u(t)$	$\frac{1}{(s+a)^2}$
$\frac{1}{(n-1)!}t^{n-1}e^{-at}u(t)$ ($n = 1, 2, 3, \dots$)	$\frac{1}{(s+a)^n}$
$t^n e^{-at}u(t)$ ($n = 1, 2, 3, \dots$)	$\frac{n!}{(s+a)^{n+1}}$
$\sin(bt)u(t)$	$\frac{b}{s^2+b^2}$
$\cos(bt)u(t)$	$\frac{s}{s^2+b^2}$
$e^{-at}\sin(bt)u(t)$	$\frac{b}{(s+a)^2+b^2}$
$e^{-at}\cos(bt)u(t)$	$\frac{(s+a)}{(s+a)^2+b^2}$

2 Laplace Transform Review

In this course we will be using Laplace transforms extensively. Although we do not often go from the s-plane to the time domain, it is important to be able to do this and to understand what is going on. In what follows is a brief review of some results with Laplace transforms.

2.1 Poles and Zeros

Assume we have the transfer function

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

where $N(s)$ and $D(s)$ are polynomials in s with no common factors. The roots of $N(s)$ are the **zeros** of the system, while the roots of $D(s)$ are the **poles** of the system.

2.2 Proper and Strictly Proper Transfer Functions

The transfer function

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

is **proper** if the degree of the polynomial $N(s)$ is less than or equal to the degree of the polynomial $D(s)$. The transfer function $H(s)$ is **strictly proper** if the degree of $N(s)$ is less than the degree of $D(s)$.

2.3 Impulse Response and Transfer Functions

If $H(s)$ is a transfer function, the inverse Laplace transform of $H(s)$ is call the **impulse response**, $h(t)$.

$$\begin{aligned}\mathcal{L}\{h(t)\} &= H(s) \\ h(t) &= \mathcal{L}^{-1}\{H(s)\}\end{aligned}$$

2.4 Partial Fractions with Distinct Poles

Let's assume we have a transfer function

$$H(s) = \frac{N(s)}{D(s)} = \frac{N(s)}{D(s)} = \frac{K(s+z_1)(s+z_2)\dots(s+z_m)}{(s+p_1)(s+p_2)\dots(s+p_n)}$$

where we assume $m < n$ (this makes $H(s)$ a **strictly proper** transfer function). The poles of the system are at $-p_1, -p_2, \dots, -p_n$ and the zeros of the system are at $-z_1, -z_2, \dots, -z_m$. Since we have distinct poles, $p_i \neq p_j$ for all i and j . Also, since we assumed $N(s)$ and $D(s)$ have no common factors, we know that $z_i \neq p_j$ for all i and j .

We would like to find the corresponding impulse response, $h(t)$. To do this, we assume

$$H(s) = \frac{N(s)}{D(s)} = a_1 \frac{1}{s+p_1} + a_2 \frac{1}{s+p_2} + \dots + a_n \frac{1}{s+p_n}$$

If we can find the a_i , it will be easy to determine $h(t)$ since the only inverse Laplace transform we need is that of $\frac{1}{s+p}$, and we know (or can look up) $\frac{1}{s+p} \leftrightarrow e^{-pt}u(t)$. To find a_1 , we first multiply by $(s+p_1)$,

$$(s+p_1)H(s) = a_1 + a_2 \frac{s+p_1}{s+p_2} + \dots + a_n \frac{s+p_1}{s+p_n}$$

and then let $s \rightarrow -p_1$. Since the poles are all distinct, we will get

$$\lim_{s \rightarrow -p_1} (s+p_1)H(s) = a_1$$

Similarly, we will get

$$\lim_{s \rightarrow -p_2} (s+p_2)H(s) = a_2$$

and in general

$$\lim_{s \rightarrow -p_i} (s+p_i)H(s) = a_i$$

Example 1. Let's assume we have

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

and we want to determine $h(t)$. Since the poles are distinct, we have

$$H(s) = \frac{(s+1)}{(s+2)(s+3)} = a_1 \frac{1}{s+2} + a_2 \frac{1}{s+3}$$

Then

$$a_1 = \lim_{s \rightarrow -2} (s+2) \frac{(s+1)}{(s+2)(s+3)} = \lim_{s \rightarrow -2} \frac{(s+1)}{(s+3)} = \frac{-1}{1} = -1$$

and

$$a_2 = \lim_{s \rightarrow -3} (s+3) \frac{(s+1)}{(s+2)(s+3)} = \lim_{s \rightarrow -3} \frac{(s+1)}{(s+2)} = \frac{-2}{-1} = 2$$

Then

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

and hence

$$h(t) = -e^{-2t}u(t) + 2e^{-3t}u(t)$$

It is often unnecessary to write out all of the steps in the above example. In particular, when we want to find a_i we will always have a cancellation between $(s+p_i)$ in the numerator with the $(s+p_i)$ in the denominator. Using this fact, when we want to find a_i we can just ignore (or *cover up*) the factor $(s+p_i)$ in the denominator. For our example above, we then have

$$\begin{aligned} a_1 &= \lim_{s \rightarrow -2} \frac{(s+1)}{\blacksquare(s+3)} = \frac{-1}{1} = -1 \\ a_2 &= \lim_{s \rightarrow -3} \frac{(s+1)}{(s+2)\blacksquare} = \frac{-2}{-1} = 2 \end{aligned}$$

where we have covered up the poles associated with a_1 and a_2 , respectively.

Example 2. Let's assume we have

$$H(s) = \frac{s^2 - s + 2}{(s+2)(s+3)(s+4)}$$

and we want to determine $h(t)$. Since the poles are distinct, we have

$$H(s) = \frac{(s^2 - s + 2)}{(s+2)(s+3)(s+4)} = a_1 \frac{1}{s+2} + a_2 \frac{1}{s+3} + a_3 \frac{1}{s+4}$$

Using the *cover up method*, we then determine

$$\begin{aligned} a_1 &= \lim_{s \rightarrow -2} \frac{(s^2 - s + 2)}{\blacksquare(s+3)(s+4)} = \frac{8}{(1)(2)} = 4 \\ a_2 &= \lim_{s \rightarrow -3} \frac{(s^2 - s + 2)}{(s+2)\blacksquare(s+4)} = \frac{14}{(-1)(1)} = -14 \\ a_3 &= \lim_{s \rightarrow -4} \frac{(s^2 - s + 2)}{(s+2)(s+3)\blacksquare} = \frac{22}{(-2)(-1)} = 11 \end{aligned}$$

and hence

$$h(t) = 4e^{-2t}u(t) - 14e^{-3t}u(t) + 11e^{-4t}u(t)$$

Example 3. Let's assume we have

$$H(s) = \frac{1}{(s+1)(s+5)}$$

and we want to determine $h(t)$. Since the poles are distinct, we have

$$H(s) = \frac{1}{(s+1)(s+5)} = a_1 \frac{1}{s+1} + a_2 \frac{1}{s+5}$$

Using the coverup method, we then determine

$$a_1 = \lim_{s \rightarrow -1} \frac{1}{\blacksquare(s+5)} = \frac{1}{4}$$
$$a_2 = \lim_{s \rightarrow -5} \frac{1}{(s+1)\blacksquare} = \frac{1}{-4}$$

and hence

$$h(t) = \frac{1}{4}e^{-t}u(t) - \frac{1}{4}e^{-5t}u(t)$$

Although we have only examined real poles, this method is also valid for complex poles, although there are usually easier ways to deal with complex poles, as we'll see.

2.5 Partial Fractions with Distinct and Repeated Poles

Whenever there are repeated poles, we need to use a different form for the partial fractions for those poles. This is probably most easily explained by means of examples.

Example 4. Assume we have the transfer function

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

and we want to find the corresponding impulse response, $h(t)$. To do this we look for a partial fraction expansion of the form

$$H(s) = \frac{1}{(s+1)(s+2)^2} = a_1 \frac{1}{s+1} + a_2 \frac{1}{s+2} + a_3 \frac{1}{(s+2)^2}$$

Example 5. Assume we have the transfer function

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

and we want to find the corresponding impulse response, $h(t)$. To do this we look for a partial fraction expansion of the form

$$H(s) = \frac{s+1}{s^2(s+2)(s+3)} = a_1 \frac{1}{s} + a_2 \frac{1}{s^2} + a_3 \frac{1}{s+2} + a_4 \frac{1}{s+3}$$

Note that there are always as many unknowns (the a_i) as the degree of the denominator polynomial.

Now we need to be able to determine the expansion coefficients. We already know how to do this for distinct poles, so we do those first.

For **Example 4**,

$$a_1 = \lim_{s \rightarrow -1} \frac{1}{\blacksquare (s+2)^2} = \frac{1}{1} = 1$$

For **Example 5**,

$$a_3 = \lim_{s \rightarrow -2} \frac{s+1}{s^2 \blacksquare (s+3)} = \frac{-1}{(-2)^2(1)} = -\frac{1}{4}$$
$$a_4 = \lim_{s \rightarrow -3} \frac{s+1}{s^2(s+2) \blacksquare} = \frac{-2}{(-3)^2(-1)} = \frac{2}{9}$$

The next set of expansion coefficients to determine are those with the highest power of the repeated poles.

For **Example 4**, multiply through by $(s+2)^2$ and let $s \rightarrow -2$,

$$a_3 = \lim_{s \rightarrow -2} (s+2)^2 \frac{1}{(s+1)(s+2)^2} = \lim_{s \rightarrow -2} \frac{1}{s+1} = -1$$

or with the coverup method

$$a_3 = \lim_{s \rightarrow -2} \frac{1}{(s+1)\blacksquare} = \frac{1}{-1} = -1$$

For **Example 5**, multiply through by s^2 and let $s \rightarrow 0$

$$a_2 = \lim_{s \rightarrow 0} s^2 \frac{s+1}{s^2(s+2)(s+3)} = \lim_{s \rightarrow 0} \frac{s+1}{(s+2)(s+3)} = \frac{1}{6}$$

or with the coverup method

$$a_2 = \lim_{s \rightarrow 0} \frac{s+1}{\blacksquare(s+2)(s+3)} = \frac{1}{6} = \frac{1}{6}$$

So far we have:

for **Example 4**

$$\frac{1}{(s+1)(s+2)^2} = \frac{1}{s+1} + a_2 \frac{1}{s+2} - \frac{1}{(s+2)^2}$$

and for **Example 5**

$$\frac{s+1}{s^2(s+2)(s+3)} = a_1 \frac{1}{s} + \frac{1}{6} \frac{1}{s^2} - \frac{1}{4} \frac{1}{s+2} + \frac{2}{9} \frac{1}{s+3}$$

We now need to determine any remaining coefficients. There are two common ways of doing this, both of which are based on the fact that both sides of the equation must be equal for any value of s . The two methods are

1. Multiply both sides of the equation by s and let $s \rightarrow \infty$. If this works it is usually very quick.
2. Select convenient values of s and evaluate both sides of the equation for these values of s

For **Example 4**, using Method 1,

$$\lim_{s \rightarrow \infty} \left[s \frac{1}{(s+1)(s+2)^2} \right] = \lim_{s \rightarrow \infty} \left[\frac{s}{s+1} + a_2 \frac{s}{s+2} - \frac{s}{(s+2)^2} \right]$$

or

$$0 = 1 + a_2 + 0$$

so $a_2 = -1$.

For **Example 5**, using Method 1,

$$\lim_{s \rightarrow \infty} \left[s \frac{s+1}{s^2(s+2)(s+3)} \right] = \lim_{s \rightarrow \infty} \left[a_1 \frac{s}{s} + \frac{1}{6} \frac{s}{s^2} - \frac{1}{4} \frac{s}{s+2} + \frac{2}{9} \frac{s}{s+3} \right]$$

or

$$0 = a_1 + 0 - \frac{1}{4} + \frac{2}{9}$$

$$\text{so } a_1 = \frac{1}{4} - \frac{2}{9} = \frac{1}{36}$$

For **Example 4**, using Method 2, let's choose $s = 0$ (note both sides of the equation must be finite!)

$$\lim_{s \rightarrow 0} \left[\frac{1}{(s+1)(s+2)^2} \right] = \lim_{s \rightarrow 0} \left[\frac{1}{s+1} + a_2 \frac{1}{s+2} - \frac{1}{(s+2)^2} \right]$$

or

$$\frac{1}{4} = 1 + \frac{a_2}{2} - \frac{1}{4}$$

$$\text{so } a_2 = 2\left(\frac{1}{4} + \frac{1}{4} - 1\right) = -1$$

For **Example 5**, using Method 2, let's choose $s = -1$ (note that $s = 0$, $s = -2$, or $s = -3$ will not work)

$$\lim_{s \rightarrow -1} \left[\frac{s+1}{s^2(s+2)(s+3)} \right] = \lim_{s \rightarrow -1} \left[a_1 \frac{1}{s} + \frac{1}{6} \frac{1}{s^2} - \frac{1}{4} \frac{1}{s+2} + \frac{2}{9} \frac{1}{s+3} \right]$$

or

$$0 = -a_1 + \frac{1}{6} - \frac{1}{49}$$

$$\text{so } a_1 = \frac{1}{6} - \frac{1}{4} + \frac{1}{9} = \frac{1}{36}$$

Then for **Example 4**,

$$h(t) = e^{-t}u(t) - e^{-2t}u(t) - te^{-2t}u(t)$$

and for **Example 5**

$$h(t) = \frac{1}{36}u(t) + \frac{1}{6}tu(t) - \frac{1}{4}e^{-2t}u(t) + \frac{2}{9}e^{-3t}u(t)$$

In summary, for repeated and distinct poles, go through the following steps:

1. Determine the form of the partial fraction expansion. There must be as many unknowns as the highest power of s in the denominator.
2. Determine the coefficients associated with the distinct poles using the coverup method.
3. Determine the coefficient associated with the highest power of a repeated pole using the coverup method.

4. Determine the remaining coefficients by

- Multiplying both sides by s and letting $s \rightarrow \infty$
- Setting s to a convenient value in both sides of the equations. Both sides must remain finite

Example 6. Assuming

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

determine the corresponding impulse response $h(t)$.

First, we determine the correct form

$$H(s) = \frac{s^2}{(s+1)^2(s+3)} = a_1 \frac{1}{s+1} + a_2 \frac{1}{(s+1)^2} + a_3 \frac{1}{s+3}$$

Second, we determine the coefficient(s) of the distinct pole(s)

$$a_3 = \lim_{s \rightarrow -3} \frac{(s^2)}{(s+1)^2} = \frac{9}{4}$$

Third, we determine the coefficient(s) of the highest power of the repeated pole(s)

$$a_2 = \lim_{s \rightarrow -1} \frac{(s^2)}{(s+3)} = \frac{1}{2}$$

Fourth, we determine any remaining coefficients

$$\lim_{s \rightarrow \infty} \left[s \frac{s^2}{(s+1)^2(s+3)} \right] = \lim_{s \rightarrow \infty} \left[a_1 \frac{s}{s+1} + \frac{1}{2} \frac{s}{(s+1)^2} + \frac{9}{4} \frac{s}{s+3} \right]$$

or

$$1 = a_1 + 0 + \frac{9}{4}$$

$$\text{or } a_1 = 1 - \frac{9}{4} = -\frac{5}{4}.$$

Putting it all together, we have

$$h(t) = -\frac{5}{4}e^{-t}u(t) + \frac{1}{2}te^{-t}u(t) + \frac{9}{4}e^{-3t}u(t)$$

Example 7. Assume we have the transfer function

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

find the corresponding impulse response, $h(t)$.

First we determine the correct form

$$H(s) = \frac{s+3}{s(s+1)^2(s+2)^2} = a_1 \frac{1}{s} + a_2 \frac{1}{s+1} + a_3 \frac{1}{(s+1)^2} + a_4 \frac{1}{s+2} + a_5 \frac{1}{(s+2)^2}$$

Second, we determine the coefficient(s) of the distinct pole(s)

$$a_1 = \lim_{s \rightarrow 0} \frac{s+3}{(s+1)^2(s+2)^2} = \frac{3}{(1)(4)} = \frac{3}{4}$$

Third, we determine the coefficient(s) of the highest power of the repeated pole(s)

$$a_3 = \lim_{s \rightarrow -1} \frac{s+3}{s(s+2)^2} = \frac{2}{(-1)(1)} = -2$$

$$a_5 = \lim_{s \rightarrow -2} \frac{s+3}{s(s+1)^2} = \frac{1}{(-2)(1)} = -\frac{1}{2}$$

Fourth, we determine any remaining coefficients

$$\lim_{s \rightarrow \infty} \left[s \frac{s+3}{s(s+1)^2(s+2)^2} \right] = \lim_{s \rightarrow \infty} \left[\frac{3s}{4s} + a_2 \frac{s}{s+1} - 2 \frac{s}{(s+1)^2} + a_4 \frac{s}{s+2} - \frac{1}{2} \frac{s}{(s+2)^2} \right]$$

or

$$0 = \frac{3}{4} + a_2 + a_4$$

We need one more equation, so let's set $s = -3$

$$\lim_{s \rightarrow -3} \left[\frac{s+3}{s(s+1)^2(s+2)^2} \right] = \lim_{s \rightarrow -3} \left[\frac{3}{8s} + a_2 \frac{1}{s+1} - 2 \frac{1}{(s+1)^2} + a_4 \frac{1}{s+2} - \frac{1}{2} \frac{1}{(s+2)^2} \right]$$

or

$$0 = -\frac{1}{4} - a_2 \frac{1}{2} - \frac{1}{2} - a_4 - \frac{1}{2}$$

This gives us the set of equations

$$\begin{bmatrix} 1 & 1 \\ \frac{1}{2} & -1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_4 \end{bmatrix} = \begin{bmatrix} \frac{-3}{4} \\ \frac{5}{4} \end{bmatrix}$$

with solution $a_2 = 1$ and $a_4 = -\frac{7}{4}$. Putting it all together we have

$$h(t) = \frac{3}{4}u(t) + e^{-t}u(t) - 2te^{-t}u(t) + \frac{-7}{4}e^{-2t}u(t) - \frac{1}{2}te^{-2t}u(t)$$

2.6 Complex Conjugate Poles: Completing the Square

Before using partial fractions on systems with complex conjugate poles, we need to review one property of Laplace transforms:

$$\text{if } x(t) \Leftrightarrow X(s), \text{ then } e^{-at}x(t) \Leftrightarrow X(s+a)$$

To show this, we start with what we are given:

$$\mathcal{L}\{x(t)\} = \int_0^{\infty} x(t)e^{-st} dt = X(s)$$

Then

$$\mathcal{L}\{e^{-at}x(t)\} = \int_0^{\infty} e^{-at}x(t)e^{-st} dt = \int_0^{\infty} x(t)e^{-(s+a)t} dt = X(s+a)$$

The other relationships we need are the Laplace transform pairs for sines and cosines

$$\begin{aligned} \cos(bt)u(t) &\Leftrightarrow \frac{s}{s^2 + b^2} \\ \sin(bt)u(t) &\Leftrightarrow \frac{b}{s^2 + b^2} \end{aligned}$$

Finally, we need to put these together, to get the Laplace transform pair:

$$\begin{aligned} e^{-at} \cos(bt)u(t) &\Leftrightarrow \frac{s+a}{(s+a)^2 + b^2} \\ e^{-at} \sin(bt)u(t) &\Leftrightarrow \frac{b}{(s+a)^2 + b^2} \end{aligned}$$

Complex poles always result in sines and cosines. We will be trying to make terms with complex poles look like these terms by completing the square in the denominator.

In order to get the denominators in the correct form when we have complex poles, we need to complete the square in the denominator. That is, we need to be able to write the denominator as

$$D(s) = (s+a)^2 + b^2$$

To do this, we always first find a using the fact that the coefficient of s will be $2a$. Then we use whatever is needed to construct b . A few examples will hopefully make this clear.

Example 8. Let's assume

$$D(s) = s^2 + s + 2$$

and we want to write this in the correct form. First we recognize that the coefficient of s is 1, so we know $2a = 1$ or $a = \frac{1}{2}$. We then have

$$D(s) = s^2 + s + 2 = \left(s + \frac{1}{2}\right)^2 + b^2$$

To find b we expand the right hand side of the above equations and then equate powers of s :

$$D(s) = s^2 + s + 2 = \left(s + \frac{1}{2}\right)^2 + b^2 = s^2 + s + \frac{1}{4} + b^2$$

clearly $2 = b^2 + \frac{1}{4}$, or $b^2 = \frac{7}{4}$, or $b = \frac{\sqrt{7}}{2}$. Hence we have

$$D(s) = s^2 + s + 2 = \left(s + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{7}}{2}\right)^2$$

and this is the form we need.

Example 9. Let's assume

$$D(s) = s^2 + 3s + 5$$

and we want to write this in the correct form. First we recognize that the coefficient of s is 3, so we know $2a = 3$ or $a = \frac{3}{2}$. We then have

$$D(s) = s^2 + 3s + 5 = \left(s + \frac{3}{2}\right)^2 + b^2$$

To find b we expand the right hand side of the above equations and then equate powers of s :

$$D(s) = s^2 + 3s + 5 = \left(s + \frac{3}{2}\right)^2 + b^2 = s^2 + 3s + \frac{9}{4} + b^2$$

clearly $5 = b^2 + \frac{9}{4}$, or $b^2 = \frac{11}{4}$, or $b = \frac{\sqrt{11}}{2}$. Hence we have

$$D(s) = s^2 + 3s + 5 = \left(s + \frac{3}{2}\right)^2 + \left(\frac{\sqrt{11}}{2}\right)^2$$

and this is the form we need.

Now that we know how to complete the square in the denominator, we are ready to look at complex poles. We will start with two simple examples, and then explain how to deal with more complicated examples.

Example 10. Assuming

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

and we want to find the corresponding impulse response $h(t)$. In this simple case, we first complete the square, as we have done above, to write

$$H(s) = \frac{1}{\left(s + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{7}}{2}\right)^2}$$

This almost has the form we want, which is

$$e^{-at} \sin(bt)u(t) \Leftrightarrow \frac{b}{(s+a)^2 + b^2}$$

However, to use this form we need b in the numerator. To achieve this we will multiply and divide by $b = \frac{\sqrt{7}}{2}$

$$\begin{aligned} H(s) &= \frac{1}{(s + \frac{1}{2})^2 + \left(\frac{\sqrt{7}}{2}\right)^2} \\ &= \frac{1}{\frac{\sqrt{7}}{2}} \frac{\frac{\sqrt{7}}{2}}{(s + \frac{1}{2})^2 + \left(\frac{\sqrt{7}}{2}\right)^2} \end{aligned}$$

or

$$h(t) = \frac{2}{\sqrt{7}} e^{-\frac{1}{2}t} \sin\left(\frac{\sqrt{7}}{2}t\right)u(t)$$

The examples we have done so far have been fairly straightforward. In general, when we have complex conjugate poles we will look for a combination of sines and cosines. Specifically, whenever we have complex conjugate poles, we will assume the correct form for a sine or a cosine

$$\frac{\textit{whatever}}{(s+a)^2 + b^2} = \frac{Ab}{(s+a)^2 + b^2} + \frac{B(s+b)}{(s+a)^2 + b^2}$$

Here A and B are the two unknown coefficients.

Example 11. Assuming

$$H(s) = \frac{s}{s^2 + 3s + 5}$$

and we want to find the corresponding impulse response $h(t)$. In this simple case, we first complete the square, as we have done above, to write

$$H(s) = \frac{s}{(s + \frac{3}{2})^2 + \left(\frac{\sqrt{11}}{2}\right)^2}$$

Now we expand using the assumed form for the complex conjugate poles,

$$H(s) = \frac{s}{(s + \frac{3}{2})^2 + \left(\frac{\sqrt{11}}{2}\right)^2} = \frac{A\frac{\sqrt{11}}{2}}{(s + \frac{3}{2})^2 + \left(\frac{\sqrt{11}}{2}\right)^2} + \frac{B(s + \frac{3}{2})}{(s + \frac{3}{2})^2 + \left(\frac{\sqrt{11}}{2}\right)^2}$$

Now if we multiply both sides by s and let $s \rightarrow \infty$ we get $B = 1$. Next, if we substitute $s = -\frac{3}{2}$ we get

$$\frac{-\frac{3}{2}}{\frac{11}{4}} = \frac{A\frac{\sqrt{11}}{2}}{\frac{11}{4}}$$

which gives $A = -3/\sqrt{11}$. Now we have all of the parameters we need, and we have

$$h(t) = e^{-\frac{3}{2}t} \cos\left(\frac{\sqrt{11}}{2}t\right)u(t) - \frac{3}{\sqrt{11}}e^{-\frac{3}{2}t} \sin\left(\frac{\sqrt{11}}{2}t\right)u(t)$$

Note that it is possible to combine the sine and cosine terms into a single cosine with a phase angle, but we will not pursue that here. The examples we have done so far only contain complex roots. In general, we need to be able to deal with systems that have both complex and real roots. Since we are dealing with real systems in this course, all complex poles will occur in complex conjugate pairs.

Example 12. Assuming

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

and we want to determine the corresponding impulse response $h(t)$. First we need to find the correct form for the partial fractions, which means we need to complete the square in the denominator

$$\begin{aligned} H(s) &= \frac{1}{(s+2)(s^2+s+1)} = \frac{1}{(s+2)\left((s+\frac{1}{2})^2 + (\frac{\sqrt{3}}{2})^2\right)} \\ &= \frac{A}{s+2} + \frac{B\frac{\sqrt{3}}{2}}{(s+\frac{1}{2})^2 + (\frac{\sqrt{3}}{2})^2} + \frac{C(s+\frac{3}{2})}{(s+\frac{1}{2})^2 + (\frac{\sqrt{3}}{2})^2} \end{aligned}$$

Note that we have three unknowns since the highest power of s in the denominator is 3. Since there is an isolated pole at -2, we find coefficient A first using the coverup method

$$A = \lim_{s \rightarrow -2} \frac{1}{(s^2+s+1)} = \frac{1}{(-2)^2 + (-2) + 1} = \frac{1}{3}$$

To find C , multiply both sides by s and let $s \rightarrow \infty$, which gives us

$$0 = A + 0 + C$$

which give us $C = -A = -1/3$. Finally, we substitute $s = -1/2$ into the equation to get

$$\frac{1}{\frac{3}{4}} = \frac{A}{\frac{3}{2}} + \frac{B}{\frac{\sqrt{3}}{2}} + \frac{C}{\frac{3}{4}}$$

This is simplified as

$$\frac{8}{9} = \frac{2}{9} + B\frac{3}{\sqrt{3}}$$

which yields $B = \frac{1}{\sqrt{3}}$. Finally we have

$$h(t) = \frac{1}{3}e^{-2t}u(t) - \frac{1}{3}e^{-\frac{1}{2}t} \cos\left(\frac{\sqrt{3}}{2}t\right)u(t) + \frac{1}{\sqrt{3}}e^{-\frac{1}{2}t} \sin\left(\frac{\sqrt{3}}{2}t\right)u(t)$$

2.7 Common Denominator/Cross Multiplying

As a last method, we'll look at a method of doing partial fractions based on using a common denominator. This method is particularly useful for simple problems like finding the step response of a second order system. However, for many other types of problems it is not very useful, since it generates a system of equations that must be solved, much like substituting values of s will do.

Example 13 Let's assume we have the second order system

$$H(s) = \frac{b}{s^2 + cs + d}$$

and we want to find the step response of this system,

$$\begin{aligned} Y(s) &= H(s) \frac{1}{s} \\ &= \frac{b}{s(s^2 + bs + c)} = a_1 \frac{1}{s} + \frac{a_2s + a_3}{s^2 + bs + c} \\ &= \frac{b}{s(s^2 + bs + c)} = \frac{a_1(s^2 + bs + c) + s(a_2s + a_3)}{s(s^2 + bs + c)} \\ &= \frac{b}{s(s^2 + bs + c)} = \frac{(a_1c)s^0 + (a_1b + a_3)s^1 + (a_1 + a_2)s^2}{s(s^2 + bs + c)} \end{aligned}$$

Since we have made the denominator common for both sides, we just need to equate powers of s in the numerator:

$$\begin{aligned} a_1c &= b \\ a_1b + a_3 &= 0 \\ a_1 + a_2 &= 0 \end{aligned}$$

Since c and b are known, we can easily solve for a_1 in the first equation, then a_2 and a_3 in the remaining equations.

Example 14. Find the *step response* of

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

using the common denominator method. $Y(s)$ is given by

$$Y(s) = \frac{1}{s} \frac{1}{s^2 + 2s + 2} = a_1 \frac{1}{s} + \frac{a_2s + a_3}{s^2 + 2s + 2}$$

If we put everything over a common denominator we will have the equation

$$\begin{aligned} 1 &= a_1(s^2 + 2s + 2) + s(a_2s + a_3) \\ &= (2a_1)s^0 + (2a_1 + a_3)s^1 + (a_1 + a_2)s^2 \end{aligned}$$

Equating powers of s we get $a_1 = \frac{1}{2}$, then $a_3 = -1$ and $a_2 = -\frac{1}{2}$. Then we have

$$\begin{aligned} Y(s) &= \frac{1}{2} \frac{1}{s} + \frac{-\frac{1}{2}s - 1}{s^2 + 2s + 2} \\ &= \frac{1}{2} \frac{1}{s} - \frac{1}{2} \frac{s + 2}{s^2 + 2s + 2} \\ &= \frac{1}{2} \frac{1}{s} - \frac{1}{2} \frac{(s + 1)}{(s + 1)^2 + 1} - \frac{1}{2} \frac{1}{(s + 1)^2 + 1} \end{aligned}$$

In the time-domain we have then

$$y(t) = \frac{1}{2}u(t) - \frac{1}{2}e^{-t} \cos(t)u(t) - \frac{1}{2}e^{-t} \sin(t)u(t)$$

Example 15. Find the step response of

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

using the common denominator method. Partial fractions will only work if the denominator is *monic*, which means the leading coefficient must be a 1. Hence we rewrite $H(s)$ as

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

$Y(s)$ is then given by

$$Y(s) = \frac{1}{s} \frac{\frac{3}{2}}{s^2 + \frac{3}{2}s + \frac{3}{2}} = a_1 \frac{1}{s} + \frac{a_2s + a_3}{s^2 + \frac{3}{2}s + \frac{3}{2}}$$

If we put everything over a common denominator we will have the equation

$$\begin{aligned} \frac{3}{2} &= a_1 \left(s^2 + \frac{3}{2}s + \frac{3}{2} \right) + s(a_2s + a_3) \\ &= \left(\frac{3}{2}a_1 \right) s^0 + \left(\frac{3}{2}a_1 + a_3 \right) s^1 + (a_1 + a_2) s^2 \end{aligned}$$

Equating powers of s we get $a_1 = 1$, then $a_2 = -1$ and $a_3 = -\frac{3}{2}$. Then we have

$$\begin{aligned} Y(s) &= \frac{1}{s} + \frac{-s - \frac{3}{2}}{s^2 + \frac{3}{2}s + \frac{3}{2}} \\ &= \frac{1}{s} - \frac{s + \frac{3}{4} + \frac{3}{4}}{s^2 + \frac{3}{2}s + \frac{3}{2}} \\ &= \frac{1}{s} - \frac{(s + \frac{3}{4})}{(s + \frac{3}{4})^2 + \left(\sqrt{\frac{15}{16}}\right)^2} - \frac{3}{4} \sqrt{\frac{16}{15}} \frac{\sqrt{\frac{15}{16}}}{(s + \frac{3}{4})^2 + \left(\sqrt{\frac{15}{16}}\right)^2} \end{aligned}$$

In the time-domain we have then

$$y(t) = u(t) - e^{-3t/4} \cos\left(\sqrt{\frac{15}{16}}t\right)u(t) - \frac{3}{\sqrt{15}}e^{-3t/4} \sin\left(\sqrt{\frac{15}{16}}t\right)u(t)$$

2.8 Complex Conjugate Poles-Again

It is very important to understand the basic structure of complex conjugate poles. For a system with complex poles at $-a \pm jb$, the characteristic equation (denominator of the transfer function) will be

$$\begin{aligned} D(s) &= [s - (-a + jb)][s - (-a - jb)] \\ &= [s + (a - jb)][s + (a + jb)] \\ &= s^2 + [(a - jb) + (a + jb)]s + (a - jb)(a + jb) \\ &= s^2 + 2as + a^2 + b^2 \\ &= (s + a)^2 + b^2 \end{aligned}$$

We know that this form leads to terms of the form $e^{-at} \cos(bt)u(t)$ and $e^{-at} \sin(bt)u(t)$. Hence we have the general relationship that complex poles at $-a \pm jb$ lead to time domain functions that

- decay like e^{-at} (the real part determines the decay rate)
- oscillate like $\cos(bt)$ or $\sin(bt)$ (the imaginary part determines the oscillation frequency)

These relationships, relating the imaginary and real parts of the poles with corresponding time domain functions, are very important to remember.

3 Final Value Theorem and the Static Gain of a System

The final value theorem for Laplace transforms can generally be stated as follows:

If $Y(s)$ has all of its poles in the open left half plane, with the possible exception of a single pole at the origin, then

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

provided the limits exists.

Example 1. For $y(t) = e^{-at}u(t)$ with $a > 0$ we have

$$\begin{aligned}\lim_{t \rightarrow \infty} y(t) &= \lim_{t \rightarrow \infty} e^{-at} = 0 \\ \lim_{s \rightarrow 0} sY(s) &= \lim_{s \rightarrow 0} s \frac{1}{s+a} = \lim_{s \rightarrow 0} \frac{s}{s+a} = 0\end{aligned}$$

Example 2. For $y(t) = \sin(bt)u(t)$ we have

$$\begin{aligned}\lim_{t \rightarrow \infty} y(t) &= \lim_{t \rightarrow \infty} \sin(bt) \\ \lim_{s \rightarrow 0} sY(s) &= \lim_{s \rightarrow 0} s \frac{b}{s^2 + b^2} = \lim_{s \rightarrow 0} \frac{sb}{s^2 + b^2} = 0\end{aligned}$$

Clearly $\lim_{t \rightarrow \infty} y(t) \neq \lim_{s \rightarrow 0} sY(s)$. Why? Because the final value theorem is not valid since $Y(s)$ has two poles on the $j\omega$ axis.

Example 3. For $y(t) = u(t)$ we have

$$\begin{aligned}\lim_{t \rightarrow \infty} y(t) &= \lim_{t \rightarrow \infty} u(t) = 1 \\ \lim_{s \rightarrow 0} sY(s) &= \lim_{s \rightarrow 0} s \frac{1}{s} = \lim_{s \rightarrow 0} \frac{s}{s} = 1\end{aligned}$$

Example 4. For $y(t) = e^{-at} \cos(bt)u(t)$ with $a > 0$ we have

$$\begin{aligned}\lim_{t \rightarrow \infty} y(t) &= \lim_{t \rightarrow \infty} e^{-at} \cos(bt)u(t) = 0 \\ \lim_{s \rightarrow 0} sY(s) &= \lim_{s \rightarrow 0} s \frac{(s+a)}{(s+a)^2 + b^2} = \lim_{s \rightarrow 0} \frac{s(s+a)}{(s+a)^2 + b^2} = 0\end{aligned}$$

One of the common ways in which we use the Final Value Theorem is to compute the static gain of a system. The response of a transfer function $G(s)$ to a step input of amplitude A ,

$$Y(s) = G(s) \frac{A}{s}$$

If we want the final value of $y(t)$ then we can use the Final Value Theorem

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

$$\begin{aligned}
&= \lim_{s \rightarrow 0} sG(s) \frac{A}{s} \\
&= AG(0) \\
&= AK_{static}
\end{aligned}$$

provided $G(0)$ exists. $G(0)$ is referred to as the *gain* or *static gain* of the system. This is a very convenient way of determining the static gain of a system. It is important to remember that the steady state value of a system is the static gain of a system multiplied by the amplitude of the step input.

Example 5. For the transfer function

$$G(s) = \frac{s + 2}{s^2 + 3s + 1}$$

the static gain is 2, and if the step input has an amplitude of 0.1, the final value will be 0.2.

Example 6. For the transfer function

$$G(s) = \frac{s^2 + 1}{s^3 + 2s^2 + 3s + 4}$$

the static gain is $\frac{1}{4}$, and if the step input has an amplitude of 3, the final value will be 0.75.

4 Step Response, Ramp Response, and Steady State Errors

In control systems, we are often most interested in the response of a system to the following types of inputs:

- a step
- a ramp
- a sinusoid

Although in reality control systems have to respond to a large number of different inputs, these are usually good models for the range of input signals a control system is likely to encounter.

4.1 Step Response and Steady State Error

The *step response* of a system is the response of the system to a step input. In the time domain, we compute the step response as

$$y(t) = h(t) \star Au(t)$$

where A is the amplitude of the step and $u(t)$ is the unit step function and \star is the convolution operator. In the s domain, we compute the step response as

$$\begin{aligned} Y(s) &= H(s) \frac{A}{s} \\ y(t) &= \mathcal{L}^{-1}\{Y(s)\} \end{aligned}$$

The *steady state error*, e_{ss} , is the difference between the input and the resulting response as $t \rightarrow \infty$. For a step input of amplitude A we have

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} [Au(t) - y(t)] \\ &= A - \lim_{t \rightarrow \infty} y(t) \end{aligned}$$

Note that the steady state error can be both positive (the final value of the output is not as large as the input) or negative (the final value of the output is larger than the input).

Example 1. Consider the system with transfer function $H(s) = \frac{4}{s^2 + 2s + 5}$. Determine step response and the steady state error for this system.

First we find the step response,

$$\begin{aligned} Y(s) &= \frac{4}{s^2 + 2s + 5} \frac{A}{s} = a_1 \frac{1}{s} + \frac{a_2 s + a_3}{(s + 1)^2 + 2^2} \\ &= A \left[\frac{4}{5} \frac{1}{s} - \frac{\frac{4}{5}s + \frac{8}{5}}{(s + 1)^2 + 2^2} \right] \\ &= A \left[\frac{4}{5} \frac{1}{s} - \frac{\frac{4}{5}(s + 1)}{(s + 1)^2 + 2^2} - \frac{2}{5} \frac{2}{(s + 1)^2 + 2^2} \right] \end{aligned}$$

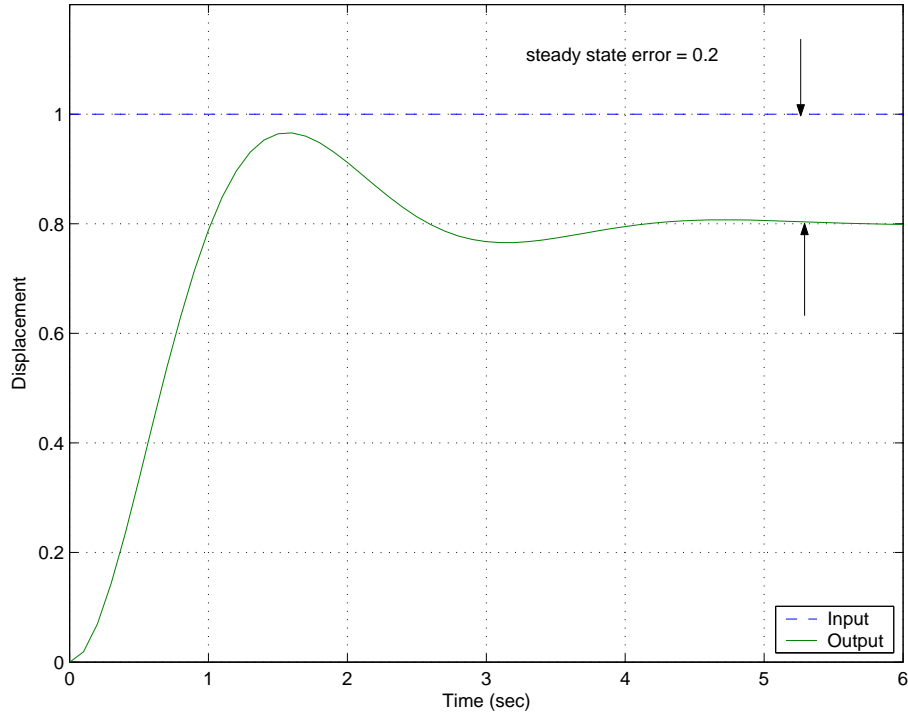


Figure 1: The unit step response and position error for the system in Example 1. This system has a *positive position error*.

or

$$y(t) = A \left[\frac{4}{5}u(t) - \frac{4}{5}e^{-t} \cos(2t)u(t) - \frac{2}{5}e^{-t} \sin(2t)u(t) \right]$$

Then the steady state error is

$$\begin{aligned} e_{ss} &= A - \lim_{t \rightarrow \infty} A \left[\frac{4}{5}u(t) - \frac{4}{5}e^{-t} \cos(2t)u(t) - \frac{2}{5}e^{-t} \sin(2t)u(t) \right] \\ &= A - \frac{4A}{5} \\ &= \frac{A}{5} \end{aligned}$$

The step response and steady state error of this system are shown in Figure 1 for a unit step ($A = 1$) input. Note that the positive steady state error indicates the final value of the output is smaller than the final value of the input.

Example 2. Consider the system with transfer function $H(s) = \frac{1}{(s+1)(s+3)}$. Determine the step response and steady state error for this system.

First we find the step response,

$$\begin{aligned} Y(s) &= \frac{5}{(s+1)(s+3)} \frac{A}{s} = a_1 \frac{1}{s} + a_2 \frac{1}{s+1} + a_3 \frac{1}{s+3} \\ &= \frac{5A}{3} \frac{1}{s} - \frac{5A}{2} \frac{1}{s+1} + \frac{5A}{6} \frac{1}{s+3} \end{aligned}$$

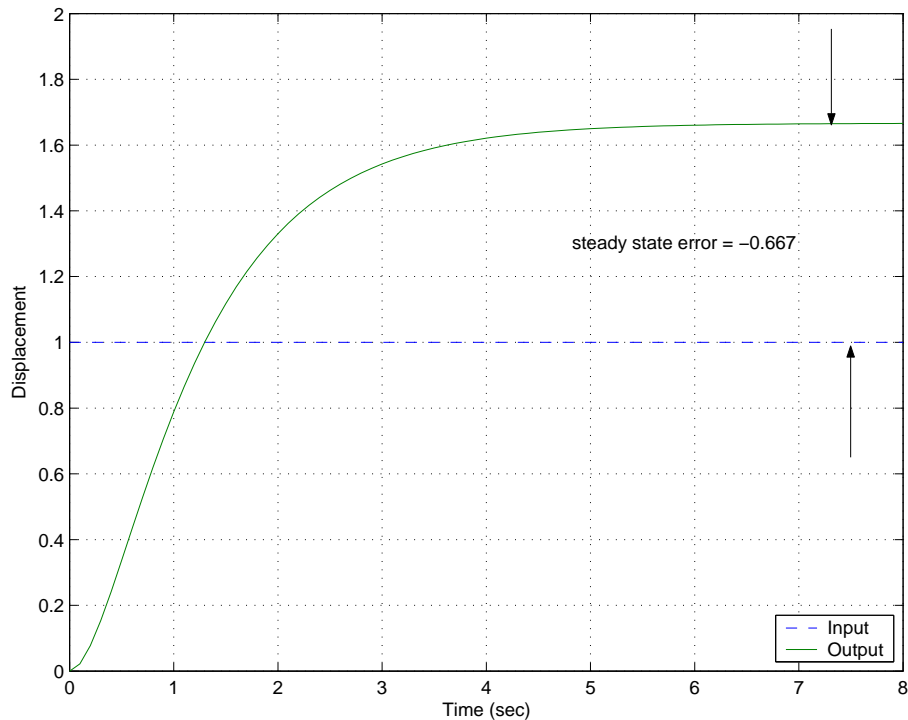


Figure 2: The unit step response and position error for the system in Example 2. This system has a *negative position error*.

or

$$y(t) = A \left[\frac{5}{3}u(t) - \frac{5}{2}e^{-t}u(t) + \frac{5}{6}e^{-3t}u(t) \right]$$

Then the steady state error is

$$\begin{aligned} e_p &= A - \lim_{t \rightarrow \infty} A \left[\frac{5}{3}u(t) - \frac{5}{2}e^{-t}u(t) + \frac{5}{6}e^{-3t}u(t) \right] \\ &= A - \frac{5A}{3} \\ &= \frac{-2A}{3} \end{aligned}$$

The step response and steady state error of this system are shown in Figure 2 for a unit step ($A = 1$) input. Note that the negative steady state error indicates the final value of the output is larger than the final value of the input.

Now, as much as I'm sure you like completing the square and doing partial fractions, there is an easier way to do this. We already have learned that if $Y(s)$ has all of its poles in the open left half plane (with the possible exception of a single pole at the origin), we can use the final value theorem to find the steady state value of the step response. Specifically,

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

$$\begin{aligned}
&= \lim_{s \rightarrow 0} s \left[H(s) \frac{A}{s} \right] \\
&= \lim_{s \rightarrow 0} AH(s) \\
&= AH(0)
\end{aligned}$$

and then, for stable $H(s)$ we can compute the steady state error as

$$e_{ss} = A - AH(0)$$

where A is the amplitude of the step input. For a *unit step response* $A = 1$.

Example 3. From Example 1, we compute

$$\begin{aligned}
e_{ss} &= A - AH(0) \\
&= A - A \frac{4}{5} \\
&= \frac{A}{5}
\end{aligned}$$

Example 4. From Example 2, we compute

$$\begin{aligned}
e_{ss} &= A - AH(0) \\
&= A - A \frac{5}{3} \\
&= \frac{-2A}{3}
\end{aligned}$$

There is yet another way to compute the steady state error, which is useful to know. Let's assume we write the transfer function as

$$H(s) = \frac{n_m s^m + n_{m-1} s^{m-1} + \dots + n_2 s^2 + n_1 s + n_0}{s^n + d_{n-1} s^{n-1} + \dots + d_2 s^2 + d_1 s + d_0}$$

To compute the steady state error for a step input we need to compute

$$e_{ss} = \lim_{s \rightarrow 0} A[1 - H(s)]$$

Let's write $1 - H(s)$ and put it all over a common denominator. Then we have

$$\begin{aligned}
1 - H(s) &= \frac{(s^n + d_{n-1} s^{n-1} + \dots + d_2 s^2 + d_1 s + d_0) - (n_m s^m + n_{m-1} s^{m-1} + \dots + n_2 s^2 + n_1 s + n_0)}{s^n + d_{n-1} s^{n-1} + \dots + d_2 s^2 + d_1 s + d_0} \\
&= \frac{\dots + (d_2 - n_2) s^2 + (d_1 - n_1) s + (d_0 - n_0)}{s^n + d_{n-1} s^{n-1} + \dots + d_2 s^2 + d_1 s + d_0}
\end{aligned}$$

Then

$$\begin{aligned}
e_{ss} &= \lim_{s \rightarrow 0} A[1 - H(s)] \\
&= A \frac{d_0 - n_0}{d_0}
\end{aligned}$$

Example 5. From Example 1, we have $n_0 = 4$ and $d_0 = 5$, so the steady state error for a step input is $e_{ss} = A \frac{5-4}{5} = \frac{A}{5}$.

Example 6. From Example 2, we have $n_0 = 5$, $d_0 = 3$, so the steady state error for a step input is $e_{ss} = A \frac{3-5}{3} = -\frac{2A}{3}$.

4.2 Ramp Response and Steady State Error

The *ramp response* of a system is the response of the system to a ramp input. In the time domain, we compute the ramp response as

$$y(t) = h(t) \star Atu(t)$$

where A is the amplitude of the step and $u(t)$ is the unit step function. In the s domain, we compute the step response as

$$\begin{aligned} Y(s) &= H(s) \frac{A}{s^2} \\ y(t) &= \mathcal{L}^{-1}\{Y(s)\} \end{aligned}$$

The *steady state error*, e_{ss} , is the difference between the input ramp and the resulting response as $t \rightarrow \infty$,

$$e_{ss} = \lim_{t \rightarrow \infty} [Atu(t) - y(t)]$$

It should be clear that unless $y(t)$ has a term like $Atu(t)$, the steady state error will be infinite. *Note that the steady state error can be both positive (the final value of the output is not as large as the input) or negative (the final value of the output is larger than the input).*

Example 7. Consider the system with transfer function $H(s) = \frac{1}{s+1}$. Determine the ramp response and steady state error for this system.

First we find the ramp response

$$\begin{aligned} Y(s) &= \frac{1}{s+1} \frac{A}{s^2} = a_1 \frac{1}{s} + a_2 \frac{1}{s^2} + a_3 \frac{1}{s+1} \\ &= A \left[-\frac{1}{s} + \frac{1}{s^2} + \frac{1}{s+1} \right] \end{aligned}$$

or

$$y(t) = A \left[-u(t) + tu(t) + e^{-t}u(t) \right]$$

Then the steady state error is

$$\begin{aligned} e_{ss} &= Atu(t) - \lim_{t \rightarrow \infty} A \left[-u(t) + tu(t) + e^{-t}u(t) \right] \\ &= At - At + A \\ &= A \end{aligned}$$

Example 8. Consider the system with transfer function $H(s) = \frac{s+2}{s^2+2s+2}$. Determine the ramp response and steady state error for this system.

First we find the ramp response

$$\begin{aligned} Y(s) &= \frac{s+2}{s^2+2s+2} \frac{A}{s^2} = a_1 \frac{1}{s} + a_2 \frac{1}{s^2} + \frac{a_3s + a_4}{s^2+2s+2} \\ &= A \left[-\frac{1}{2} \frac{1}{s} + \frac{1}{s^2} + \frac{1}{2} \frac{s}{(s+1)^2+1} \right] \\ &= A \left[-\frac{1}{2} \frac{1}{s} + \frac{1}{s^2} + \frac{1}{2} \frac{s+1}{(s+1)^2+1} - \frac{1}{2} \frac{1}{(s+1)^2+1} \right] \end{aligned}$$

or

$$y(t) = A \left[-\frac{1}{2}u(t) + tu(t) + \frac{1}{2}e^{-t} \cos(t)u(t) - \frac{1}{2}e^{-t} \sin(t)u(t) \right]$$

Then the steady state error is

$$\begin{aligned} e_{ss} &= At u(t) - \lim_{t \rightarrow \infty} A \left[-\frac{1}{2}u(t) + tu(t) + \frac{1}{2}e^{-t} \cos(t)u(t) - \frac{1}{2}e^{-t} \sin(t)u(t) \right] \\ &= At - At + \frac{1}{2}A \\ &= \frac{A}{2} \end{aligned}$$

The ramp response and steady state error for this system are shown in Figure 3 for a unit ramp input. Note that the steady state error is positive, indicating the output of the system is smaller than the input in steady state.

We can try and use the Final Value Theorem again, but it becomes a bit more complicated. We want to find

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} [At u(t) - y(t)] \\ &= \lim_{s \rightarrow 0} s \left[\frac{A}{s^2} - \frac{A}{s^2} H(s) \right] \\ &= \lim_{s \rightarrow 0} \frac{A}{s} [1 - H(s)] \end{aligned}$$

Let's assume again we can write the transfer function as

$$H(s) = \frac{n_m s^m + n_{m-1} s^{m-1} + \dots + n_2 s^2 + n_1 s + n_0}{s^n + d_{n-1} s^{n-1} + \dots + d_2 s^2 + d_1 s + d_0}$$

If we compute $1 - H(s)$ and put things over a common denominator, we have

$$\begin{aligned} 1 - H(s) &= \frac{(s^n + d_{n-1} s^{n-1} + \dots + d_2 s^2 + d_1 s + d_0) - (n_m s^m + n_{m-1} s^{m-1} + \dots + n_2 s^2 + n_1 s + n_0)}{s^n + d_{n-1} s^{n-1} + \dots + d_2 s^2 + d_1 s + d_0} \\ &= \frac{\dots + (d_2 - n_2) s^2 + (d_1 - n_1) s + (d_0 - n_0)}{s^n + d_{n-1} s^{n-1} + \dots + d_2 s^2 + d_1 s + d_0} \end{aligned}$$

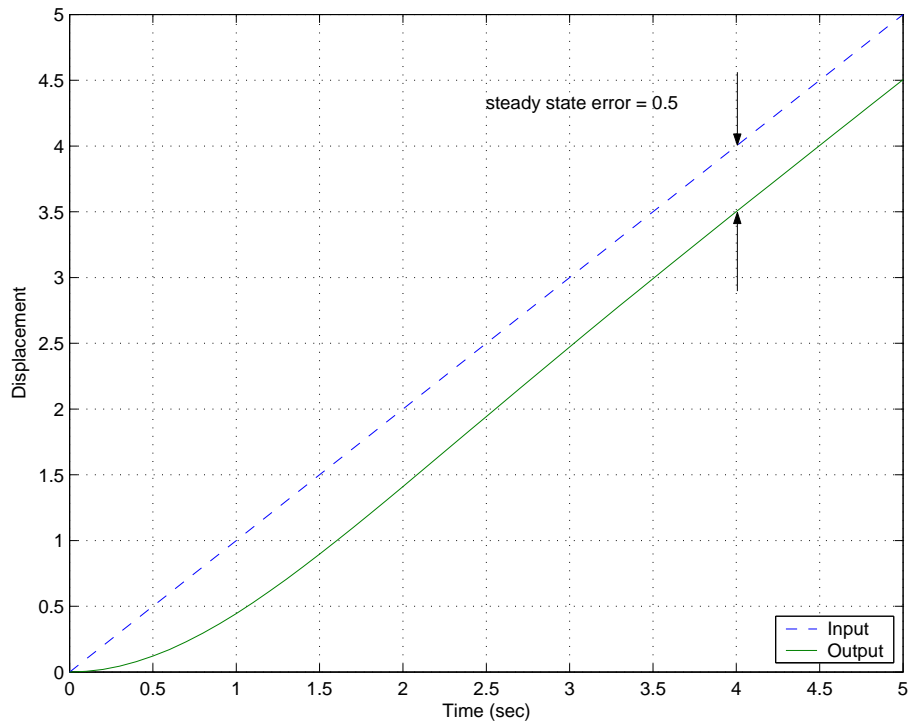


Figure 3: The unit ramp response and steady state error for the system in Example 8. Note that the steady state error is positive in this case, since the steady state value of the output is smaller than the steady state value of the input.

and

$$\frac{1}{s}[1 - H(s)] = \frac{\dots + (d_2 - n_2)s^+(d_1 - n_1) + (d_0 - n_0)\frac{1}{s}}{s^n + d_{n-1}s^{n-1} + \dots + d_2s^2 + d_1s + d_0}$$

Now, in order to have e_{ss} be finite, we must get a finite value as $s \rightarrow 0$ in this expression. The value of the denominator will be d_0 as $s \rightarrow 0$, so the denominator will be OK. All of the terms in the numerator will be zero except the last two: $(d_1 - n_1) + (d_0 - n_0)\frac{1}{s}$. In order to get a finite value from these terms, we must have $n_0 = d_0$, that is, constant terms in the numerator and denominator must be the same. This also means that the system must have a zero steady state error for a step input. *Important!! If the system does not have a zero steady state error for a step input, the steady state error for a ramp input will be infinite! Conversely, if a system has finite steady state error for a ramp input, the steady state error for a step input must be zero!* If $n_0 = d_0$, then we have

$$e_{ss} = \lim_{s \rightarrow 0} \frac{A}{s}[1 - H(s)] = A \frac{d_1 - n_1}{d_0}$$

Example 9. For the system in Example 7, $H(s) = \frac{1}{s+1}$. Here $n_0 = d_0 = 1$, so the system has zero steady state error for a step input, and $n_1 = 0$, $d_1 = 1$. Hence for a ramp input $e_{ss} = A \frac{d_1 - n_1}{d_0} = A$.

Example 10. For the system in Example 8, $H(s) = \frac{s+2}{s^2+2s+2}$. Here $n_0 = d_0 = 2$, so the system has zero steady state error for a step input, and $n_1 = 1$, $d_1 = 2$. Hence for a ramp input $e_{ss} = A \frac{d_1 - n_1}{d_0} = \frac{A}{2}$.

4.3 Summary

Assume we write the transfer function of a system as

$$H(s) = \frac{n_ms^m + n_{m-1}s^{m-1} + \dots + n_2s^2 + n_1s + n_0}{s^n + d_{n-1}s^{n-1} + \dots + d_2s^2 + d_1s + d_0}$$

The *step response* of a system is the response of the system to a step input. The *steady state error*, e_{ss} , for a step input is the difference between the input and the output of the system in steady state. We can compute the steady state error for a step input in a variety of ways:

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} [Au(t) - y(t)] \\ &= A - \lim_{t \rightarrow \infty} y(t) \\ &= A(1 - H(0)) \\ &= A \frac{d_0 - n_0}{d_0} \end{aligned}$$

The *ramp response* of a system is the response of the system to a ramp input. The *steady state error*, e_{ss} , for a ramp input is the difference between the input and output of the system in steady state. A system has infinite steady state error for a ramp input unless the steady state

error for a step input is zero. We can compute the steady state error for a ramp input in a variety of ways:

$$\begin{aligned} e_{ss} &= \lim_{t \rightarrow \infty} [At - y(t)] \\ &= A \frac{d_1 - n_1}{d_0} \end{aligned}$$

5 Response of a Ideal Second Order System

This is an important example, which you have probably seen before. Let's assume we have an ideal second order system with transfer function

$$H(s) = \frac{K_{static}}{\frac{1}{\omega_n^2}s^2 + \frac{2\zeta}{\omega_n}s + 1} = \frac{K_{static} \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where ζ is the damping ratio, ω_n is the natural frequency, and K_{static} is the static gain. The poles of the transfer function are the roots of the denominator, which are given by the quadratic formula

$$\begin{aligned} roots &= \frac{-2\zeta\omega_n \pm \sqrt{(2\zeta\omega_n)^2 - 4\omega_n^2}}{2} \\ &= -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \\ &= -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2} \\ &= -\zeta\omega_n \pm j\omega_d \\ &= -\sigma \pm j\omega_d \\ &= -1/\tau \pm j\omega_d \end{aligned}$$

where we have used the damped frequency $\omega_d = \omega_n\sqrt{1 - \zeta^2}$ and $\sigma = \frac{1}{\tau} = \zeta\omega_n$. As we start to talk about systems with more than two poles, it is easier to remember to use the form of the poles $-\sigma \pm j\omega_d$ or $-1/\tau \pm j\omega_d$.

5.1 Step Response of an Ideal Second Order System

To find the step response,

$$Y(s) = H(s)U(s) = \frac{K_{static} \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \frac{1}{s}$$

We then look for a partial fraction expansion in the form

$$Y(s) = \frac{K_{static} \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \frac{1}{s} = a_1 \frac{1}{s} + \frac{a_2 s + a_3}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

From this, we can determine that $a_1 = K_{static}$, $a_2 = -K_{static}$, and $a_3 = -2\zeta\omega_n K_{static}$. Hence we have

$$Y(s) = K_{static} \frac{1}{s} - K_{static} \frac{s + 2\zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Completing the square in the denominator we have

$$Y(s) = K_{static} \frac{1}{s} - K_{static} \frac{s + 2\zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2}$$

or

$$\begin{aligned} Y(s) &= K_{static} \frac{1}{s} - K_{static} \frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} - K_{static} \frac{\zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} \\ &= K_{static} \frac{1}{s} - K_{static} \frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} - K_{static} \frac{\zeta\omega_n}{\omega_d} \frac{\omega_d}{(s + \zeta\omega_n)^2 + \omega_d^2} \end{aligned}$$

or in the time domain

$$y(t) = K_{static} \left[1 - e^{-\zeta\omega_n t} \cos(\omega_d t) - \frac{\zeta\omega_n}{\omega_d} e^{-\zeta\omega_n t} \sin(\omega_d t) \right] u(t)$$

We would now like to write the sine and cosine in terms of a sine and a phase angle. To do this, we use the identity

$$r \sin(\omega_d t + \theta) = r \cos(\omega_d t) \sin(\theta) + r \sin(\omega_d t) \cos(\theta)$$

Hence we have

$$\begin{aligned} r \sin(\theta) &= 1 \\ r \cos(\theta) &= \frac{\zeta\omega_n}{\omega_d} = \frac{\zeta}{\sqrt{1 - \zeta^2}} \end{aligned}$$

Hence

$$\begin{aligned} \theta &= \tan^{-1} \left(\frac{\sqrt{1 - \zeta^2}}{\zeta} \right) \\ r &= \frac{1}{\sqrt{1 - \zeta^2}} \end{aligned}$$

Note that

$$\cos(\theta) = \frac{\zeta}{\sqrt{1 - \zeta^2}} \frac{1}{r} = \frac{\zeta}{\sqrt{1 - \zeta^2}} \sqrt{1 - \zeta^2}$$

or $\theta = \cos^{-1}(\zeta)$. Finally we have

$$y(t) = K_{static} \left[1 - \frac{1}{\sqrt{1 - \zeta^2}} e^{-\zeta\omega_n t} \sin(\omega_d t + \theta) \right] u(t)$$

5.2 Time to Peak, T_p

From our solution of the response of the ideal second order system to a unit step, we can compute the time to peak by taking the derivative of $y(t)$ and setting it equal to zero. This will give us the maximum value of $y(t)$ and the time that this occurs at is called the time to peak, T_p .

$$\frac{dy(t)}{dt} = -\frac{K_{static}}{\sqrt{1 - \zeta^2}} \left[-\zeta\omega_n e^{-\zeta\omega_n t} \sin(\omega_d t + \theta) + \omega_d e^{-\zeta\omega_n t} \cos(\omega_d t + \theta) \right] = 0$$

or

$$\begin{aligned}\zeta\omega_n \sin(\omega_d t + \theta) &= \omega_d \cos(\omega_d t + \theta) \\ \tan(\omega_d t + \theta) &= \frac{\sqrt{1-\zeta^2}}{\zeta} \\ \theta + \omega_d t &= \tan^{-1}\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)\end{aligned}$$

but we already have $\theta = \tan^{-1}\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)$, hence $\omega_d t$ must be equal to one period of the tangent, which is π . Hence

$$\boxed{T_p = \frac{\pi}{\omega_d}}$$

Remember that ω_d is equal to the imaginary part of the complex poles.

5.3 Percent Overshoot, PO

Evaluating $y(t)$ at the peak time T_p we get the maximum value of $y(t)$,

$$\begin{aligned}y(T_p) &= K_{static} \left[1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n T_p} \sin(\omega_d T_p + \theta) \right] \\ &= K_{static} \left[1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n \pi / \omega_d} \sin\left(\omega_d \frac{\pi}{\omega_d} + \theta\right) \right] \\ &= K_{static} \left[1 + \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\pi / \sqrt{1-\zeta^2}} \sin(\theta) \right]\end{aligned}$$

since $\sin(\theta + \pi) = -\sin(\theta)$. Then $\sin(\theta) = \sqrt{1-\zeta^2}$, hence

$$y(t) = K_{static} \left[1 + e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \right]$$

The percent overshoot is defined as

$$\text{Percent Overshoot} = P.O. = \frac{y(T_p) - y(\infty)}{y(\infty)} \times 100\%$$

For our second order system we have $y(\infty) = K_{static}$, so

$$P.O. = \frac{K_{static} \left[1 + e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \right] - K_{static}}{K_{static}} \times 100\%$$

or

$$\boxed{P.O. = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \times 100\%}$$

5.4 Settling Time, T_s

The settling time is defined as the time it takes for the output of a system with a step input to stay within a given percentage of its final value. In this course, we use the 2% settling time criteria, which is generally four time constants. For any exponential decay, the general form is written as $e^{-t/\tau}$, where τ is the time constant. For the ideal second order system response, we have $\tau = 1/\zeta\omega_n$ or $\sigma = \zeta\omega_n$. Hence, for an ideal second order system, we estimate the settling time as

$$T_s = 4\tau = \frac{4}{\sigma} = \frac{4}{\zeta\omega_n}$$

For systems other than second order systems we will want to talk about the settling time, hence the use of the forms

$$\boxed{T_s = 4\tau = \frac{4}{\sigma}}$$

are often more appropriate to remember.

Example 1. Consider the system with transfer function given by

$$H(s) = \frac{9}{s^2 + \beta s + 9}$$

determine the range of β so that $T_s \leq 5$ seconds and $T_p \leq 1.2$ seconds.

For the transfer function, we see that $\omega_n = 3$ and $2\zeta\omega_n = \beta$, so $\zeta = \beta/(2\omega_n) = \beta/6$. For the settling time constraint we have

$$\begin{aligned} T_s &= \frac{4}{\zeta\omega_n} \leq 5 \\ \frac{4}{\frac{\beta}{6} \cdot 3} &\leq 5 \\ \frac{8}{\beta} &\leq 5 \end{aligned}$$

so $\beta \geq 1.60$. For the time to peak constraint, we have

$$\begin{aligned} T_p &= \frac{\pi}{\omega_d} \leq 1.2 \\ \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} &\leq 1.2 \\ \frac{\pi}{1.2\omega_n} &\leq \sqrt{1 - \zeta^2} \\ \left(\frac{\pi}{1.2\omega_n}\right)^2 &\leq 1 - \zeta^2 \\ \zeta^2 &\leq 1 - \left(\frac{\pi}{1.2\omega_n}\right)^2 \\ \zeta &\leq \sqrt{1 - \left(\frac{\pi}{1.2\omega_n}\right)^2} \\ \beta &\leq 6\sqrt{1 - \left(\frac{\pi}{1.2\omega_n}\right)^2} \end{aligned}$$

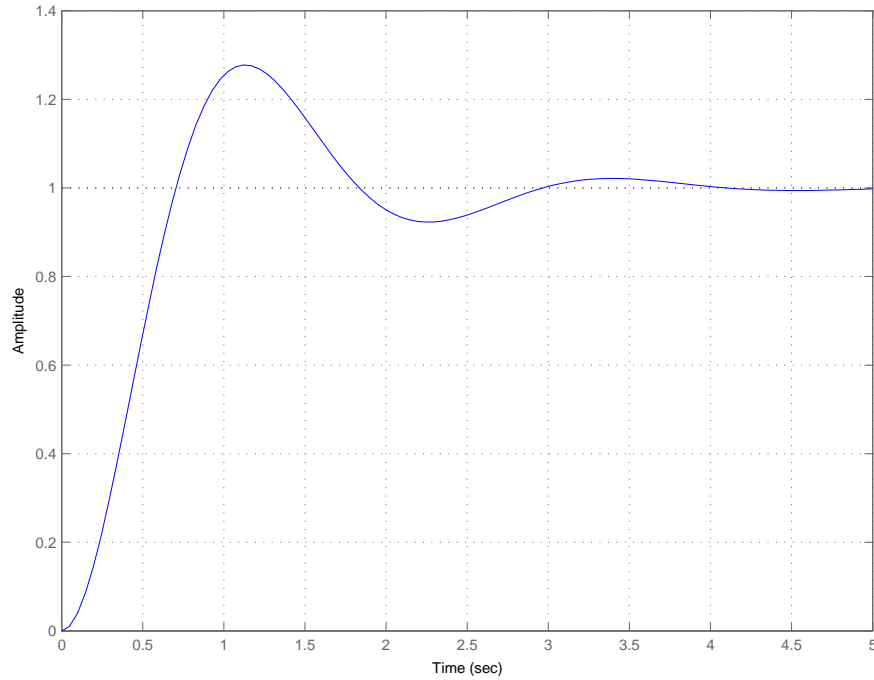


Figure 4: Step response for the system $H(s) = \frac{9}{s^2 + 2.265s + 9}$. The settling time should be less than 5 seconds, the time to peak should be less than 1.2 seconds, and the percent overshoot should be 27.8%.

or $\beta \leq 2.93$. To meet both constraints we need $1.60 \leq \beta \leq 2.93$. Let's choose the average, so $\beta = 2.265$. Then $\zeta = 0.3775$ and the percent overshoot is 27.8%. The step response of this system is shown in Figure 4.

Example 2. Consider the system with transfer function given by

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

determine the range of K so that $PO \leq 20\%$. Is there any value of K so that $T_s \leq 2$ seconds?

For the transfer function, we see that $\omega_n = \sqrt{K}$ and $2\zeta\omega_n = 2$, so $\zeta\omega_n = 1$ and $\zeta = \frac{1}{\sqrt{K}}$. For the percent overshoot we have $b = 20/100 = 0.2$ and

$$\begin{aligned} e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} &\leq b \\ -\frac{\zeta\pi}{\sqrt{1-\zeta^2}} &\leq \ln(b) \\ -\frac{\pi}{\sqrt{K}} \frac{1}{\sqrt{1-\frac{1}{K}}} &\leq \ln(b) \\ -\frac{\pi}{\sqrt{K-1}} &\leq \ln(b) \end{aligned}$$

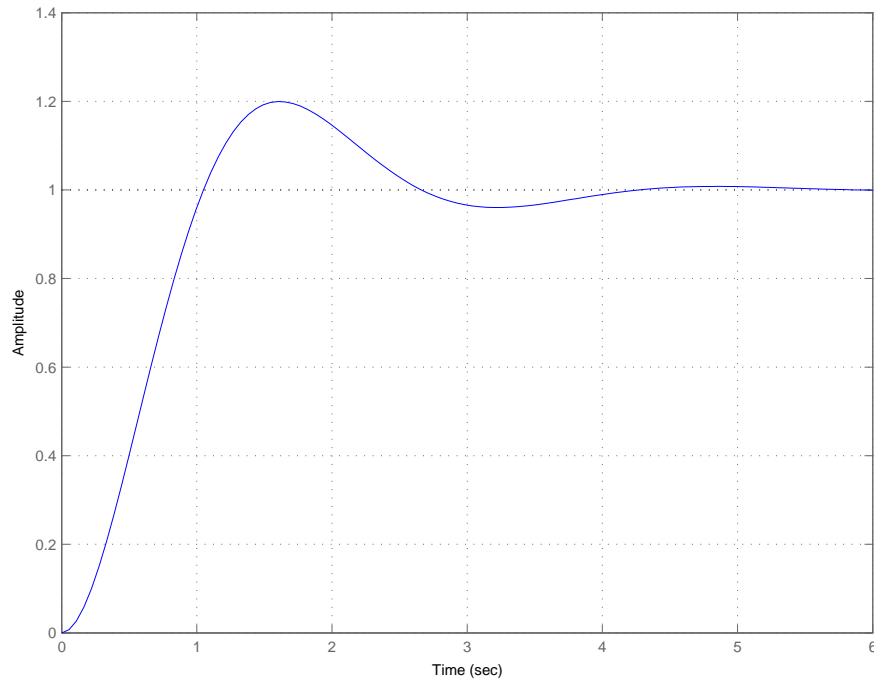


Figure 5: Step response for the system $H(s) = \frac{K}{s^2+2s+K}$. The percent overshoot should be less than or equal to 20% and the settling time should be 4 seconds.

$$\begin{aligned}
 -\frac{\pi}{\ln(b)} &\leq \sqrt{K-1} \\
 \left(\frac{\pi}{\ln(b)}\right)^2 &\leq K-1 \\
 1 + \left(\frac{\pi}{\ln(b)}\right)^2 &\leq K
 \end{aligned}$$

Hence we need $K \geq 4.8$ to meet the percent overshoot requirement. Now we try to meet the settling time requirement

$$T_s = \frac{4}{\zeta\omega_n} \leq 2$$

but $\frac{4}{\zeta\omega_n} = \frac{4}{1} = 4$. Thus, we cannot meet the settling time constraint for any value of K . The step response of this system for $K = 2.8$ is shown in Figure 5.

5.5 Constraint Regions in the s -Plane

Sometimes, instead of looking at a transfer function and trying to determine the percent overshoot, settling time, or time to peak, we can take the opposite approach and try to determine

the region in the s -plane the poles of the system should be located in to achieve a given criteria. Each one of the three criteria will determine a region of space in the s -plane.

Time to Peak (T_p) Let's assume we have a maximum time to peak given, T_p^{max} , and we want to know where to find all of the poles that will meet this constraint. We have

$$T_p = \frac{\pi}{\omega_d} \leq T_p^{max}$$

we can rearrange this as

$$\frac{\pi}{T_p^{max}} \leq \omega_d$$

Since we can write the complex poles as $-\sigma \pm j\omega_d$, this means that the imaginary part of the poles must be greater than $\frac{\pi}{T_p^{max}}$.

Example 3. Determine all acceptable pole location so that the time to peak will be less than 2 seconds. We have $T_p^{max} = 2$, so $\omega_d \geq \frac{\pi}{2} = 1.57$. The acceptable pole locations are shown in the shaded region of Figure 6.

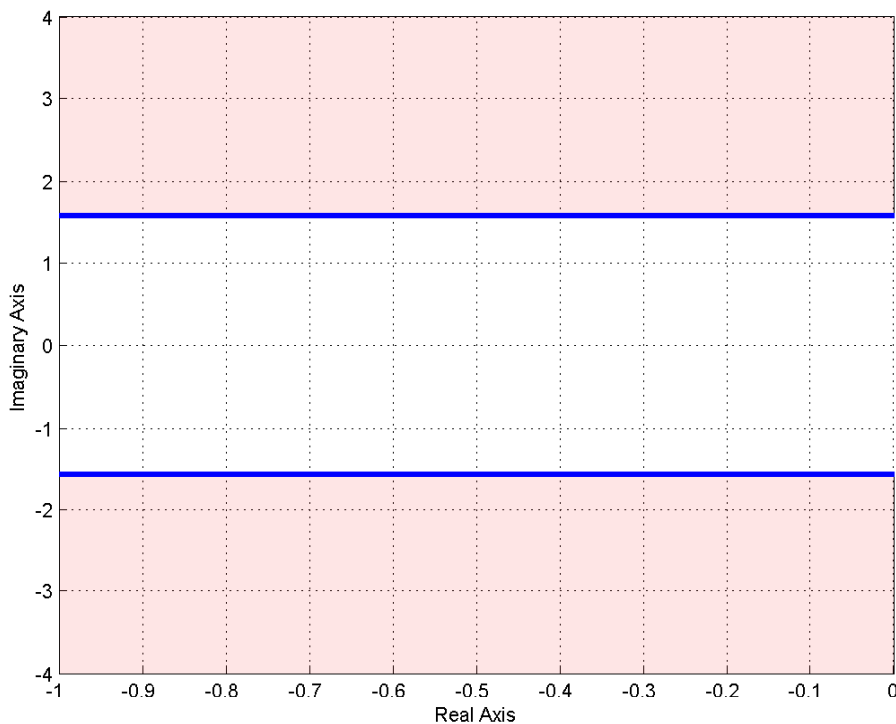


Figure 6: Acceptable pole locations for $T_p \leq 2$ seconds are shown in the shaded region.

Percent Overshoot (P.O.) Let's assume we have a maximum percent overshoot given, PO^{max} , and we want to know where to find all of the poles that will meet this constraint. We have

$$P.O. = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \times 100\% \leq PO^{max}$$

or

$$e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \leq \frac{PO^{max}}{100} = b$$

where we have defined the parameter $b = PO^{max}/100$ for notational convenience. We need to first solve the above expression for ζ .

$$\begin{aligned} -\frac{\zeta\pi}{\sqrt{1-\zeta^2}} &\leq \ln(b) \\ \frac{\zeta}{\sqrt{1-\zeta^2}} &\geq \frac{-\ln(b)}{\pi} \\ \frac{\zeta^2}{1-\zeta^2} &\geq \left(\frac{-\ln(b)}{\pi}\right)^2 \\ \zeta^2 &\geq \left(\frac{-\ln(b)}{\pi}\right)^2 - \zeta^2 \left(\frac{-\ln(b)}{\pi}\right)^2 \\ \zeta^2 \left[1 + \left(\frac{-\ln(b)}{\pi}\right)^2\right] &\geq \left(\frac{-\ln(b)}{\pi}\right)^2 \\ \zeta &\geq \frac{\frac{-\ln(b)}{\pi}}{\sqrt{1 + \left(\frac{-\ln(b)}{\pi}\right)^2}} \end{aligned}$$

Now we use the relationship

$$\theta = \cos^{-1}(\zeta)$$

In summary, we have

$$\theta \leq \cos^{-1}(\zeta), \zeta \geq \frac{\frac{-\ln(b)}{\pi}}{\sqrt{1 + \left(\frac{-\ln(b)}{\pi}\right)^2}}, b = \frac{PO^{max}}{100}$$

This angle θ is measured from the negative real axis. Hence an angle of 90 degrees indicates $\zeta = 0$ and there is no damping (the poles are on the $j\omega$ axis), while an angle of 0 degrees means the system has a damping ratio of 1, and the poles are purely real.

Example 4. Determine all acceptable pole locations so that the percent overshoot will be less than 10%. We have $b = 0.1$, so $\zeta \geq 0.59$ and $\theta \leq 53.8^\circ$. The acceptable pole locations are shown in the shaded region of Figure 7.

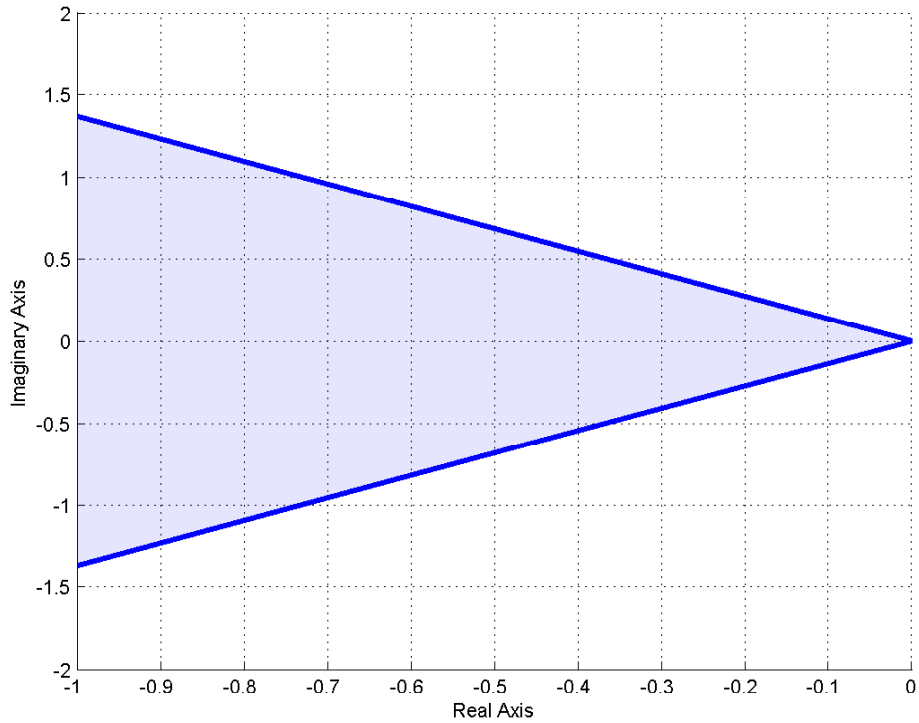


Figure 7: Acceptable pole locations for Percent Overshoot less than or equal to 10%. The acceptable pole locations are shown in the shaded region.

Example 5. Determine all acceptable pole locations so that the percent overshoot will be less than 20% and the time to peak will be less than 3 seconds. We have $b = 0.2$, so $\zeta \geq 0.46$ and $\theta \leq 62.9^\circ$. We also have $T_p^{max} = 3$, so $\omega_d \geq \frac{\pi}{3} = 1.04$. The acceptable pole locations for each constraint are shown in Figure 8. The overlapping regions are the acceptable pole locations to meet both the percent overshoot and time to peak constraints.

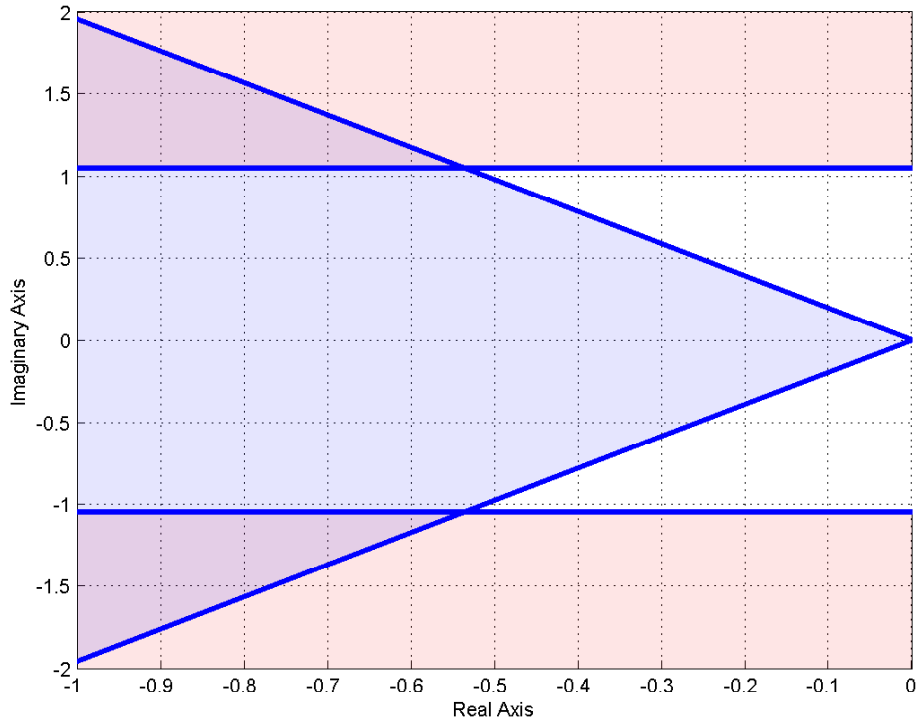


Figure 8: Acceptable pole locations for Percent Overshoot less than or equal to 20% and time to peak less than or equal to 3 seconds. The acceptable pole locations for each constraint are shown in the shaded regions. The overlapping regions are those pole locations that will meet both constraints.

Settling Time (T_s) Let's assume we have a maximum settling time T_s^{max} , and we want to know where to find all of the poles that will meet this constraint. We have

$$T_s = \frac{4}{\sigma} \leq T_s^{max}$$

or

$$\frac{4}{T_s^{max}} \leq \sigma$$

Since we can write the complex poles as $-\sigma \pm j\omega_d$, this means that the real part of the poles must be greater (in magnitude) than $\frac{4}{T_s^{max}}$. In other words, the poles must have real parts less than $-\frac{4}{T_s^{max}}$

Example 6. Determine all acceptable pole locations so that the settling time will be less than 3 seconds. We have $T_s^{max} = 3$, so $\sigma \geq \frac{4}{T_s^{max}} = \frac{4}{3} = 1.333$. The acceptable pole locations are shown in Figure 9.

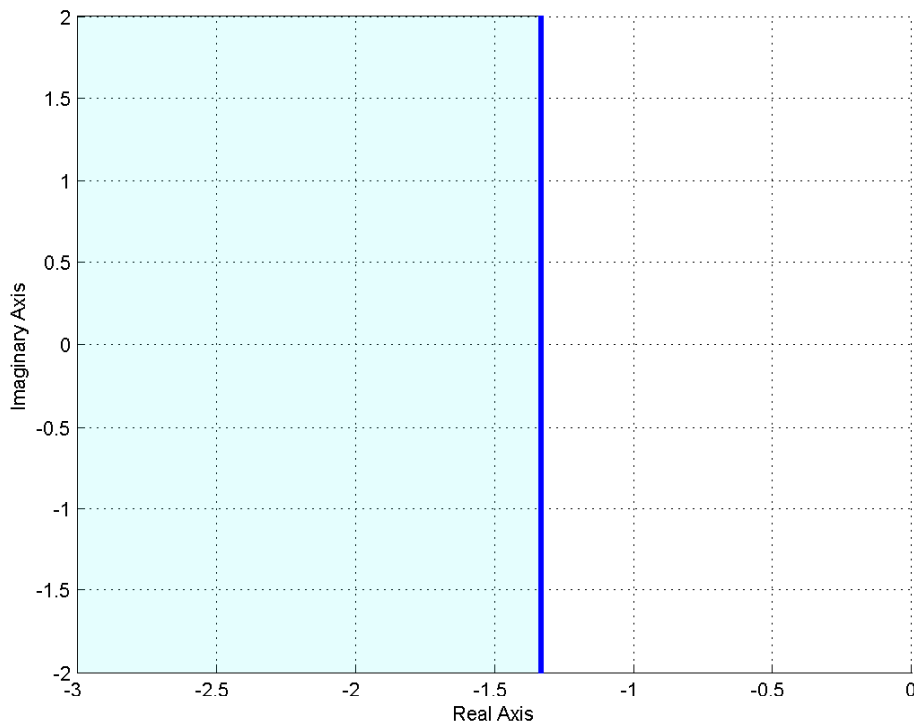


Figure 9: Acceptable pole locations for settling time less than or equal to 3 seconds. The acceptable pole locations are shown in the shaded region.

Example 7. Determine all acceptable pole locations so that the settling time will be less than 1 second and the time to peak will be less than or equal to 0.5 seconds. We have $T_s^{max} = 1$, so $\sigma \geq \frac{4}{T_s^{max}} = \frac{4}{1} = 4$. We also have $T_p^{max} = 0.5$, so $\omega_d \geq \frac{\pi}{T_p^{max}} = \frac{\pi}{0.5} = 6.28$. The acceptable pole locations for each constraint are shown in Figure 10. The overlapping regions (upper left corner, lower left corner) are the acceptable pole locations to meet both the settling time and time to peak constraints.

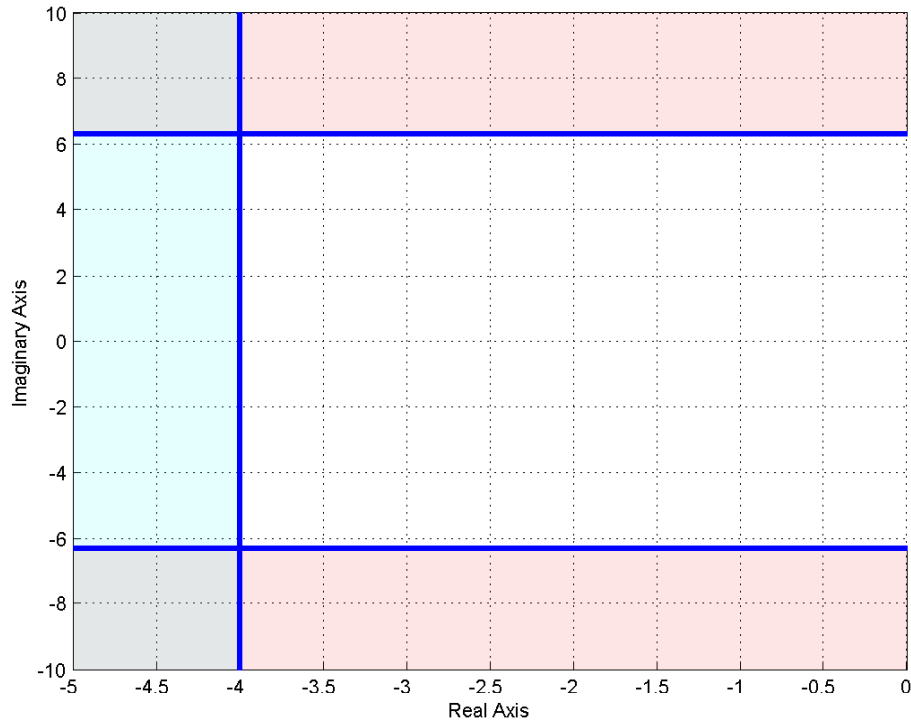


Figure 10: Acceptable pole locations for settling time less than or equal to 1 second and time to peak less than 0.5 seconds. The acceptable pole locations for each constraint are shown in the shaded regions. The overlapping regions are those pole locations that will meet both constraints.

Example 8. Determine all acceptable pole locations so that the settling time will be less than 5 seconds, the time to peak will be less than or equal to 2 seconds, and the percent overshoot will be less than 5%. We have $T_s^{max} = 5$, so $\sigma \geq \frac{4}{T_s^{max}} = \frac{4}{5} = 0.8$. We also have $T_p^{max} = 2$, so $\omega_d \geq \frac{\pi}{T_p^{max}} = \frac{\pi}{2} = 1.57$. Finally, $b = 0.05$, $\zeta \geq 0.69$ or $\theta < 46.4^\circ$. The acceptable pole locations for each constraint are shown in Figure 11. The overlapping regions (two triangular wedges) are the acceptable pole locations to meet all three constraints.

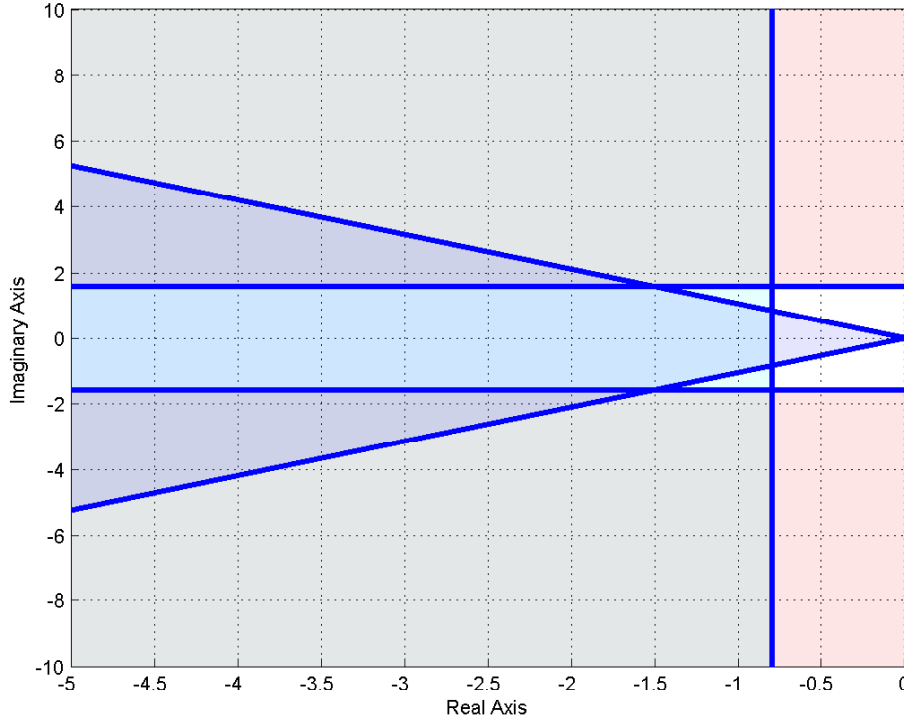


Figure 11: Acceptable pole locations for settling time less than 5 seconds, time to peak less than or equal to 2 seconds, and the percent overshoot less than 5%. The acceptable pole locations for each constraint are shown in the shaded regions. The overlapping regions (two triangular wedges) are those pole locations that will meet all three constraints.

5.6 Summary

For an *ideal* second order system with transfer function

$$H(s) = \frac{K_{static}}{\frac{1}{\omega_n^2} s^2 + \frac{2\zeta}{\omega_n} s + 1} = \frac{K_{static} \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

the poles are located at $-\zeta\omega_n \pm j\omega_d$, which is commonly written as either $-\sigma \pm j\omega_d$ or $-\frac{1}{\tau} \pm j\omega_d$. We can compute the percent overshoot (PO), the settling time (T_s), and the time to peak (T_p)

$$PO = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \times 100\%$$

$$T_s = \frac{4}{\zeta\omega_n} = 4\tau = \frac{4}{\sigma}$$

$$T_p = \frac{\pi}{\omega_d}$$

It is important to remember that *these relationships are only valid for **ideal** second order systems!*

What is generally more useful to us is to use these relationships to determine acceptable pole locations to meet the various design criteria. If the maximum desired settling time is T_s^{max} , then all poles must have real parts less than $-4/T_s^{max}$. If the maximum desired time to peak is T_p^{max} , then the imaginary parts of the dominant poles must have imaginary parts larger than π/T_p^{max} , or less than $-\pi/T_p^{max}$ (since poles come in complex conjugate pairs). If the maximum percent overshoot is PO^{max} , then the poles must lie in a wedge determined by $\theta = \cos^{-1}(\zeta)$ where θ is measured from the negative real axis and

$$\zeta \geq \frac{\frac{-\ln(b)}{\pi}}{\sqrt{1 + \left(\frac{-\ln(b)}{\pi}\right)^2}}, \quad b = \frac{PO^{max}}{100}$$

Each of these constraints can be used to define a region of acceptable pole locations for an ideal second order system. However, they are often used as a guide (or starting point) for higher order systems, and systems with zeros.

6 Characteristic Polynomial, Modes, and Stability

In this section, we first introduce the concepts of the characteristic polynomial, characteristic equation, and characteristic modes. You'll obviously note the word *characteristic* is used quite a lot here. Then, we utilize these concepts to define stability of our systems.

6.1 Characteristic Polynomial, Equation, and Modes

Consider a transfer function

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

where $N(s)$ and $D(s)$ are polynomials in s with no common factors. $D(s)$ is called the *characteristic polynomial* of the system, and the equation $D(s) = 0$ is called the *characteristic equation*. The time functions associated with the roots of the characteristic equation (the poles of the system) are called the *characteristic modes*. To determine the characteristic modes, it is often easiest to think of doing partial fraction expansion and looking at the resulting time functions. Some examples will probably help.

Example 1. Consider the transfer function

$$H(s) = \frac{s+2}{s^2(s+1)(s+3)} = a_1 \frac{1}{s} + a_2 \frac{1}{s^2} + a_3 \frac{1}{s+1} + a_4 \frac{1}{s+3}$$

Then we have:

$$\begin{aligned} \text{Characteristic Polynomial:} & \quad s^2(s+1)(s+3) \\ \text{Characteristic Equation:} & \quad s^2(s+1)(s+3) = 0 \\ \text{Characteristic Modes:} & \quad u(t), tu(t), e^{-t}u(t), e^{-3t}u(t) \end{aligned}$$

The impulse response is a linear combination of characteristic modes:

$$h(t) = a_1u(t) + a_2tu(t) + a_3e^{-t}u(t) + a_4e^{-3t}u(t)$$

Example 2. Consider the transfer function

$$H(s) = \frac{s-3}{s(s+1)^2(s+3)} = a_1 \frac{1}{s} + a_2 \frac{1}{s+1} + a_3 \frac{1}{(s+1)^2} + a_4 \frac{1}{s+3}$$

Then we have:

$$\begin{aligned} \text{Characteristic Polynomial:} & \quad s(s+1)^2(s+3) \\ \text{Characteristic Equation:} & \quad s(s+1)^2(s+3) = 0 \\ \text{Characteristic Modes:} & \quad u(t), e^{-t}u(t), te^{-t}u(t), e^{-3t}u(t) \end{aligned}$$

The impulse response is a linear combination of characteristic modes:

$$h(t) = a_1u(t) + a_2e^{-t}u(t) + a_3te^{-t}u(t) + a_4e^{-3t}u(t)$$

Example 3. Consider the transfer function

$$\begin{aligned} H(s) &= \frac{1}{s^2 + s + 1} = \frac{1}{(s + \frac{1}{2})^2 + (\frac{\sqrt{3}}{2})^2} \\ &= a_1 \frac{s + \frac{1}{2}}{(s + \frac{1}{2})^2 + (\frac{\sqrt{3}}{2})^2} + a_2 \frac{\frac{\sqrt{3}}{2}}{(s + \frac{1}{2})^2 + (\frac{\sqrt{3}}{2})^2} \end{aligned}$$

Then we have:

$$\begin{aligned} \text{Characteristic Polynomial:} & \quad s^2 + s + 1 \\ \text{Characteristic Equation:} & \quad s^2 + s + 1 = 0 \\ \text{Characteristic Modes:} & \quad e^{-t/2} \cos(\frac{\sqrt{3}}{2}t)u(t), \quad e^{-t/2} \sin(\frac{\sqrt{3}}{2}t)u(t) \end{aligned}$$

The impulse response is going to be a linear combination of characteristic modes:

$$h(t) = a_1 e^{-t/2} \cos(\frac{\sqrt{3}}{2}t)u(t) + a_2 e^{-t/2} \sin(\frac{\sqrt{3}}{2}t)u(t)$$

6.2 Characteristic Mode Reminders

There are a few things to keep in mind when finding characteristic modes

- There are as many characteristic modes as there are poles of the transfer function. Each characteristic mode must be different from the others.
- For any complex poles $-\sigma \pm j\omega_d$, the characteristic mode will be of the form $e^{-\sigma t} \cos(\omega_d t)u(t)$, and $e^{-\sigma t} \sin(\omega_d t)u(t)$.
- Assume pole p_i corresponds to characteristic mode $\phi_i(t)$. If there are two poles at p_i , the characteristic modes associated with pole p_i will be $\phi_i(t)$ and $t\phi_i(t)$. If there are three poles at p_i , the characteristic modes associated with p_i will be $\phi_i(t)$, $t\phi_i(t)$, and $t^2\phi_i(t)$. If pole p_i is repeated n times, the characteristic modes associated with pole p_i will be $\phi_i(t)$, $t\phi_i(t)$, $t^2\phi_i(t)$, ... $t^{n-1}\phi_i(t)$
- The impulse response is a linear combination of the characteristic modes of a system.

Example 4. If a transfer function has poles at $-1, -1, -2 \pm 3j, -5 \pm 2j$, the characteristic modes are given by $e^{-t}u(t)$, $te^{-t}u(t)$, $e^{-2t} \cos(3t)u(t)$, $e^{-2t} \sin(3t)u(t)$, $e^{-5t} \cos(2t)u(t)$, and $e^{-5t} \sin(2t)u(t)$.

Example 5. If a transfer function has poles at $-2, -2, -2, -3 \pm 2j, -3 \pm 2j$, the characteristic modes are at $e^{-2t}u(t)$, $te^{-2t}u(t)$, $t^2e^{-2t}u(t)$, $e^{-3t} \cos(2t)u(t)$, $e^{-3t} \sin(2t)u(t)$, $te^{-3t} \cos(2t)u(t)$, and $te^{-3t} \sin(2t)u(t)$.

6.3 Stability

A system is defined to be stable if **all** of its characteristic modes go to zero as $t \rightarrow \infty$. A system is defined to be marginally stable if **all** of its characteristic modes are bounded as $t \rightarrow \infty$. A system is unstable if **any** of its characteristic modes is unbounded as $t \rightarrow \infty$. There are other definitions of stability, each with their own purpose. For the systems we will be studying in this course, generally linear time invariant systems, these are the most appropriate. Note that the stability of a system is independent of the input.

In determining stability, the following mathematical truths should be remembered

$$\begin{aligned}\lim_{t \rightarrow \infty} t^n e^{-at} &= 0 \text{ for all positive } a \text{ and } n \\ \lim_{t \rightarrow \infty} e^{-at} \cos(\omega_d t + \phi) &= 0 \text{ for all positive } a \\ \lim_{t \rightarrow \infty} e^{-at} \sin(\omega_d t + \phi) &= 0 \text{ for all positive } a \\ u(t) &\text{ is bounded} \\ \cos(\omega_d t + \phi) &\text{ is bounded} \\ \sin(\omega_d t + \phi) &\text{ is bounded}\end{aligned}$$

Example 6. Assume a system has poles at $-1, 0, -2$. Is the system stable?

The characteristic modes of the system are $e^{-t}u(t)$, $u(t)$, and $e^{-2t}u(t)$. Both $e^{-t}u(t)$ and $e^{-2t}u(t)$ go to zero as $t \rightarrow \infty$. $u(t)$ does not go to zero, but it is bounded. Hence the system is *marginally stable*.

Example 7. Assume a system has poles at $-1, 1, -2 \pm 3j$. Is the system stable?

The characteristic modes of the system are $e^{-t}u(t)$, $e^t u(t)$, $e^{-2t} \cos(3t)u(t)$, and $e^{-2t} \sin(3t)u(t)$. All of these modes go to zero as t goes to infinity, except the mode $e^t u(t)$. This mode is unbounded as $t \rightarrow \infty$. Hence the system is *unstable*.

Example 8. Assume a system has poles at $-1, -1, -2 \pm j, -2 \pm j$. Is the system stable?

The characteristic modes of the system are $e^{-t}u(t)$, $te^{-t}u(t)$, $e^{-2t} \cos(t)u(t)$, $e^{-2t} \sin(t)u(t)$, $te^{-2t} \cos(t)u(t)$, and $te^{-2t} \sin(t)u(t)$. All of the characteristic modes go to zero as t goes to infinity, so the system is *stable*.

6.4 Settling Time and Dominant Poles

For an ideal second order system, we have already shown that the (2%) settling time is given by

$$T_s = \frac{4}{\zeta \omega_n}$$

We need to be able to deal with systems with more than two poles. To do this, we first make the following observations:

- We normally write decaying exponentials in the form $e^{-t/\tau}$, where τ is the *time constant*. Using the 2% settling time, we set the settling time equal to four time constants, $T_s = 4\tau$.
- If a system has a real pole at $-\sigma$, the corresponding mode is $e^{-\sigma t}u(t)$. Hence the time constant τ is equal to $\frac{1}{\sigma}$. The settling time for this pole is then $T_s = 4\tau = 4\frac{1}{\sigma}$.
- If a system has complex conjugate poles at $-\sigma \pm j\omega_d$, the corresponding modes are $e^{-\sigma t} \cos(\omega_d t)u(t)$ and $e^{-\sigma t} \sin(\omega_d t)u(t)$. Although these modes oscillate, the settling time depends on the time constants, which again leads to $\tau = \frac{1}{\sigma}$, and the settling time for this type of mode is given by $T_s = 4\frac{1}{\sigma}$.

Hence, to determine the settling time associated with the i^{th} pole of the system, p_i , we compute

$$T_s^i = 4 \frac{1}{\text{Re}\{-p_i\}} = \frac{4}{\sigma}$$

where we have written the real part of the pole, $\text{Re}\{-p_i\}$, is equal to σ .

*To determine the settling time of a system with multiple poles, determine the characteristic mode associated with each pole, and then compute the settling time corresponding to that mode. The largest such settling time is the settling time of the system. The poles associated with the largest settling time are the **dominant poles** of the system.*

Example 9. Assume we have a system with poles at $-5, -4, -3 \pm 2j$. Determine the settling time and the dominant poles of the system.

We have the settling times $T_s^1 = \frac{4}{5}$, $T_s^2 = \frac{4}{4}$, and $T_s^3 = \frac{4}{3}$. The largest of these is $T_s = \frac{4}{3}$, so this is the estimated settling time of the system. This settling time is associated with the poles at $-3 \pm 2j$, so these are the dominant poles.

Example 10. Assume we have a system with poles at $-2 \pm 3j, -1, -5 \pm 2j$. Determine the settling time and the dominant poles of the system.

We have the settling times $T_s^1 = \frac{4}{2}$, $T_s^2 = \frac{4}{1}$, and $T_s^3 = \frac{4}{5}$. The largest of these is $T_s = \frac{4}{1}$, so this is the estimated settling time of the system. This settling time is associated with the pole at -1 , so this is the dominant pole.

While the poles of the system determine the characteristic modes of the system, the amplitudes that multiply these modes (the a_i in the partial fraction expansion) are determined by both the poles and zeros of the system. In addition, when a pole is repeated, the form of the characteristic mode is $t^n e^{-\sigma t}$ (multiplied by sine or cosine for complex poles). Neither of these affects, the zeros of a system and the effects of repeated poles, was considered in estimating the settling time for a system. However, the approximation we have made is usually fairly reasonable.

Dominant poles are the slowest responding poles in a system. If we want faster response, these are the poles we must move away from the $j\omega$ axis.

7 Time Domain Response and System Bandwidth

The relationship between the time domain and frequency domain is something we must be aware of when designing control systems. While we want our system to respond quickly, i.e., have a small settling time, we have to realize what effects this has in the frequency domain. We will be dealing predominantly with *lowpass* systems in this course. For these systems we will define the *bandwidth* of a system to be that frequency ω_b where the magnitude has fallen 3 dB from the magnitude at dc, or zero frequency. Hence the bandwidth defines the the half power frequency of the system, or that frequency when

$$\frac{1}{2}|H(0)|^2 = |H(j\omega_b)|^2$$

Consider a first order system described by the transfer function

$$G(s) = \frac{K}{\tau s + 1} = \frac{\left(\frac{K}{\tau}\right)}{s + \frac{1}{\tau}}$$

where K is the static gain and τ is the time constant. The pole of the system is a $-\frac{1}{\tau}$. Assuming the system is initially at rest, the unit step response of the system will be given by

$$y(t) = K(1 - e^{-\frac{t}{\tau}})u(t)$$

If we want faster response, we want the time constant τ to become smaller, which means the magnitude of poles of the system become larger (the poles move farther away from the $j\omega$ axis. Figure 12 displays the step response and corresponding frequency response (more precisely, the magnitude portion of the frequency response) for $K/\tau = 1$ (this ratio is fixed) and $\tau = 1$, $\tau = 1/10$ and $\tau = 1/100$, which corresponds to poles at -1, -10, and -100. As this figure indicates, as the response of the system becomes faster (in the time domain), the *bandwidth* of the system increases. For this system the bandwidth will be determined by the pole location, or $\omega_b = 1/\tau$. Thus *the speed of response is directly related to the bandwidth of the system*.

Now let's consider a transfer function with two distinct poles, say at $-p_1$ and $-p_2$, so the transfer function is

$$G(s) = \frac{K}{(s + p_1)(s + p_2)}$$

and the unit step response for $p_1 \neq p_2$ is given by

$$y(t) = \left[\frac{K}{p_1 p_2} + \frac{K}{(p_1 - p_2)p_1} e^{-p_1 t} + \frac{K}{(p_2 - p_1)p_2} e^{-p_2 t} \right] u(t)$$

Figure 13 displays the step response and corresponding frequency response when $K/p_1 p_2 = 1$ and $p_1 = 1, p_2 = 2$, $p_1 = 1, p_2 = 10$, and $p_1 = 1, p_2 = 100$ and Figure 14 for $K/p_1 p_2 = 1$ and $p_1 = 6, p_2 = 7$, $p_1 = 6, p_2 = 20$, and $p_1 = 6, p_2 = 40$.

As these figures demonstrate, the speed of response is determined by the pole closest to the $j\omega$ axis, the *dominant* pole. The *bandwidth* of the system is also determined by the dominant pole. While the second pole affects the shape of both the time and frequency response, it is the dominant pole that really determines the speed of response and the bandwidth. Here the

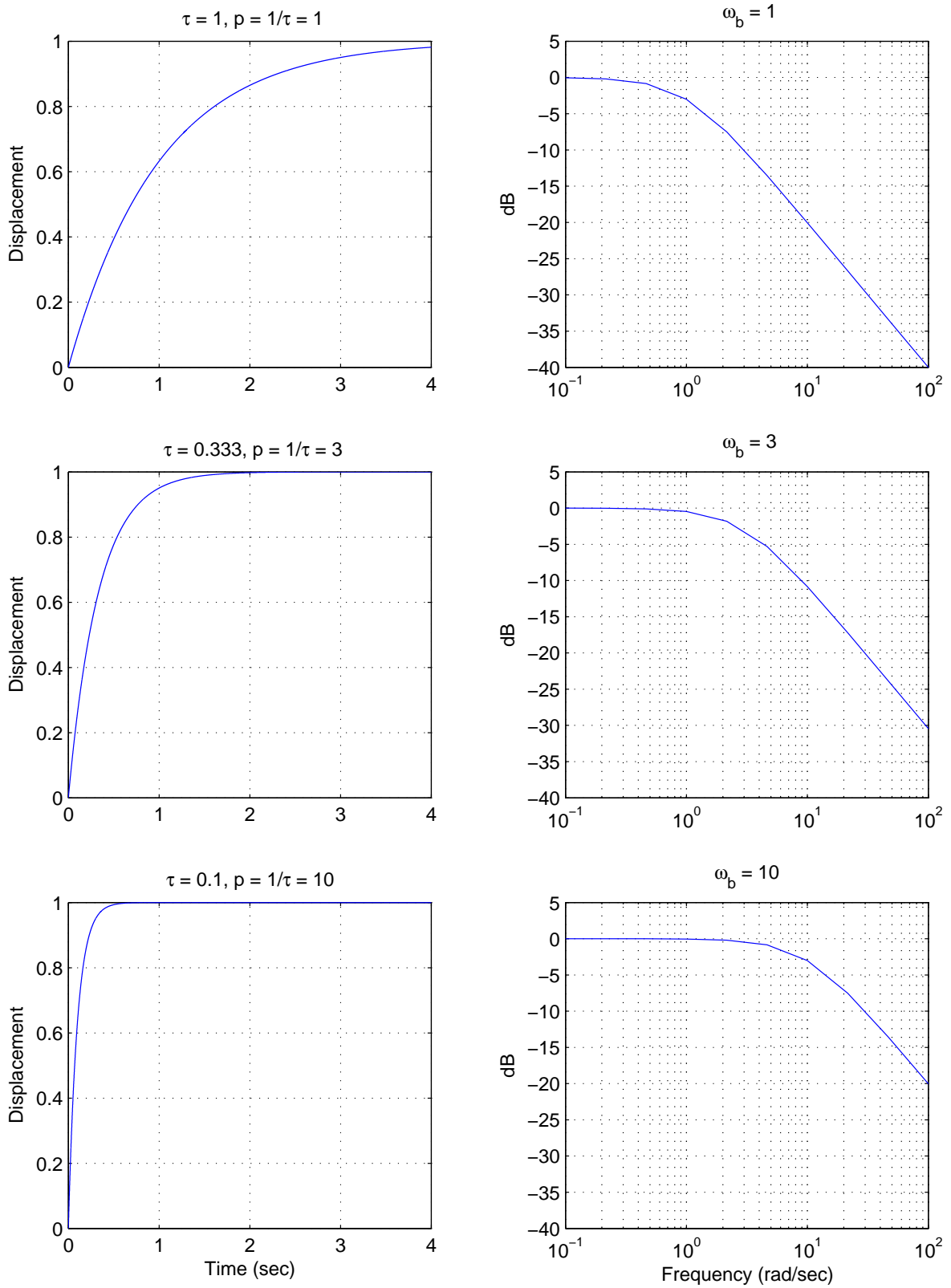


Figure 12: The unit step response and bandwidth for three first order systems. The magnitude of the system pole p is equal to the bandwidth ω_b .

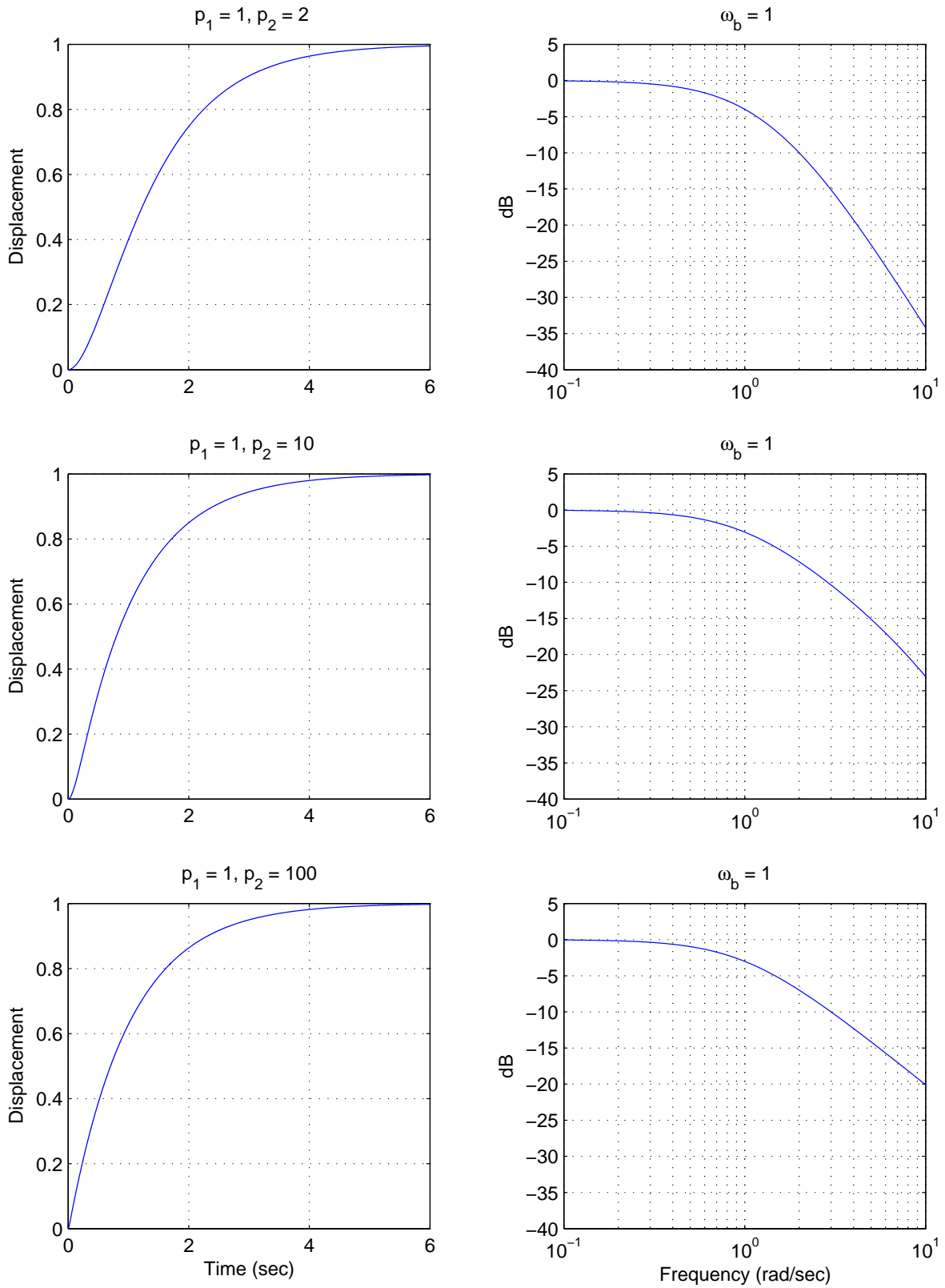


Figure 13: The unit step response and bandwidth for three second order systems with distinct poles. The rate of response is dominated by the *dominant pole* at -1, and the bandwidth (-3 dB point) is determined by this dominant pole.

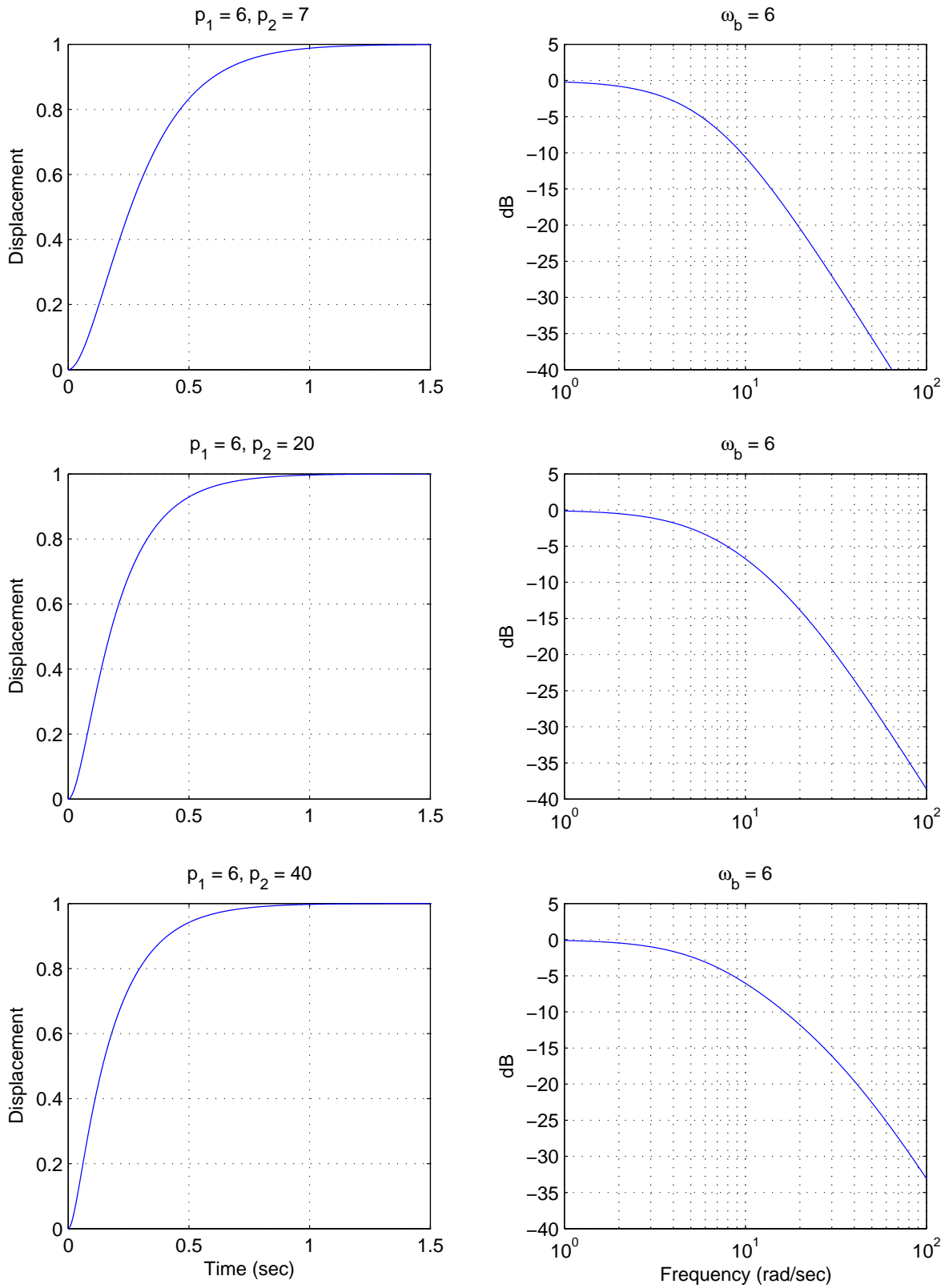


Figure 14: The unit step response and bandwidth for three second order systems with distinct poles. The rate of response is dominated by the *dominant pole* at -6, and the bandwidth (-3 dB point) is determined by this dominant pole.

bandwidth is determined by $\omega_b = \min(p_1, p_2)$. Clearly, if we were to add additional distinct poles to this system, the response would be determined by the dominant poles.

Now let's look at a system with complex conjugate poles, such as our ideal second order system. For an *ideal* second order system with transfer function

$$\begin{aligned} G(s) &= \frac{K}{\frac{1}{\omega_n^2}s^2 + \frac{2\zeta}{\omega_n}s + 1} \\ &= \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \end{aligned}$$

the poles are located at $-\zeta\omega_n \pm j\omega_d$, which is commonly written as $-\sigma \pm j\omega_d$. The characteristic modes that go with these poles are of the form

$$\begin{aligned} e^{-\sigma t} \cos(\omega_d t) \\ e^{-\sigma t} \sin(\omega_d t) \end{aligned}$$

Hence the speed of response will be governed by σ , which is the real part of the pole. The bandwidth of the system is more complicated to determine. As a simple rule, for a fixed ω_d (the imaginary part of the pole), as σ gets larger the bandwidth gets larger. Figures 15 and 16 display the both the step and frequency responses (magnitude only) of an ideal second order systems with complex poles at $[-5 \pm 4j, -10 \pm 4j, -50 \pm 4j]$ and $[-5 \pm 15j, -10 \pm 15j, -50 \pm 15j]$, respectively. Note again, comparing these figures, that it is the real part of the pole that determines the settling time, not the imaginary part.

Why, you might ask, do we care about the bandwidth? There are two reasons. The first is that the bandwidth tells us the types of signals our system will be able to follow. We all know that if the input to a system $G(s)$ is $x(t) = A\cos(\omega_0 t)$, that in steady state the output of the system will be given by

$$y(t) = A|G(j\omega_0)| \cos(\omega_0 t + \angle G(j\omega_0))$$

Hence if the input to our system oscillates "faster" than $\cos(\omega_b t)$, or has higher frequency content than ω_b , where ω_b is the bandwidth, our system will not be able to follow this input very well. More accurately, the output of the system will oscillate at the same frequency as the input, but with a substantially reduced amplitude.

The second reason we care about bandwidth is that all real systems have noise in them. This noise is often introduced to the system by the sensors we need to make measurements, such as measuring the system position or velocity. A fairly reasonable model for noise is white noise. White noise is basically modelled as having constant power spectral density (the power/frequency) of $N_0/2$, or

$$S_{xx}(\omega) = \frac{N_0}{2}$$

If the noise is the input to a system with transfer function $G(\omega)$, then the output power spectral density $S_{yy}(\omega)$ is given by

$$\begin{aligned} S_{yy}(\omega) &= |G(\omega)|^2 S_{xx}(\omega) \\ &= |G(\omega)|^2 \frac{N_0}{2} \end{aligned}$$

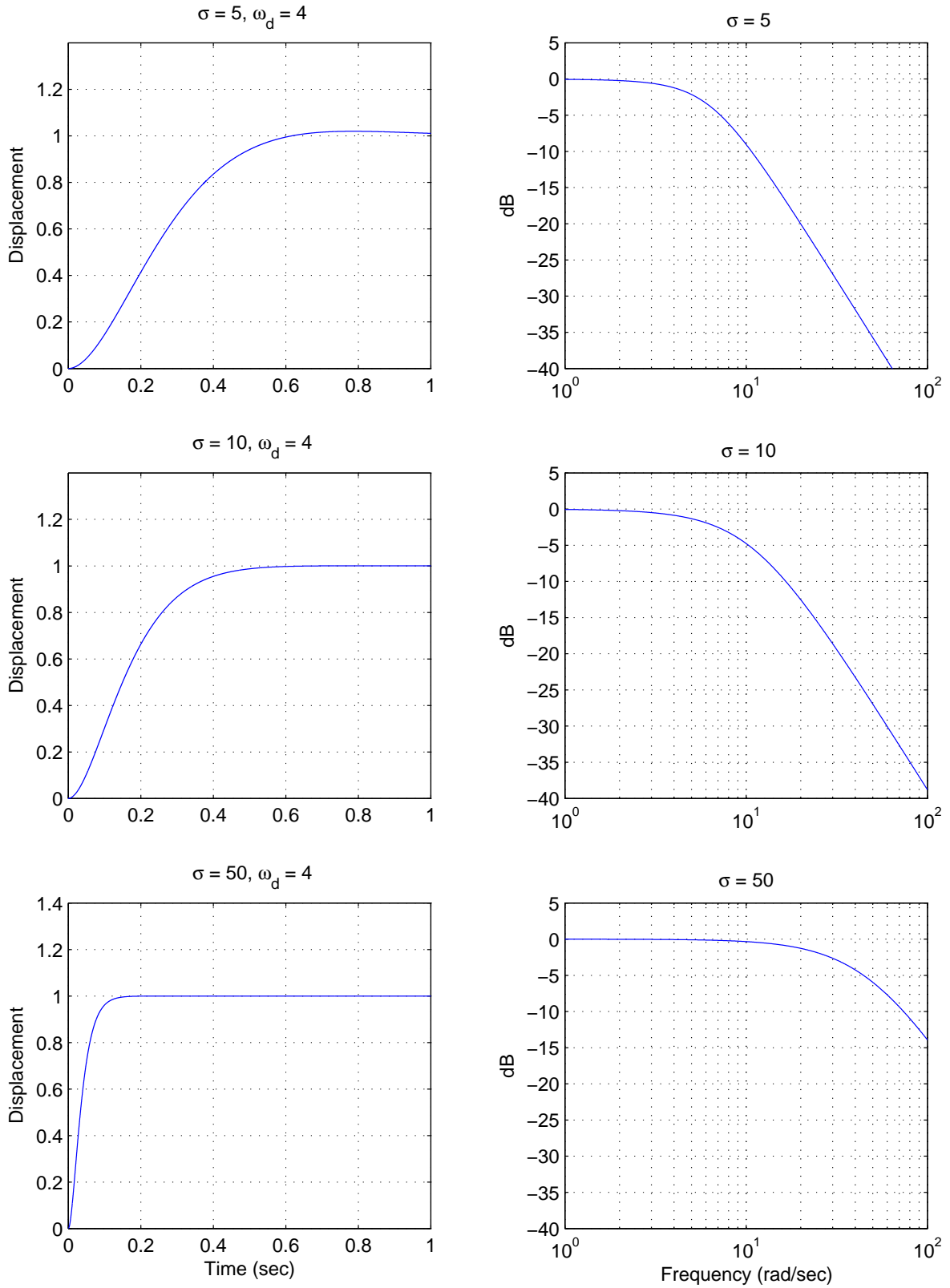


Figure 15: The unit step response and bandwidth for three second order systems with distinct complex conjugate poles. The imaginary parts of the poles are fixed at ± 4 . Note that the faster the time domain response the larger the system bandwidth.

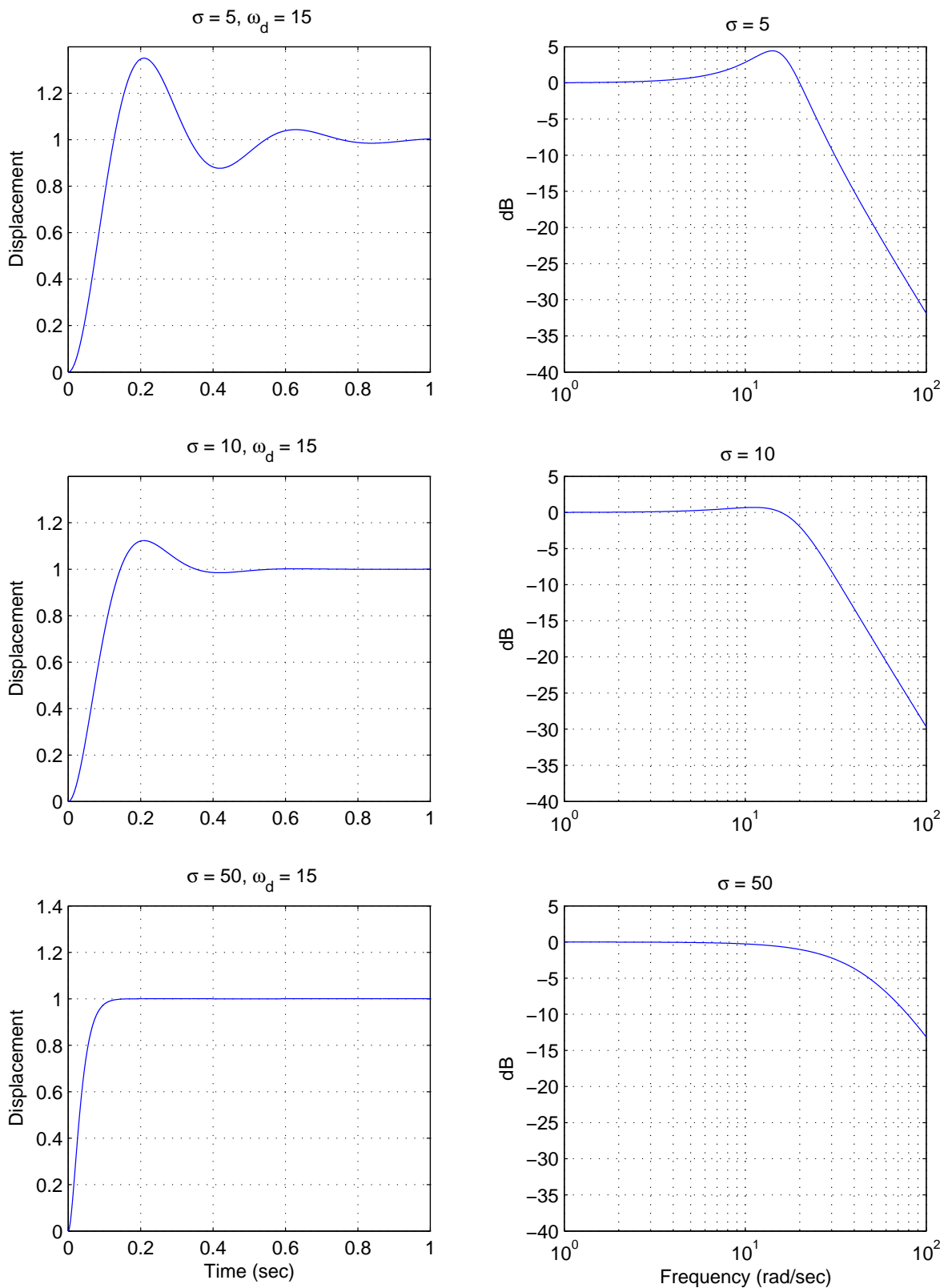


Figure 16: The unit step response and bandwidth for three second order systems with distinct complex conjugate poles. The imaginary parts of the poles are fixed at ± 15 . Note that the faster the time domain response the larger the system bandwidth.

If we want the average noise power, P_N , we need to integrate the power spectral density over the frequency range on interest. Since we are mostly considering *low pass* systems, the range we are interested in is from $-\omega_b$ to ω_b , or

$$\begin{aligned} P_N &= \int_{-\omega_b}^{\omega_b} S_{yy}(\omega) d\omega \\ &= \int_{-\omega_b}^{\omega_b} |G(\omega)|^2 \frac{N_0}{2} d\omega \end{aligned}$$

If we model the magnitude of the transfer function, $|G(\omega)|$, as a constant B over the bandwidth,

$$|G(\omega)| = \begin{cases} 0 & |\omega| > |\omega_b| \\ B & |\omega| < |\omega_b| \end{cases}$$

we get the average noise power estimate

$$P_A = N_0 B^2 \omega_b$$

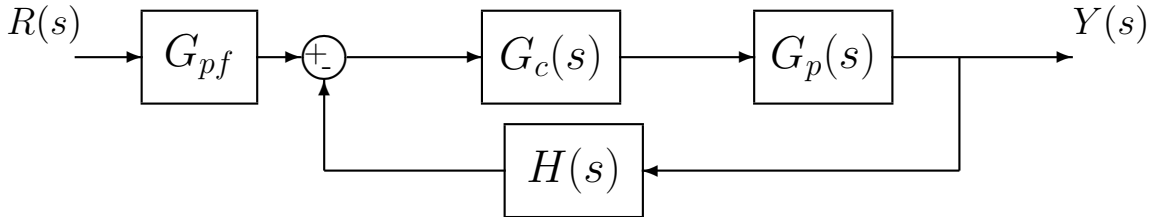
Hence to reduce the average noise power in our system, we want ω_b small. However, this usually means a slower response in the time domain. Bummer.

8 Block Diagrams

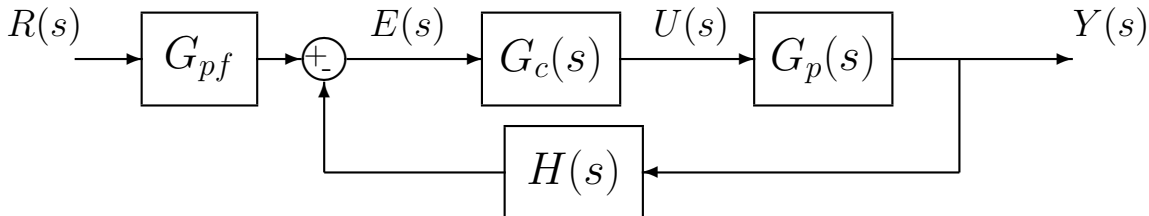
In this chapter we will first review the input-output relationship for the most basic feedback configuration, and then review Mason's gain formula for analyzing more complicated block diagrams.

8.1 Basic Feedback Configuration

The most basic feedback configuration we will deal with is shown below



Here $R(s)$ is the *reference input*. This is usually the signal we are trying to follow. $G_{pf}(s)$ is a *prefilter* which is usually used to condition the signal (change units) or to scale the input to fix the final value of the output. $G_p(s)$ is a model of the *plant* we are trying to control. $G_c(s)$ is a *controller* (or product of controllers) we have designed to improve performance. $Y(s)$ is the system output, and $H(s)$ is a signal conditioner often used to change the units of the output into more convenient units. Two other variables that are usually identified in the block diagram are shown below:



Here $U(s)$ is the input to the plant, so $Y(s) = G_p(s)U(s)$. Finally, $E(s)$ is the error signal, or actuating error.

To determine the overall transfer function, we find

$$\begin{aligned} Y(s) &= G_p(s)U(s) \\ &= G_p(s)G_c(s)E(s) \end{aligned}$$

and

$$E(s) = G_{pf}(s)R(s) - H(s)Y(s)$$

Combining these we get

$$Y(s) = G_p(s)G_c(s) [G_{pf}(s)R(s) - H(s)Y(s)]$$

$$\begin{aligned}
&= G_{pf}(s)G_c(s)G_p(s)R(s) - G_c(s)G_p(s)H(s)Y(s) \\
Y(s) + G_c(s)G_p(s)H(s)Y(s) &= G_{pf}(s)G_c(s)G_p(s)R(s) \\
Y(s) [1 + G_c(s)G_p(s)H(s)] &= G_{pf}(s)G_c(s)G_p(s)R(s)
\end{aligned}$$

or the closed-loop transfer function is

$$G_0(s) = \frac{Y(s)}{R(s)} = \frac{G_{pf}(s)G_c(s)G_p(s)}{1 + G_c(s)G_p(s)H(s)}$$

8.2 Mason's Gain Formula

When we have more complicated block diagrams to analyze, we can continue to analyze them by looking at error signals and using algebra to relate the input to the output signal. An alternative approach is to write the block diagram as a *signal flow graph* and then use Mason's gain rule, which is a systematic method for analyzing signal flow graphs.

A signal flow graph is composed of *directed branches* and *nodes*. Just like block diagrams, these graphs show the flow of signals throughout an interconnected system. However, rather than showing transfer function blocks, the transfer functions (or *transmittances*) are written over the directed branches. Finally, in a signal flow graph all of the branches into a node are summed. Hence if you want negative feedback, you need to include the negative sign in one of the transfer functions (transmittances). Before getting into too much detail, you should examine the examples of block diagrams and the equivalent signal flow graphs in Figures 17-19. In these following diagrams a '1' indicates the transmittance is just unity.

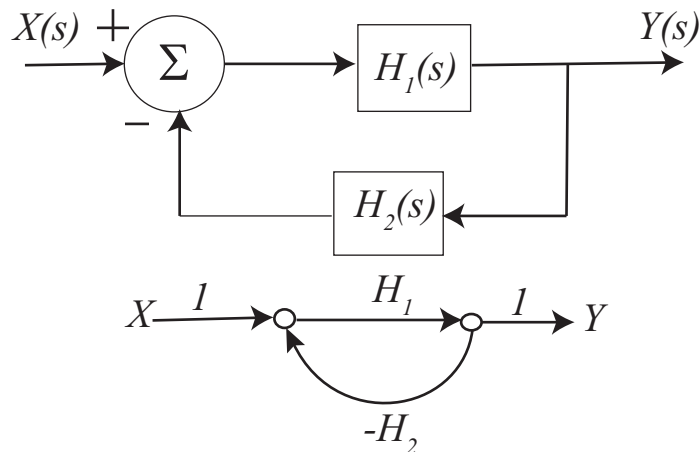


Figure 17: Signal flow graph of a simple feedback system.

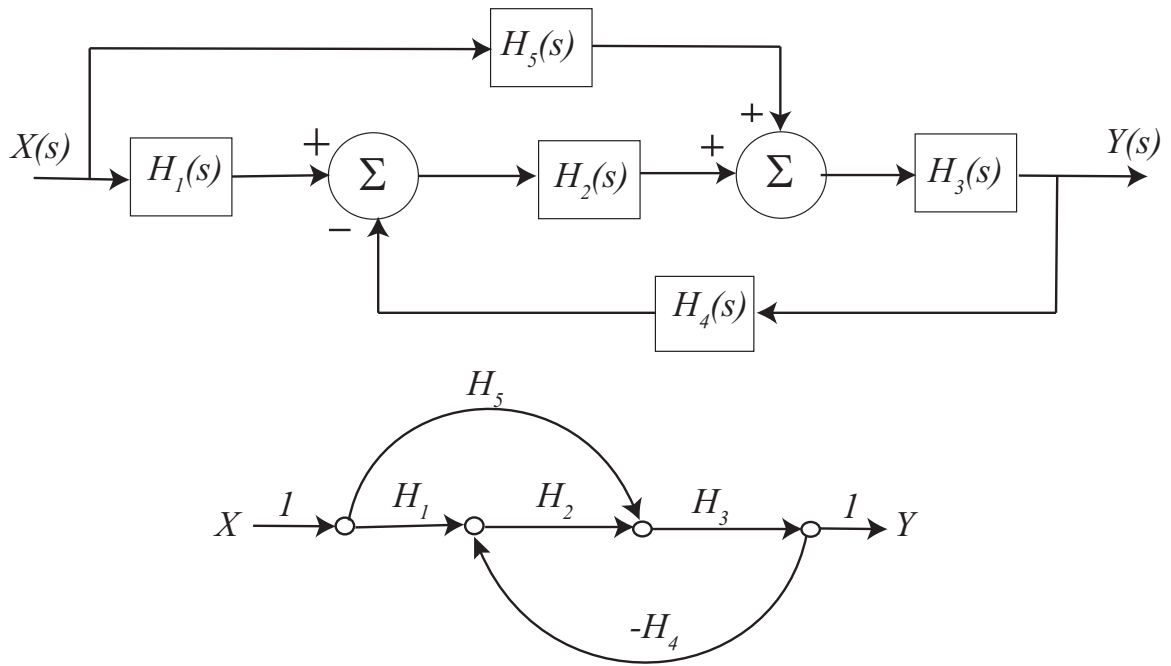


Figure 18: Second example of a block diagram and the equivalent signal flow graph.

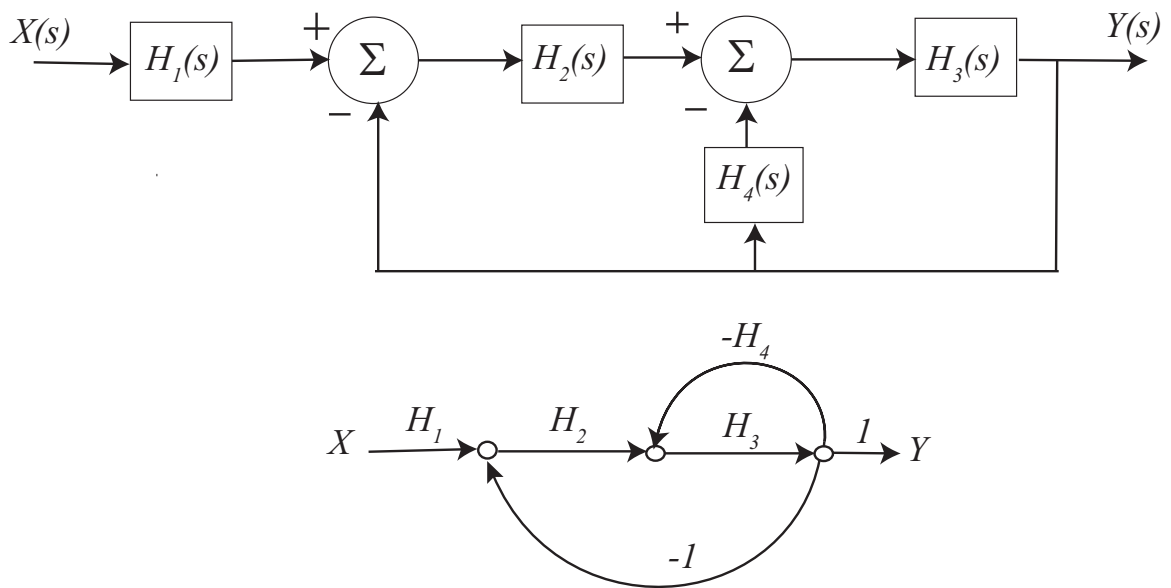


Figure 19: Third example of a block diagram and the equivalent signal flow graph.

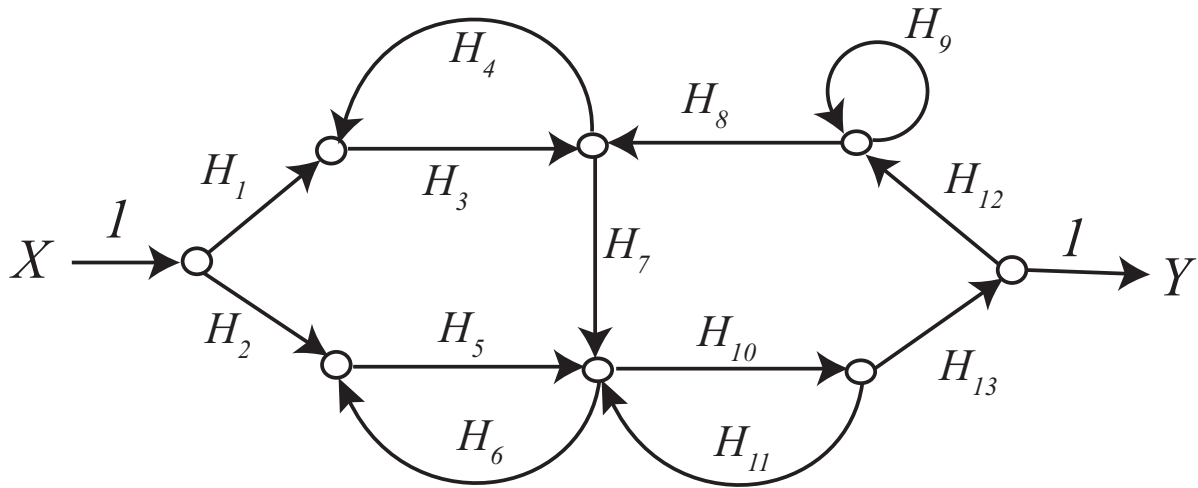


Figure 20: Example signal flow graph used in the definitions example.

We would like to be able to determine the overall system transfer function from the input to the output of a system represented by a signal flow graph. In order to do this we will need to introduce some definitions and apply them to an example problem. It is usually helpful to label some of the *paths* and *loops* on the signal flow graph to try and keep track of them. As you are going through the following definitions and steps, try to label these on the graphs. Let's assume we have a (fairly complicated) signal flow graph like that depicted in Figure 20.

Definition: A *path* is any succession of branches, from input to output, in the direction of the arrows, which does not pass through any node more than once. A *path gain* is the product of the transfer functions (transmittances) of the branches comprising the path.

For the signal flow graph in Figure 20, the path gains are arbitrarily numbered):

$$P_1 = H_1 H_3 H_7 H_{10} H_{13} \quad P_2 = H_2 H_5 H_{10} H_{13}$$

Definition: A *loop* is any closed succession of branches, in the directions of the arrows, which does not pass through any node more than once. The *loop gain* is the product of the transfer functions (transmittances) of the branches comprising the loops.

For the signal flow graph in Figure 20, the loop gains are (arbitrarily numbered):

$$L_1 = H_3 H_4, \quad L_2 = H_5 H_6, \quad L_3 = H_{10} H_{11}, \quad L_4 = H_9, \quad L_5 = H_{10} H_{13} H_{12} H_8 H_7$$

Definition: Two loops are *touching* if they have any node in common. A path and a loop are *touching* if they have any node in common.

Definition: The *determinant* of a signal flow graph is denoted by Δ , and is computed by the formula

$$\begin{aligned}\Delta &= 1 - (\text{sum of all loop gains}) \\ &\quad + (\text{sum of all products of gains of all combinations of 2 nontouching loops}) \\ &\quad - (\text{sum of all products of gains of all combinations of 3 nontouching loops}) \\ &\quad + \dots\end{aligned}$$

For the signal flow graph in Figure 20, the determinant is

$$\begin{aligned}\Delta &= 1 - (L_1 + L_2 + L_3 + L_4 + L_5) \\ &\quad + (L_1L_2 + L_1L_3 + L_1L_4 + L_2L_4 + L_3L_4) \\ &\quad - (L_1L_2L_4 + L_1L_3L_4)\end{aligned}$$

Definition: The *cofactor of a path* is the determinant of the signal flow graph with all of the loops touching the path set to zero.

For the signal flow graph in Figure 20, the cofactors are

$$\begin{aligned}\text{path } P_1 : \quad \Delta_1 &= 1 - L_4 \\ \text{path } P_2 : \quad \Delta_2 &= 1 - (L_1 + L_4) + (L_1L_4)\end{aligned}$$

Definition: The *transfer function* of the signal flow graph is given by the formula

$$H_{system} = \frac{P_1\Delta_1 + P_2\Delta_2 + P_3\Delta_3 + \dots}{\Delta}$$

For the signal flow graph in Figure 20, the system transfer function is

$$H_{system} = \frac{P_1\Delta_1 + P_2\Delta_2}{\Delta}$$

In most instances our systems are not as complicated as this one, but this method will work for any system provided we are systematic. let's do some examples now. You should try these and then check your answers.

Example. For the signal flow diagram shown in Figure 17 we have

$$P_1 = H_1, \quad L_1 = H_1(-H_2) = -H_1H_2, \quad \Delta = 1 + H_1H_2, \quad \Delta_1 = 1$$

which produces the closed loop transfer function

$$H_{system} = \frac{H_1}{1 + H_1H_2}$$

Example. For the signal flow graph shown in Figure 18 we have

$$P_1 = H_3H_5, \quad P_2 = H_1H_2H_3, \quad L_1 = -H_2H_3H_4, \quad \Delta = 1 - L_1, \quad \Delta_1 = 1, \quad \Delta_2 = 1$$

which produces the closed loop transfer function

$$H_{system} = \frac{H_3H_5 + H_1H_2H_3}{1 + H_2H_3H_4}$$

Example. for the signal flow graph shown in Figure 19 we have

$$P_1 = H_1H_2H_3, L_1 = -H_2H_3, L_2 = -H_3H_4, \Delta = 1 - L_1 - L_2, \Delta_1 = 1$$

which produces the closed loop transfer function

$$H_{system} = \frac{H_1H_2H_3}{1 + H_1H_3 + H_3H_4}$$

Example. For the signal flow graph in Figure 21, the closed loop transfer function is

$$H_{system} = \frac{H_1 + H_3}{1 + H_1H_2}$$

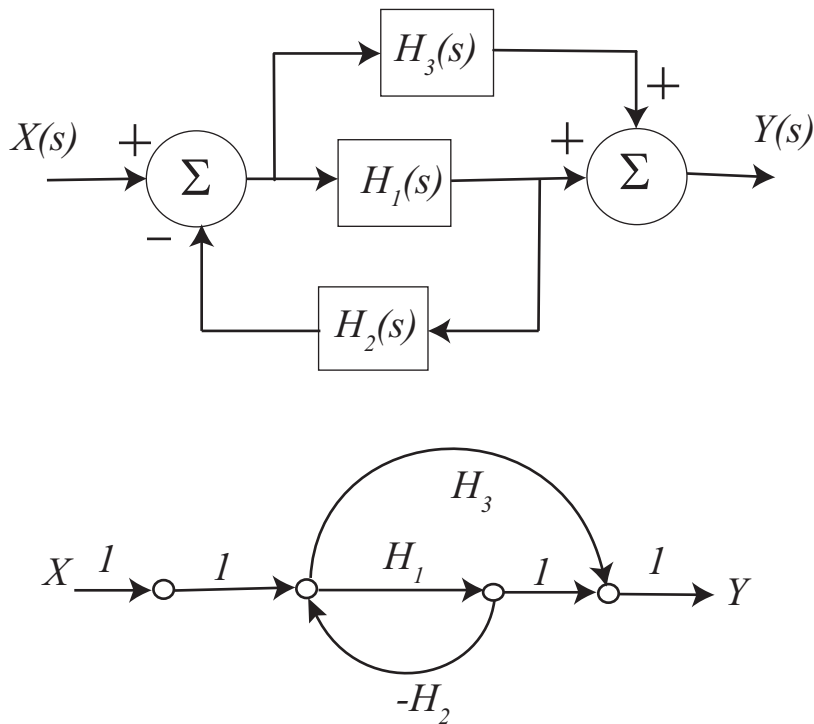


Figure 21: First practice block diagram and corresponding signal flow graph.

Example. For the signal flow graph in Figure 22, the closed loop transfer function is

$$H_{system} = \frac{H_3(1 + H_1H_2) + H_1}{1 + H_1H_2}$$

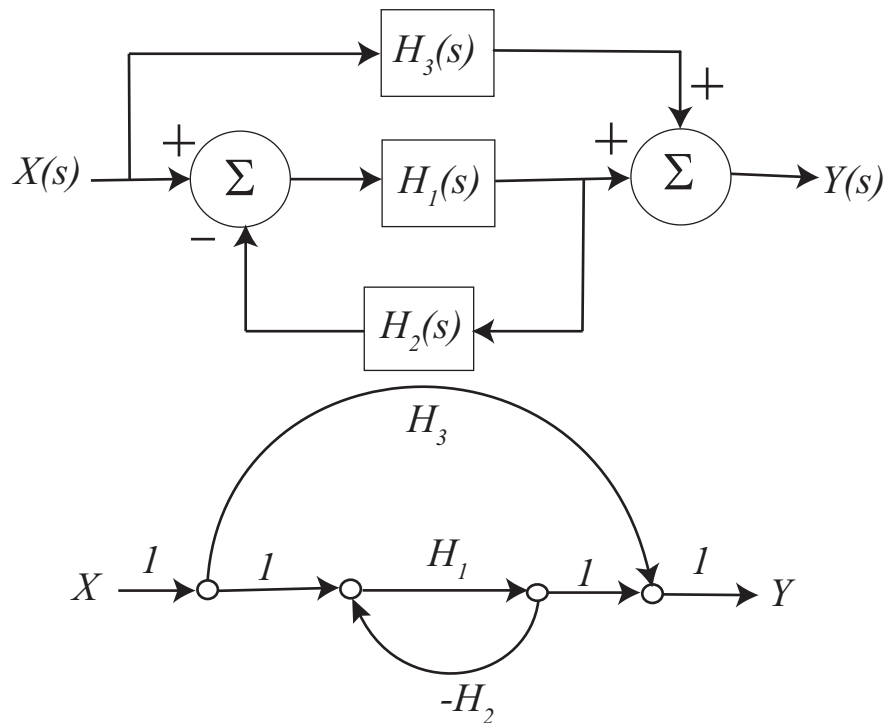


Figure 22: Second practice block diagram and corresponding signal flow graph.

Example. For the signal flow graph in Figure 23, the closed loop transfer function is

$$H_{system} = \frac{H_1 H_4 + H_1 H_2 H_3}{1 - H_1 H_2 H_6 - H_2 H_3 H_5 - H_4 H_5 + H_1 H_2 H_3 + H_1 H_4}$$

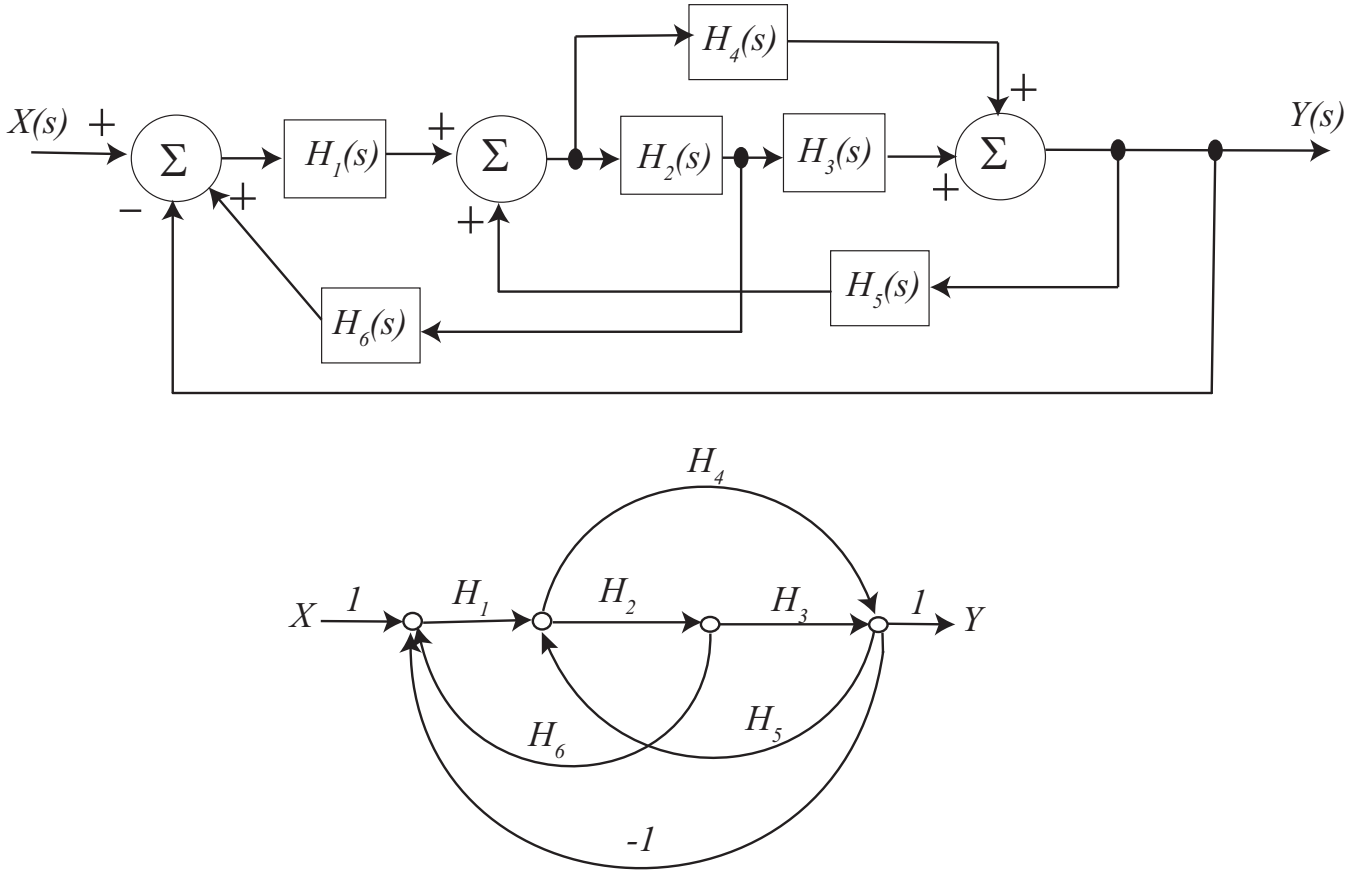
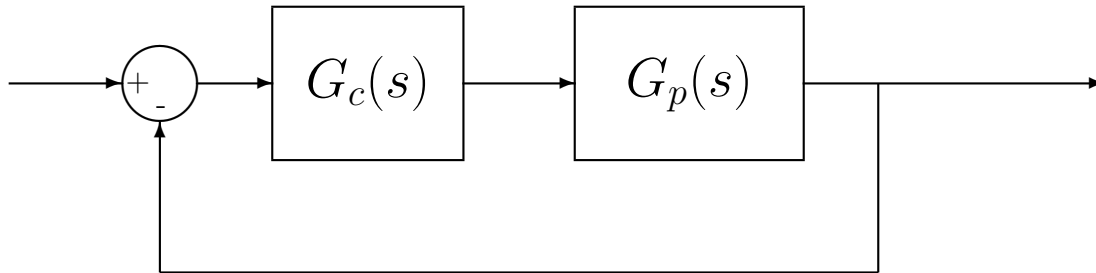


Figure 23: Third practice block diagram and corresponding signal flow graph.

9 Model Matching

The first type of control scheme we will discuss is that of model matching. Here, we assume we have a plant $G_p(s)$ with a controller $G_c(s)$ in a unity feedback scheme, as shown below.



For this closed-loop feedback system, the closed-loop transfer function $G_0(s)$ is given by

$$G_0(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}$$

The object of this course is to determine how to choose the controller $G_c(s)$ so the overall system meets some design criteria. The idea behind model matching is to assume we know what we want the closed loop transfer function $G_0(s)$ to be. Then, since $G_0(s)$ and $G_p(s)$ are known, we can determine the controller $G_c(s)$ as

$$\begin{aligned} [1 + G_c(s)G_p(s)]G_0(s) &= G_c(s)G_p(s) \\ G_0(s) + G_c(s)G_p(s)G_0(s) &= G_c(s)G_p(s) \\ G_0(s) &= G_c(s)G_p(s) - G_c(s)G_p(s)G_0(s) \\ G_0(s) &= G_c(s)G_p(s) [1 - G_0(s)] \end{aligned}$$

or

$$G_c(s) = \frac{G_0(s)}{G_p(s) [1 - G_0(s)]}$$

If we write $G_p(s) = \frac{N(s)}{D(s)}$ and $G_0(s) = \frac{N_0(s)}{D_0(s)}$, we can rewrite this as

$$G_c(s) = \frac{N_0(s)D(s)}{D_0(s)N(s) - N(s)N_0(s)}$$

This form is often easier to work with. However, we still have to be careful to eliminate common factors from the numerator and denominator.

While this type of model matching looks simple, there are certain restrictions on when this will work. The closed loop transfer function $G_0(s)$ is said to be implementable if¹

1. The controller $G_c(s)$ is a proper rational transfer function
2. The controller $G_c(s)$ is stable

Consider a plant with proper transfer function $G_p(s) = \frac{N(s)}{D(s)}$ where we want the closed loop transfer function to be $G_0(s) = \frac{N_0(s)}{D_0(s)}$. We can find a $G_c(s)$ so that $G_0(s)$ is implementable only under the following conditions:

1. The degree of $D_0(s)$ - the degree of $N_0(s) \geq$ the degree $D(s)$ - the degree of $N(s)$
2. All right half plane zeros of $N(s)$ are retained in $N_0(s)$ (the RHP zeros of the plant must also be in the closed-loop transfer function)
3. $G_0(s)$ is stable, i.e., all poles of $G_0(s)$ are in the left half plane (none on the axes)

Example 1. Consider the system with plant

$$G_p(s) = \frac{(s+2)(s-1)}{s(s^2-2s+2)}$$

Are the following closed loop transfer functions implementable?

1. $G_0(s) = 1$. (*No, it violates (1) and (2)*)
2. $G_0(s) = \frac{(s+2)}{(s+3)(s+1)}$. (*No, violates (2)*)
3. $G_0(s) = \frac{(s-1)}{(s+3)(s+1)}$. (*Yes*)
4. $G_0(s) = \frac{(s-1)}{s(s+2)}$. (*No, violates (3)*)
5. $G_0(s) = \frac{(s-1)}{(s+3)(s+1)^2}$. (*Yes*)
6. $G_0(s) = \frac{(s-1)(2s-3)}{(s+2)^3}$. (*Yes*)

Now that we know when we can use model matching, we need to find some good models. That is, how do we find a desirable $G_0(s)$? We will look at four possible choices, ITAE optimal systems, deadbeat systems, quadratic optimal systems, and steady state frequency response matching.

¹There are other restrictions, but they are not important in this course.

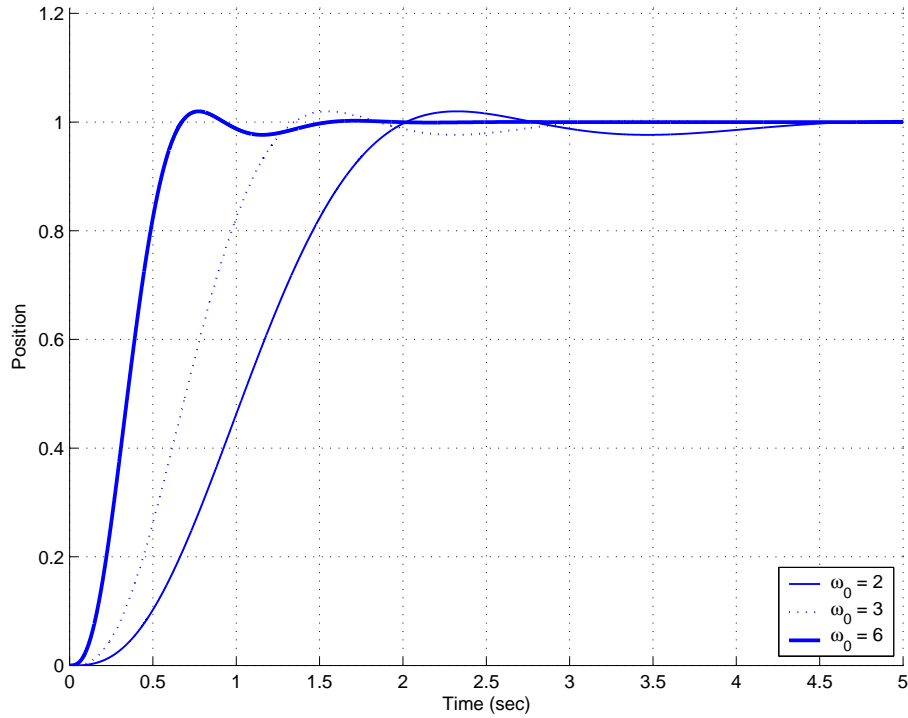


Figure 24: The unit step response for a third order ITAE system with $\omega_0 = 2, 3$ and 6 radians/second. This system will have a steady state error of zero for a step input.

9.1 ITAE Optimal Systems

ITAE optimal systems minimize the **I**ntegral of **T**ime multiplied by the **A**bsolute **E**rror. These have been determined numerically. The second, third, and fourth order ITAE systems with a steady state error of zero for a step input have the following closed loop transfer functions

$$G_0(s) = \frac{\omega_0^2}{s^2 + 1.4\omega_0 s + \omega_0^2}$$

$$G_0(s) = \frac{\omega_0^3}{s^3 + 1.75\omega_0 s^2 + 2.15\omega_0^2 s + \omega_0^3}$$

$$G_0(s) = \frac{\omega_0^4}{s^4 + 2.1\omega_0 s^3 + 3.4\omega_0^2 s^2 + 2.7\omega_0^3 s + \omega_0^4}$$

Since $G_0(0) = 1$ for each of these closed loop transfer functions the steady state error will clearly be 0 for a step input. Figure 24 shows the step response for a third order ITAE system with $\omega_0 = 2, 3$ and 6 radians/second. As this figure shows, the larger the value of ω_0 , the faster the response of the system.

The second, third, and fourth order ITAE systems that have zero steady state errors for *both* step and ramp inputs have the following closed loop transfer functions

$$G_0(s) = \frac{3.2\omega_0 s + \omega_0^2}{s^2 + 3.2\omega_0 s + \omega_0^2}$$

$$G_0(s) = \frac{3.25\omega_0^2 s + \omega_0^3}{s^3 + 1.75\omega_0 s^2 + 3.25\omega_0^2 s + \omega_0^3}$$

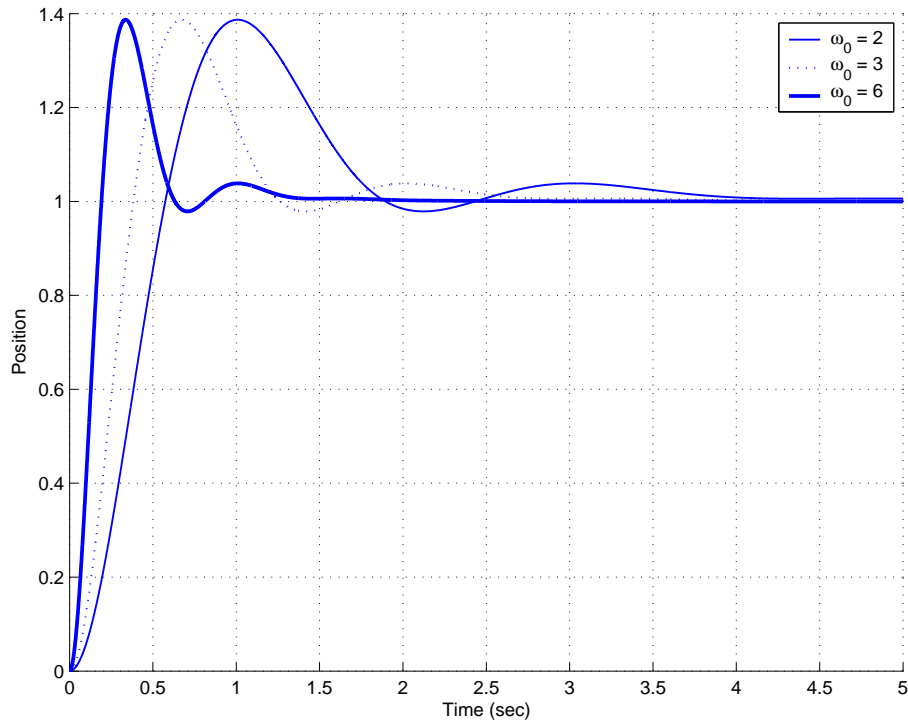


Figure 25: The unit step response for a third order ITAE system that has zero steady state error for both a step and ramp input with $\omega_0 = 2, 3$ and 6 radians/second.

$$G_0(s) = \frac{5.14\omega_0^3 s + \omega_0^4}{s^4 + 2.41\omega_0 s^3 + 4.93\omega_0^2 s^2 + 5.14\omega_0^3 s + \omega_0^4}$$

Since the coefficients of s^1 and s^0 are identical for both the numerator and denominator of these closed loop transfer functions, the steady state errors will be zero for both step and ramp inputs. Figure 25 shows the step response for a third order ITAE system with $\omega_0 = 2, 3$ and 6 radians/second. As this figure shows, we are paying for the steady state errors for *both* a step and ramp input with a substantial overshoot. As with the ITAE systems with zero steady state error for only a step input, the larger the value of ω_0 , the faster the response of the system.

You, the designer, need to choose the value of ω_0 . The larger the ω_0 , the faster the system responds (good) and the larger the control effort (bad).

9.2 Deadbeat Systems

A *deadbeat* response is a response that that proceeds rapidly to the desired level and holds that level with minimal overshoot. These have also been determined numerically. The second, third, and fourth order closed loop transfer functions for deadbeat control are

$$G_0(s) = \frac{\omega_0^2}{s^2 + 1.82\omega_0 s + \omega_0^2}$$

$$G_0(s) = \frac{\omega_0^3}{s^3 + 1.90\omega_0 s^2 + 2.20\omega_0^2 s + \omega_0^3}$$

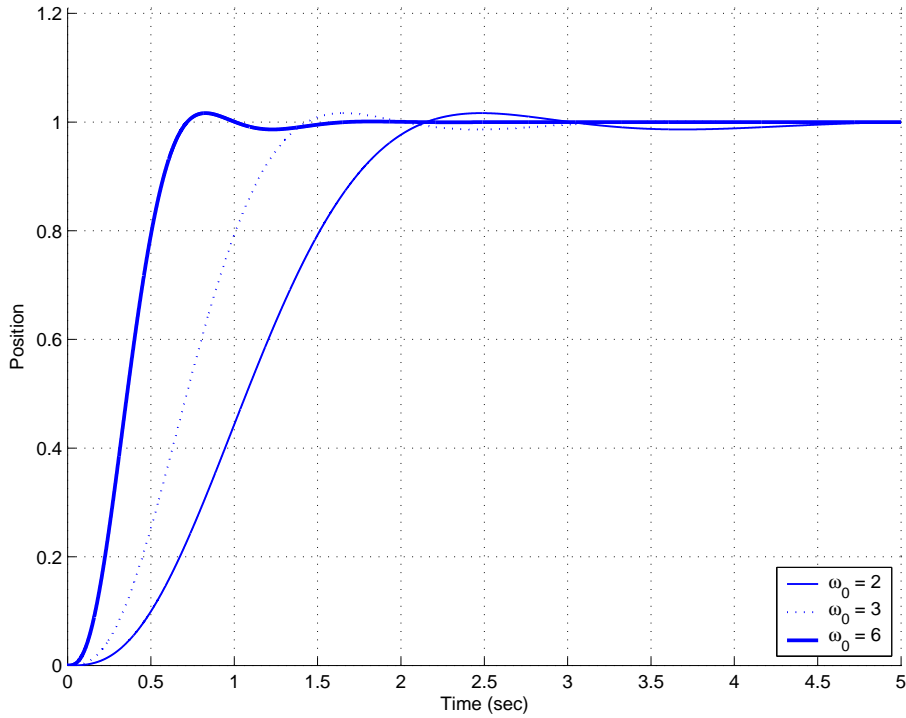


Figure 26: The unit step response for a third order deadbeat system with $\omega_0 = 2, 3$ and 6 radians/second.

$$G_0(s) = \frac{\omega_0^4}{s^4 + 2.20\omega_0 s^3 + 3.50\omega_0^2 s^2 + 2.80\omega_0^3 s + \omega_0^4}$$

Since $G_0(0) = 1$ for each of these closed loop transfer functions, it should be clear that we will have $e_{ss} = 0$ for a step input. However the steady state error will not be zero for a ramp input. Figure 26 shows the step response for a third order deadbeat response for $\omega_0 = 2, 3$ and 6 radians/second. Comparing Figure 26 with Figure 24, it is clear that the ITAE and deadbeat responses are quite similar. This is also clear by examining the transfer functions. As with the ITAE systems, the larger the value of ω_0 the faster the response of the system (good) and the more control effort (bad) required.

9.3 Summary and Caveates

In the first part of this section, the conditions under which it is possible to obtain an implementable closed loop transfer function $G_0(s)$ have been given, it may not be possible to find such a $G_0(s)$ using the ITAE, deadbeat, or quadratic optimal criteria. In particular, the ITAE and deadbeat methods will not work for a system with RHP zeros in the plant.

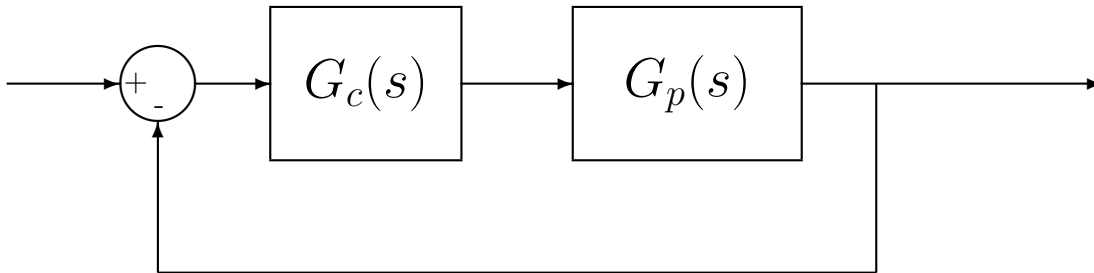
The model matching methods we have discussed often utilize pole-zero cancellations between the controller $G_c(s)$ and the plant $G_p(s)$ to achieve the desired closed-loop transfer function. This requires that we have a fairly good model of the plant being cancelled. If we have a good model of the plant and a stable pole is being cancelled, this is usually OK. However cancelling an unstable pole is not acceptable. In addition, the plant may change over time, and we are

dealing with models of the plant in the first place. Hence the pole-zero cancellations may not be very effective for some systems.

10 System Type and Steady State Errors

10.1 Review

Let's assume we have a control system in a unity feedback configuration as shown below:



where $G_p(s)$ is the plant transfer function and $G_c(s)$ is a controller. Note that there is no prefilter! The closed-loop transfer function is given by

$$G_0(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}$$

We already know that if we write

$$G_0(s) = \frac{n_m s^m + n_{m-1} s^{m-1} + \dots + n_2 s^2 + b_1 s + b_0}{s^n + d_{n-1} s^{n-1} + \dots + d_2 s^2 + d_1 s + d_0}$$

that the steady state error for an input step of amplitude A is given by

$$e_{ss} = A \frac{d_0 - n_0}{d_0}$$

If $G_0(0) = 1$ (the constant terms in the numerator and denominator are the same), then $e_{ss} = 0$. The steady state error for an input of tA is given by

$$e_{ss} = A \frac{d_1 - n_1}{d_0}$$

If the coefficients of s^1 and s^0 are the same, then the steady state error for a ramp input is zero.

10.2 System Type For a Unity Feedback Configuration

Unity feedback configurations are very common, and we would like to be able to analyze this type of system very quickly without computing the closed-loop transfer function.

Let's assume we group all of the transfer functions together into one transfer function, which we will call $G(s)$, so $G(s) = G_c(s)G_p(s)$. Assume we write $G(s)$ as²

$$G(s) = \frac{K(T_a s + 1)(T_b s + 1) \dots (T_m s + 1)}{s^N (T_1 s + 1)(T_2 s + 1) \dots (T_n s + 1)}$$

²We do not actually want to rewrite $G(s)$, this is just used for illustrative purposes.

This is said to be a type N system, where N is the number of poles at the origin of $G(s)$. (These poles at the origin are also called “free integrators” or just “integrators”.) If the system output is $Y(s)$ and the system input is $R(s)$, then the system transfer function is

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)}$$

Let’s define the error $E(s)$ to be the difference between the input $R(s)$ and the output $Y(s)$,

$$\begin{aligned} E(s) &= R(s) - Y(s) \\ &= R(s) - \frac{G(s)}{1 + G(s)} R(s) \\ &= \frac{R(s) \{ [1 + G(s)] - G(s) \}}{1 + G(s)} \\ &= \frac{R(s)}{1 + G(s)} \end{aligned}$$

The steady state error is then

$$e_{ss} = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \frac{sR(s)}{1 + G(s)}$$

We will use this expression to determine expressions for the steady state error for both step and ramp inputs for unity feedback systems.

10.3 Steady State Errors for Step and Ramp Inputs

As we have previously defined, the steady state error for a step input is the difference between a step input $r(t)$ and the corresponding output $y(t)$ as we let $t \rightarrow \infty$. Hence, $e_{ss} = \lim_{s \rightarrow 0} sE(s)$ for $R(s) = \frac{A}{s}$, or

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} sE(s) \\ &= \lim_{s \rightarrow 0} \frac{s \frac{A}{s}}{1 + G(s)} \\ &= \lim_{s \rightarrow 0} \frac{A}{1 + G(s)} \\ &= \frac{A}{1 + G(0)} \\ &= \frac{A}{1 + K_p} \end{aligned}$$

The position error constant K_p is defined to be $G(0)$. For a type 0 system, $K_p = K$ and $e_p = \frac{A}{1+K}$, while for a type 1 or higher system, $K_p = \infty$ and $e_{ss} = 0$.

The steady state error for a ramp input is the difference between the ramp input $r(t)$ and the corresponding output $y(t)$ as we let $t \rightarrow \infty$. Hence $e_{ss} = \lim_{s \rightarrow 0} sE(s)$ for $R(s) = \frac{A}{s^2}$, or

$$\begin{aligned} e_{ss} &= \lim_{s \rightarrow 0} sE(s) \\ &= \lim_{s \rightarrow 0} \frac{s \frac{A}{s^2}}{1 + G(s)} \\ &= \lim_{s \rightarrow 0} \frac{A}{s + sG(s)} \\ &= \lim_{s \rightarrow 0} \frac{A}{sG(s)} \\ &= \frac{A}{K_v} \end{aligned}$$

The *velocity error constant* K_v is defined to be $\lim_{s \rightarrow 0} sG(s)$. For a type 0 system, $K_v = 0$ and $e_{ss} = \infty$. For a type 1 system, $K_v = K$ and $e_{ss} = \frac{A}{K}$. For a type 2 or higher system, $K_v = \infty$ and $e_{ss} = 0$.

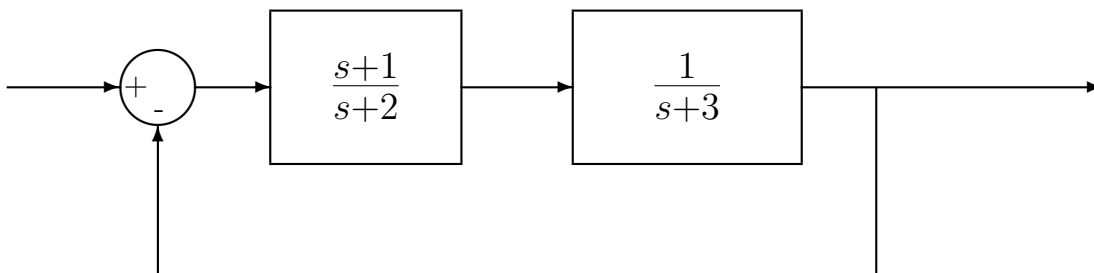
We can summarize these results in Table 1 below.

System Type	e_{ss} for step input $Au(t)$	e_{ss} for ramp input $Atu(t)$
0	$\frac{A}{1+K_p}$	∞
1	0	$\frac{A}{K_v}$
2	0	0
3	0	0

Table 1: Summary of system type (number of poles at the origin), steady state error for a step input $Au(t)$, and steady state error for a ramp input $Atu(t)$ for a unity feedback system.

10.4 Examples

Example 1. For the unity feedback system shown below, determine the system type, the steady state error for a step input and the steady state error for a ramp input.



Here

$$G(s) = \frac{(s+1)}{(s+2)(s+3)}$$

there are no poles at zero so this is a type 0 system. The position error constant is then

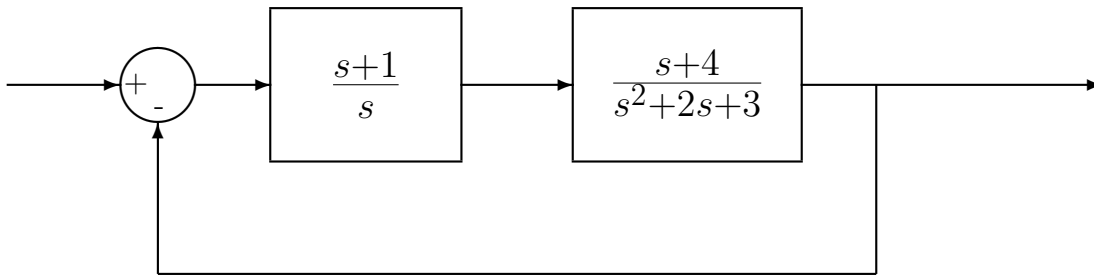
$$\begin{aligned} K_p &= \lim_{s \rightarrow 0} G(s) \\ &= \frac{1}{(2)(3)} \\ &= \frac{1}{6} \end{aligned}$$

so for a step input

$$\begin{aligned} e_{ss} &= \frac{A}{1 + K_p} \\ &= \frac{A}{1 + 0.1667} \\ &= 0.857A \end{aligned}$$

Since e_{ss} is not zero for a step input, $e_{ss} = \infty$ for a ramp input. ($e_{ss} = \infty$ for a ramp input since this is a type 0 system).

Example 2. For the unity feedback system shown below, determine the system type, the steady state error for a step input and the steady state error for a ramp input.



Here

$$G(s) = \frac{(s+1)(s+4)}{s(s^2+2s+3)}$$

so there is one pole at zero, hence this is a type 1 system. The steady state error for a step input is then $e_{ss} = 0$. Note that we do not need to do any computation for this once we recognize this as a type 1 system! The velocity error constant is then

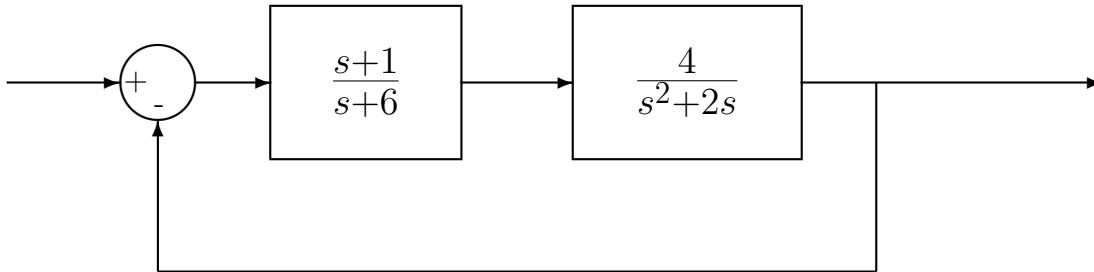
$$\begin{aligned} K_v &= \lim_{s \rightarrow 0} sG(s) \\ &= \frac{(1)(4)}{3} \\ &= \frac{4}{3} \end{aligned}$$

so the steady state error for a ramp input is

$$e_{ss} = \frac{A}{K_v}$$

$$\begin{aligned}
 &= \frac{A}{\left(\frac{4}{3}\right)} \\
 &= 0.75A
 \end{aligned}$$

Example 3. For the unity feedback system shown below, determine the system type, the steady state error for a step input and the steady state error for a ramp input.



Here

$$G(s) = \frac{(s+1)(4)}{(s+6)s(s+2)}$$

there is one pole at zero so this is a type 1 system. The steady state error for a step input is then $e_{ss} = 0$. The velocity error constant is then

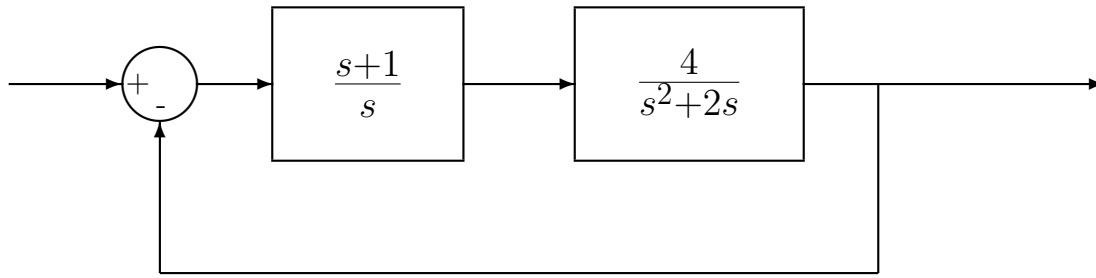
$$\begin{aligned}
 K_v &= \lim_{s \rightarrow 0} sG(s) \\
 &= \frac{(1)(4)}{(6)(2)} \\
 &= \frac{1}{3}
 \end{aligned}$$

so the steady state error for a ramp input is

$$\begin{aligned}
 e_{ss} &= \frac{A}{K_v} \\
 &= \frac{A}{\left(\frac{1}{3}\right)} \\
 &= 3A
 \end{aligned}$$

Example 4. For the unity feedback system shown below, determine the system type, the steady

state error for a step input and the steady state error for a ramp input.



Here

$$G(s) = \frac{(s+1)(4)}{s^2(s+2)}$$

there are two poles at zero so this is a type 2 system. Hence the steady state error for both a step input and a ramp input are zero.

11 Controller Design Using the Root Locus

The *root locus* is a graph in the complex plane that shows how the *poles* of a closed loop system vary as a single parameter varies. *Points on the root locus are the only possible closed loop pole locations!* If a point is not on the root locus, it is not possible for a closed loop pole to be located there (with the controller under consideration).

Knowledge of the location of the closed loop poles helps estimated the step response of a system and if a system is likely to become unstable for some values of a varying parameter. However, the zeros of the system also affect the step response, and how they affect the step response is not as apparent from the root locus. Matlab's *sisotool* is a very powerful tool for utilizing the root locus to produce the desired step response. In this chapter some of the more basic rules for constructing the root locus are included. The goal is not to necessarily be able to draw a root locus in all it's detail, but rather to understand what is going on and determine if a particular type of controller is likely to ever work. With the exception of the first rule, there will be no motivation or derivation for the rule. As far as we are concerned, they just are.

After the root locus is introduced, we will introduce proportional, integral, derivative, and lead controllers, as well as various useful combinations. We will then utilize the root locus in conjunction with these controller to determine if it is possible to use one of these controllers to accomplish what we want. Mostly we want to rule out controllers that we know won't work.

Special Note: *In the following examples, it cannot be stressed enough that our guidelines for settling time and percent overshoot are based on a closed loop transfer function which is an ideal second order system, which contains no zeros. As long as there are only poles in the closed loop transfer function these estimates tend to work fairly well. However, once the closed loop transfer function has zeros, these estimates may no longer be very accurate. This is when a tool like Matlab's sisotool is extremely valuable. In the following sections, the ability to meet the design constraints will be based soley on the locations of the closed loop poles and the assumption of an ideal second order system. The only way to really tell if a model of a system meets the design specifications is to simulate the response of the model.*

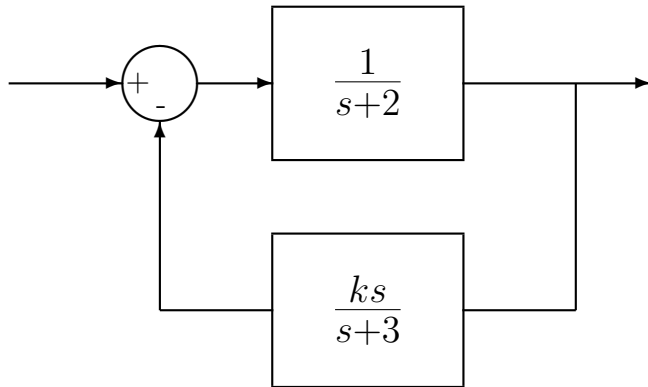
11.1 Standard Root Locus Form

In order to use the root locus techniques (and Matlab), we need to write the equation for the closed loop poles in the standard form

$$1 + kG(s) = 0$$

where k is the only free parameter. It is important to remember that we are using the root locus to determine the closed loop pole locations, not the locations of the closed loop zeros.

Example 1. Consider the following system:



We would like to use the root locus to determine the locations of the closed loop poles as the parameter k varies. To do this, we must first determine the closed loop transfer function.

$$G_0(s) = \frac{\left(\frac{1}{s+2}\right)}{1 + \left(\frac{ks}{s+3}\right)\left(\frac{1}{s+2}\right)}$$

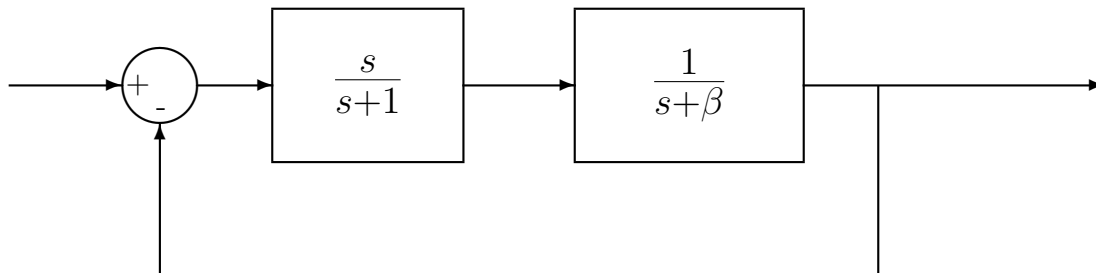
Hence the equation we need is

$$1 + k \left[\frac{s}{(s+2)(s+3)} \right] = 0$$

and

$$G(s) = \frac{s}{(s+2)(s+3)}$$

Example 2. Consider the following system:



We would like to use the root locus to determine the locations of the closed loop poles as the parameter β varies. To do this, we must first determine the closed loop transfer function.

$$G_0(s) = \frac{\left(\frac{s}{s+1}\right)\left(\frac{1}{s+\beta}\right)}{1 + \left(\frac{s}{s+1}\right)\left(\frac{1}{s+\beta}\right)}$$

$$\begin{aligned}
&= \frac{s}{(s+1)(s+\beta)+s} \\
&= \frac{s}{s^2+2s+\beta(s+1)} \\
&= \frac{\left(\frac{s}{s^2+2s}\right)}{1+\beta\left(\frac{s+1}{s^2+2s}\right)} \\
&\quad \frac{\left(\frac{1}{s+2}\right)}{1+\beta\left(\frac{s+1}{s(s+2)}\right)}
\end{aligned}$$

Hence the equation we need is

$$1 + \beta \left[\frac{s+1}{s(s+2)} \right] = 0$$

and

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

11.2 Examples

In the following sections we will determine the root locus plot for the following functions:

$$\begin{aligned}
G_1(s) &= \frac{1}{(s+1)(s+2)} \\
G_2(s) &= \frac{1}{(s+1)(s+2)(s+3)} \\
G_3(s) &= \frac{1}{(s+1)(s+2)(s+3)(s+4)} \\
G_4(s) &= \frac{(s+2)}{(s+1)(s+3)} \\
G_5(s) &= \frac{(s+2)}{(s+1)(s+3)(s+4)} \\
G_6(s) &= \frac{(s+2)(s+3)}{(s+1)(s+4)} \\
G_7(s) &= \frac{(s+3)(s+4)}{(s+1)(s+2)} \\
G_8(s) &= \frac{(s+2)}{(s+1)(s+3)(s^2+2s+2)} \\
G_9(s) &= \frac{(s+2)(s^2+5s+10.25)}{(s+1)(s+3)(s+4)} \\
G_{10}(s) &= \frac{1}{s^2+2s+2} \\
G_{11}(s) &= \frac{1}{s(s^2+2s+2)}
\end{aligned}$$

$$G_{12}(s) = \frac{s^2 + 2s + 2}{s^2 + 5s + 10}$$

As you try to sketch a root locus plot, you should go through the following steps in the order in which they are presented. You do not need all of the steps for all root locus plots. Once all poles have been paired with zeros you are done and there is no need to continue. In each case, we will assume the feedback configuration shown in Figure 27.

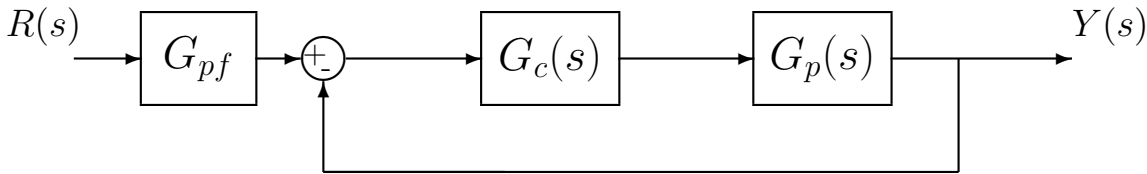


Figure 27: Assumed feedback configuration for root locus analysis in this chapter.

11.3 Loci Branches

Let's consider the expression in standard root locus form

$$1 + kG(s) = 0$$

where k is the parameter free to vary. Let's break up the product $G(s)$ into numerator and denominator polynomials

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

We can then rewrite our first expression as

$$d(s) + kn(s) = 0$$

Now when $k = 0$, this expression is true only if $d(s) = 0$, i.e., we are at the *poles* of $G(s)$. As $k \rightarrow \infty$, we need to have $n(s) = 0$, i.e. we are at the *zeros* of $G(s)$. Hence our first rule can be summarized as follows:

The root locus starts ($k = 0$) *at the poles of* $G(s)$ *and ends on the zeros of* $G(s)$.

There are a few basic points we need to make to clarify this rule and understand it's application:

- It is important to note that each pole of $G(s)$ must end at a zero of $G(s)$. Thus for each pole of $G(s)$ that a branch of the root locus begins at, we must determine the location of the corresponding zero where it will arrive as $k \rightarrow \infty$. Once we have "paired" a pole with a zero we are done with them both.

- Since we are dealing with real systems, all of our transfer functions will have real coefficients. Thus all of the poles and zeros of our system will either be purely real or occur in complex conjugate pairs. This implies that *the root locus plot will symmetric about the real axis*.
- There is usually some confusion here about *poles* and *zeros*. The root locus is a plot of how the poles of the **closed loop system** vary as a parameter is varied. However, to construct the root locus, we need to use the poles and zeros of $G(s)$. However, the poles of $G(s)$ are not the poles of the closed loop system.

11.4 Real Axis Segments

Before we look anywhere else, we always look along the real axis for points on the root locus. To determine if there are any parts of the real axis on the root locus we use the following rule

The root locus includes all points along the real axis to the left of an odd number of poles plus zeros of $G(s)$.

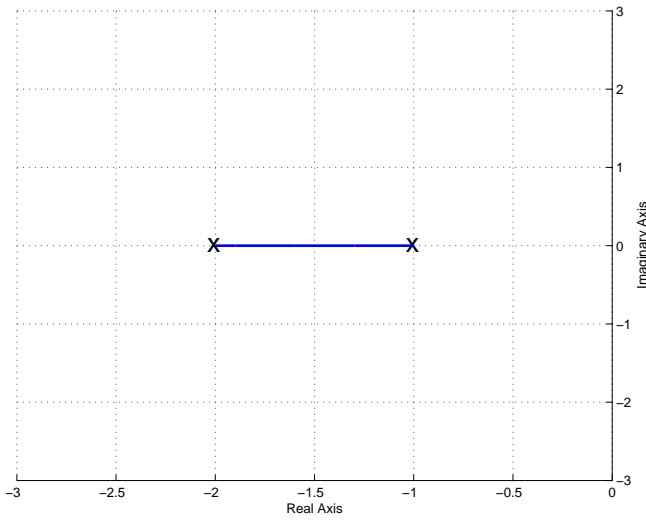
This is the starting point for all root locus construction. If, after identifying all segments of the real axis on the root locus, each pole is matched to a zero by a segment of the root locus, we are done. Note that there may be zeros at infinity. Figures 28, 29, and 30 display the real axis segments of the root locus for twelve different transfer functions, $G_1(s) - G_{12}(s)$.

At this point, let's examine these root locus plots to see if we are done. To determine if you are done, you need to be sure that every pole is matched to a zero and the root locus is symmetric about the real axis. It is acceptable for a zero (or a pole) to be at infinity, as long as you know where at infinity it is, such as along the real axis.

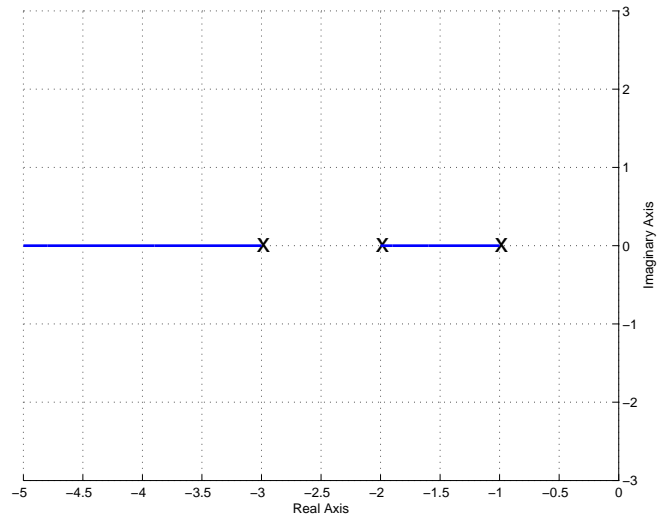
At this point, $G_4(s)$ and $G_6(s)$ are the only two completed root locus plots. In $G_4(s)$ the pole at -3 is paired with a zero at infinity (along the real axis). In $G_6(s)$ the pole at -1 is paired with the zero at -2, and the pole at -4 is paired with the zero at -3.

Before we go on to the next two rules, we need to see if we can do anything to complete any more of the root locus plots. In this case we can also complete the plots for $G_7(s)$, $G_9(s)$, and $G_{12}(s)$. How do we know this? In each one of these plots there are two zeros and two poles that are unpaired. We now need to pair them up. We cannot have any more parts of the root locus on the real axis, so all other segments must basically be off of the real axis. The basic thing to keep in mind here is that the root locus *must be symmetric about the real axis*. Figure 31 shows the completed root locus plots for these functions. Note that the segments we have added to pair the poles with the zeros are symmetric about the real axis. Three questions naturally arise here: (1) How do we know where the root locus leave the real axis? (2) At what angle do the new segments leave the real axis or poles? and (3) At what angle do they arrive at the zeros? While there are rules for computing these, we will not pursue them in this course.

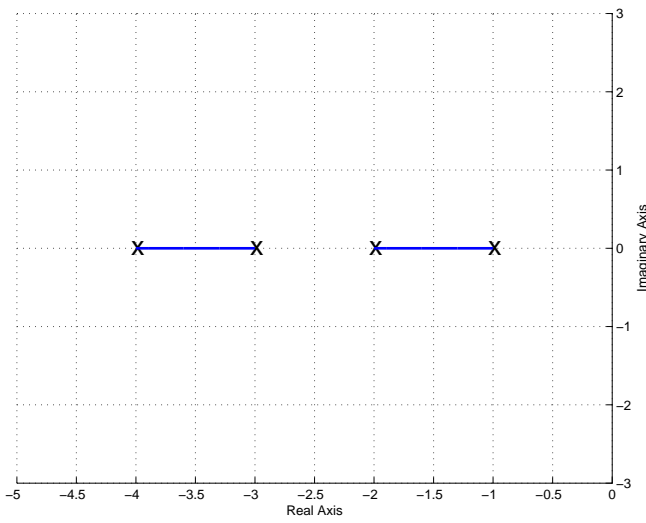
We cannot complete the remaining root locus plots because there are not enough "finite" zeros to pair with each pole. Hence these poles must pair with zeros at "infinity". However, since we are dealing with a two dimensional coordinate system, we need to figure out where (in terms of direction) this infinity is. This brings us to the next two rules, which are used as a pair.



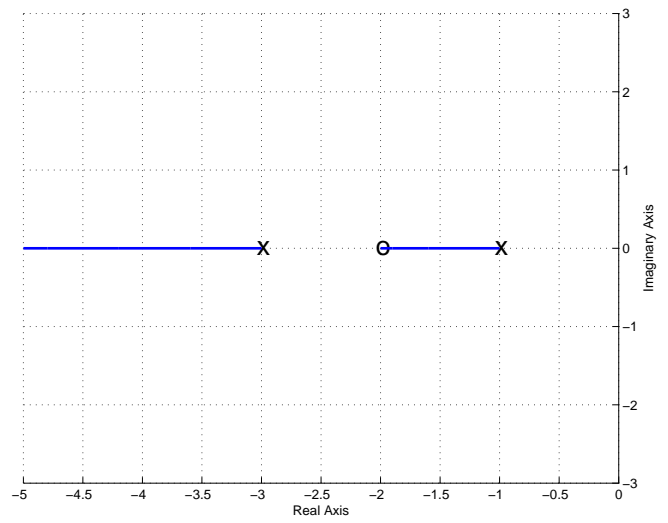
$$G_1(s) = \frac{1}{(s+1)(s+2)}$$



$$G_2(s) = \frac{1}{(s+1)(s+2)(s+3)}$$

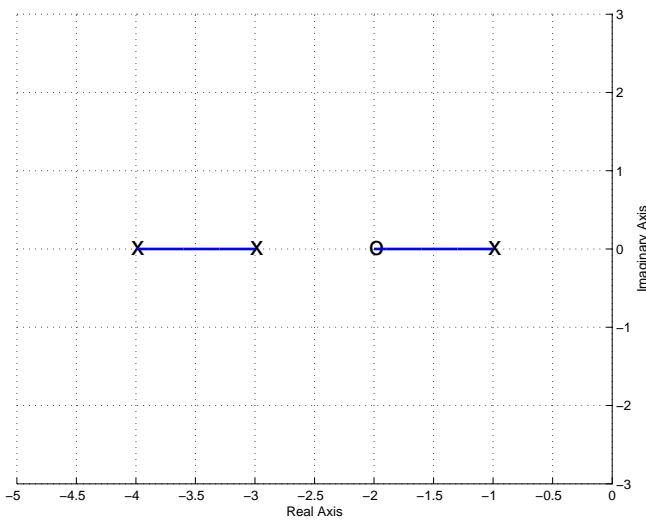


$$G_3(s) = \frac{1}{(s+1)(s+2)(s+3)(s+4)}$$

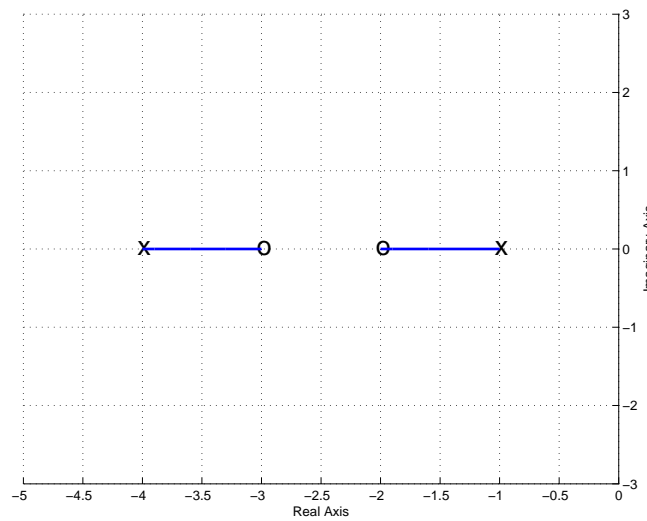


$$G_4(s) = \frac{(s+2)}{(s+1)(s+3)}$$

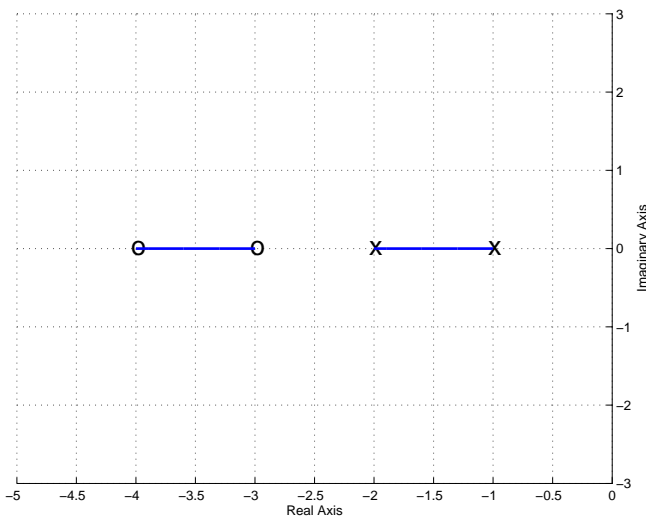
Figure 28: Root locus real-axis segments for $G_1(s) - G_4(s)$.



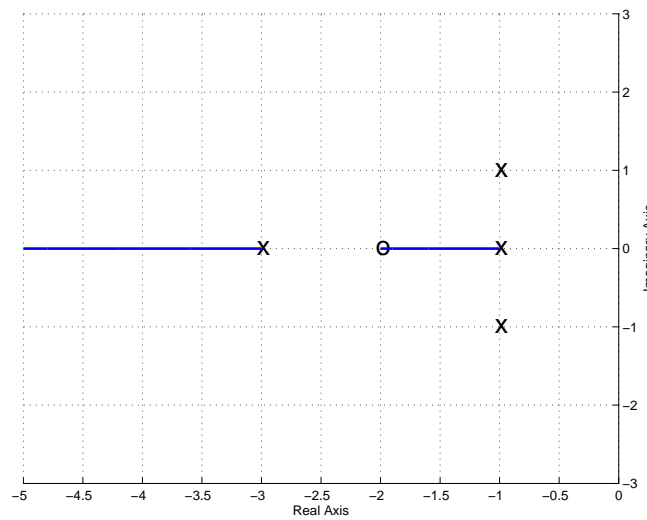
$$G_5(s) = \frac{(s+2)}{(s+1)(s+3)(s+4)}$$



$$G_6(s) = \frac{(s+2)(s+3)}{(s+1)(s+4)}$$

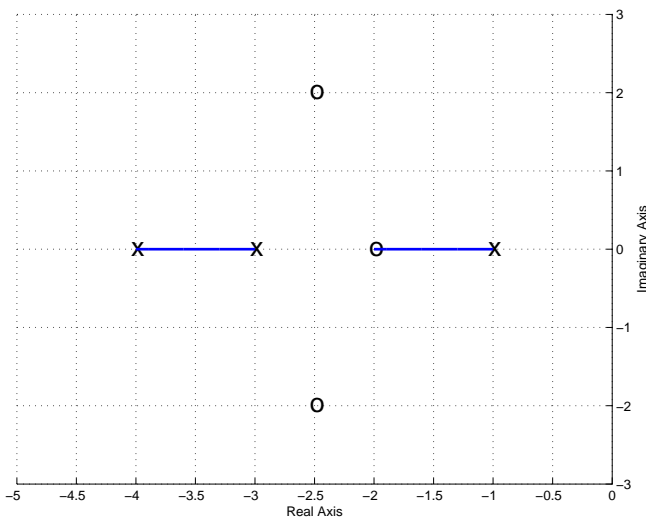


$$G_7(s) = \frac{(s+3)(s+4)}{(s+1)(s+2)}$$

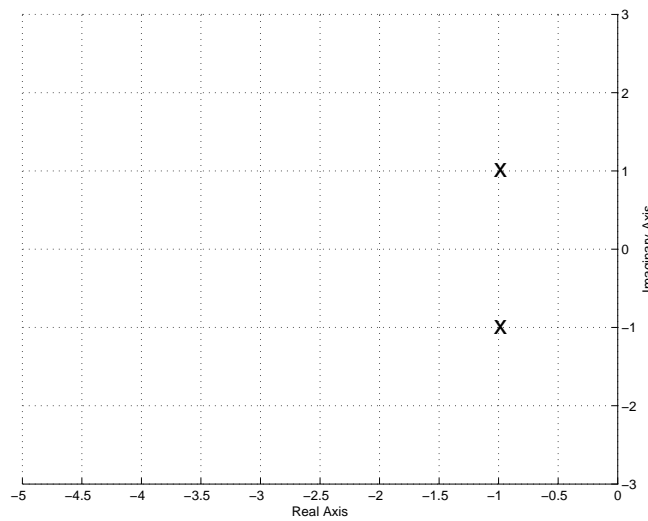


$$G_8(s) = \frac{(s+2)}{(s+1)(s+3)(s^2+2s+2)}$$

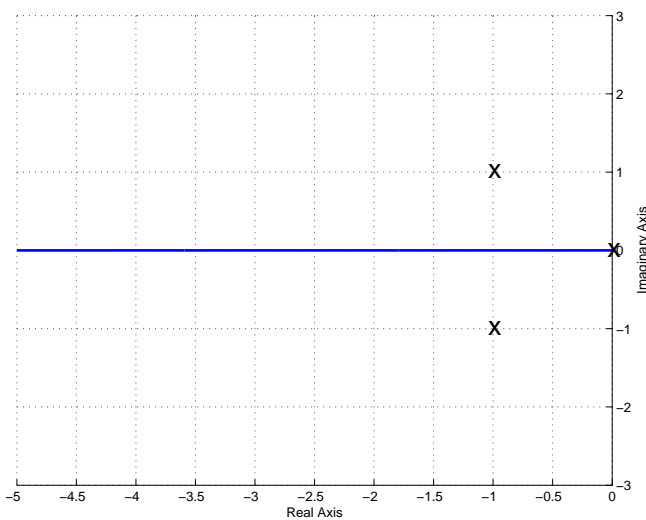
Figure 29: Root locus real-axis segments for $G_5(s) - G_8(s)$



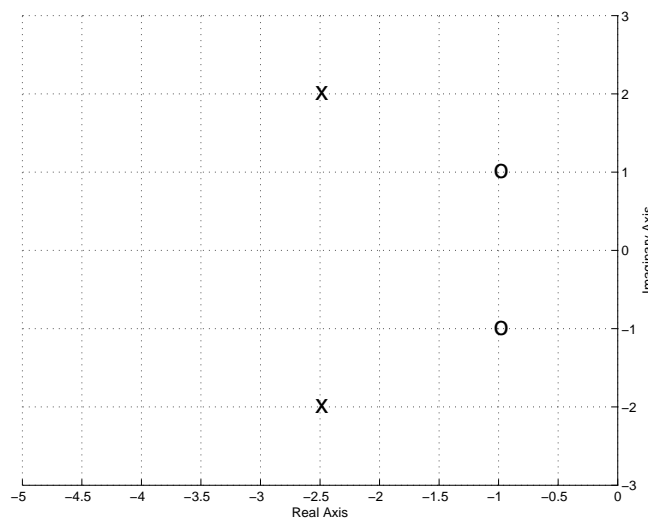
$$G_9(s) = \frac{(s+2)(s^2+5s+10.25)}{(s+1)(s+3)(s+4)}$$



$$G_{10}(s) = \frac{1}{s^2+2s+2}$$

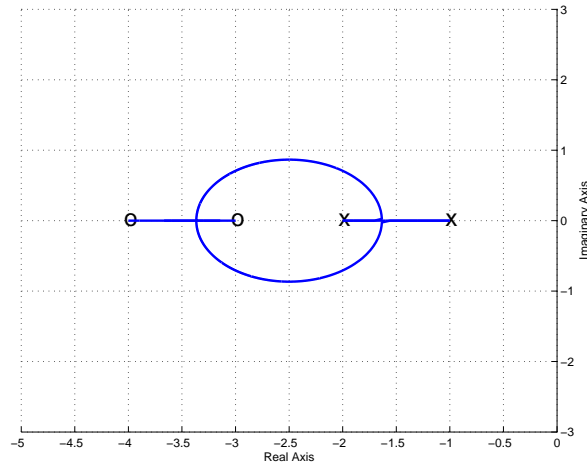


$$G_{11}(s) = \frac{1}{s(s^2+2s+2)}$$

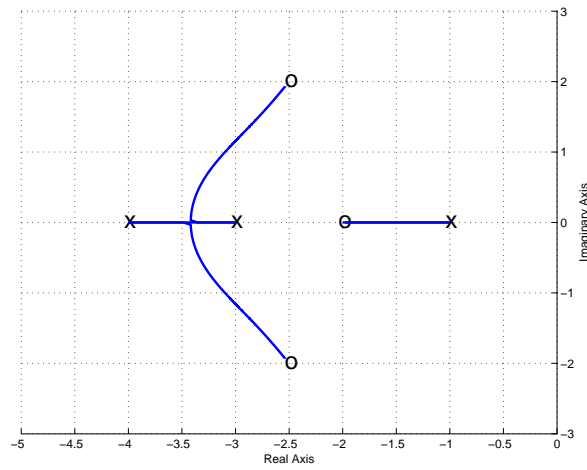


$$G_{12}(s) = \frac{s^2+2s+2}{s^2+5s+10}$$

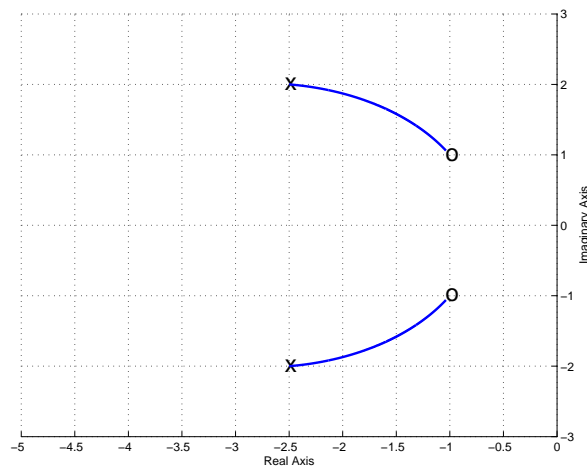
Figure 30: Root locus real-axis segments for $G_9(s) - G_{12}(s)$.



$$G_7(s) = \frac{(s+3)(s+4)}{(s+1)(s+2)}$$



$$G_9(s) = \frac{(s+2)(s^2+5s+10.25)}{(s+1)(s+3)(s+4)}$$



$$G_{12}(s) = \frac{s^2+2s+2}{s^2+5s+10.25}$$

Figure 31: Completed root locus plots for $G_7(s)$, $G_9(s)$, $G_{12}(s)$.

11.5 Asymptotic Angles and Centroid of the Asymptotes

In order to locate the zeros at infinity, we need two things: a starting point and a direction. The starting point is called the *centroid of the asymptotes* and the directions are the *asymptotes*. We determine them using the following rules

The starting point on the real axis from which the asymptotic lines radiate is given by

$$\sigma_c = \frac{\sum_i p_i - \sum_j z_j}{n - m}$$

where p_i is the i^{th} pole of $G(s)$, z_j is the j^{th} zero of $G(s)$, n is the number of poles of $G(s)$ and m is the number of zeros of $G(s)$.

As $k \rightarrow \infty$, the branches of the locus become asymptotic to straight lines with angles

$$\theta = \frac{180^\circ + i360^\circ}{n - m}, \quad i = 0, \pm 1, \pm 2, \dots$$

until all $(n - m)$ angles not differing by multiples of 360° are obtained. n is the number of poles of $G(s)$ and m is the number of zeros of $G(s)$.

Note that in order to determine how many angles to look for, we first utilize the fact that *the root locus must be symmetric about the real axis*, thus if $+45^\circ$ is one angle, we must also have -45° . Secondly, once we have found as many directions as unpaired poles we can stop since we then know where their pairs are. We will now go through the remaining $G_i(s)$ using these rules to complete the root locus plots.

$G_1(s)$ First we determine the centroid of the asymptotes, σ_c . There are two poles, one at -1 and one at -2, and there are no zeros. Hence $n = 2$ and $m = 0$. To determine σ_c ,

$$\sigma_c = \frac{[(-1) + (-2)]}{2 - 0} = -\frac{3}{2}$$

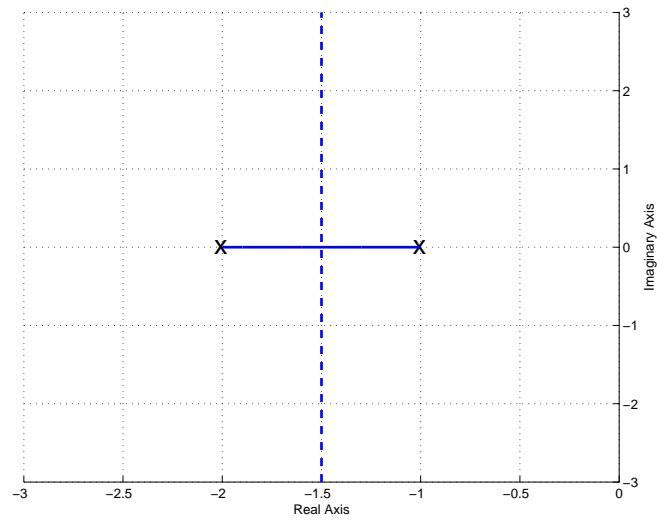
Hence the asymptotes radiate from this point. Next we compute the asymptote angles

$$\theta(i = 0) = \frac{180^\circ}{2 - 0} = 90^\circ$$

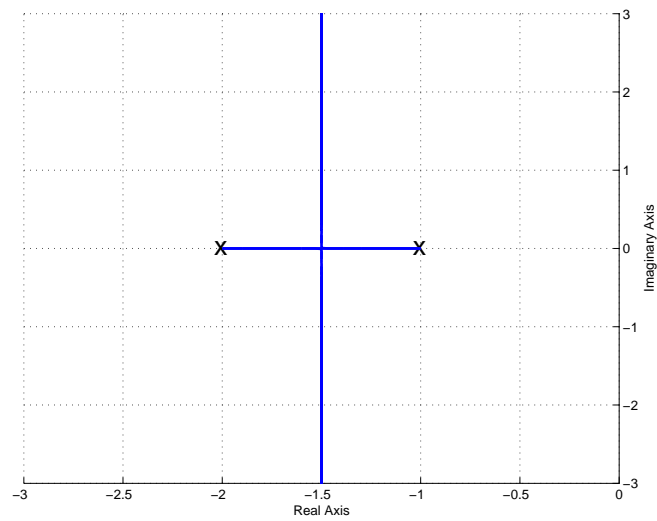
Since the root locus is symmetric, we know we must also have -90° . We are looking for zeros for two poles and we have two angles, so we are done. Figure 32 shows both the real axis segment and the asymptotes for $G_1(s)$ and then the completed root locus. Again, we don't know where the root locus leaves the real axis without more analysis, but you should know that it leaves the real axis and is symmetric.

$G_2(s)$ First we determine the centroid of the asymptotes, σ_c . There are three poles, one at -1, one at -2, and one at -3, and there are no zeros. Hence $n = 3$ and $m = 0$. To determine σ_c ,

$$\sigma_c = \frac{[(-1) + (-2) + (-3)]}{3 - 0} = -2$$



Real axis segment and the asymptotes



Completed root locus plot

Figure 32: Real axis segment with asymptotes and completed root locus plots for $G_1(s) = \frac{1}{(s+1)(s+2)}$. The centroid of the asymptotes is at $\sigma_c = -3/2$ and the angles of the asymptotes are ± 90 degrees.

Hence the asymptotes radiate from this point. Next we compute the asymptote angles

$$\begin{aligned}\theta(i = 0) &= \frac{180^\circ}{3 - 0} = 60^\circ \\ \theta(i = 1) &= \frac{180^\circ + 360^\circ}{3} = 180^\circ\end{aligned}$$

Since the root locus is symmetric, we know we must also have -60° . We are looking for zeros for three poles and we have three angles, so we are done. Figure 33 shows both the real axis segment and the asymptotes for $G_2(s)$ and then the completed root locus. Again, we don't know where the root locus leaves the real axis without more analysis, but you should know that it leaves the real axis and is symmetric.

$G_3(s)$ First we determine the centroid of the asymptotes, σ_c . There are four poles, one at -1, one at -2, one at -3, and one at -4, and there are no zeros. Hence $n = 4$ and $m = 0$. To determine σ_c ,

$$\sigma_c = \frac{[(-1) + (-2) + (-3) + (-4)]}{4 - 0} = -2.5$$

Hence the asymptotes radiate from this point. Next we compute the asymptote angles

$$\begin{aligned}\theta(i = 0) &= \frac{180^\circ}{4 - 0} = 45^\circ \\ \theta(i = 1) &= \frac{180^\circ + 360^\circ}{4} = 135^\circ\end{aligned}$$

Since the root locus is symmetric, we know we must also have -45° and -135° . We are looking for zeros for four poles and we have four angles, so we are done. Figure 34 shows both the real axis segment and the asymptotes for $G_3(s)$ and then the completed root locus. Again, we don't know where the root locus leaves the real axis without more analysis, but you should know that it leaves the real axis and is symmetric.

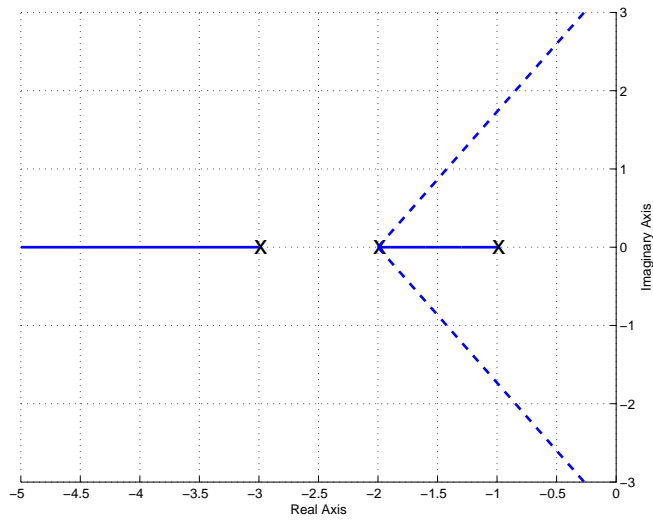
$G_5(s)$ First we determine the centroid of the asymptotes, σ_c . There are three poles, one at -1, one at -3, and one at -4, and one zero at -2. Hence $n = 3$ and $m = 1$. To determine σ_c ,

$$\sigma_c = \frac{[(-1) + (-3) + (-4)] - [(-2)]}{3 - 1} = -3$$

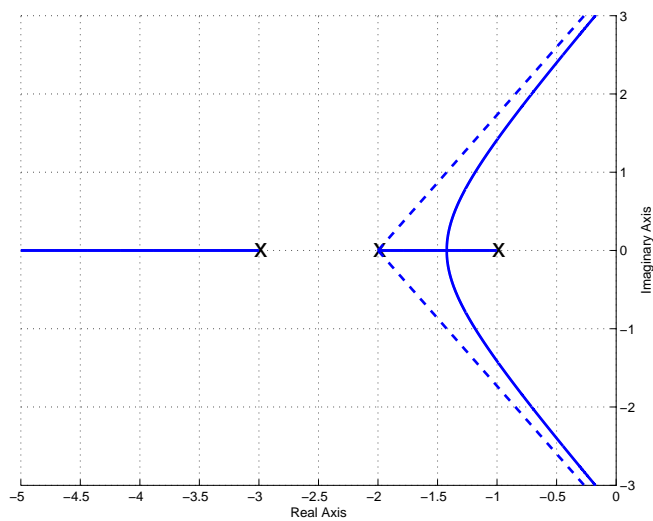
Hence the asymptotes radiate from this point. Next we compute the asymptote angles

$$\theta(i = 0) = \frac{180^\circ}{3 - 1} = 90^\circ$$

Since the root locus is symmetric, we know we must also have -90° . We are looking for zeros for two poles and we have two angles, so we are done. Figure 35 shows both the real axis segment and the asymptotes for $G_5(s)$ and then the completed root locus. Again, we don't know where the root locus leaves the real axis without more analysis, but you should know that it leaves the

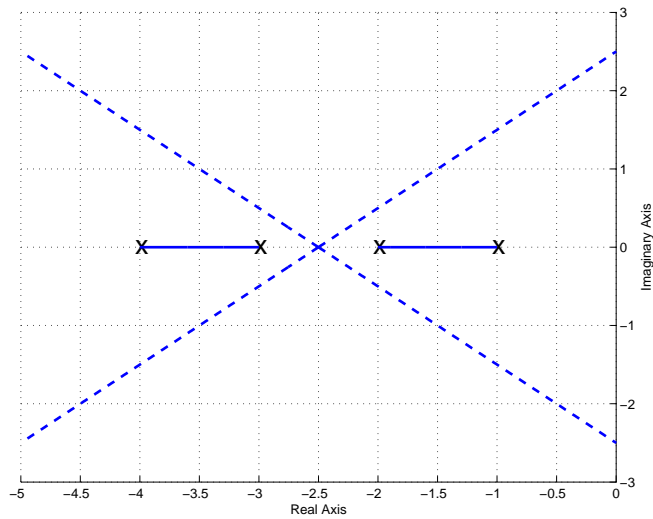


Real axis segment and the asymptotes

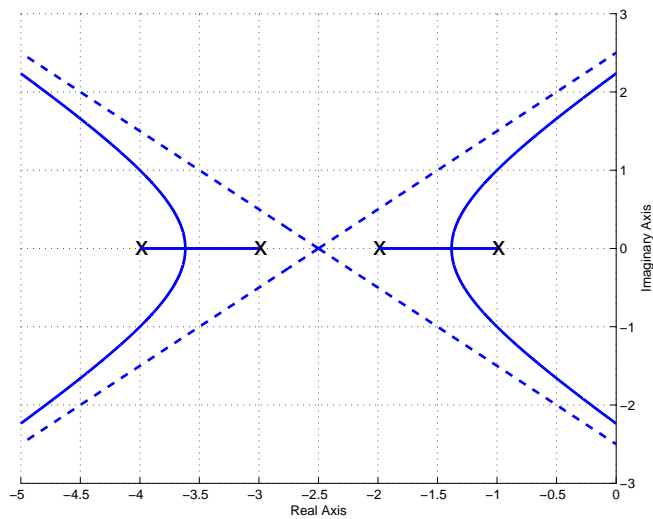


Completed root locus plot (with the asymptotes)

Figure 33: Real axis segment with asymptotes and completed root locus plots for $G_2(s) = \frac{1}{(s+1)(s+2)(s+3)}$. The centroid of the asymptotes is at $\sigma_c = -2$ and the angles of the asymptotes are 180 and ± 60 degrees.

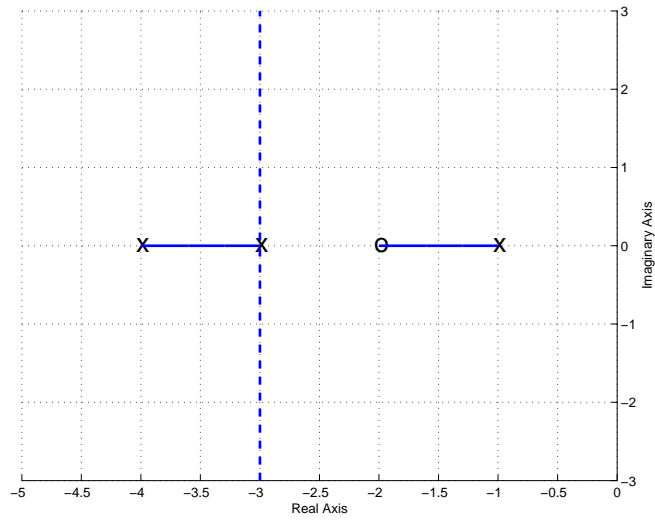


Real axis segment and the asymptotes

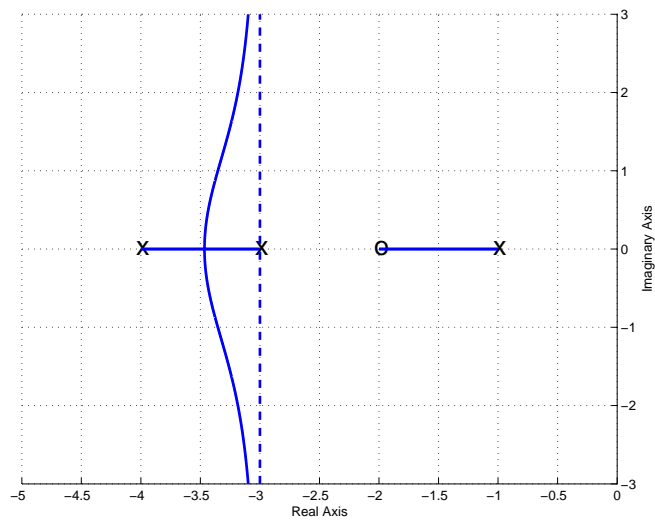


Completed root locus plot (with the asymptotes)

Figure 34: Real axis segment with asymptotes and completed root locus plots for $G_3(s) = \frac{1}{(s+1)(s+2)(s+3)(s+4)}$. The centroid of the asymptotes is at $\sigma_c = -2.5$ and the angles of the asymptotes are ± 45 and ± 135 degrees.



Real axis segment and the asymptotes



Completed root locus plot (with the asymptotes)

Figure 35: Real axis segment with asymptotes and completed root locus plots for $G_5(s) = \frac{(s+2)}{(s+1)(s+3)(s+4)}$. The centroid of the asymptotes is at $\sigma_c = -3$ and the angles of the asymptotes are ± 90 degrees.

real axis and is symmetric.

$G_8(s)$ First we determine the centroid of the asymptotes, σ_c . There are four poles, one at -1, one at -3, one at -1+j, one at -1-j, and one zero at -2. Hence $n = 4$ and $m = 1$. To determine σ_c ,

$$\sigma_c = \frac{[(-1) + (-3) + (-1 + j) + (-1 - j)] - [(-2)]}{4 - 1} = -1.3$$

Hence the asymptotes radiate from this point. Next we compute the asymptote angles

$$\begin{aligned}\theta(i = 0) &= \frac{180^\circ}{4 - 1} = 60^\circ \\ \theta(i = 1) &= \frac{180^\circ + 360^\circ}{4 - 1} = 180^\circ\end{aligned}$$

Since the root locus is symmetric, we know we must also have -60° . We are looking for zeros for three poles and we have three angles, so we are done. Figure 36 shows both the real axis segment and the asymptotes for $G_8(s)$ and then the completed root locus. Again, we don't know where the root locus leaves the real axis without more analysis, but you should know that it leaves the real axis and is symmetric.

$G_{10}(s)$ First we determine the centroid of the asymptotes, σ_c . There are two poles, one at -1+j, one at -1-j, and no zeros. Hence $n = 2$ and $m = 0$. To determine σ_c ,

$$\sigma_c = \frac{[(-1 + j) + (-1 - j)]}{2 - 0} = -1$$

Hence the asymptotes radiate from this point. Next we compute the asymptotes angles

$$\theta(i = 0) = \frac{180^\circ}{2 - 0} = 90^\circ$$

Since the root locus is symmetric, we know we must also have -90° . We are looking for zeros for two poles and we have two angles, so we are done. Figure 37 shows both the real axis segment and the asymptotes for $G_{10}(s)$ and then the completed root locus.

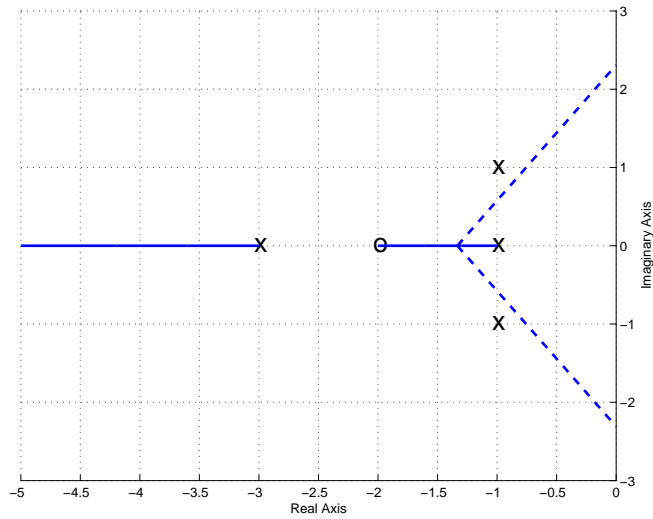
$G_{11}(s)$ First we determine the centroid of the asymptotes, σ_c . There are three poles, one at -1+j, one at -1-j, one at 0, and no zeros. Hence $n = 3$ and $m = 0$. To determine σ_c ,

$$\sigma_c = \frac{[(-1 + j) + (-1 - j) + (0)]}{3 - 0} = -\frac{2}{3}$$

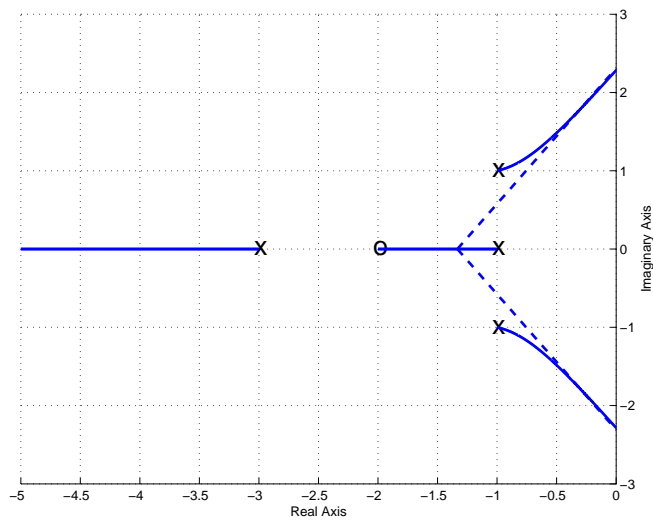
Hence the asymptotes radiate from this point. Next we compute the asymptote angles

$$\begin{aligned}\theta(i = 0) &= \frac{180^\circ}{3 - 0} = 60^\circ \\ \theta(i = 1) &= \frac{180^\circ + 360^\circ}{3 - 0} = 90^\circ\end{aligned}$$

Since the root locus is symmetric, we know we must also have -60° . We are looking for zeros for three poles and we have three angles, so we are done. Figure 38 shows both the real axis segment and the asymptotes for $G_{11}(s)$ and then the completed root locus.

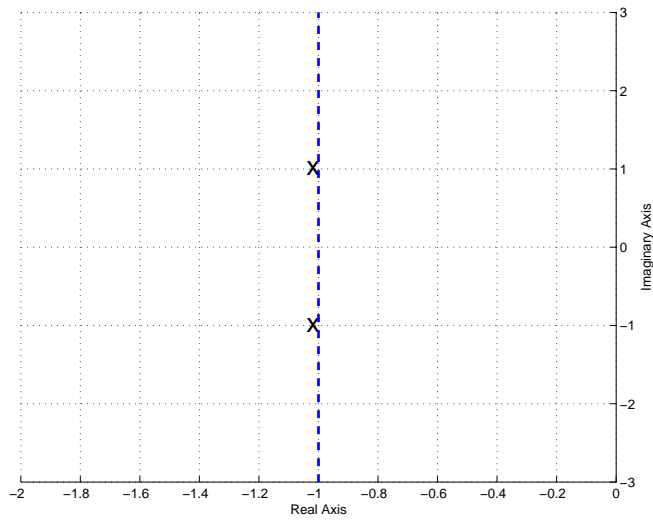


Real axis segment and the asymptotes

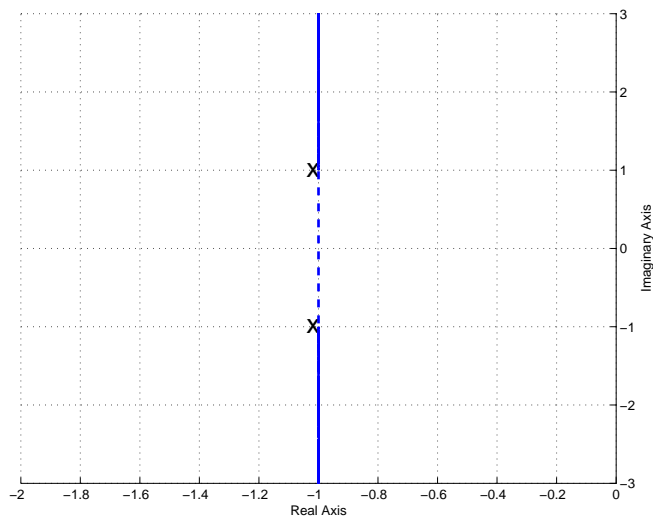


Completed root locus plot (with the asymptotes)

Figure 36: Real axis segment with asymptotes and completed root locus plots for $G_8(s) = \frac{(s+2)}{(s+1)(s+3)(s^2+2s+2)}$. The centroid of the asymptotes is at $\sigma_c = -1.3$ and the angles of the asymptotes are 180 and ± 60 degrees.

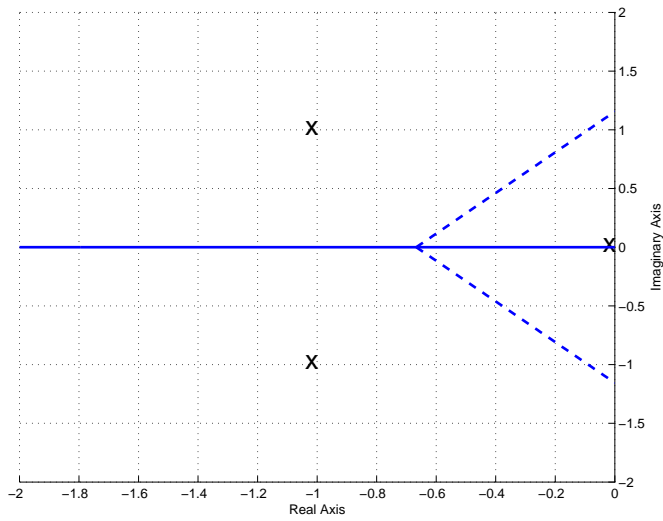


Real axis segment and the asymptotes

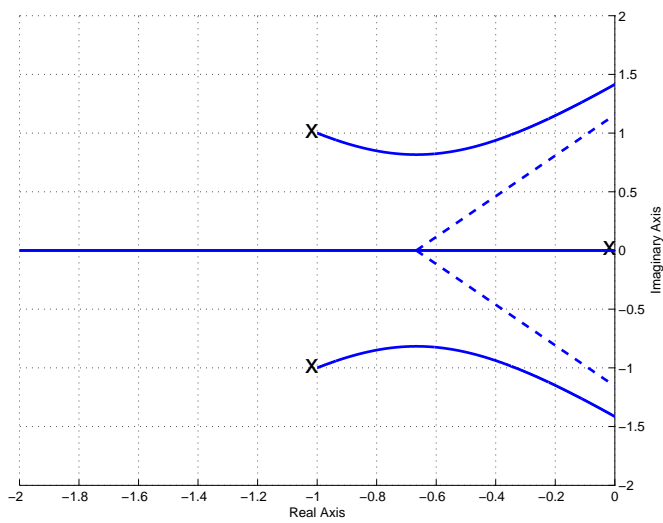


Completed root locus plot (with the asymptotes)

Figure 37: Real axis segment with asymptotes and completed root locus plots for $G_{10}(s) = \frac{1}{s^2+2s+2}$. The centroid of the asymptotes is at $\sigma_c = -1$ and the angles of the asymptotes are ± 90 degrees.



Real axis segment and the asymptotes



Completed root locus plot (with the asymptotes)

Figure 38: Real axis segment with asymptotes and completed root locus plots for $G_{11}(s) = \frac{1}{s(s^2+2s+2)}$. The centroid of the asymptotes is at $\sigma_c = -2/3$ and the angles of the asymptotes are 90 and ± 60 degrees.

11.6 Common Industrial Controller Types

In addition to the types of controllers we have already discussed using the model matching methods, we will now introduce types of controllers that are very commonly used in industrial applications. For each of these types of controllers, we assume the controller is the ratio of the signal into the plant $U(s)$ and the error signal $E(s)$, hence

$$G_c(s) = \frac{U(s)}{E(s)}$$

or

$$U(s) = G_c(s)E(s)$$

Proportional (P) Control In a proportional controller, the output of the controller is directly proportional to the error signal, or $u(t) = k_p e(t)$. Here k_p is the proportional gain, and $G_c(s) = k_p$. For this type of controller, the output tends to oscillate a lot.

Integral (I) Control In an integral controller, the output of the controller is proportional to the integral of the error signal, or

$$u(t) = k_i \int^t e(\lambda) d\lambda$$

This means

$$G_c(s) = \frac{k_i}{s}$$

Unless the plant has a zero at the origin, this type of controller makes the system a type one system and, *if the system is stable*, the steady state error for a step input will be zero. For plants with complex conjugate poles near the $j\omega$ axis, the system will likely become unstable except for small values of gain k_i . This type of controller also tends to be slow, which can be a problem.

Derivative (D) Control In a derivative controller, the output of the controller is proportional to the derivative of the error signal, or

$$u(t) = k_d \frac{d}{dt} e(t)$$

This means

$$G_c(s) = k_d s$$

A derivative controller anticipates the direction of the error signal since it is proportional to the slope, and tends to be very fast. However, it is very difficult to compute derivatives for real systems, so this type of controller is very seldom used alone. In order to filter out noise and smooth the signal before the derivative is computed this controller is often used in combination with a lowpass filter.

Proportional+Integral (PI) Controllers For this type of controller, we combine both the proportional and integral types of controllers, so

$$u(t) = k_p e(t) + k_i \int^t e(\lambda) d\lambda$$

or

$$G_c(s) = k_p + \frac{k_i}{s}$$

This can be written in the form

$$G_c(s) = \frac{k(s+z)}{s}$$

Hence a PI controller always has a pole at the origin, while both the location of the zero $-z$ and gain k can be varied to meet design requirements.

Proportional+Derivative (PD) Controllers For this type of controller, we combine both the proportional and derivative types of controllers, so

$$u(t) = k_p e(t) + k_d \frac{d}{dt} e(t)$$

or

$$G_c(s) = k_p + k_d s$$

This can be written in the form

$$G_c(s) = k(s+z)$$

The location of the zero $-z$ and gain k can be varied to meet design requirements.

Proportional+Integral+Derivative (PID) Controllers For this type of controller, we combine both the proportional, integral, and derivative types of controllers, so

$$u(t) = k_p e(t) + k_i \int^t e(\lambda) d\lambda + k_d \frac{d}{dt} e(t)$$

or

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

This can be written in the form

$$G_c(s) = \frac{k(s+z_1)(s+z_2)}{s}$$

Hence a PID controller always has a pole at the origin, while both the location of the zeros $-z_1$ and $-z_2$ and gain k can be varied to meet design requirements. Note that we can either have

real zeros or complex conjugate zeros with this type of controller.

Lead Controllers For this type of controller has the transfer function

$$G_c(s) = \frac{k(s+z)}{(s+p)}$$

where p is larger than z^3 . Note that for this type of controller we have the freedom to choose the gain k , the zero $-z$, and the pole $-p$ to meet design requirements.

Special Note. For all of these controllers, we want both the poles and zeros in the left half plane. This means that z is a positive number (so the zero is at $-z$) and p is a positive number (so the pole is at $-p$.)

11.7 Controller and Design Constraint Examples

Example 3. Assume we have the plant

$$G_p(s) = \frac{2}{s^2 + 2s + 10}$$

and we want to design a controller so that

- the system is stable
- the steady state error for a unit step is less than 0.2
- the settling time is less than 2 seconds

We will assume we are going to use the basic control structure shown in Figure 27 at the beginning of the chapter. In order for the system to remain stable the poles of the closed loop system must be in the left half plane. In order to meet the settling time requirement, we need

$$T_s = \frac{4}{\sigma} \leq 2$$

or $\sigma \geq 2$. Hence the real parts of the poles must be less than -2 (or have magnitude σ greater than 2). On our root locus plots we will denote this location by a vertical dotted line, which we will need to be sure to be to the left of in order to meet the settling time constraints. Since we are using a prefilter, we will be able to meet the steady state error requirements for any stable system. However, the prefilter is outside the feedback loop, so we would like to avoid this if possible. We will next examine each of the possible controller types to see if it meets the requirements.

Proportional (P) Controller. For this controller, $G(s) = \frac{2}{s^2+2s+10}$. First we compute the centroid of the asymptotes and the asymptote angles:

$$\begin{aligned} \theta &= \frac{180^\circ}{2} = 90^\circ \\ \sigma_c &= \frac{[(-1-3j) + (-1+3j)]}{2} = -1 \end{aligned}$$

³If p is less than z then this is a *lag* controller, which tends to slow the system down. We will discuss lag controllers in a subsequent chapter.

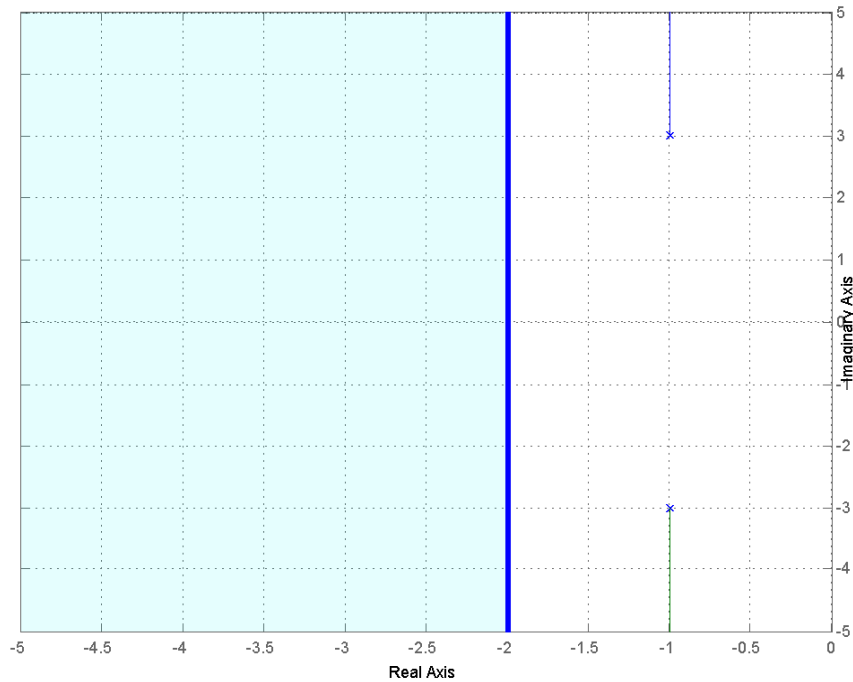


Figure 39: Root locus plot for $G_p(s) = \frac{2}{s^2+2s+10}$ with the proportional (P) controller. To match the settling time all poles must be to the left of -2.

The root locus for the system with the proportional controller is shown in Figure 39, along with the heavy line at -2, which divides the plane into the pole locations that will meet the settling time requirement and those that will not. As this figure shows, the system will remain stable for all values of k^4 , but will never meet the settling time requirement since the poles are always to the right of -2. To meet the steady state error requirement, we find

$$\begin{aligned}
 K_p &= \frac{k}{5} \\
 e_{ss} &= \frac{1}{1+K_p} \leq \frac{1}{5} \\
 1+K_p &= 1 + \frac{k}{5} \geq 5 \\
 k &\geq 20
 \end{aligned}$$

Hence to meet the steady state error constraint we need $k \geq 20$. As an alternative, we could use a prefilter with gain $G_{pf} = \frac{k+5}{k}$ for zero steady state error. However the prefilter is outside the feedback loop.

⁴Here $k = k_p$.

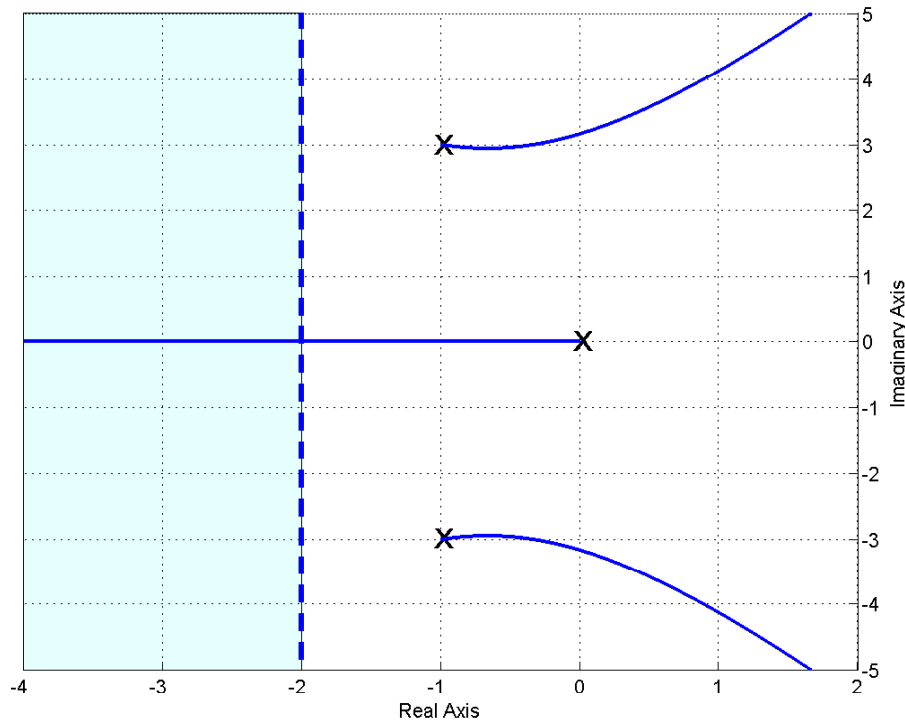


Figure 40: Root locus plot for $G_p(s) = \frac{2}{s^2+2s+10}$ with the integral (I) controller. To match the settling time all poles must be to the left of -2.

Integral (I) Controller. For this controller, $G(s) = \frac{2}{s(s^2+2s+10)}$. First we compute the centroid of the asymptotes and the asymptote angles:

$$\theta = \frac{180^\circ}{3} = 60^\circ$$

$$\sigma_c = \frac{[(-1 - 3j) + (-1 + 3j) + (0)]}{3} = -\frac{2}{3}$$

The root locus for the system with the integral controller is shown in Figure 40, along with a line at -2, which divides the plane into the pole locations that will meet the settling time requirement and those that will not. As this figure shows, the system will not remain stable for all values of k^5 , at some value of k some of the poles will be in the right half plane. If k is chosen so the system is stable, the steady state error will be zero. The system will never meet the settling time requirement since two of the poles are always to the right of -2. (In order to meet the settling time requirement *all* of the poles must be to the left of -2.)

Proportional+Integral (PI) Controller. For this controller, $G(s) = \frac{2(s+z)}{s(s^2+2s+10)}$. First we compute the centroid of the asymptotes and the asymptote angles:

$$\theta = \frac{180^\circ}{3-2} = 90^\circ$$

⁵Here $k = k_i$.

$$\sigma_c = \frac{[(-1 - 3j) + (-1 + 3j) + (0)] - [-z]}{2} = \frac{z - 2}{2}, z \geq 0.$$

Here the centroid of the asymptotes depends on where we locate the zero of the system. Figure 41 displays the root locus plot for $z = 3$ and $z = 0.5$, as well as the constraint region. We always want $z \geq 0$ and we need *all* poles to the left of -2 to meet the settling time constraint. The best we can do for the complex conjugate poles is to set z to a very small number. If $z = 0$ it cancels the pole at the origin and the root locus looks like that of the proportional controller. For the zero at -0.5 the system will remain stable for all k and will have a steady state error of zero. For the pole at -3 the system will remain stable only for a range of k , but it will have a steady state error of zero for a unit step so long as the system is stable.

Proportional+Derivative (PD) Controller For this controller, $G(s) = \frac{2(s+z)}{(s^2+2s+10)}$. For this controller, we do not need to compute the centroid, just think a bit. Once we put the zero on the real axis, the entire real axis to the left of this point will be on the root locus. The two poles will come down to the real axis to pair with the zeros. If we choose the zero to be in the left half plane, the system will be stable for all values of k . To be absolutely clear that we can choose poles that meet the settling time constraint, we should be sure $z \geq 2$, so *both* poles must be to the left of -2. It may be possible without this constraint, but this will guarantee it. Figure 42 shows the root locus for this system with $z = 1$, $z = 2$, and $z = 3$. Not that for $z = 1$, for small values of k the system does not meet the settling time constraint, then for a range of k it does meet the settling time constraint, and then once k becomes larger than some critical value one of the closed loop poles leaves the critical region (to the left of -2) on its way to the zero and the system no longer meets the settling time criteria. For both $z = 2$ and $z = 4$, the system does not meet the settling time criteria until k is larger than some critical value, then for all k larger than this critical value the system meets the settling time constraint. To meet the steady state error constraint we need

$$e_{ss} = \frac{1}{1 + K_p} \leq \frac{1}{5}$$

$$1 + K_p = 1 + \frac{kz}{5} \geq 5$$

or

$$kz \geq 20$$

If we cannot meet this condition, we can use a prefilter with gain $G_{pf} = \frac{kz+5}{kz}$ for a steady state error, but the prefilter is outside the feedback loop.

Proportional+Integral+Derivative (PID) Controller. For this controller, $G(s) = \frac{2(s+z_1)(s+z_2)}{s(s^2+2s+10)}$. For this controller, we do not need to compute the centroid, just think a bit. First of all, both zeros are in the left half plane the system will be stable for all k , and the steady state error will also be 0. We then have two cases to consider: (1) both zeros are real, and (2) the zeros are complex conjugates.

Real Zeros If we put both zeros on the real axis, and make sure both of them are to the left of -2, then we can be sure that for some value of k all poles will be to the left of -2. Figure 43 displays

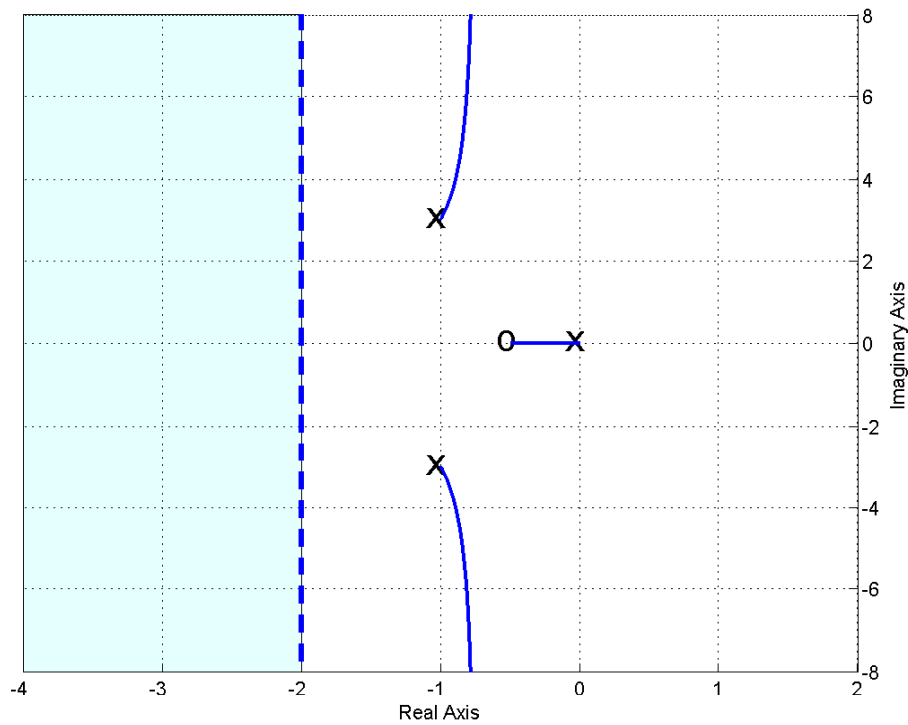
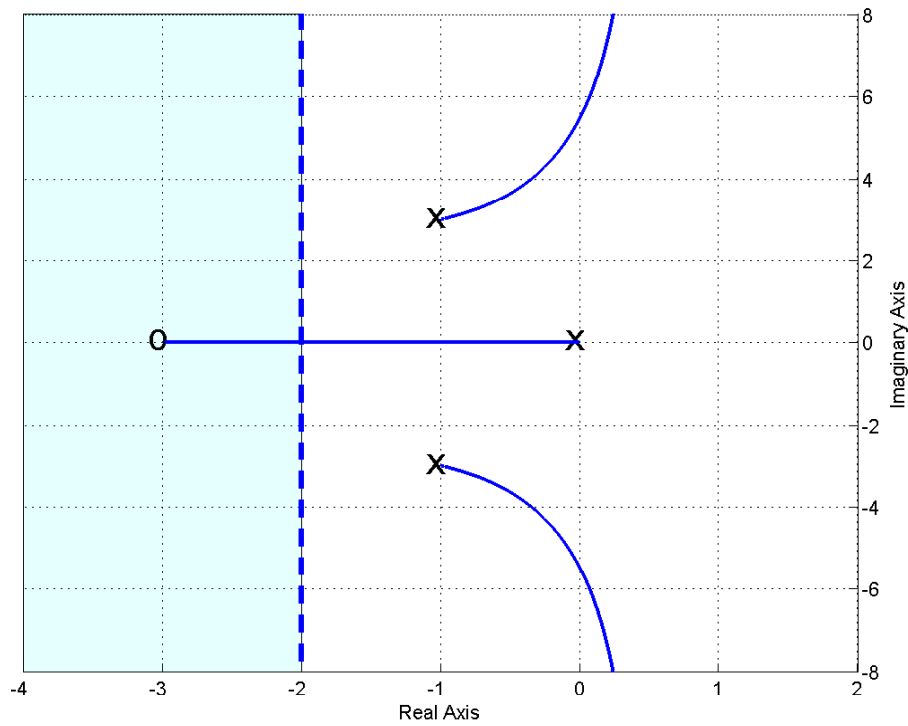


Figure 41: Root locus plots for $G_p(s) = \frac{2}{s^2+2s+10}$ with the proportional+integral (PI) controllers with the controller zero at -3 (top) and -0.5 (bottom). To match the settling time *all* poles must be to the left of -2 .

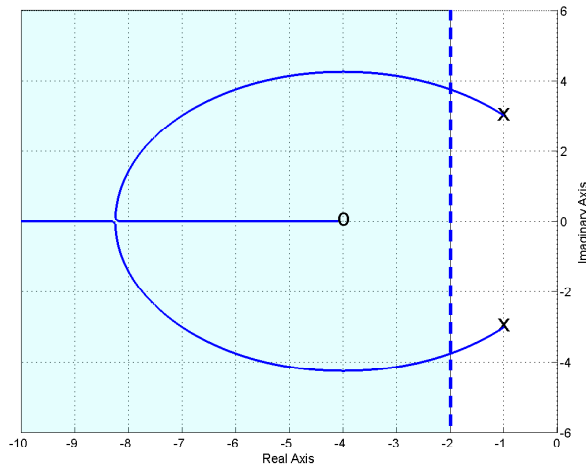
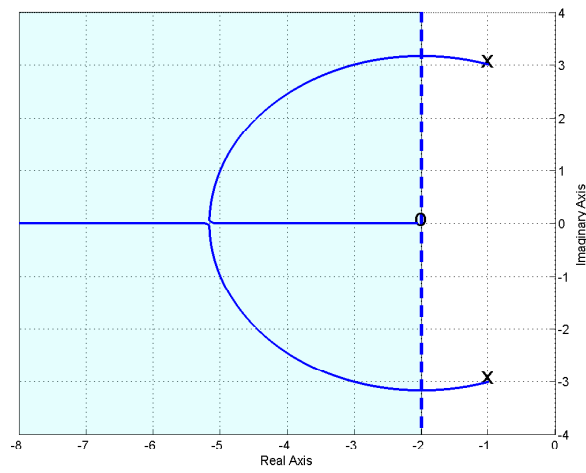
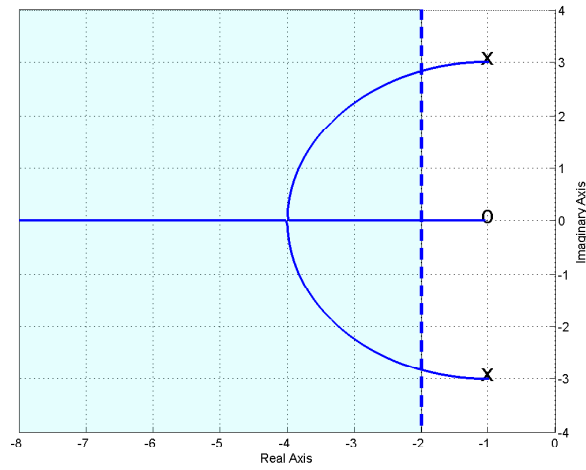


Figure 42: Root locus plots for $G_p(s) = \frac{2}{s^2+2s+10}$ with the proportional+derivative (PD) controllers with the controller zero at -1 (top), -2 (middle), and -4 (bottom). To match the settling time *all* poles must be to the left of -2.

three different examples of PID controllers with real zeros. For the choice of zero locations in the top panel, the settling time condition will never be met. For the choice of zeros in the middle and bottom panels, the settling time condition will be met if k is sufficiently large.

Complex Conjugate Zeros If the zeros are complex conjugates, we need to be sure the real part of the zero is to the left of -4, then we can be sure there is some value for k so that *all* of the poles will be to the left of -2. Figure 44 displays three different examples of PID controllers with complex conjugate zeros. For the choice of real zeros shown in the top two panels, the settling time condition will never be met. For the choice of zero locations in the bottom panel, the settling time conditions will be met for k sufficiently large.

Lead Controller. For this controller, $G(s) = \frac{2(s+z)}{(s+p)(s^2+2s+10)}$. First we compute the centroid of the asymptotes and the asymptote angles:

$$\begin{aligned}\theta &= \frac{180^\circ}{2} = 90^\circ \\ \sigma_c &= \frac{[(-1-3j) + (-1+3j) + (-p)] - [(-z)]}{2} = \frac{-2-p+z}{2} \\ &= \frac{-2-l}{2}\end{aligned}$$

where $l = p - z$, the difference between the pole and the zero of the controller. For positive l the system will be stable for all values of k . In order to meet the settling time requirement, we should have $l > 2$. Figure 45 displays the root locus for $l = 2$ for various combinations of z and p . Figure 46 displays the root locus for $l = 4$ (top), $l = 6$ (middle), and $l = 18$ (bottom) for z fixed at $z = 3$. Clearly as l increases, the asymptotes move to the left. This is what we usually use a lead controller for, to pull the poles of the closed loop system to the left. Finally, Figure 47 displays a common phenomena when using a lead controller. Here we have $l = 6$ and the asymptotes are at -4. However, as k increases the complex conjugate poles move to the left (good), however, the pole on the real axis moves to the right (bad). It may not be possible with this configuration to find a value (or range of values) of k so the system meets the settling time requirements.

In order to meet the steady state error requirements we need

$$\begin{aligned}e_{ss} &= \frac{1}{1+K_p} \\ 1+K_p &= 1 + \frac{kz}{5p} \leq \frac{1}{5}\end{aligned}$$

or

$$\frac{kz}{p} \geq 20$$

If we cannot meet this condition, we can use a prefilter if we need to. A prefilter with $G_{pf} = \frac{kz+5p}{kz}$ will produce a steady state error of zero, but it is outside the feedback loop.

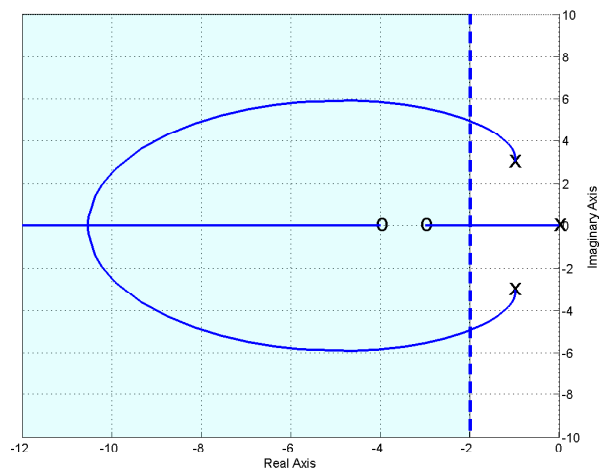
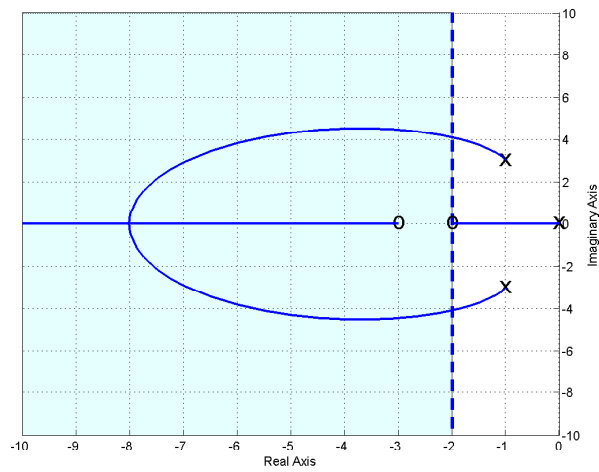
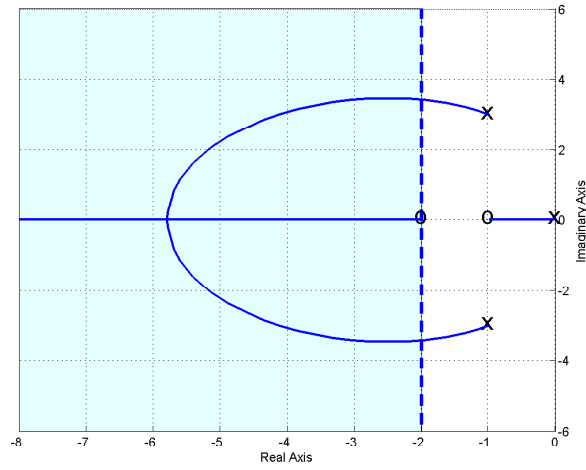


Figure 43: Root locus plots for $G_p(s) = \frac{2}{s^2+2s+10}$ with proportional+integral+derivative (PID) controllers with real zeros. To match the settling time *all* poles must be to the left of -2.

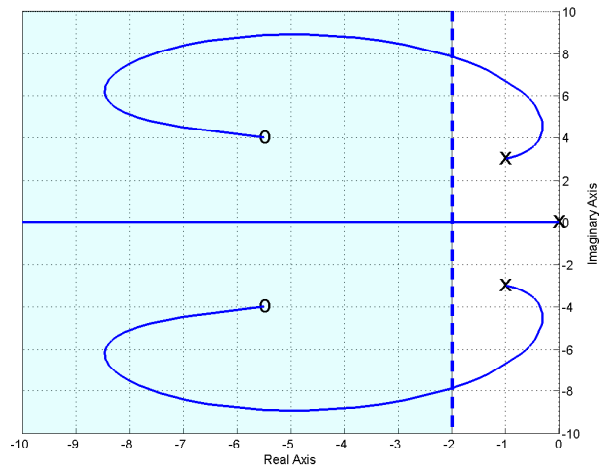
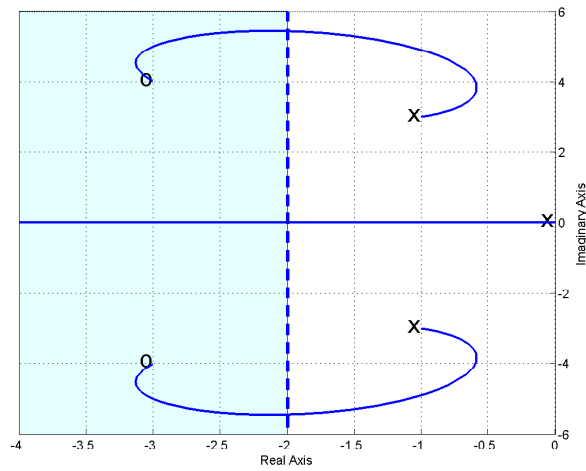
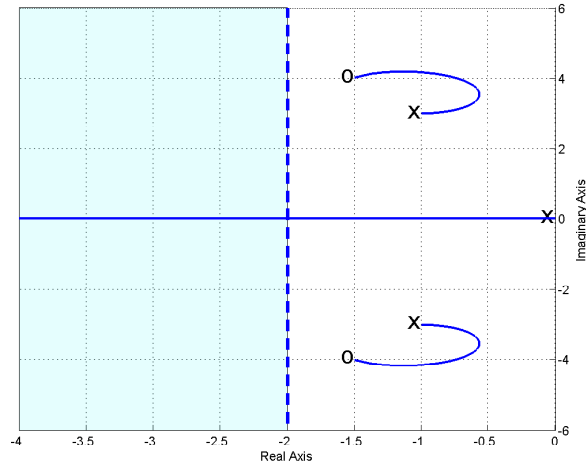


Figure 44: Root locus plots for $G_p(s) = \frac{2}{s^2+2s+10}$ with proportional+integral+derivative (PID) controllers with complex conjugate zeros. To match the settling time *all* poles must be to the left of -2.

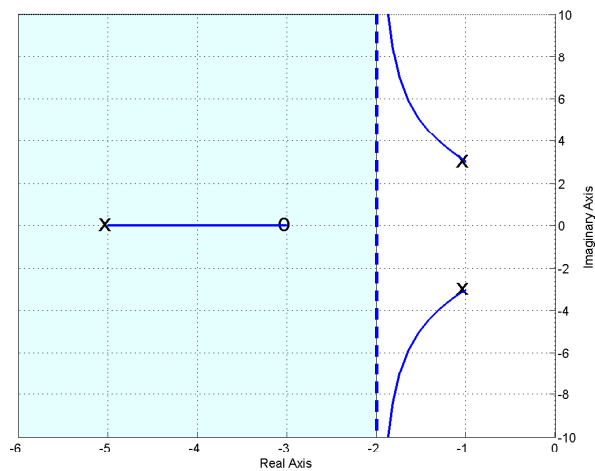
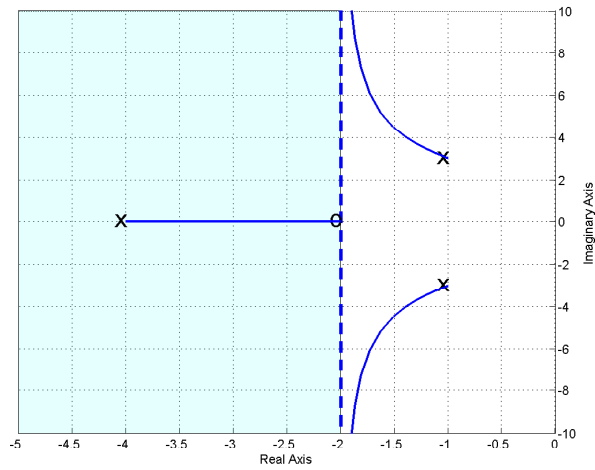
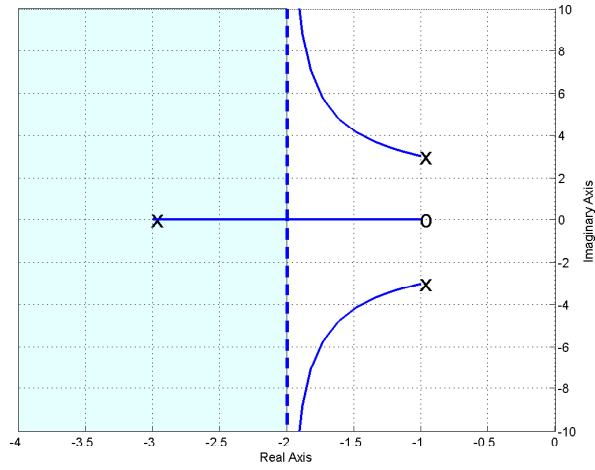


Figure 45: Root locus plots for $G_p(s) = \frac{2}{s^2+2s+10}$ with lead controllers for $l = 2$ and asymptote $\sigma_c = -2$. To match the settling time *all* poles must be to the left of -2 .

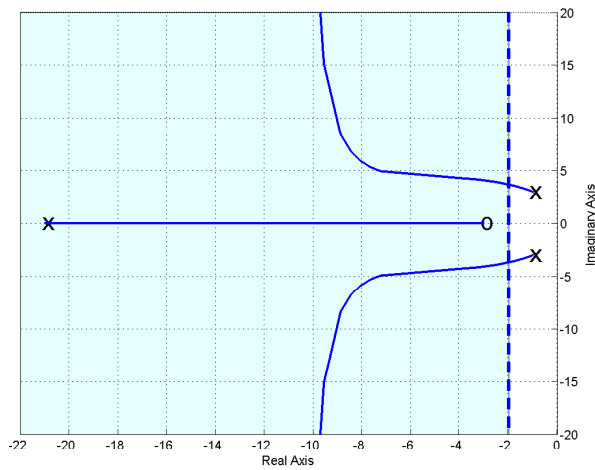
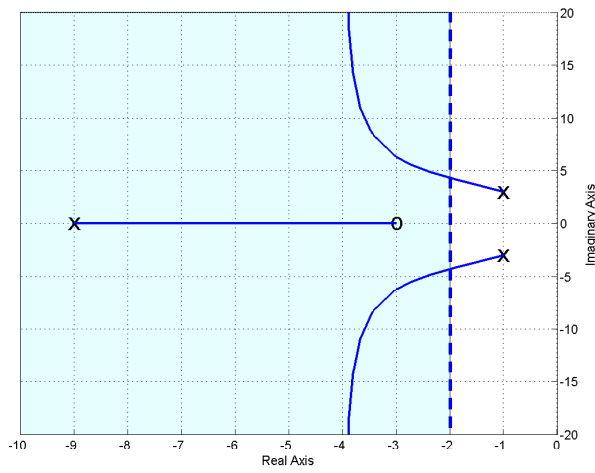
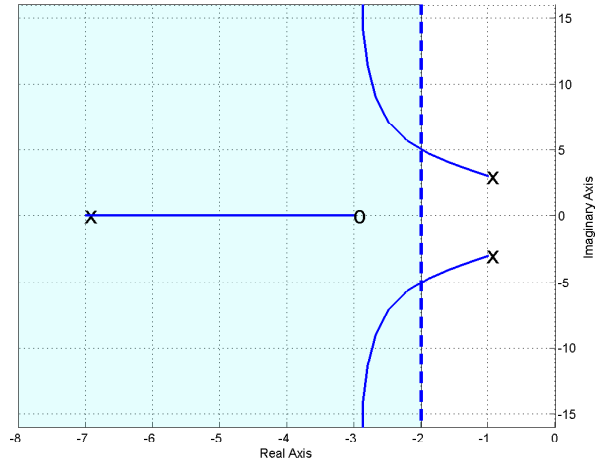


Figure 46: Root locus plots for $G_p(s) = \frac{2}{s^2+2s+10}$ with lead controllers for $l = 4$ and asymptote $\sigma_c = -3$ (top), $l = 6$ and asymptote $\sigma_c = -4$ (middle), and $l = 18$ and asymptote $\sigma_c = -10$ (bottom). To match the settling time *all* poles must be to the left of -2 .

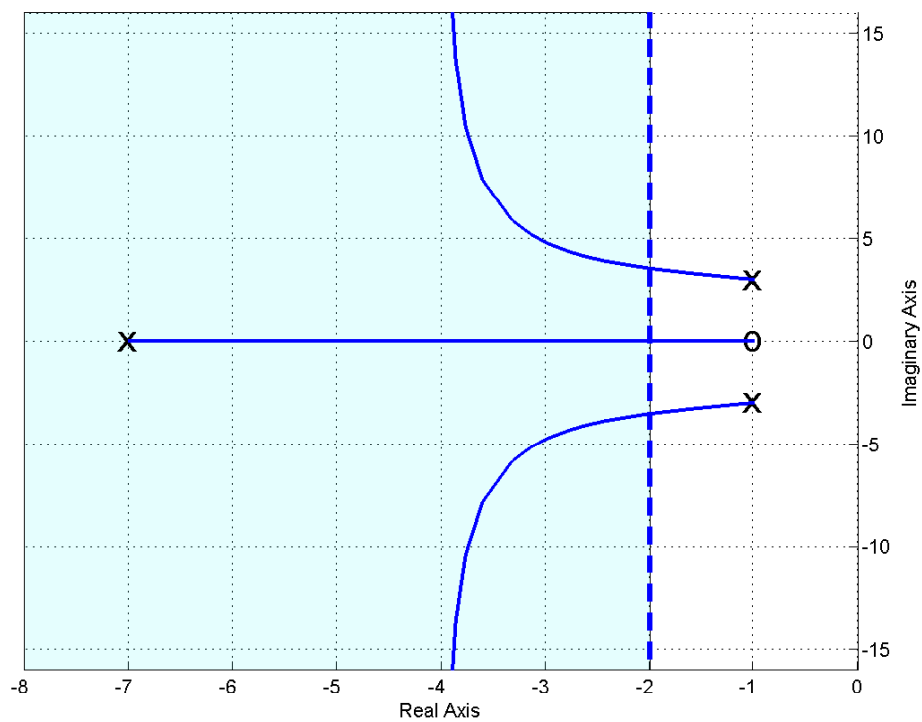


Figure 47: Root locus plot for $G_p(s) = \frac{2}{s^2+2s+10}$ with the lead controller for $l = 6$ and asymptote $\sigma_c = -4$. As the complex conjugate poles move to the left, the pole on the real axis moves to the right. It may not be possible to find a value (or range of values) for k to meet the settling time requirements. To match the settling time *all* poles must be to the left of -2 .

Example 4. Assume we have the plant

$$G_p(s) = \frac{2}{s^2 + 3s + 2}$$

and we want to design a controller so that

- the system is stable
- the steady state error is less than 0.2 (for a unit step)
- the settling time is less than 1.5 seconds
- the percent overshoot is less than 20%

We will assume we are going to use the basic control structure shown in Figure 27 at the beginning of the chapter. In order for the system to remain stable the poles of the closed loop system must be in the left half plane. In order to meet the settling time requirement, we need

$$T_s = \frac{4}{\sigma} \leq 1.5$$

or $\sigma \geq 2.67$. Hence the real parts of the poles must be less than -2.67 (or have magnitude σ greater than 2.67). On our root locus plots we will denote this location by a vertical dotted line, which we will need to be sure to be to the left of in order to meet the settling time constraints. In order to meet the percent overshoot constraint, we need the poles within a wedge of 62.8° from the negative real axis. On our root locus plot we will again denote this wedge by a dashed line. To meet both settling time and percent overshoot constraints we must have part of the root locus in the region of overlap. In addition, we must remember that, in particular, *the percent overshoot region is an approximation based on an ideal second order system*. Since we are using a prefilter, we will be able to meet the steady state error requirements for any stable system. We will next examine each of the possible controller types to see if it meets the requirements. This example will not be done in the detail of the last example, since I am really getting sick of writing this.

Proportional (P) Controller. For this controller, $G(s) = \frac{2}{s^2+3s+2}$. The root locus for the system with the proportional controller is shown in Figure 48. As this figure shows, the system will remain stable for all values of k^6 . It will never meet the settling time requirement since the poles are always to the right of -2.67. It will meet the percent overshoot requirements until k reaches a critical value, then for all values of k larger than this critical value the percent overshoot requirement will not be met. The steady state error requirement can be met if $k \geq 4$, but for this value of k the system may not meet any other requirements.

Integral (I) Controller. For this controller, $G(s) = \frac{2}{s(s^2+3s+2)}$. The root locus for the system with the integral controller is shown in Figure 49. As this figure shows, the system will not remain stable for all values of k^7 . For some value of k one set of poles crosses the $j\omega$ axis and the system will be unstable for all values of k larger than this. This system will never meet the

⁶Here $k = k_p$.

⁷Here $k = k_i$.

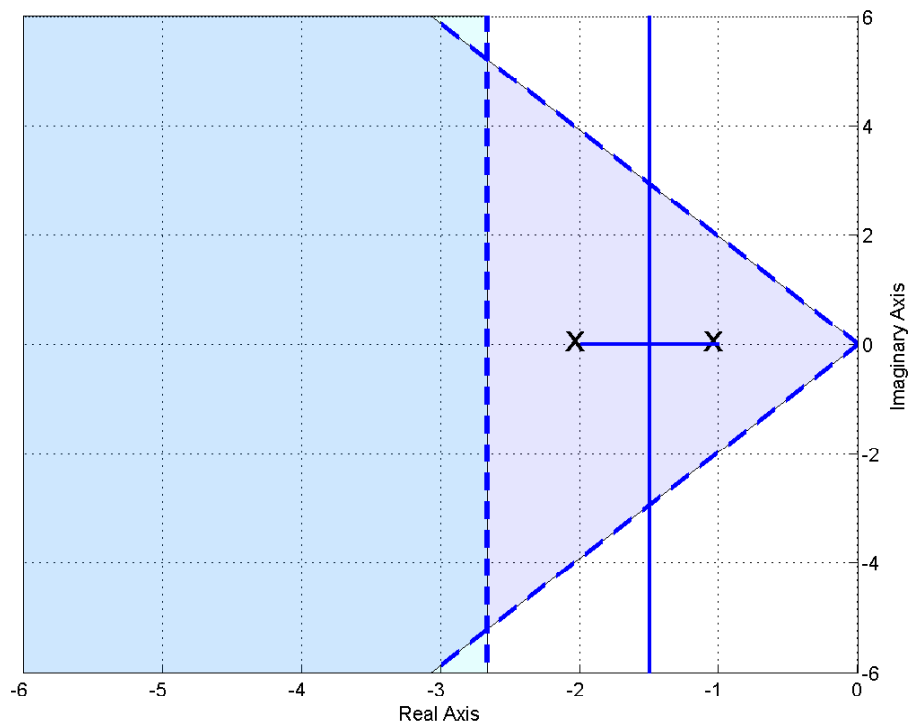


Figure 48: Root locus plot for $G_p(s) = \frac{2}{s^2+3s+2}$ with the proportional (P) controller. This system will never meet the settling time criteria, but it will meet the percent overshoot criteria as long as k is not too large.

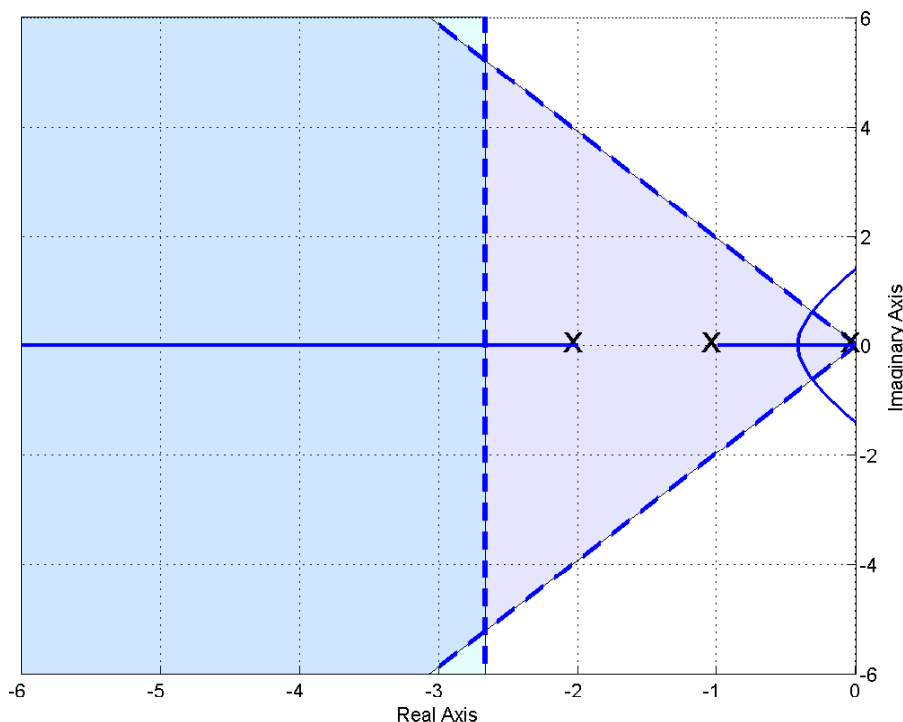


Figure 49: Root locus plot for $G_p(s) = \frac{2}{s^2+3s+2}$ with the integral (I) controller. This system will never meet the settling time criteria, but it will meet the percent overshoot criteria as long as k is not too large. This system will become unstable for k sufficiently large.

settling time requirement since two of the three poles are always to the right of -2.67. It will meet the percent overshoot requirements until k reaches a critical value, then for all values of k larger than this critical value the percent overshoot requirement will not be met. If this system is stable, the steady state error will be zero, since it is a type one system.

Proportional+Integral (PI) Controller. For this controller, $G(s) = \frac{2(s+z)}{s(s^2+3s+2)}$. The root locus for the system with the three PI controllers is shown in Figure 50. This figure shows three different values of the zero location in this controller: $z = 0.5$ (top panel), $z = 1.5$ (middle panel), and $z = 4$ (bottom panel). As this figure shows, none of the configurations (choices of z) will meet the settling time criteria. All of the configurations will meet the percent overshoot criteria if the value of k is not too large. The third configuration will become unstable for sufficiently large k . If the value of k is chosen so the system is stable, the steady state error criteria will be met since this is a type one system.

Proportional+Derivative (PD) Controller. For this controller, $G(s) = \frac{2(s+z)}{s^2+3s+2}$. The root locus for the system with three PD controller is shown in Figure 51. This figure shows three different values of the zero location in this controller: $z = 0.5$ (top panel), $z = 1.5$ (middle panel), and $z = 3$ (bottom panel). As this figure shows, only the last configuration (choices of z) will meet the settling time criteria for a sufficiently large value of k . All of the configurations

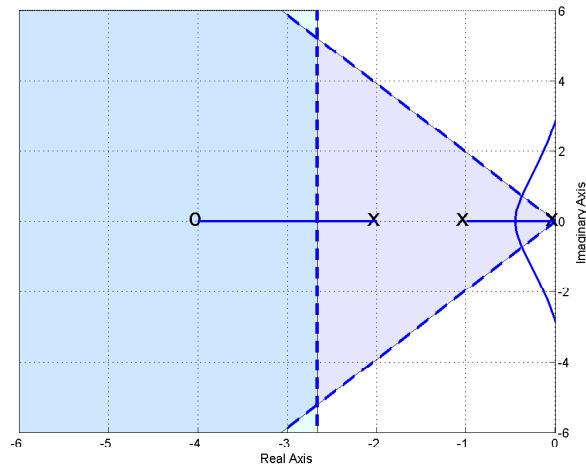
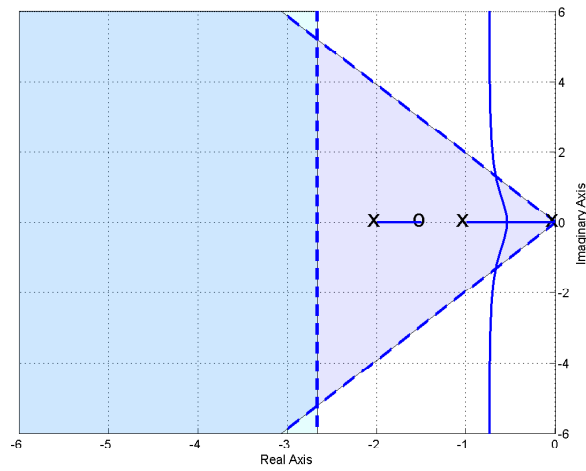
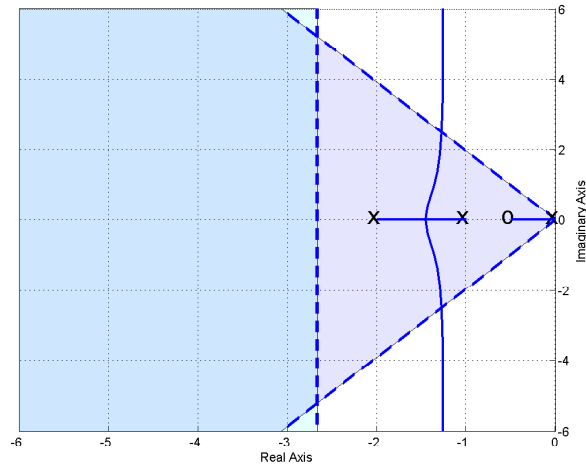


Figure 50: Root locus plot for $G_p(s) = \frac{2}{s^2+3s+2}$ with three proportional+integral (PI) controllers. None of these systems will ever meet the settling time criteria. All three will meet the percent overshoot criteria if k is small enough. The third system will become unstable for k sufficiently large.

will meet the percent overshoot criteria for all values of k . In order to meet the steady state error requirement, we need $kz \geq 4$.

Proportional+Integral+Derivative (PID) Controller. For this controller, $G(s) = \frac{2(s+z_1)(s+z_2)}{s(s^2+3s+2)}$. For this controller, we have two cases to consider: (1) both zeros are real, and (2) the zeros are complex conjugates.

Real Zeros Figure 52 and 53 displays different examples of PID controllers with real zeros. The first two systems in Figure 53 show poor choices for the locations of the controller zeros. The results in the second panel of Figure 53 shows that for some configurations, it is possible for the system to become unstable if k is chosen incorrectly.

Complex Conjugate Zeros Figure 54 and 55 displays different examples of PID controllers with complex conjugate zeros. The first two systems in Figure 54 show poor choices for the locations of the controller zeros. The bottom panel shows an acceptable choice for meeting both requirements, since for a sufficiently large k all poles will be in the acceptable region. The results in the second panel of Figure 55 shows that for some configurations, it is possible for the system to become unstable if k is chosen incorrectly.

Lead Controller. For this controller, $G(s) = \frac{2(s+z)}{(s+p)(s^2+3s+2)}$. For this case, it is worthwhile to examine the location of the centroid of the asymptotes to determine how far apart we need the pole and the zero to be.

$$\begin{aligned}\theta &= \frac{180}{2} = 90 \\ \sigma_c &= \frac{[(-1) + (-2) + (-p)] - [(-z)]}{2} = \frac{-3 - p + z}{2} \\ &= \frac{-3 - l}{2}\end{aligned}$$

where $l = p - z$, the difference between the pole and the zero of the controller. In order to meet the settling time constraint, we will need

$$\sigma_c = \frac{-3 - l}{2} \leq -2.67$$

or

$$l \geq 2.34$$

The root locus plots for the system with various lead controllers are shown in Figures 56 and 57. Figure 56 shows root locus plots for $l = 3$ and three different choices of pole and zero locations. Although each one of these systems has the same value of $l = 3$, so one set of asymptotes meets the settling criteria, unless care is taken the system will not meet the settling time criteria. *The first two choices in this figure are clearly stupid choices for the location of the controller pole and the controller zero!* In this figure, only the third choice will produce a system that meets all of the criteria for k less than a critical value. If k becomes too large, the percent overshoot criteria will not be met by any of these systems. Figure 57 shows two more root locus plots for $l = 6$ and $l = 9$, keeping $z = 3$. These plots demonstrate again how the lead controller bends the root locus to the left. In order to meet the steady state error requirement, we need $\frac{kz}{p} \geq 6$.

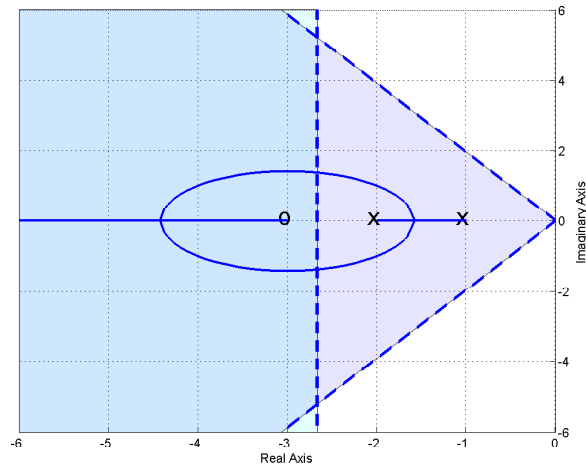
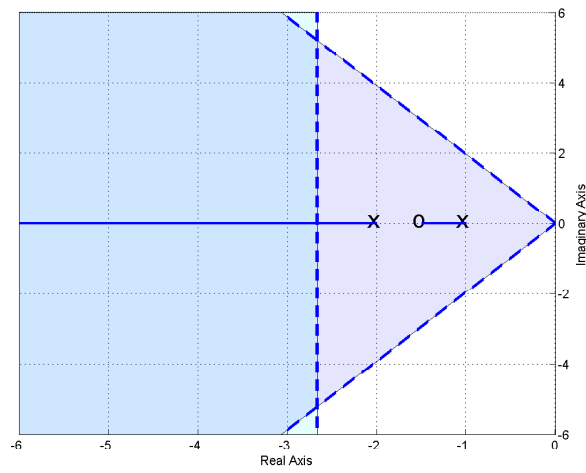
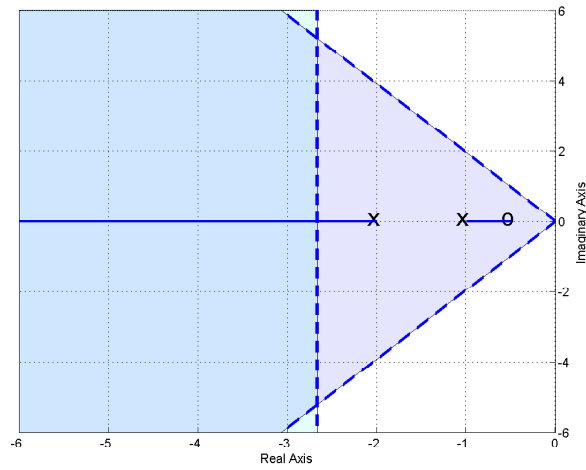


Figure 51: Root locus plots for $G_p(s) = \frac{2}{s^2+3s+2}$ with three proportional+derivative (PD) controllers. Only the third system meets the settling time criteria (for k large enough). All three will meet the percent overshoot criteria.

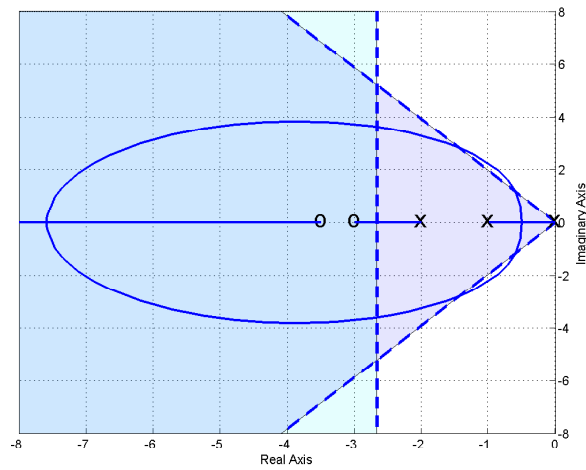
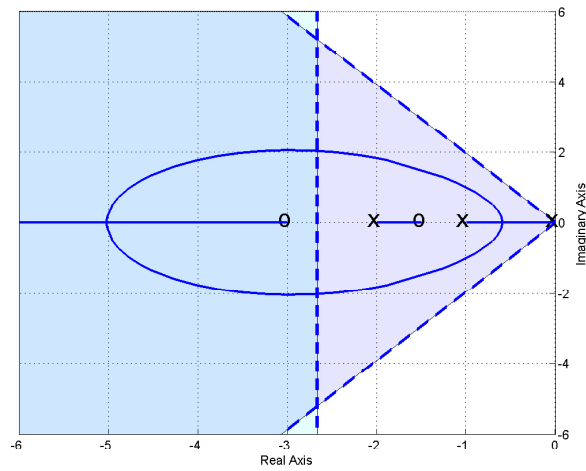
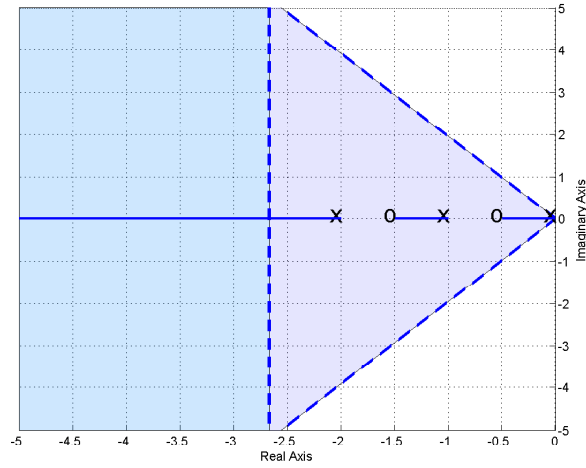


Figure 52: Root locus plots for $G_p(s) = \frac{2}{s^2+3s+2}$ with proportional+integral+derivative (PID) controllers with real. The systems in the top two panels demonstrate bad choices for the location of the controller zeros.

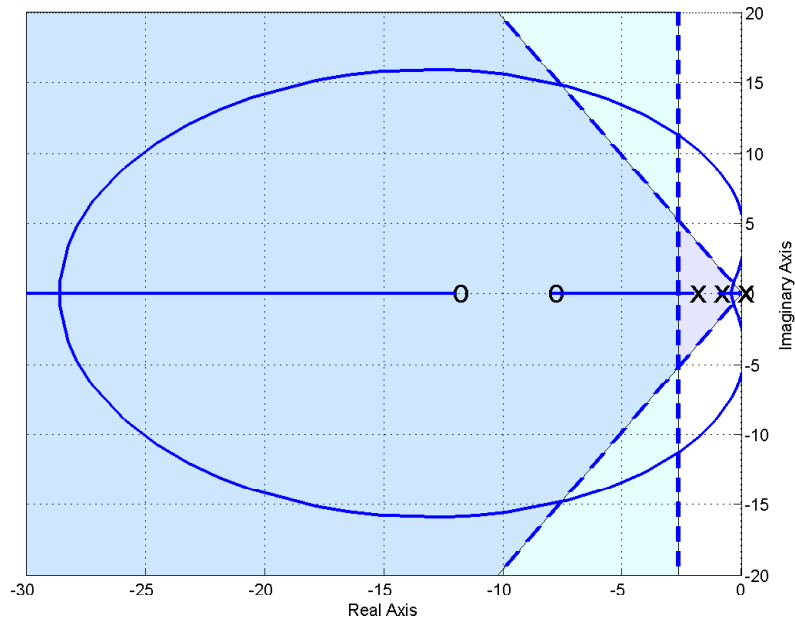
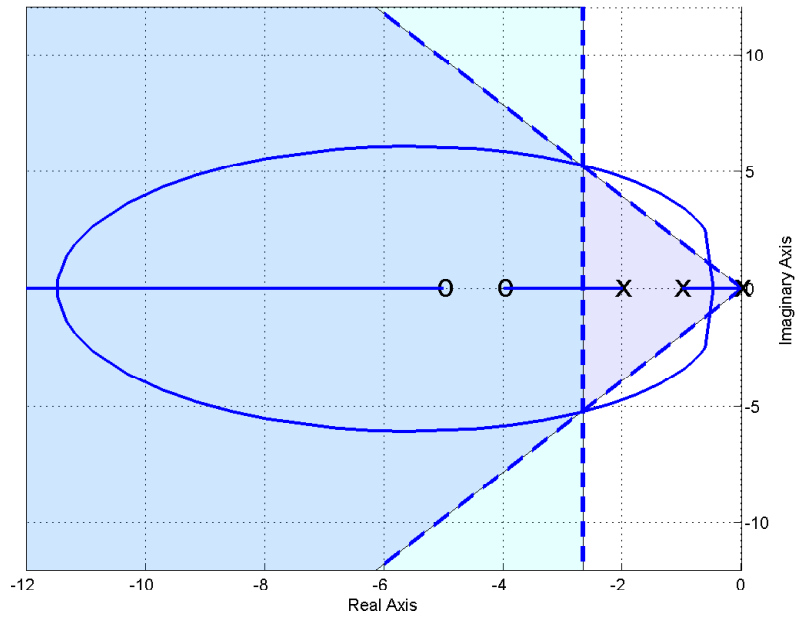


Figure 53: Root locus plots for $G_p(s) = \frac{2}{s^2 + 3s + 2}$ with proportional+integral+derivative (PID) controllers with real zeros.

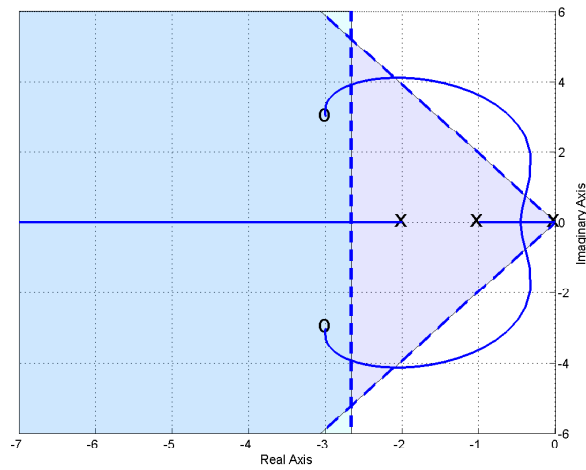
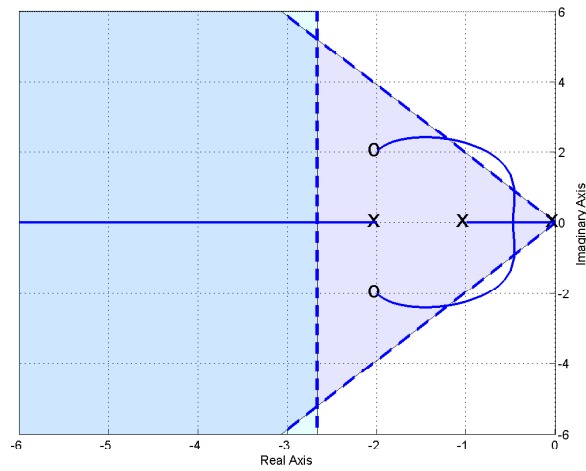
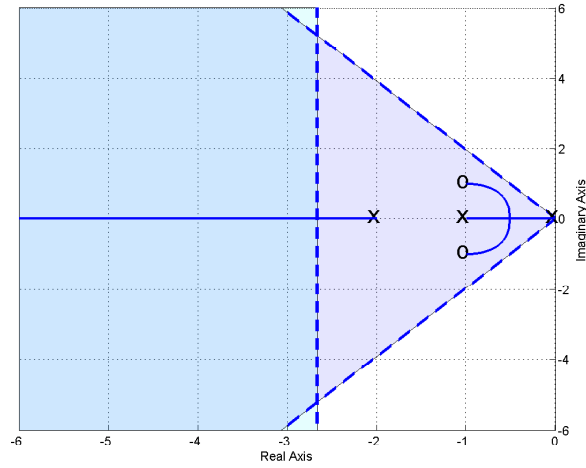


Figure 54: Root locus plots for $G_p(s) = \frac{2}{s^2+3s+2}$ with proportional+integral+derivative (PID) controllers with complex conjugate zeros. The systems in the top two panels demonstrate bad choices for the location of the controller zeros.

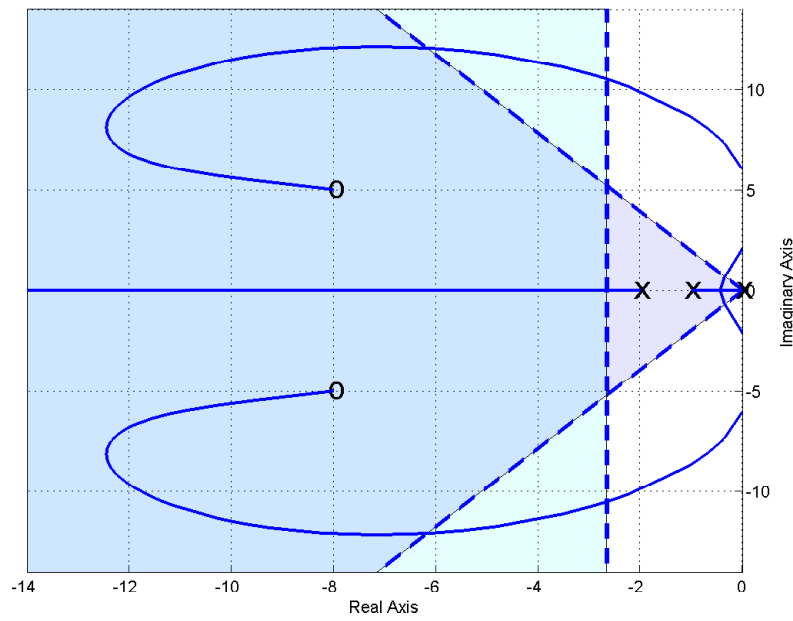
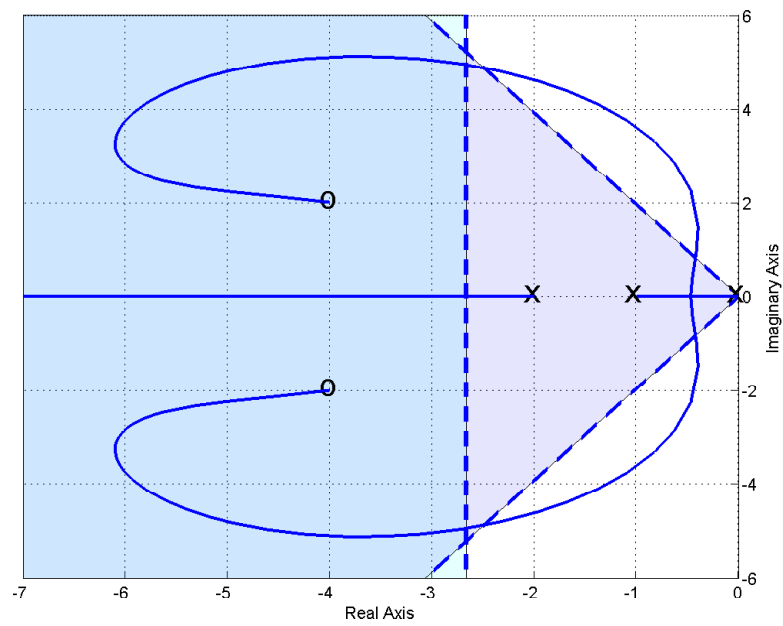


Figure 55: Root locus plots for $G_p(s) = \frac{2}{s^2 + 3s + 2}$ with proportional+integral+derivative (PID) controllers with complex conjugate zeros.

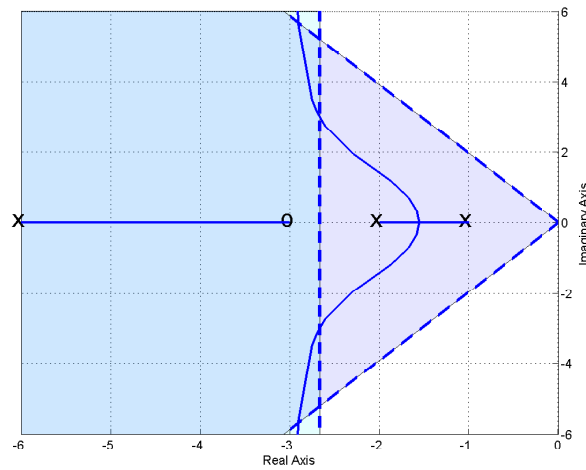
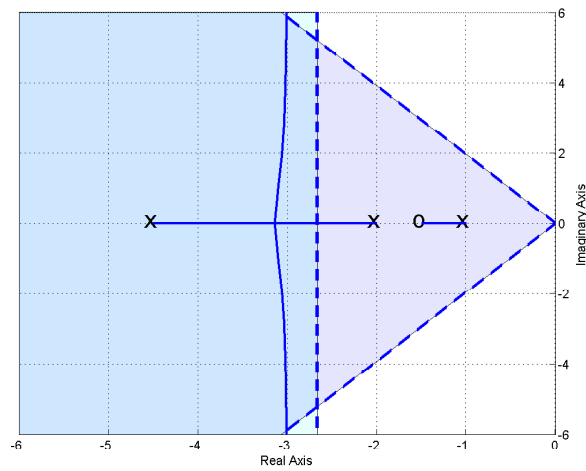
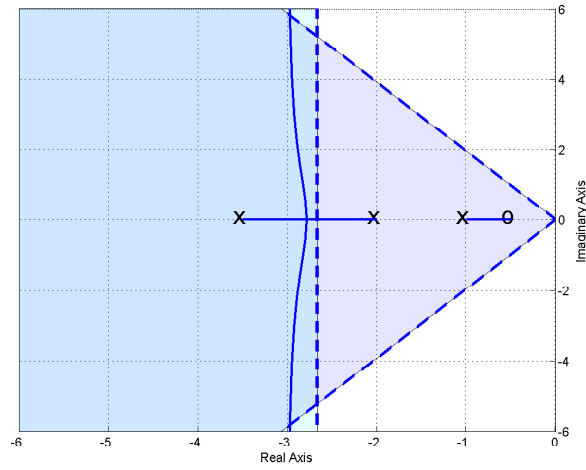


Figure 56: Root locus plots for $G_p(s) = \frac{2}{s^2+3s+2}$ with lead controllers. Here $l = 3$ for all three controllers. However, only the third system is capable of meeting both the settling time and percent overshoot criteria. If k becomes too large in these systems the percent overshoot criteria will not be met.

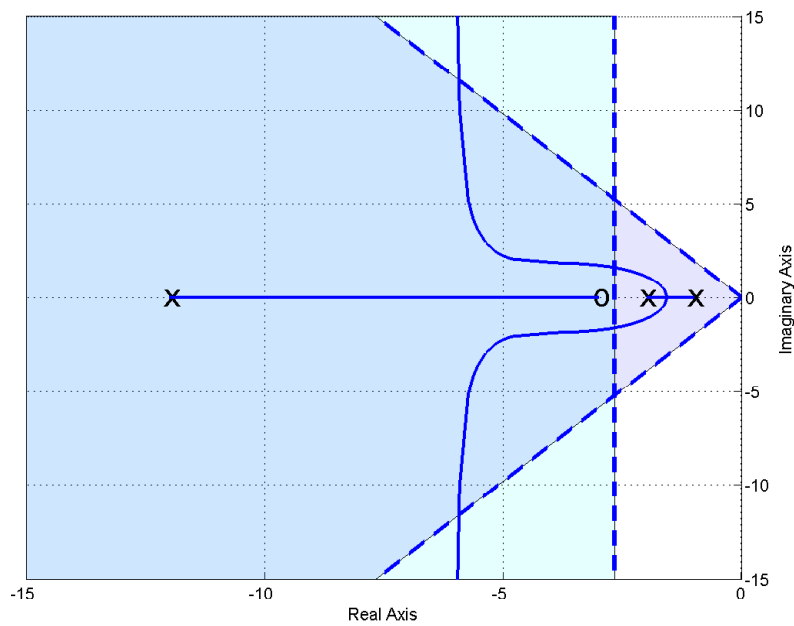
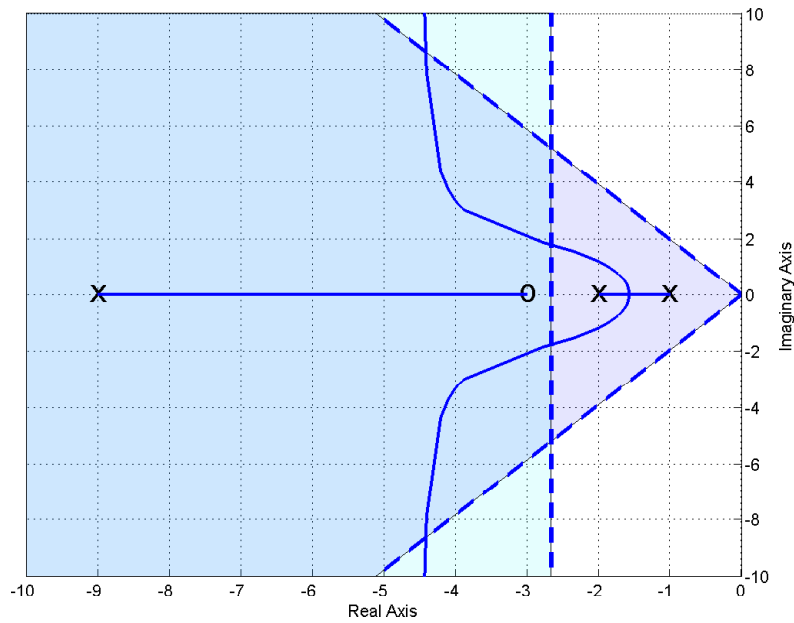


Figure 57: Root locus plots for $G_p(s) = \frac{2}{s^2+3s+2}$ with lead controllers. Here $l = 6$ (top panel) and $l = 9$ (bottom panel). If k becomes too large in these systems the percent overshoot criteria will not be met.

11.8 Seemingly Odd Root Locus Behavior

Up to this point, you have been seeing only “normal” root locus behavior, and these are the types of plots you should be able to produce by hand sketches. However, there are times when one can get some fairly odd looking shapes, and you should be aware that these are possible. None of these root locus plots violate any of the rules you have been given, and if we had gone over the rules for determining the locations the root locus leave the real axis and the location where the root locus enters the real axis (“break-in/break-out” points) these root loci would not seem so strange.

To begin with, let’s continue with the lead controller from Example 3. As we showed before, as the distance between the pole and the zero, $l = p - z$, becomes larger, the asymptotes of the complex poles move to the right. However, at some point we will begin to see some of this odd behavior. Figure 58 displays this behavior as l is varied from 23 to 25.

Figure 59 displays strange behavior for PID controllers for Example 3 as the imaginary parts of the zeros move closer to the real axis. Again, none of our rules are violated, but this is unexpected behavior.

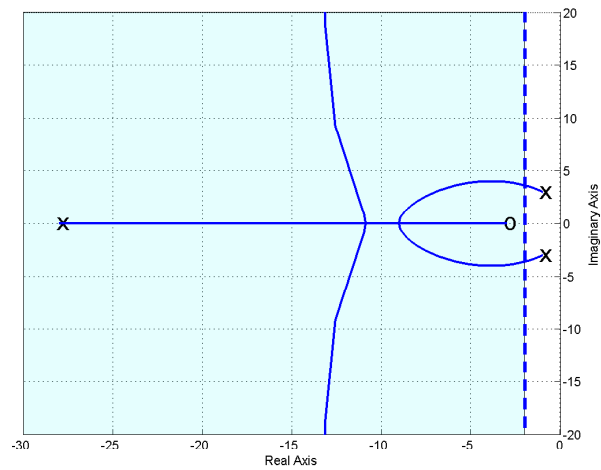
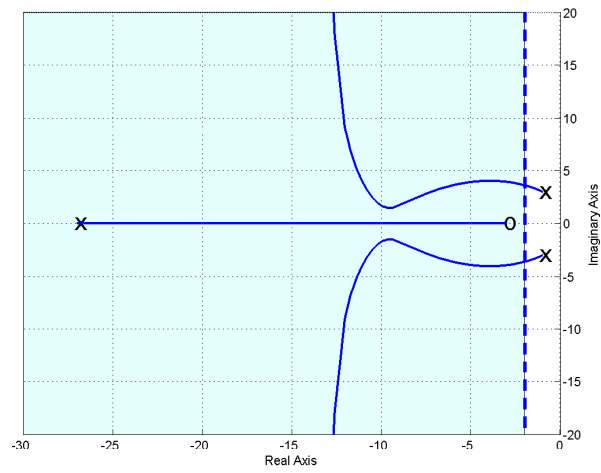
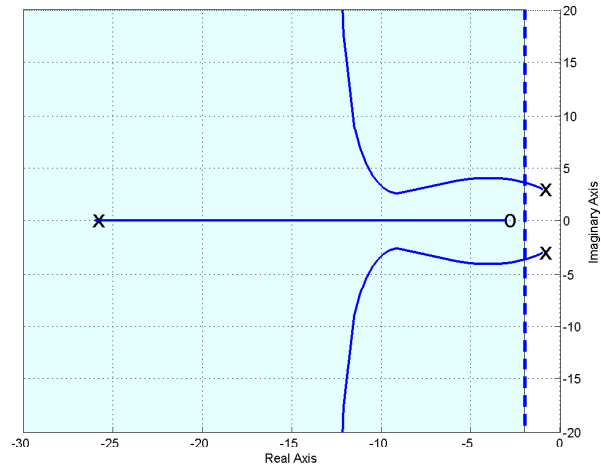


Figure 58: Root locus plot for $G_p(s) = \frac{2}{s^2+2s+10}$ with the lead controller for $l = 23, 24,$ and 25 . This graph exemplifies some of the possible root locus behavior we have not seen yet. To match the settling time *all* poles must be to the left of -2 .

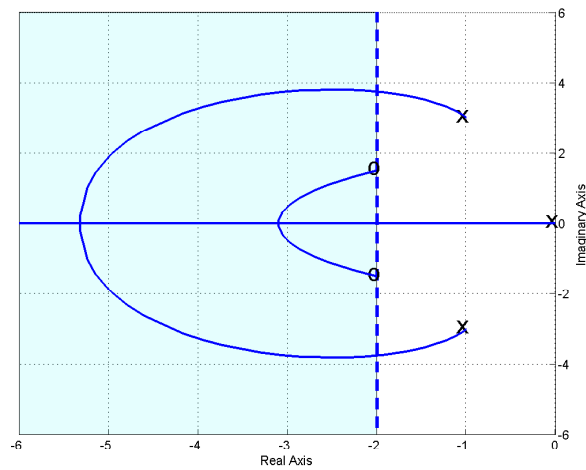
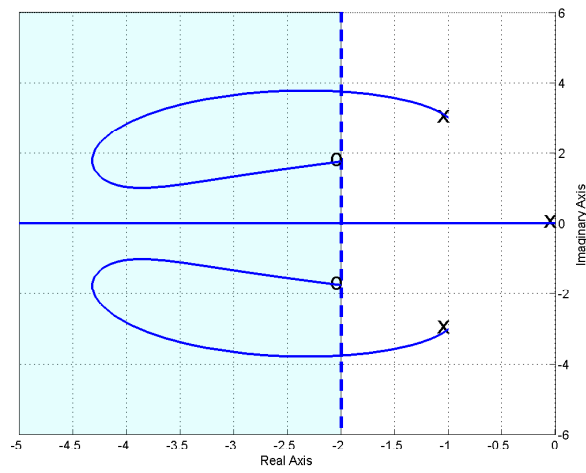
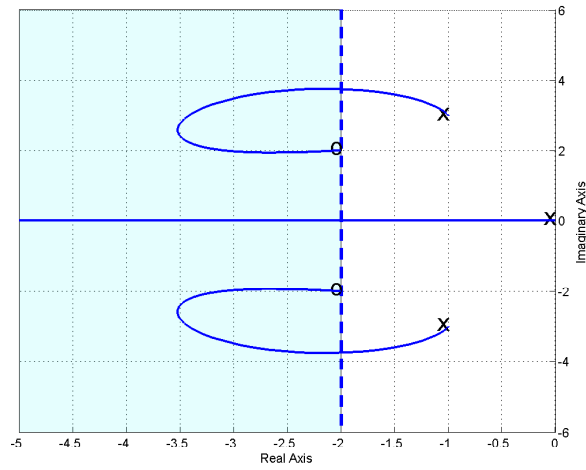


Figure 59: Root locus plot for $G_p(s) = \frac{2}{s^2+2s+10}$ with the pid controller as the imaginary part of the zeros move closer to the real axis. This graph exemplifies some of the possible root locus behavior we have not seen yet. To match the settling time *all* poles must be to the left of -2.

12 System Sensitivity

There are generally two kinds of sensitivity used in control systems. The first type of sensitivity refers to the sensitivity of a system to variations in a parameter or transfer function. (This type of sensitivity is important to study since we need to be able to determine how to design a control system to reduce the sensitivity of the system to changes in the plant, since we often have to estimate the plant and this estimation will contain some errors.) The other type of sensitivity usually refers to how sensitive the system is to outside disturbances. Again, this is important to understand so we can design a control system to reduce the effects of external disturbances. Finally, it is important to understand that sensitivity is a function of frequency, and you need to understand the range of frequencies you expect to be operating your system under (i.e., the frequency content of the input signal). For example, a system may be very sensitive to a parameter at frequencies near 100 Hz, but if your system is typically operating in the 1-10 Hz range this sensitivity is not very important.

12.1 Sensitivity to Parameter Variations

The system's sensitivity to changes in a parameter α is defined as the ratio of the percentage change in the system transfer function $G_0(s)$ to the percentage change in the parameter α to its nominal value α_0 . Note that α may itself be a transfer function or a block in the block diagram representation of a system.

To mathematically define the sensitivity, let's denote the system transfer function as

$$G_0(s) = \frac{N_0(s)}{D_0(s)}$$

Then, the sensitivity of G_0 with respect to changes in α is

$$\begin{aligned} S_\alpha^{G_0}(s) &= \left. \frac{\Delta G_0(s)/G_0(s)}{\Delta\alpha/\alpha} \right|_{\alpha_0} \\ &= \left. \frac{\alpha}{G_0(s)} \frac{\Delta G_0(s)}{\Delta\alpha} \right|_{\alpha_0} \\ &= \left. \frac{\alpha}{G_0(s)} \frac{\partial G_0(s)}{\partial\alpha} \right|_{\alpha_0} \end{aligned}$$

A simpler formula for this can be derived as follows:

$$\begin{aligned} \frac{\partial G_0(s)}{\partial\alpha} &= \frac{\partial}{\partial\alpha} \frac{N_0(s)}{D_0(s)} \\ &= \frac{D_0(s) \frac{\partial N_0(s)}{\partial\alpha} - N_0(s) \frac{\partial D_0(s)}{\partial\alpha}}{D_0(s)^2} \\ &= \frac{1}{D_0(s)} \frac{\partial N_0(s)}{\partial\alpha} - \frac{N_0(s)}{D_0(s)^2} \frac{\partial D_0(s)}{\partial\alpha} \\ &= \frac{N_0(s)}{D_0(s)} \left(\frac{1}{N_0(s)} \frac{\partial N_0(s)}{\partial\alpha} - \frac{1}{D_0(s)} \frac{\partial D_0(s)}{\partial\alpha} \right) \end{aligned}$$

Hence

$$\begin{aligned} S_{\alpha}^{G_0}(s) &= \left. \frac{\alpha}{G_0(s)} \frac{\partial G_0(s)}{\partial \alpha} \right|_{\alpha_0} \\ &= \left. \frac{\alpha}{G_0(s)} G_0(s) \left(\frac{1}{N_0(s)} \frac{\partial N_0(s)}{\partial \alpha} - \frac{1}{D_0(s)} \frac{\partial D_0(s)}{\partial \alpha} \right) \right|_{\alpha_0} \end{aligned}$$

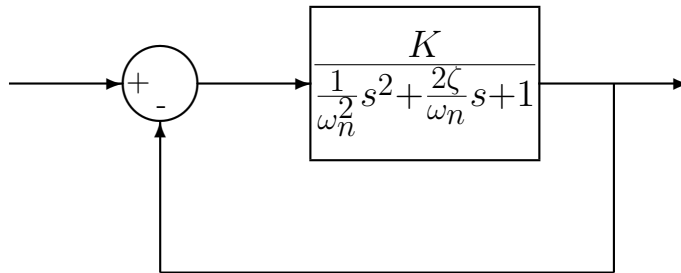
or

$$S_{\alpha}^{G_0}(s) = \left. \left(\frac{\alpha}{N_0(s)} \frac{\partial N_0(s)}{\partial \alpha} - \frac{\alpha}{D_0(s)} \frac{\partial D_0(s)}{\partial \alpha} \right) \right|_{\alpha_0}$$

It is important to note that:

- The sensitivity is really a function of frequency $s = j\omega$, and we normally look at the magnitude as a function of frequency, $|S_{\alpha_0}^{G_0}(j\omega)|$
- We are looking at variations from the nominal values of α_0

Example 1. Consider the closed-loop system shown below:



where the nominal values of the parameters are $\omega_n = 20$, $\zeta = 0.1$, and $K = 0.1$. To compute the sensitivity of the closed-loop system to variations in ω_n (from the nominal value) we first determine the close loop transfer function

$$\begin{aligned} G_0(s) &= \frac{K}{\frac{1}{\omega_n^2} s^2 + \frac{2\zeta}{\omega_n} s + 1 + K} \\ &= \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2(K + 1)} \end{aligned}$$

Hence

$$\begin{aligned} N_0(s) &= K\omega_n^2 \\ D_0(s) &= s^2 + 2\zeta\omega_n s + \omega_n^2(K + 1) \end{aligned}$$

We then compute

$$\begin{aligned}
\frac{\partial N_0(s)}{\partial \omega_n} &= 2\omega_n K \\
\frac{\partial D_0(s)}{\partial \omega_n} &= 2\zeta s + 2\omega_n(K+1) \\
S_{\omega_n}^{G_0}(s) &= \left(\frac{\omega_n}{N_0(s)}\right)(2\omega_n K) + \left(\frac{\omega_n}{D_0(s)}\right)[2\zeta s + 2\omega_n(K+1)] \\
&= \frac{2\omega_n^2 K}{\omega_n^2 K} - \frac{2\zeta\omega_n s + 2\omega_n^2(K+1)}{s^2 + 2\zeta\omega_n s + \omega_n^2(K+1)} \\
&= 2 - \frac{2\zeta\omega_n s + 2\omega_n^2(K+1)}{s^2 + 2\zeta\omega_n s + \omega_n^2(K+1)} \\
&= \frac{[2s^2 + 4\zeta\omega_n s + 2\omega_n^2(K+1)] - [2\zeta\omega_n s + 2\omega_n^2(K+1)]}{s^2 + 2\zeta\omega_n s + \omega_n^2(K+1)} \\
&= \frac{2s^2 + 2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2(K+1)}
\end{aligned}$$

In terms of frequency this is

$$S_{\omega_n}^{G_0}(j\omega) = \frac{-2\omega^2 + 2j\zeta\omega_n\omega}{-\omega^2 + j2\zeta\omega_n\omega + \omega_n^2(K+1)}$$

In terms of the magnitude this is

$$|S_{\omega_n}^{G_0}(j\omega)| = \frac{\sqrt{(2\omega^2)^2 + (2\zeta\omega_n\omega)^2}}{\sqrt{(\omega_n^2(K+1) - \omega^2)^2 + (2\zeta\omega_n\omega)^2}}$$

Figure 60 shows a graph of the sensitivity function $|S_{\omega_n}^{G_0}(j\omega)|$ as a function of frequency, for the nominal values $K = 0.1$, $\omega_n = 20$, and $\zeta = 0.1$. As the figure shows, the system is not very sensitive to changes in ω_n until ω is around 10 rad/sec.

Example 2. Consider the following two systems, the first is an open-loop system with a prefilter (G_{pf}) and controller ($G_c(s)$) before the plant ($G_p(s)$), and the second is a closed-loop system with a prefilter outside of the closed-loop and a controller inside the loop before the plant. Let's

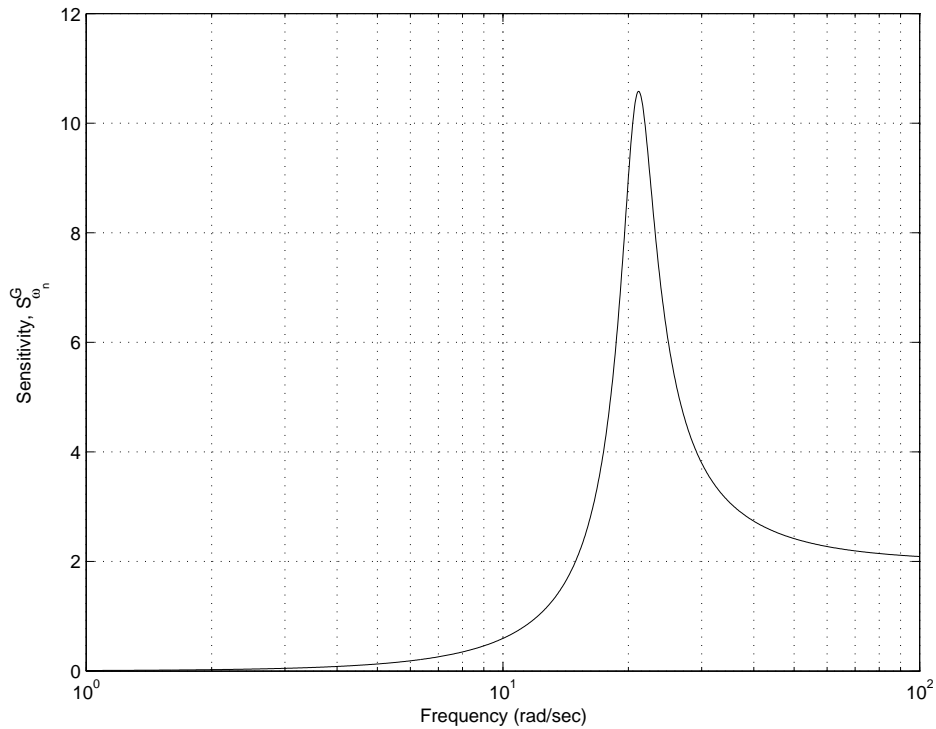
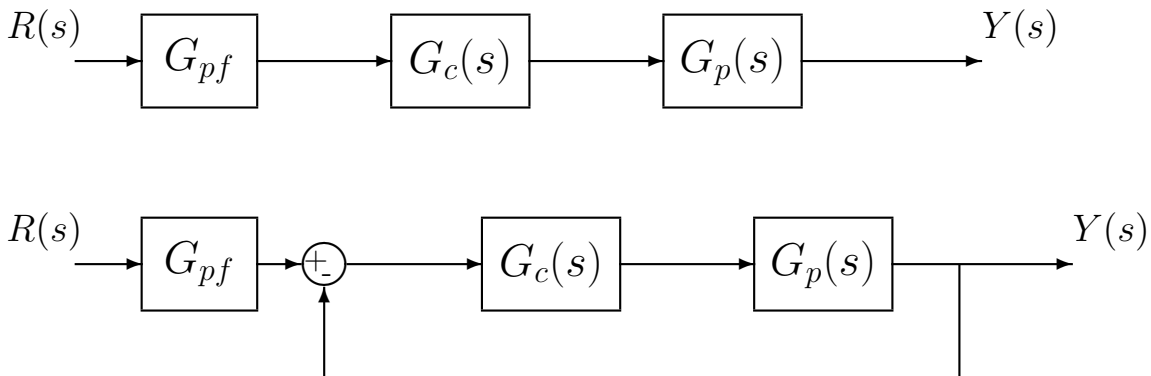


Figure 60: The sensitivity function of Example 1, $S_{\omega_n}^{G_0}(j\omega)$, as a function of frequency for the nominal values $K = 0.1$, $\omega_n = 20$, and $\zeta = 0.1$.

examine the sensitivity of each system to variations in the prefilter and controller.



First we need to determine expressions for the transfer function between the input $R(s)$ and output $Y(s)$ for the two systems. For the open-loop system we have

$$G_0^{open}(s) = G_{pf}(s)G_c(s)G_p(s)$$

while for the closed-loop system we have

$$G_0^{closed}(s) = \frac{G_{pf}(s)G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}$$

Let's first compute the sensitivity to variations in the prefilter, $G_{pf}(s)$. For the open-loop system

$$\begin{aligned} S_{G_{pf}}^{G_0^{open}} &= \frac{G_{pf}(s)}{N_0(s)} \frac{\partial N_0(s)}{\partial G_{pf}(s)} - \frac{G_{pf}(s)}{D_0(s)} \frac{\partial D_0(s)}{\partial G_{pf}(s)} \\ &= \frac{G_{pf}(s)}{G_{pf}(s)G_c(s)G_p(s)} G_c(s)G_p(s) - 0 \\ &= 1 \end{aligned}$$

For the close loop system

$$\begin{aligned} S_{G_{pf}}^{G_0^{closed}} &= \frac{G_{pf}(s)}{N_0(s)} \frac{\partial N_0(s)}{\partial G_{pf}(s)} - \frac{G_{pf}(s)}{D_0(s)} \frac{\partial D_0(s)}{\partial G_{pf}(s)} \\ &= \frac{G_{pf}(s)}{G_{pf}(s)G_c(s)G_p(s)} G_c(s)G_p(s) - 0 \\ &= 1 \end{aligned}$$

Hence both the open and closed-loop systems are equally sensitive to variations in the prefilter $G_{pf}(s)$. *This is because the prefilter is outside of the close loop. Feedback cannot help compensate for variations outside of the closed-loop!*

Now let's compute the sensitivity to variations in the plant, $G_p(s)$. For the open-loop system

$$\begin{aligned} S_{G_p}^{G_0^{open}} &= \frac{G_p(s)}{N_0(s)} \frac{\partial N_0(s)}{\partial G_p(s)} - \frac{G_p(s)}{D_0(s)} \frac{\partial D_0(s)}{\partial G_p(s)} \\ &= \frac{G_p(s)}{G_{pf}(s)G_c(s)G_p(s)} G_{pf}(s)G_c(s) - 0 \\ &= 1 \end{aligned}$$

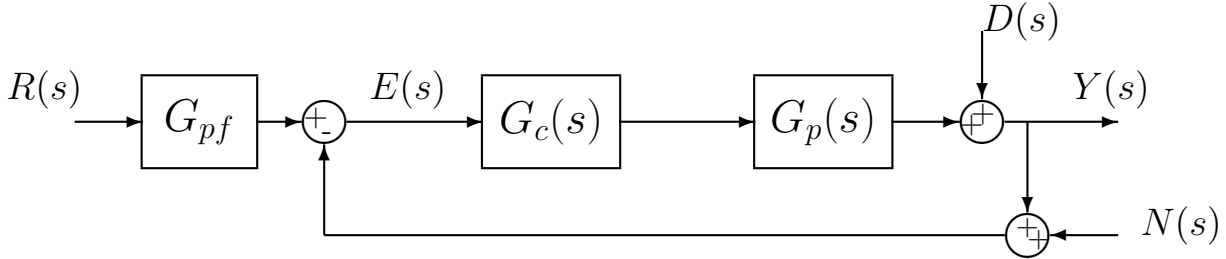
For the close loop system

$$\begin{aligned} S_{G_p}^{G_0^{closed}} &= \frac{G_p(s)}{N_0(s)} \frac{\partial N_0(s)}{\partial G_p(s)} - \frac{G_p(s)}{D_0(s)} \frac{\partial D_0(s)}{\partial G_p(s)} \\ &= \frac{G_p(s)}{G_{pf}(s)G_c(s)G_p(s)} G_{pf}(s)G_c(s) - \frac{G_p(s)}{1 + G_c(s)G_p(s)} G_c(s) \\ &= 1 - \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \\ &= \frac{[1 + G_c(s)G_p(s)] - [G_c(s)G_p(s)]}{1 + G_c(s)G_p(s)} \\ &= \frac{1}{1 + G_c(s)G_p(s)} \end{aligned}$$

In order to reduce the sensitivity of the system to variations in the plant, we want $|1 + G_c(j\omega)G_p(j\omega)|$ to be large. In this case, the closed-loop system can be made much less sensitive to variations in the plant than the open-loop systems. *This is because the plant is inside of the close loop. Feedback can help compensate for parameter/plant variations inside of the closed-loop!*

12.2 Sensitivity to External Disturbances

In addition to the sensitivity of a system to variation in a parameter, we need to also look at the sensitivity of a system to external disturbances. The two most common models of external disturbances are (1) a disturbance that changes the controlled variable, and (2) additive noise in a sensor. Consider the system shown below, with additive disturbances $D(s)$, which models an *output disturbance*, and $N(s)$, which models a *noise disturbance*. When analyzing each of these disturbances we assume there is only one input to the system at a time.



For the output disturbance, we compute the transfer function from $D(s)$ to $Y(s)$ (assuming $N(s)$ and $R(s)$ are zero) as

$$\begin{aligned} E(s) &= 0 - Y(s) \\ Y(s) &= E(s)G_c(s)G_p(s) + D(s) \\ &= -G_c(s)G_p(s)Y(s) + D(s) \end{aligned}$$

or the closed-loop transfer function from $D(s)$ to $Y(s)$ is

$$G_0^D(s) = \frac{1}{1 + G_c(s)G_p(s)}$$

Hence, to reduce the sensitivity of the system to output disturbances, we need $|1 + G_c(j\omega)G_p(j\omega)|$ to be large. This is the same condition we had to reduce the system's sensitivity to variations in $G_p(s)$.

For the noise disturbance, we compute the transfer function from $N(s)$ to $Y(s)$ (assuming $D(s)$ and $R(s)$ are zero) as

$$\begin{aligned} E(s) &= 0 - [N(s) + Y(s)] \\ Y(s) &= E(s)G_c(s)G_p(s) \\ &= -G_c(s)G_p(s)Y(s) - G_c(s)G_p(s)N(s) \end{aligned}$$

or the closed-loop transfer function from $N(s)$ to $Y(s)$ is

$$G_0^N(s) = \frac{-G_c(s)G_p(s)}{1 + G_c(s)G_p(s)}$$

Hence to reduce the sensitivity of the system to noise disturbances we need $|G_c(j\omega)G_p(j\omega)|$ to be small. This is essentially the opposite of the condition we need to reduce the system sensitivity to variations in $G_p(s)$ or to output disturbances.

12.3 Summary

There are generally two kinds of sensitivity used in control systems. The first type of sensitivity refers to the sensitivity of a system to variations in a parameter or transfer function. We compute this sensitivity as

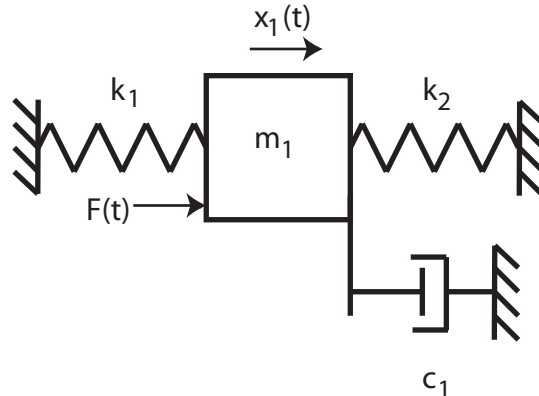
$$S_{\alpha}^{G_0}(s) = \left(\frac{\alpha}{N_0(s)} \frac{\partial N_0(s)}{\partial \alpha} - \frac{\alpha}{D_0(s)} \frac{\partial D_0(s)}{\partial \alpha} \right) \Big|_{\alpha_0}$$

We usually compute the sensitivity as a function of frequency, ω , $|S_{\alpha_0}^{G_0}(j\omega)|$. We are generally only concerned with the sensitivity within a range of frequencies that our system will be operating in. From the examples we see that, from a system sensitivity view, a closed-loop system has no advantages over an open-loop system for parameters or transfer functions outside the feedback loop. For a closed-loop system with plant $G_p(s)$, to minimize the sensitivity of the closed-loop system to variations in the plant we want $|1 + G_c(j\omega)G_p(j\omega)|$ to be *large*.

The other type of sensitivity usually refers to how sensitive the system is to *output disturbances* or *noise disturbances*. To reduce the effects of output disturbances, we again want $|1 + G_c(j\omega)G_p(j\omega)|$ to be *large*. To reduce the effects of noise disturbances we want $|G_c(j\omega)G_p(j\omega)|$ to be *small*. These are contradictory conditions. The relative importance of the different disturbances depends on the particular system being analyzed.

13 State Variables and State Variable Feedback

Consider the model of the rectilinear spring-mass-damper system we have been using in lab.



The equations of motion can be written

$$m_1 \ddot{x}_1(t) + c_1 \dot{x}_1(t) + (k_1 + k_2)x_1(t) = F(t)$$

or

$$\frac{1}{\omega_n^2} \ddot{x}_1(t) + \frac{2\zeta}{\omega_n} \dot{x}_1(t) + x_1(t) = \frac{1}{k_1 + k_2} F(t) \equiv K_{static} u(t)$$

where $u(t)$ is the motor input in volts, and K_{static} is the static gain for the system. Note that this gain also includes the open-loop motor gain. We can also write this as

$$\ddot{x}(t) + 2\zeta\omega_n \dot{x}(t) + \omega_n^2 x(t) = \omega_n^2 K_{static} u(t)$$

We can then take Laplace transforms to get the transfer function

$$G_p(s) = \frac{X_1(s)}{U(s)} = \frac{K_{static}}{\frac{1}{\omega_n^2} s^2 + \frac{2\zeta}{\omega_n} s + 1}$$

We can also write the model in *state variable form*. For linear, time-invariant models, a state variable model has the general form

$$\begin{aligned} \dot{\underline{x}}(t) &= A\underline{x}(t) + B\underline{u}(t) \\ \underline{y}(t) &= C\underline{x}(t) + D\underline{u}(t) \end{aligned}$$

where $\underline{x}(t)$ is the state vector, $\underline{u}(t)$ is the input vector, $\underline{y}(t)$ is the output vector, and A, B, C , and D are constant matrices.

For our system, let's let $q_1(t) = x(t)$ and $q_2(t) = \dot{x}(t)$. Then we can write

$$\begin{aligned} \dot{q}_1(t) &= q_2(t) \\ \dot{q}_2(t) &= -2\zeta\omega_n \dot{x}(t) - \omega_n^2 x(t) + \omega_n^2 K_{static} u(t) \\ &= -2\zeta\omega_n q_2(t) - \omega_n^2 q_1(t) + \omega_n^2 K_{static} u(t) \\ &= -\omega_n^2 q_1(t) - 2\zeta\omega_n q_2(t) + \omega_n^2 K_{static} u(t) \end{aligned}$$

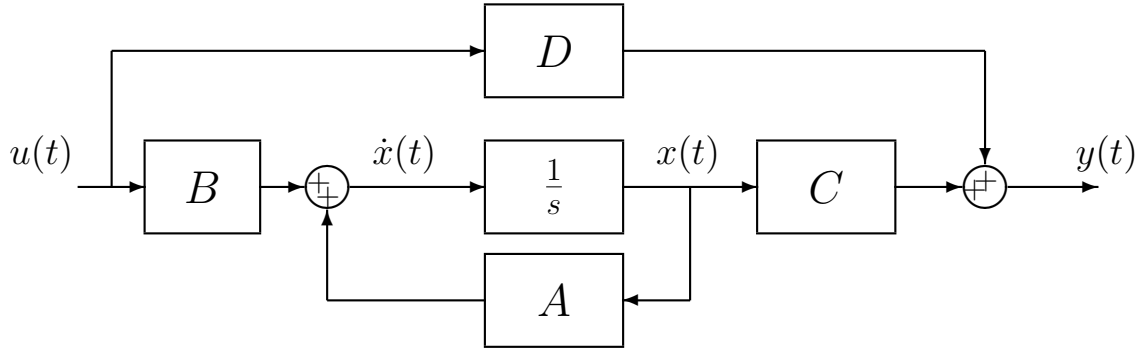


Figure 61: General state variable form for an open-loop plant

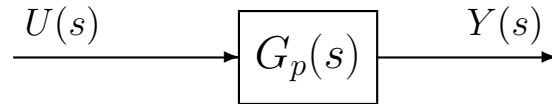


Figure 62: General transfer function form for an open-loop plant

If the output is considered to be the position of the cart, the correct state variable form is

$$\frac{d}{dt} \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 K_{static} \end{bmatrix} u(t)$$

$$y(t) = [1 \ 0] \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}$$

If the output was considered to be the velocity of the cart, the output equation would be

$$y(t) = [0 \ 1] \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}$$

while, if both the position of the cart and the velocity of the cart were the desired outputs, the output equation would be

$$\underline{y}(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}$$

We would like to be able to go between a state variable model of a system to a transfer function model. Each type of model has its benefits. Figure 61 shows the general form for an open-loop state variable model of a plant, while Figure 62 shows the equivalent transfer function form.

13.1 State Variable to Transfer Function Model

Assume we have the state variable description written in scalar form:

$$\begin{aligned}\dot{x}_1(t) &= a_{11}x_1(t) + a_{12}x_2(t) + b_1u(t) \\ \dot{x}_2(t) &= a_{21}x_1(t) + a_{22}x_2(t) + b_2u(t) \\ y(t) &= c_1x_1(t) + c_2x_2(t) + du(t)\end{aligned}$$

In matrix/vector form, this is

$$\begin{aligned}\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + [d]u(t)\end{aligned}$$

or

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

Taking the Laplace transform of the scalar equations (assuming zero initial conditions) we get

$$\begin{aligned}\begin{bmatrix} sX_1(s) \\ sX_2(s) \end{bmatrix} &= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} U(s) \\ Y(s) &= \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} X_1(s) \\ X_2(s) \end{bmatrix} + [d]U(s)\end{aligned}$$

We can write this new system of equations in matrix form as

$$\begin{aligned}sX(s) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s)\end{aligned}$$

We can rewrite the first equation as

$$(sI - A)X(s) = BU(s)$$

or

$$X(s) = (sI - A)^{-1}BU(s)$$

We can then solve for $Y(s)$ as

$$Y(s) = [C(sI - A)^{-1}B + D]U(s)$$

or

$$Y(s) = G(s)U(s)$$

Hence, the transfer function (or transfer matrix, if there is more than one input or output), is given by

$$G(s) = [C(sI - A)^{-1}B + D]$$

In going from a state variable model to a transfer function model, you need to be able to compute the inverse of a matrix. You are expected to be able to compute the inverse of a 2x2 matrix without a computer (or calculator). If matrix P is given as

$$P = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

then

$$P^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

and the *determinant* of P is given by $ad - bc$.

Example 1. Assume we have the state variable model

$$\begin{aligned} \dot{\underline{x}} &= \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} \underline{x} + \begin{bmatrix} 5 \\ 0 \end{bmatrix} u \\ y &= [1 \ 2] \underline{x} \end{aligned}$$

and we want to find the transfer function model. We need to compute

$$G(s) = [C(sI - A)^{-1}B + D]$$

First we compute $sI - A$ as

$$sI - A = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} s-1 & 0 \\ -2 & s-3 \end{bmatrix}$$

Next we compute

$$(sI - A)^{-1} = \frac{1}{(s-1)(s-3) - (-2)(0)} \begin{bmatrix} s-3 & 0 \\ 2 & s-1 \end{bmatrix}$$

then

$$\begin{aligned} C(sI - A)^{-1} &= [1 \ 2] \frac{1}{(s-1)(s-3)} \begin{bmatrix} s-3 & 0 \\ 2 & s-1 \end{bmatrix} \\ &= \frac{1}{(s-1)(s-3)} [(1)(s-3) + (2)(2) \quad (1)(0) + (2)(s-1)] \\ &= \frac{1}{(s-1)(s-3)} [s+1 \quad 2s-2] \end{aligned}$$

and finally

$$\begin{aligned}
 G(s) &= C(sI - A)^{-1}B \\
 &= \frac{1}{(s-1)(s-3)} [s+1 \quad 2s-2] \begin{bmatrix} 5 \\ 0 \end{bmatrix} \\
 &= \frac{1}{(s-1)(s-3)} [5(s+1) + 0(2s-2)] \\
 &= \frac{5(s+1)}{(s-1)(s-3)}
 \end{aligned}$$

The poles of the transfer function are at 1 and 3, and there is a zero at -1. The system is clearly unstable.

Example 2. Assume we have the state variable model

$$\begin{aligned}
 \dot{\underline{x}} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \underline{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u \\
 y &= [1 \ 2] \underline{x}
 \end{aligned}$$

and we want to find the transfer function model. We need to compute

$$G(s) = [C(sI - A)^{-1}B + D]$$

First we compute $sI - A$ as

$$sI - A = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} s-1 & 0 \\ 0 & s \end{bmatrix}$$

Next we compute

$$(sI - A)^{-1} = \frac{1}{(s-1)(s) - (0)(0)} \begin{bmatrix} s & 0 \\ 0 & s-1 \end{bmatrix}$$

then

$$\begin{aligned}
 C(sI - A)^{-1} &= [1 \ 2] \frac{1}{s(s-1)} \begin{bmatrix} s & 0 \\ 0 & s-1 \end{bmatrix} \\
 &= \frac{1}{s(s-1)} [(1)(s) \quad (2)(s-1)] \\
 &= \frac{1}{s(s-1)} [s \quad 2s-2]
 \end{aligned}$$

and finally

$$\begin{aligned}
 G(s) &= C(sI - A)^{-1}B \\
 &= \frac{1}{s(s-1)} [s \quad 2s-2] \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
 &= \frac{1}{s(s-1)} [s + (2s-2)] \\
 &= \frac{3s-2}{s(s-1)}
 \end{aligned}$$

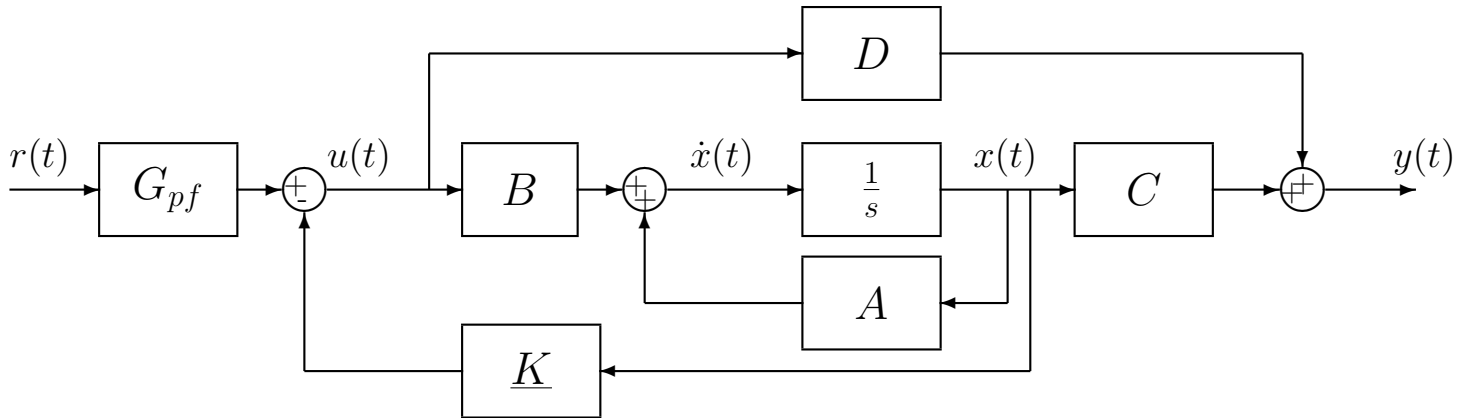


Figure 63: State variable model of a plant with state variable feedback.

The poles of the transfer function are at 0 and 1, and there is a zero at $-\frac{2}{3}$. The system is clearly unstable.

13.2 State Variable Feedback

Up to this point, we have shown how we can go from a state variable description of an open-loop system to a transfer function model. In particular, we can model a plant using either a transfer function description or a state variable description. We can then implement any of the single-input single-output controllers we have been utilizing in this course. However, each of these methods assumes we are feeding back only one variable, usually the output. However, a state variable model allows us a much more powerful method of control, that of feeding back all of the states, which is called *state variable feedback*.

Let's assume the input to the plant, $u(t)$, is the difference between the scaled reference input, $G_{pf}r(t)$, and scaled states, $\underline{K}\underline{x}(t)$, or

$$u(t) = G_{pf}r(t) - \underline{K}\underline{x}(t)$$

Here G_{pf} is a prefilter, much like we used $G_{pf}(s)$ for the transfer function feedback systems. Figure 63 displays a state variable model of a plant with state variable feedback.

With the state variable feedback the state equations become

$$\begin{aligned} \dot{\underline{x}}(t) &= A\underline{x}(t) + Bu(t) \\ &= A\underline{x}(t) + B[G_{pf}r(t) - \underline{K}\underline{x}(t)] \\ &= [A - B\underline{K}]\underline{x}(t) + [BG_{pf}]r(t) \\ &= \tilde{A}\underline{x}(t) + \tilde{B}r(t) \end{aligned}$$

where

$$\begin{aligned}\tilde{A} &= [A - B\underline{K}] \\ \tilde{B} &= BG_{pf}\end{aligned}$$

The output equation is then

$$\begin{aligned}y(t) &= C\underline{x}(t) + Du(t) \\ &= C\underline{x}(t) + D[G_{pf}r(t) - \underline{K}\underline{x}(t)] \\ &= [C - D\underline{K}]\underline{x}(t) + [DG_{pf}]r(t) \\ &= \tilde{C}\underline{x}(t) + \tilde{D}r(t)\end{aligned}$$

where

$$\begin{aligned}\tilde{C} &= [C - D\underline{K}] \\ \tilde{D} &= DG_{pf}\end{aligned}$$

Under most circumstances $D = 0$ so $\tilde{C} = C$ and $\tilde{D} = 0$.

The new input to our system is $r(t)$. The transfer function between the input $R(s)$ and the output $Y(s)$ for the state variable model with state variable feedback is given by

$$G(s) = \tilde{C}(sI - \tilde{A})^{-1}\tilde{B} + \tilde{D}$$

Example 3. Assume we again have the state variable model

$$\begin{aligned}\dot{\underline{x}} &= \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} \underline{x} + \begin{bmatrix} 5 \\ 0 \end{bmatrix} u \\ y &= [1 \ 2] \underline{x}\end{aligned}$$

but now we have state variable feedback. We want to find the transfer function model for the system with the state variable feedback. We need to compute

$$G(s) = [\tilde{C}(sI - \tilde{A})^{-1}\tilde{B} + \tilde{D}]$$

First we compute

$$\begin{aligned}\tilde{A} &= A - B\underline{K} = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} - \begin{bmatrix} 5 \\ 0 \end{bmatrix} [K_1 \ K_2] \\ &= \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} - \begin{bmatrix} 5K_1 & 5K_2 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 - 5K_1 & -5K_2 \\ 2 & 3 \end{bmatrix}\end{aligned}$$

and

$$\tilde{B} = BG_{pf} = \begin{bmatrix} 5G_{pf} \\ 0 \end{bmatrix}$$

Since $D = 0$ we have $\tilde{C} = C$ and $\tilde{D} = 0$.

Next we compute

$$\begin{aligned} sI - \tilde{A} &= \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 1 - 5K_1 & -5K_2 \\ 2 & 3 \end{bmatrix} \\ &= \begin{bmatrix} s - 1 + 5K_1 & 5K_2 \\ -2 & s - 3 \end{bmatrix} \end{aligned}$$

and

$$(sI - \tilde{A})^{-1} = \frac{1}{(s - 1 + 5K_1)(s - 3) - (-2)(5K_2)} \begin{bmatrix} s - 3 & -5K_2 \\ 2 & s - 1 + 5K_1 \end{bmatrix}$$

At this point it is probably easiest to postmultiply by \tilde{B} first

$$\begin{aligned} (sI - \tilde{A})^{-1} \tilde{B} &= \frac{1}{(s - 1 + 5K_1)(s - 3) - (-2)(5K_2)} \begin{bmatrix} s - 3 & -5K_2 \\ 2 & s - 1 + 5K_1 \end{bmatrix} \begin{bmatrix} 5G_{pf} \\ 0 \end{bmatrix} \\ &= \frac{1}{(s - 1 + 5K_1)(s - 3) + 10K_2} \begin{bmatrix} 5G_{pf}(s - 3) \\ 10G_{pf} \end{bmatrix} \end{aligned}$$

Finally, premultiplying by C we get

$$\begin{aligned} G(s) &= [1 \ 2] \frac{1}{(s - 1 + 5K_1)(s - 3) + 10K_2} \begin{bmatrix} 5G_{pf}(s - 3) \\ 10G_{pf} \end{bmatrix} \\ &= \frac{5G_{pf}(s - 3) + (2)(10G_{pf})}{(s - 1 + 5K_1)(s - 3) + 10K_2} \\ &= \frac{G_{pf}5(s + 1)}{s^2 + (5K_1 - 4)s + (10K_2 - 15K_1 + 3)} \end{aligned}$$

You should note

- the state variable feedback *did not change the zeros of the system*
- G_{pf} is just a scaling factor
- For $K_1 = K_2 = 0$ (open-loop) and $G_{pf} = 1$ (no prefilter), we get

$$G(s) = \frac{5(s + 1)}{(s - 1)(s - 3)}$$

as before.

Example 4. Assume we again have the state variable model

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u \\ y &= [1 \ 2] \mathbf{x} \end{aligned}$$

but now we have state variable feedback. We want to find the transfer function model for the system with the state variable feedback. We need to compute

$$G(s) = [\tilde{C}(sI - \tilde{A})^{-1}\tilde{B} + \tilde{D}]$$

First we compute

$$\begin{aligned}\tilde{A} &= A - BK = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} [K_1 \ K_2] \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} K_1 & K_2 \\ K_1 & K_2 \end{bmatrix} \\ &= \begin{bmatrix} 1 - K_1 & -K_2 \\ -K_1 & -K_2 \end{bmatrix}\end{aligned}$$

and

$$\tilde{B} = BG_{pf} = \begin{bmatrix} G_{pf} \\ G_{pf} \end{bmatrix}$$

Since $D = 0$ we have $\tilde{C} = C$ and $\tilde{D} = 0$.

Next we compute

$$\begin{aligned}sI - \tilde{A} &= \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 1 - K_1 & -K_2 \\ -K_1 & -K_2 \end{bmatrix} \\ &= \begin{bmatrix} s - 1 + K_1 & K_2 \\ K_1 & s + K_2 \end{bmatrix}\end{aligned}$$

and

$$(sI - \tilde{A})^{-1} = \frac{1}{(s - 1 + K_1)(s + K_2) - (K_1)(K_2)} \begin{bmatrix} s + K_2 & -K_2 \\ -K_1 & s - 1 + K_1 \end{bmatrix}$$

At this point it is probably easiest to postmultiply by \tilde{B} first

$$\begin{aligned}(sI - \tilde{A})^{-1} \tilde{B} &= \frac{1}{(s - 1 + K_1)(s + K_2) - K_1 K_2} \begin{bmatrix} s + K_2 & -K_2 \\ -K_1 & s - 1 + K_1 \end{bmatrix} \begin{bmatrix} G_{pf} \\ G_{pf} \end{bmatrix} \\ &= \frac{G_{pf}}{(s - 1 + K_1)(s + K_2) - K_1 K_2} \begin{bmatrix} s \\ s - 1 \end{bmatrix}\end{aligned}$$

Finally, premultiplying by C we get

$$\begin{aligned}G(s) &= [1 \ 2] \frac{G_{pf}}{(s - 1 + K_1)(s + K_2) - K_1 K_2} \begin{bmatrix} s \\ s - 1 \end{bmatrix} \\ &= \frac{G_{pf}(3s - 2)}{(s - 1 + K_1)(s + K_2) - K_1 K_2} \\ &= \frac{G_{pf}(3s - 2)}{s^2 + (K_1 + K_2 - 1)s - K_2}\end{aligned}$$

You should note

- the state variable feedback *did not change the zeros of the system*
- G_{pf} is just a scaling factor
- For $K_1 = K_2 = 0$ (open-loop) and $G_{pf} = 1$ (no prefilter), we get

$$G(s) = \frac{3s - 2}{s(s - 1)}$$

as before.

Example 5. Assume we have the state variable model

$$\begin{aligned}\dot{\underline{x}} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \underline{x} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u \\ y &= [3 \ 4] \underline{x}\end{aligned}$$

We want to find the transfer function model for the system with the state variable feedback. We need to compute

$$G(s) = [\tilde{C}(sI - \tilde{A})^{-1}\tilde{B} + \tilde{D}]$$

First we compute

$$\begin{aligned}\tilde{A} &= A - BK = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} [K_1 \ K_2] \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_1 & K_2 \\ 2K_1 & 2K_2 \end{bmatrix} \\ &= \begin{bmatrix} 1 - K_1 & -K_2 \\ -2K_1 & 1 - 2K_2 \end{bmatrix}\end{aligned}$$

and

$$\tilde{B} = BG_{pf} = \begin{bmatrix} G_{pf} \\ 2G_{pf} \end{bmatrix}$$

Since $D = 0$ we have $\tilde{C} = C$ and $\tilde{D} = 0$.

Next we compute

$$\begin{aligned}sI - \tilde{A} &= \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 1 - K_1 & -K_2 \\ -2K_1 & 1 - 2K_2 \end{bmatrix} \\ &= \begin{bmatrix} s - 1 + K_1 & K_2 \\ 2K_1 & s - 1 + 2K_2 \end{bmatrix}\end{aligned}$$

and

$$(sI - \tilde{A})^{-1} = \frac{1}{(s - 1 + K_1)(s - 1 + 2K_2) - (2K_1)(K_2)} \begin{bmatrix} s - 1 + 2K_2 & -K_2 \\ -2K_1 & s - 1 + K_1 \end{bmatrix}$$

Let's postmultiply by \tilde{B} first

$$\begin{aligned} (sI - \tilde{A})^{-1} \tilde{B} &= \frac{1}{(s-1+K_1)(s-1+2K_2) - (2K_1)(K_2)} \begin{bmatrix} s-1+2K_2 & -K_2 \\ -2K_1 & s-1+K_1 \end{bmatrix} \begin{bmatrix} G_{pf} \\ 2G_{pf} \end{bmatrix} \\ &= \frac{G_{pf}}{(s-1+K_1)(s-1+2K_2) - 2K_1K_2} \begin{bmatrix} s-1 \\ 2s-2 \end{bmatrix} \end{aligned}$$

Finally, premultiplying by C we get

$$\begin{aligned} G(s) &= [3 \ 4] \frac{G_{pf}}{(s-1+K_1)(s-1+2K_2) - 2K_1K_2} \begin{bmatrix} s-1 \\ 2s-2 \end{bmatrix} \\ &= \frac{G_{pf}[3(s-1) + 4(2s-2)]}{(s-1+K_1)(s-1+2K_2) - 2K_1K_2} \\ &= \frac{11G_{pf}(s-1)}{(s-1+K_1)(s-1+2K_2) - 2K_1K_2} \\ &= \frac{11G_{pf}(s-1)}{[(s-1)+K_1][(s-1)+2K_2] - 2K_1K_2} \\ &= \frac{11G_{pf}(s-1)}{(s-1)^2 + (K_1+2K_2)(s-1) + 2K_1K_2 - 2K_1K_2} \\ &= \frac{11G_{pf}}{s-1+K_1+2K_2} \end{aligned}$$

Note that this transfer function has only one pole.

13.3 Controllability for State Variable Systems

A single-input single-output state variable system is said to be *controllable*⁸ if we can place as many poles of the closed-loop transfer function as there are states of the state variable model. For example, if there are two states in the state variable model we assume we want the closed-loop characteristic equation to be $s^2 + a_1s + a_0$ and see if we can find K_1 and K_2 to achieve any possible values for a_1 and a_0 . If, when the transfer function is simplified as much as possible, the order of the characteristic equation (the denominator of the transfer function) is less than the number of states of the system the system is *not controllable* or *uncontrollable*.

Example 6. For the state variable system in Example 3, we set the characteristic polynomial (after all pole/zero cancellations) equal to an arbitrary second order polynomial (since there are two states)

$$s^2 + (5K_1 - 4)s + (10K_2 - 15K_1 + 3) = s^2 + a_1s + a_0$$

from which we get

$$\begin{aligned} 5K_1 - 4 &= a_1 \\ 5K_1 &= a_1 + 4 \\ K_1 &= \frac{a_1 + 4}{5} \end{aligned}$$

⁸This is one of many possible (and equivalent) definitions.

and

$$\begin{aligned}10K_2 - 15K_1 + 3 &= a_0 \\10K_2 &= a_0 + 15K_1 - 3 \\10K_2 &= a_0 + 3(a_1 + 4) - 3 \\10K_2 &= a_0 + 3a_1 - 9 \\K_2 &= \frac{a_0 + 3a_1 - 9}{10}\end{aligned}$$

Hence we can determine a K_1 and K_2 to achieve any possible values of a_0 and a_1 . This system is *controllable*.

Example 7. For the state variable system in Example 4, we set the characteristic polynomial (after all pole/zero cancellations) equal to an arbitrary second order polynomial (since there are two states)

$$s^2 + (K_1 + K_2 - 1)s - K_2 = s^2 + a_1s + a_0$$

from which we get

$$K_2 = -a_0$$

and

$$\begin{aligned}K_1 + K_2 - 1 &= a_1 \\K_1 &= a_1 - K_2 + 1 \\K_1 &= a_1 + a_0 + 1\end{aligned}$$

Hence we can determine a K_1 and K_2 to achieve any possible values of a_0 and a_1 . This system is *controllable*.

Example 8. For the state variable system in Example 5, we set the characteristic polynomial (after all pole/zero cancellations) equal to an arbitrary second order polynomial (since there are two states)

$$s - 1 + K_1 + 2K_2 = s^2 + a_1s + a_0$$

Clearly it is not possible to find constant values of K_1 and K_2 so these two equations to be equal. Hence the system is not controllable.

13.4 Summary

State variable models are an alternative method of modelling a system. However, we can derive transfer function models from state variable models and state variable models from transfer function models. State variable models have an advantage over transfer function models in that we can utilize state variable feedback to place all of the poles of the system if the system is controllable. Unlike the coefficient matching (Diophantine equation) transfer function methods, state variable feedback does not add zeros to the closed-loop system.

16 z -Transforms

In this course we will assume we are sampling the continuous time signal $x(t)$ at a uniform sampling rate. The time interval between samples will be denoted by T . Thus we will denote the discrete-time (sampled) values as $x(0), x(T), x(2T), \dots, x(kT)$. This is shown graphically in Figure 64. Sometimes we remove the explicit dependence on the sampling interval T and just write $x(0), x(1), x(2), \dots, x(k)$, since the sample interval is the same for all of the different samples.

16.1 Special Functions

Just as in continuous-time, there are certain special functions that are used very often in discrete-time. Specifically, we will be concerned with the unit impulse function, the unit step function, and the unit ramp function.

The *unit impulse* or *delta* function is defined as

$$\delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

or

$$\delta(n - k) = \begin{cases} 1 & k - n = 0 \\ 0 & k - n \neq 0 \end{cases}$$

The *unit step* or *Heaviside* function is defined as

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

or

$$u(n - k) = \begin{cases} 1 & n - k \geq 0 \\ 0 & n - k < 0 \end{cases}$$

The *unit ramp* function is defined as

$$r(k) = ku(k)$$

or

$$r(n - k) = (n - k)u(n - k)$$

While there are other special function, these are the special functions we will be utilizing the most.

16.2 Impulse Response and Convolution

The unit impulse response of a Linear Time-Invariant (LTI) system, $h(k)$, is the response of the system at rest (no initial energy, the initial conditions are all zero), to a unit impulse at time zero. Symbolically, we can write

$$\delta(k) \rightarrow h(k)$$

Since the system is assumed to be time-invariant we also have

$$\delta(n - k) \rightarrow h(n - k)$$

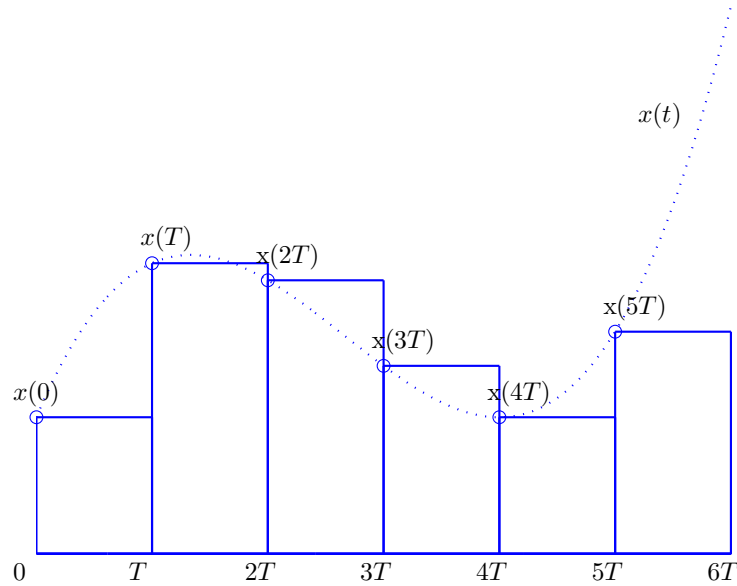


Figure 64: Continuous-time signal $x(t)$ and its samples $x(kT)$. We assume the samples are taken at the beginning of each time interval.

Since the system is also linear, we have

$$\alpha\delta(k) \rightarrow \alpha h(k)$$

and

$$\alpha\delta(n-k) \rightarrow \alpha h(n-k)$$

Now we can write $x(n)$ as

$$x(n) = \dots + x(-2)\delta(n+2) + x(-1)\delta(n+1) + x(0)\delta(n) + x(1)\delta(n-1) + x(2)\delta(n-2) + \dots$$

If $x(n)$ is the input to an LTI system, then we can compute the output as

$$y(n) = \dots + x(-2)h(n+2) + x(-1)h(n+1) + x(0)h(n) + x(1)h(n-1) + x(2)h(n-2) + \dots$$

since we treat the $x(k)$'s as constants, and the system is only responding to the impulses. We can write this expression as

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

or, by making a change of variable in the sum, as

$$y(n) = \sum_{k=-\infty}^{\infty} x(n-k)h(k)$$

Thus the output of an LTI system is the *convolution* of the input with the impulse response of the system,

$$\begin{aligned} y(n) &= h(n) \star x(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \\ &= x(n) \star h(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \end{aligned}$$

If both the system and input are *causal* (both are zero for $n \leq 0$), then we can simplify the convolution sums as

$$\begin{aligned} y(n) &= h(n) \star x(n) = \sum_{k=0}^n h(k)x(n-k) \\ &= x(n) \star h(n) = \sum_{k=0}^n x(k)h(n-k) \end{aligned}$$

16.3 A Useful Summation

In what follows, we will make extensive use of a simple summation formula, which you probably saw in high school. Let's denote the sum S_n as

$$S_n = 1 + a + a^2 + \dots + a^{n-1} + a^n = \sum_{k=0}^{k=n} a^k$$

If we multiply by a we have

$$\begin{aligned} aS_n &= a(1 + a + a^2 + \dots + a^{n-1} + a^n) \\ &= a + a^2 + a^3 + \dots + a^n + a^{n+1} \\ &= S_n - 1 + a^{n+1} \end{aligned}$$

Solving for S_n we have

$$\begin{aligned} aS_n - S_n &= a^{n+1} - 1 \\ S_n &= \frac{a^{n+1} - 1}{a - 1} \end{aligned}$$

or

$$\sum_{k=0}^{k=n} a^k = \frac{1 - a^{n+1}}{1 - a}$$

Note that if $n = \infty$ and $|a| < 1$, then we have

$$\sum_{k=0}^{\infty} a^k = \frac{1}{1 - a}$$

Now let's do some convolution examples.

Example. Assume the input to a system is $x(n) = u(n)$ and the impulse response of the system is $h(n) = u(n)$. Determine the output of the system. We have

$$\begin{aligned}
 y(n) &= x(k) \star h(k) \\
 &= \sum_{k=-\infty}^{\infty} x(k)h(n-k) \\
 &= \sum_{k=-\infty}^n u(k)u(n-k) \\
 &= \sum_{k=0}^n (1)(1) \\
 &= (n+1)u(n)
 \end{aligned}$$

Note that we need the final $u(n)$ since our method is only valid if $n \geq 0$.

Example. Assume the input to a system is $x(n) = \alpha^n u(n)$ and the impulse response is $h(n) = u(n)$. Determine the output of the system. We have

$$\begin{aligned}
 y(n) &= x(k) \star h(k) \\
 &= \sum_{k=-\infty}^{\infty} x(k)h(n-k) \\
 &= \sum_{k=-\infty}^n \alpha^k u(k)u(n-k) \\
 &= \sum_{k=0}^n \alpha^k \\
 &= \frac{1 - \alpha^{n+1}}{1 - \alpha} u(n)
 \end{aligned}$$

Note that we need the final $u(n)$ since our method is only valid if $n \geq 0$.

Example. Assume the input to a system is $x(n) = \alpha^{n+2} u(n+2)$ and the impulse response is $h(n) = \beta^{n-1} u(n-1)$. Determine the output of the system. We have

$$\begin{aligned}
 y(n) &= x(k) \star h(k) \\
 &= \sum_{k=-\infty}^{\infty} x(k)h(n-k) \\
 &= \sum_{k=-\infty}^n \alpha^{k+2} u(k+2) \beta^{n-k-1} u(n-k-1)
 \end{aligned}$$

Now let's look at our sums. We know $u(k+2) = 1$ for $k \geq -2$ and $u(n-k-1) = 1$ for $n-k-1 \geq 0$ or $n-1 \geq k$. Hence our sum is

$$y(n) = \sum_{k=-2}^{n-1} \alpha^{k+2} \beta^{n-k-1}$$

In order to use our summation formula, let's change variables in this sum. We want the lower limit to be zero, so let's define $l = k+2$, or $k = l-2$. In terms of l , our new sum is

$$y(n) = \sum_{l=0}^{l=n+1} \alpha^l \beta^{n-l+1}$$

$$\begin{aligned}
&= \beta^{n+1} \sum_{l=0}^{n+1} \left(\frac{\alpha}{\beta}\right)^l \\
&= \beta^{n+1} \left[\frac{1 - \left(\frac{\alpha}{\beta}\right)^{n+2}}{1 - \frac{\alpha}{\beta}} \right] u(n+1)
\end{aligned}$$

Example. Assume the input to a system is $x(n) = \alpha^{-n}u(-n)$ and the impulse response is $h(n) = \beta^{n-1}u(n-1)$. Determine the output of the system. We have

$$\begin{aligned}
y(n) &= x(k) \star h(k) \\
&= \sum_{k=-\infty}^{\infty} x(k)h(n-k) \\
&= \sum_{k=-\infty}^{\infty} \alpha^{-k}u(-k)\beta^{n-k-1}u(n-k-1)
\end{aligned}$$

We know that $u(-k) = 1$ for $k \leq 0$ and $u(n-k-1) = 1$ for $n-k-1 \geq 0$ or $k \leq n-1$. Now we have to be very careful since our answers will depend on what we assume about n . For $n \geq 1$, the most restrictive condition is $k \leq 0$. Our new sum becomes

$$\begin{aligned}
y(n) &= \sum_{k=-\infty}^0 \alpha^{-k}\beta^{n-k-1} \\
&= \beta^{n-1} \sum_{k=-\infty}^0 (\alpha\beta)^{-k}
\end{aligned}$$

Letting $l = -k$ and assuming the sum is *absolutely convergent* so that we can rearrange the order of the summation, we have

$$y(n) = \beta^{n-1} \sum_{l=0}^{l=\infty} (\alpha\beta)^l = \beta^{n-1} \frac{1}{1 - \alpha\beta}$$

Next, let's look at what happens for $n \leq 0$. Here the more restrictive condition is $k \leq n-1$, and we have the sum

$$\begin{aligned}
y(n) &= \sum_{k=-\infty}^{n-1} \alpha^{-k}\beta^{n-k-1} \\
&= \beta^{n-1} \sum_{k=-\infty}^{n-1} (\alpha\beta)^{-k}
\end{aligned}$$

Letting $l = k - (n-1) = k - n + 1$ we have

$$y(n) = \beta^{n-1}(\alpha\beta)^{-(n-1)} \sum_{l=-\infty}^0 (\alpha\beta)^{-l}$$

Finally, let $p = -l$ and assume absolute convergence, and we have

$$y(n) = \alpha^{-n+1} \frac{1}{1 - \alpha\beta}$$

Combining our two regions we have

$$y(n) = \beta^{n-1} \frac{1}{1 - \alpha\beta} u(n-1) + \alpha^{-n+1} \frac{1}{1 - \alpha\beta} u(-n)$$

Note that there is no overlap in the definition regions (values of n).

16.4 z -Transforms

In continuous-time we can either compute the output of a system using the convolution directly, or we can resort to transform methods. In continuous-time control systems we generally utilize Laplace transforms and block diagrams, since this is a much easier way to determine the output than by using convolution in the time domain. Similarly, in discrete-time we can utilize the z -transform to avoid using the convolution sum. Just as with Laplace transforms, we can define one and two-sided z -transforms.

The *one-sided* z -transform is defined as

$$X(z) = \mathcal{Z}\{x(nT)\} = \sum_{k=0}^{\infty} x(kT)z^{-k}$$

and the *two-sided* z -transform is defined as

$$X(z) = \mathcal{Z}\{x(nT)\} = \sum_{k=-\infty}^{\infty} x(kT)z^{-k}$$

Just as with Laplace transforms, the only difference between the one and two-sided z -transform is the lower limit. Often we do not directly utilize the dependence on T and leave it out of the summation. In this course we are generally concerned with causal systems and causal inputs, so we generally will use the one-sided z -transform. Finally, the sum often only converges for values of $|z|$ within a specific region. This is called the *region of convergence* or *ROC*. Now let's do some examples.

Example. Determine the one-sided z -transform of $x(n) = \delta(n - l)$. We have

$$X(z) = \sum_{k=0}^{k=\infty} \delta(k - l)z^{-k} = z^{-l}$$

This sum converges for all values of $|z|$, so the region of convergence is the entire z -plane.

Example. Determine the one-sided z -transform of $x(n) = u(n)$, the unit step function. We have

$$X(z) = \sum_{k=0}^{\infty} u(k)z^{-k} = \sum_{k=0}^{\infty} \left(\frac{1}{z}\right)^k = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}$$

This is only true if the sum is absolutely convergent, which means $|\frac{1}{z}| < 1$ or if $|z| > 1$. Hence the ROC is defined by $|z| > 1$.

Example. Determine the one-sided z -transform of $x(n) = nu(n)$, the unit ramp function. We have

$$X(z) = \sum_{k=0}^{\infty} ku(k)z^{-k} = \sum_{k=0}^{\infty} kz^{-k}$$

At this point, we need to use something we all ready know, and some calculus. For the unit step we have

$$\frac{z}{z-1} = \sum_{k=0}^{\infty} z^{-k}$$

Taking derivatives of both sides with respect to z yields

$$\begin{aligned} \frac{d}{dz} \left[\frac{z}{z-1} \right] &= \frac{(z-1) - z}{(z-1)^2} = \frac{-1}{(z-1)^2} \\ \frac{d}{dz} \left[\sum_{k=0}^{\infty} z^{-k} \right] &= \sum_{k=0}^{\infty} -kz^{-k-1} = -z^{-1} \sum_{k=0}^{\infty} kz^{-k} \end{aligned}$$

Hence we have

$$\frac{-1}{(z-1)^2} = -z^{-1} \sum_{k=0}^{\infty} kz^{-k}$$

or

$$\sum_{k=0}^{\infty} kz^{-k} = \frac{z}{(1-z)^2}$$

which is the result we wanted. Just as with the z -transform of the unit step, the ROC for this function is $|z| > 1$.

Example. Determine the one-sided z -transform for the function $x(n) = a^n u(n)$. We have

$$\begin{aligned} X(z) &= \sum_{k=-\infty}^{\infty} a^k u(k) z^{-k} \\ &= \sum_{k=0}^{\infty} \left(\frac{a}{z} \right)^k \\ &= \frac{1}{1 - \frac{a}{z}} = \frac{z}{z-a} \end{aligned}$$

The region of convergence is given by $|\frac{a}{z}| < 1$ or $|z| > |a|$.

Example. Determine the z -transform of the $x(t) = e^{-at} u(t)$ when $x(t)$ is sampled with sampling interval T . The sampled version of the signal is given by $x(nT) = e^{-anT} u(nT) = (e^{-aT})^n u(n)$. Computing the z -transform we have

$$\begin{aligned} X(z) &= \sum_{k=-\infty}^{\infty} (e^{-aT})^k u(k) z^{-k} \\ &= \sum_{k=0}^{\infty} \left(\frac{1}{e^{aT} z} \right)^k \\ &= \frac{1}{1 - \frac{1}{e^{aT} z}} \\ &= \frac{z}{z - e^{-aT}} \end{aligned}$$

The region of convergence is given by $|\frac{1}{e^{aT}z}| < 1$ or $e^{-aT} < |z|$.

Example. Find the one-sided z -transform of $x(n) = a^n \cos(bnT)u(n)$. We have

$$X(z) = \sum_{k=-\infty}^{\infty} a^k \cos(bkT)u(k)z^{-k} = \sum_{k=0}^{\infty} \left(\frac{a}{z}\right)^k \cos(bkT)$$

At this point we need to use Euler's identity,

$$\cos(bkT) = \frac{e^{jbkT} + e^{-jbkT}}{2} = \frac{1}{2} (e^{jbT})^k + \frac{1}{2} (e^{-jbT})^k$$

Inserting this identity into our sum we have

$$\begin{aligned} X(z) &= \frac{1}{2} \sum_{k=0}^{\infty} \left(\frac{a}{z} e^{jbT}\right)^k + \frac{1}{2} \sum_{k=0}^{\infty} \left(\frac{a}{z} e^{-jbT}\right)^k \\ &= \frac{1}{2} \frac{1}{1 - \frac{ae^{jbT}}{z}} + \frac{1}{2} \frac{1}{1 - \frac{ae^{-jbT}}{z}} \end{aligned}$$

After some algebra we arrive at

$$X(z) = \frac{z(z - a \cos(bT))}{z^2 - 2a \cos(bT)z + a^2}$$

the region of convergence is given by

$$\left|\frac{ae^{jbT}}{z}\right| < 1, \quad \left|\frac{ae^{-jbT}}{z}\right| < 1$$

or $|z| > |a|$ since $|e^{jbT}| = |e^{-jbT}| = 1$.

16.5 z -Transform Properties

Just as with Laplace and Fourier transforms, there are a number of useful properties of z -transforms. We will cover a few of them in this section. Most of the properties we will cover are straightforward to prove, and you should be able to prove most of them. Unless otherwise stated, we will assume only one-sided z transforms.

Linearity. If we have the z -transform pairs $g(n) \leftrightarrow G(z)$ and $h(n) \leftrightarrow H(z)$, then for all constants a and b we have

$$\mathcal{Z} \{ag(n) + bh(n)\} = aG(z) + bH(z)$$

Multiplication by a^n . If $g(n) \leftrightarrow G(z)$, then $a^n g(n) \leftrightarrow G(\frac{z}{a})$.

Proof:

$$\begin{aligned} \mathcal{Z} \{a^n g(n)\} &= \sum_{k=0}^{\infty} a^k g(k) z^{-k} \\ &= \sum_{k=0}^{\infty} g(k) \left(\frac{z}{a}\right)^{-k} \\ &= G\left(\frac{z}{a}\right) \end{aligned}$$

Shifting. If $g(n) \leftrightarrow G(z)$ and $g(n)$ is causal, then

$$\begin{aligned}\mathcal{Z}\{g(n-l)\} &= z^{-l}G(z) \\ \mathcal{Z}\{g(n+l)\} &= z^l \left[G(z) - \sum_{k=0}^{l-1} g(k)z^{-k} \right]\end{aligned}$$

Proof of first part: Taking the z -transforms we have

$$\begin{aligned}\mathcal{Z}\{g(n-l)\} &= \sum_{k=0}^{\infty} g(k-l)z^{-k} \\ &= \sum_{k=l}^{\infty} g(k-l)z^{-k}\end{aligned}$$

since $g(n)$ is causal. Now let $m = k - l$, or $k = m + l$. The sum then becomes

$$\begin{aligned}\mathcal{Z}\{g(n-l)\} &= \sum_{m=0}^{\infty} g(m)z^{-(m+l)} \\ &= z^{-l} \sum_{m=0}^{\infty} g(m)z^{-m} \\ &= z^{-l}G(z)\end{aligned}$$

Proof of the second part: Let's start by trying to find the z -transform of $g(n+1)$. We have then

$$\begin{aligned}\mathcal{Z}\{g(n+1)\} &= \sum_{k=0}^{\infty} g(k+1)z^{-k} \\ &= g(1) + g(2)z^{-1} + g(3)z^{-2} + g(4)z^{-3} + g(5)z^{-4} + \dots\end{aligned}$$

Now we know

$$\begin{aligned}G(z) &= \sum_{k=0}^{\infty} g(k)z^{-k} \\ &= g(0) + g(1)z^{-1} + g(2)z^{-2} + g(3)z^{-3} + g(4)z^{-4} + g(5)z^{-5} + \dots\end{aligned}$$

Rearranging this expression we get

$$\begin{aligned}zG(z) &= g(0)z + g(1) + g(2)z^{-1} + g(3)z^{-2} + g(4)z^{-3} + g(5)z^{-4} + \dots \\ zG(z) - g(0) &= g(1) + g(2)z^{-1} + g(3)z^{-2} + g(4)z^{-3} + g(5)z^{-4} + \dots \\ &= \mathcal{Z}\{g(n+1)\}\end{aligned}$$

Now consider taking the z -transform of $g(n+2)$. We have

$$\begin{aligned}\mathcal{Z}\{g(n+2)\} &= \sum_{k=0}^{\infty} g(k+2)z^{-k} \\ &= g(2) + g(3)z^{-1} + g(4)z^{-2} + g(5)z^{-3} + \dots\end{aligned}$$

Starting from the definition of $G(z)$ from above

$$G(z) = g(0) + g(1)z^{-1} + g(2)z^{-2} + g(3)z^{-3} + g(4)z^{-4} + g(5)z^{-5} + \dots$$

we can find

$$\begin{aligned} z^2G(z) &= g(0)z^2 + g(1)z + g(2) + g(3)z^{-1} + g(4)z^{-2} + g(5)z^{-3} + \dots \\ z^2G(z) - z^2g(0) - zg(1) &= g(2) + g(3)z^{-1} + g(4)z^{-2} + g(5)z^{-3} + \dots \\ &= \mathcal{Z}\{g(n+2)\} \end{aligned}$$

In general we have

$$\begin{aligned} \mathcal{Z}\{g(n+l)\} &= \sum_{k=0}^{\infty} g(k+l)z^{-k} \\ &= g(l) + g(l+1)z^{-1} + g(l+2)z^{-2} + g(l+3)z^{-3} + \dots \end{aligned}$$

and

$$\begin{aligned} G(z) &= g(0) + g(1)z^{-1} + g(2)z^{-2} + \dots + g(l-1)z^{l-1} + g(l)z^{-l} + g(l+1)z^{-l-1} + g(l+2)z^{-l-2} + \dots \\ z^lG(z) &= z^lg(0) + z^{l-1}g(1) + z^{l-2}g(2) + \dots + z^{-1}g(l-1) + g(l) + g(l+1)z^{-1} + g(l+2)z^{-2} + \dots \end{aligned}$$

Rearranging we get

$$\begin{aligned} z^lG(z) - z^lg(0) - z^{l-1}g(1) + z^{l-2}g(2) + \dots + z^{-1}g(l-1) &= g(l) + g(l+1)z^{-1} + g(l+2)z^{-2} + \dots \\ z^l \left[G(z) - \sum_{k=0}^{l-1} g(k)z^{-k} \right] &= \mathcal{Z}\{g(n+l)\} \end{aligned}$$

Final Value Theorem. Just as with Laplace transforms, the Final Value Theorem is very useful for determining steady state errors in discrete-time control systems. If we assume $g(k)$ is causal, and all poles of $G(z)$ are inside the unit circle (with exception of a possible simple pole at $z = 1$), then

$$\lim_{t \rightarrow \infty} x(t) = \lim_{z \rightarrow 1} [1 - z^{-1}] G(z) = \lim_{z \rightarrow 1} \frac{z-1}{z} G(z)$$

Example. If $g(t) = (1 - e^{-3t})u(t)$, then $G(z) = \frac{z(1-e^{-3T})}{(z-1)(z-e^{-3T})}$. By the Final Value Theorem we have

$$\lim_{t \rightarrow \infty} g(t) = 1 = \lim_{z \rightarrow 1} \frac{z-1}{z} G(z) = \frac{1 - e^{-3T}}{1 - e^{-3T}} = 1$$

since all of the poles of the system are within the unit circle with the exception of a simple pole at $z = 0$.

Example. If $g(t) = e^{-t}u(t)$, then $G(z) = \frac{z}{z-e^{-T}}$. By the Final Value Theorem

$$\lim_{t \rightarrow \infty} g(t) = 0 = \lim_{z \rightarrow 1} \frac{z-1}{z} G(z) = 0$$

since all of the poles of the system are within the unit circle.

Example. If $g(t) = u(t)$, then $G(z) = \frac{z}{z-1}$. By the Final Value Theorem

$$\lim_{t \rightarrow \infty} g(t) = 1 = \lim_{z \rightarrow 1} \frac{z-1}{z} G(z) = 1$$

Note that in this example there is a simple pole at $z = 1$.

Initial Value Theorem. Just as with laplace transforms, the Initial Value Theorem for z -transforms is very useful for determining the initial required control effort for different types of controllers. The Initial Value Theorem states that if the limits exist, then

$$\lim_{t \rightarrow 0} g(t) = \lim_{z \rightarrow \infty} G(z)$$

Example. If $g(t) = (1 - e^{-3t})u(t)$, then $G(z) = \frac{z(1-e^{-3T})}{(z-1)(z-e^{-3T})}$. By the Initial Value Theorem we have

$$\lim_{t \rightarrow 0} g(t) = 0 = \lim_{z \rightarrow \infty} G(z) = 0$$

Example. If $g(t) = e^{-t}u(t)$, then $G(z) = \frac{z}{z-e^{-T}}$. By the Initial Value Theorem

$$\lim_{t \rightarrow 0} g(t) = 1 = \lim_{z \rightarrow \infty} G(z) = 1$$

Table 2 summarizes some common z -transform pairs, while Table 3 summarizes some common z -transform properties.

16.6 Inverse z -Transforms

There are four common methods for going from the z -domain back to the (discrete) time domain. These are

- Computational methods (Matlab/Maple, etc)
- Contour integration in the complex plane
- Partial fractions
- Long division

In this course we will emphasize the use of partial fractions, which you should be pretty familiar with. It is generally easiest to find the partial fraction expansion for $\frac{G(z)}{z}$ when you want to find the inverse z -transform of $G(z)$, since simple poles have the form $a^k u(k) \leftrightarrow \frac{z}{z-a}$.

Example. Determine $g(n)$ for $G(z) = \frac{z(z-4)}{(z-3)(z-2)}$. We have

$$\frac{G(z)}{z} = \frac{z-4}{(z-3)(z-2)} = \frac{A}{z-3} + \frac{B}{z-2}$$

$X(s)$	$x(t)$	$x(nT)$ or $x(n)$	$X(z)$
—	—	$\delta(n)$	1
—	—	$\delta(n-l)$	z^{-l}
$\frac{1}{s}$	$u(t)$	$u(nT)$	$\frac{z}{z-1}$
$\frac{1}{s+a}$	$e^{-at}u(t)$	$e^{-anT}u(nT)$	$\frac{z}{z-e^{-aT}}$
$\frac{1}{s^2}$	$tu(t)$	$(nT)u(nT)$	$\frac{Tz}{(z-1)^2}$
$\frac{1}{s^3}$	$\frac{1}{2}t^2u(t)$	$\frac{1}{2}(nT)^2u(nT)$	$\frac{T^2z(z+1)}{(z-1)^3}$
$\frac{a}{s(s+a)}$	$(1-e^{-at})u(t)$	$(1-e^{-anT})u(nT)$	$\frac{z(1-e^{-aT})}{(z-1)(z-e^{-aT})}$
$\frac{b-a}{(s+a)(s+b)}$	$(e^{-at}-e^{-bt})u(t)$	$(e^{-anT}-e^{-bnT})u(nT)$	$\frac{z(e^{-aT}-e^{-bT})}{(z-e^{-aT})(z-e^{-bT})}$
$\frac{1}{(s+a)^2}$	$te^{-at}u(t)$	$(nT)e^{-anT}u(nT)$	$\frac{zTe^{-aT}}{(z-e^{-aT})^2}$
$\frac{s}{(s+a)^2}$	$(1-at)e^{-at}u(t)$	$(1-anT)e^{-anT}u(nT)$	$\frac{z[z-(1+aT)e^{-aT}]}{(z-e^{-aT})^2}$
$\frac{\omega}{s^2+\omega^2}$	$\sin(\omega t)u(t)$	$\sin(\omega nT)u(nT)$	$\frac{z \sin(\omega T)}{z^2-2z \cos(\omega T)+1}$
$\frac{s}{s^2+\omega^2}$	$\cos(\omega t)u(t)$	$\cos(\omega nT)u(nT)$	$\frac{z[z-\cos(\omega T)]}{z^2-2z \cos(\omega T)+1}$
		$b^n u(n)$	$\frac{z}{z-b}$
		$nb^n u(n)$	$\frac{bz}{(z-b)^2}$
		$(1-b^n)u(n)$	$\frac{z(1-b)}{(z-1)(z-b)}$

Table 2: Common (one sided) z -transforms. Often we set $T = 1$ and $e^{-aT} = b$ to find results when the sampling time is not important.

$x(t)$	$x(nT)$	$\mathcal{Z}[x(t)]$ or $\mathcal{Z}[x(nT)]$
$\alpha x(t)$	$\alpha x(nT)$	$\alpha X(z)$
$\alpha x_1(t) + \beta x_2(t)$	$\alpha x_1(nT) + \beta x_2(nT)$	$\alpha X_1(z) + \beta X_2(z)$
$x(t+T)$	$x(nT+T)$	$zX(z) - zx(0)$
$x(t+2T)$	$x(nT+2T)$	$z^2X(z) - z^2x(0) - zx(T)$
$x(t+kT)$	$x(nT+kT)$	$z^kX(z) - z^kx(0) - z^{k-1}x(T) - z^{k-2}x(2T) - \dots - zx(kT-T)$
$x(t-kT)$	$x(nT-kT)$	$z^{-k}X(z)$
$tx(t)$	$(nT)x(nT)$	$-Tz \frac{d}{dz}X(z)$
$e^{-at}x(t)$	$e^{-anT}x(nT)$	$X(ze^{aT})$
—	$a^n x(nT)$	$X\left(\frac{z}{a}\right)$
$x(0)$	$x(0)$	$\lim_{z \rightarrow \infty} X(z)$ if the limit exists
$x(\infty)$	$x(\infty)$	$\lim_{z \rightarrow 1} [(1-z^{-1})X(z)]$ if all poles of $X(z)$ are inside unit circle (a simple pole at $z=1$ is allowed)
—	$\sum_{k=0}^n x(k)$	$\frac{z}{z-1}X(z)$
$x(t) \star h(t)$	$x(nT) \star h(nT)$	$X(z)H(z)$

Table 3: Common properties of (one sided) z -transforms. Often we set $T = 1$ and $e^{-aT} = b$ to find results when the sampling time is not important

Using the "cover up" method we quickly arrive at

$$A = \frac{3-4}{3-2} = -1 \quad B = \frac{2-4}{2-3} = 2$$

This leads to

$$\frac{G(z)}{z} = \frac{-1}{z-3} + \frac{2}{z-2}$$

or

$$G(z) = -\frac{z}{z-3} + 2\frac{z}{z-2}$$

From this form we can quickly conclude that

$$g(n) = -(3)^n u(n) + 2(2)^n u(n)$$

The first few terms of this impulse response are $g(0) = 1$, $g(1) = 1$, $g(2) = -1$, and $g(3) = -11$.

Example. Determine $g(n)$ if $G(z) = \frac{z-4}{(z-3)(z-2)}$. For this $G(z)$, there is no z in the numerator to divide by to create the $\frac{G(z)}{z}$. One solution is to multiply by $\frac{z}{z}$ and then do the partial fractions, as follows:

$$G(z) = \frac{z-4}{(z-3)(z-2)} = \frac{z-4}{(z-3)(z-2)} \frac{z}{z}$$

so we have

$$\frac{G(z)}{z} = \frac{z-4}{z(z-3)(z-2)} = \frac{A}{z} + \frac{B}{z-3} + \frac{C}{z-2}$$

Using the "cover up" method we get

$$A = \frac{-4}{(-3)(-2)} = \frac{-2}{3}, \quad B = \frac{3-4}{(3-2)(3)} = \frac{-1}{3}, \quad C = \frac{2-4}{(2-3)(2)} = 1$$

so

$$G(z) = -\frac{2}{3} - \frac{1}{3} \frac{z}{z-3} + \frac{z}{z-2}$$

which yields

$$g(n) = -\frac{2}{3}\delta(n) - \frac{1}{3}(3)^n u(n) + (2)^n u(n)$$

If we compute the first few terms of $g(n)$ we get $g(0) = 0$, $g(1) = 1$, $g(2) = 1$, $g(3) = -1$, $g(4) = -11$.

Example. Determine $g(n)$ if $G(z) = \frac{z-4}{(z-3)(z-2)}$. For this $G(z)$, there is no z in the numerator to divide by to create the $\frac{G(z)}{z}$. One solution is to construct and $F(z)$ so that $G(z) = z^{-1}F(z)$, as follows:

$$F(z) = \frac{z(z-4)}{(z-3)(z-2)}$$

so we have

$$\frac{F(z)}{z} = \frac{z-4}{(z-3)(z-2)} = \frac{A}{z-3} + \frac{B}{z-2}$$

Using the "cover up" method we get

$$F(z) = -\frac{z}{z-3} + 2\frac{z}{z-2}$$

and

$$f(n) = -(3)^n u(n) + 2(2)^n u(n)$$

Since we know $G(z) = z^{-1}F(z)$, we have $g(n) = f(n-1)$ using the delay property. Hence

$$g(n) = -(3)^{n-1} u(n-1) + 2(2)^{n-1} u(n-1)$$

Although this answer looks much different than in the previous example, the two solutions are in fact the same.

Example. Determine $g(n)$ for $G(z) = \frac{z(-5z+22)}{(z+1)(z-2)^2}$. We have

$$\frac{G(z)}{z} = \frac{-5z+22}{(z+1)(z-2)^2} = \frac{A}{z+1} + \frac{B}{z-2} + \frac{C}{(z-2)^2}$$

Using the "cover-up" method we can quickly find A and C ,

$$A = \frac{-5(-1)+22}{(-1-2)^2} = \frac{27}{9} = 3, \quad C = \frac{-5(2)+22}{(2+1)} = \frac{12}{3} = 4$$

To get B , we multiply both sides of the original equation by z and then let $z \rightarrow \infty$, to get

$$0 = A + B, \quad B = -3$$

So we have

$$G(z) = 3\frac{z}{z+1} - 3\frac{z}{z-2} + 4\frac{z}{(z-2)^2}$$

and

$$g(n) = 3(-1)^n u(n) - 3(2)^n u(n) + 4n(2)^{n-1} u(n)$$

The first few terms of the impulse response are $g(0) = 0$, $g(1) = -5$, $g(2) = 7$, $g(3) = 21$, and $g(4) = 83$.

While we can normally use partial fractions to determine the time response for all sample times, sometimes we only want to determine the time response for the first few terms. We can use the method of long division to determine the first few terms of the time domain response by utilizing the fact that if we compute the z -transform of $g(n)$ we have

$$G(z) = g(0) + g(1)z^{-1} + g(2)z^{-2} + g(3)z^{-3} + g(4)z^{-4} + \dots$$

This method of long division is particularly important as a method to check your answer. However, it is not very useful for determining a closed form solution for all time.

Example. Use long division to determine the first few terms of the impulse response for the transfer function $G(z) = \frac{z(z-4)}{(z-3)(z-2)}$. We first need to multiply out the numerator and the denominator, so we have $G(z) = \frac{z^2-4z}{z^2-5z+6}$. Next we do the long division,

$$\begin{array}{r}
z^2 - 5z + 6 \overline{) \begin{array}{r} 1 \quad +z^{-1} \quad -z^{-2} - 11z^{-3} \\ z^2 \quad -4z \\ \hline z^2 \quad -5z \quad +6 \\ \hline z \quad -6 \\ z \quad -5 \quad +6z^{-1} \\ \hline -1 \quad -6z^{-1} \\ -1 \quad +5z^{-1} \quad -6z^{-2} \\ \hline -11z^{-1} \quad +6z^{-2} \end{array}
\end{array}$$

Hence we have the approximations

$$\begin{aligned}
G(z) &= \frac{z(z-4)}{z^2-5z+6} = 1 + z^{-1} - z^{-2} - 11z^{-3} + \dots \\
g(n) &= \delta(n) + \delta(n-1) - \delta(n-2) - 11\delta(n-3) + \dots
\end{aligned}$$

which agrees with our previous result.

Example. Use long division to determine the first few terms of the impulse response for the transfer function $G(z) = \frac{z-4}{(z-3)(z-2)}$. We first need to multiply out the denominator, so we have $G(z) = \frac{z-4}{z^2-5z+6}$. Next we do the long division,

$$\begin{array}{r}
z^2 - 5z + 6 \overline{) \begin{array}{r} z^{-1} \quad +z^{-2} \quad -z^{-3} - 11z^{-4} \\ z \quad -4 \\ \hline z \quad -5 \quad +6z^{-1} \\ \hline 1 \quad -6z^{-1} \\ 1 \quad -5z^{-1} \quad +6z^{-2} \\ \hline -z^{-1} \quad -6z^{-2} \\ -z^{-1} \quad +5z^{-2} \quad -6z^{-3} \\ \hline -11z^{-2} \quad +6z^{-3} \end{array}
\end{array}$$

Hence we have the approximations

$$\begin{aligned}
G(z) &= \frac{z-4}{z^2-5z+6} = z^{-1} + z^{-2} - z^{-3} - 11z^{-4} + \dots \\
g(n) &= \delta(n-1) + \delta(n-2) - \delta(n-3) - 11\delta(n-4) + \dots
\end{aligned}$$

which agrees with our previous result.

Example. Use long division to determine the first few terms of the impulse response for the transfer function $G(z) = \frac{z(-5z+22)}{(z+1)(z-2)^2}$. We first need to multiply out the numerator and denominator, so we have $G(z) = \frac{-5z^2+22z}{z^3-3z^2+4}$. Next we do the long division,

$$\begin{array}{r}
z^3 - 3z^2 + 4 \mid \begin{array}{r}
-5z^{-1} \quad +7z^{-2} \quad +21z^{-3} + 83z^{-4} \\
\hline
-5z^2 \quad +22z \\
-5z^2 \quad +15z \qquad \qquad \qquad -20z^{-1} \\
\hline
7z \qquad \qquad \qquad \qquad \qquad \qquad +20z^{-1} \\
7z \qquad \qquad \qquad \qquad \qquad \qquad -21 \qquad \qquad \qquad +28z^{-2} \\
\hline
\qquad \qquad \qquad \qquad \qquad \qquad 21 \quad +20z^{-1} \quad -28z^{-2} \\
\qquad \qquad \qquad \qquad \qquad \qquad 21 \quad -63z^{-1} \qquad \qquad \qquad +84z^{-3} \\
\hline
\qquad \qquad \qquad \qquad \qquad \qquad 83z^{-1} \quad -28z^{-3} \quad -84z^{-3}
\end{array}
\end{array}$$

Hence we have the approximations

$$\begin{aligned}
G(z) &= \frac{z(-5z + 22)}{z^3 - 3z^2 + 4} = -5z^{-1} + 7z^{-2} + 21z^{-3} + 83z^{-4} + \dots \\
g(n) &= -5\delta(n-1) + 7\delta(n-2) + 21\delta(n-3) + 83\delta(n-4) + \dots
\end{aligned}$$

which agrees with our previous result.

16.7 Second Order Transfer Functions with Complex Conjugate Poles

Consider the continuous-time impulse response response that corresponds to a transfer function with complex conjugate poles at $-a \pm jb$,

$$G(s) = \frac{r}{(s+a)^2 + b^2} \leftrightarrow g(t) = re^{-at} \cos(bt + \theta)u(t)$$

If we look at samples of the impulse response, we get

$$\begin{aligned}
g(nT) &= re^{-anT} \cos(bnT + \theta)u(nT) \\
&= r \left(e^{-aT} \right)^n \cos((bT)n + \theta)u(nT) \\
&= r\gamma^n \cos(\beta n + \theta)u(n)
\end{aligned}$$

Hence, the discrete-time impulse response that corresponds to a continuous-time system with complex conjugate poles is of the form

$$g(n) = r\gamma^n \cos(\beta n + \theta)u(n)$$

As you might expect, this form shows up a great deal when we are modeling many types of systems. Hence we should know the corresponding z -transform of this impulse response. As you will see, it is much more complex than you might expect.

Using Euler's identity we can write $g(n)$ as

$$g(n) = \frac{r}{2}\gamma^n e^{j\beta n} e^{j\theta} + \frac{r}{2}\gamma^n e^{-j\beta n} e^{-j\theta}$$

and taking the z -transform of this we have

$$\begin{aligned}
G(z) &= \frac{r}{2} \sum_{k=0}^{\infty} \gamma^k e^{j\beta k} e^{j\theta} z^{-k} + \frac{r}{2} \sum_{k=0}^{\infty} \gamma^k e^{-j\beta k} e^{-j\theta} z^{-k} \\
&= \frac{r}{2} e^{j\theta} \sum_{k=0}^{\infty} \left(\frac{\gamma e^{j\beta}}{z} \right)^k + \frac{r}{2} e^{-j\theta} \sum_{k=0}^{\infty} \left(\frac{\gamma e^{-j\beta}}{z} \right)^k \\
&= \frac{r e^{j\theta}}{2} \left[\frac{1}{1 - \gamma e^{j\beta} z^{-1}} \right] + \frac{r e^{-j\theta}}{2} \left[\frac{1}{1 - \gamma e^{-j\beta} z^{-1}} \right] \\
&= \frac{r e^{j\theta}}{2} \left[\frac{z}{z - \gamma e^{j\beta}} \right] + \frac{r e^{-j\theta}}{2} \left[\frac{z}{z - \gamma e^{-j\beta}} \right] \\
&= \frac{r e^{j\theta}}{2} \left[\frac{z(z - \gamma e^{-j\beta})}{(z - \gamma e^{j\beta})(z - \gamma e^{-j\beta})} \right] + \frac{r e^{-j\theta}}{2} \left[\frac{z(z - \gamma e^{j\beta})}{(z - \gamma e^{j\beta})(z - \gamma e^{-j\beta})} \right] \\
&= \frac{r}{2} \left[\frac{z^2 e^{j\theta} - z \gamma e^{-j\beta} e^{j\theta} + z^2 e^{-j\theta} - z \gamma e^{j\beta} e^{-j\theta}}{z^2 - \gamma z e^{j\beta} - \gamma z e^{-j\beta} + \gamma^2} \right] \\
&= \frac{r}{2} \left[\frac{z^2 (e^{j\theta} + e^{-j\theta}) - z \gamma (e^{j(\beta-\theta)} + e^{-(\beta-\theta)})}{z^2 - \gamma z (e^{j\beta} + e^{-j\beta}) + \gamma^2} \right] \\
&= r \left[\frac{z^2 \cos(\theta) - z \gamma \cos(\beta - \theta)}{z^2 - 2\gamma z \cos(\beta) + \gamma^2} \right]
\end{aligned}$$

Unfortunately, what we really need is a form that we can (easily?) use in doing partial fractions. This is going to take a bit more work. Let's write

$$G(z) = r \left[\frac{z^2 \cos(\theta) - z \gamma \cos(\beta - \theta)}{z^2 - 2\gamma z \cos(\beta) + \gamma^2} \right] = \frac{Az^2 + Bz}{z^2 + 2Cz + \gamma^2}$$

This second form is much easier to use when doing partial fractions, but we need to be able to relate the parameters A , B , and C , to the parameters r , θ , and β . Clearly we have already identified the parameter γ .

Let's start with the denominator. Equating powers of z in the denominator we get

$$-2C = 2\gamma \cos(\beta)$$

which means

$$\beta = \cos^{-1} \left(\frac{-C}{\gamma} \right)$$

Next we identify

$$\begin{aligned}
A &= r \cos(\theta) \\
B &= -r \gamma \cos(\beta - \theta)
\end{aligned}$$

Expanding the cosine expression out we get

$$\begin{aligned}
B &= -r \gamma \cos(\beta - \theta) \\
&= -r \gamma \cos(\beta) \cos(\theta) - r \gamma \sin(\beta) \sin(\theta) \\
&= -\gamma \cos(\beta) (r \cos(\theta)) - r \gamma \sin(\beta) \sin(\theta) \\
&= CA - r \gamma \sin(\beta) \sin(\theta)
\end{aligned}$$

But

$$\begin{aligned}\frac{-C}{\gamma} &= \cos(\beta) \\ \sin(\beta) &= \sqrt{1 - \cos^2(\beta)} = \sqrt{1 - \frac{C^2}{\gamma^2}} = \frac{1}{\gamma}\sqrt{\gamma^2 - C^2}\end{aligned}$$

So

$$B = CA - \gamma \left(\frac{1}{\gamma} \sqrt{\gamma^2 - C^2} \right) r \sin(\theta)$$

or

$$r \sin(\theta) = \frac{CA - B}{\sqrt{\gamma^2 - C^2}}$$

Recall that we also have $r \cos(\theta) = A$, so dividing these we get

$$\tan(\theta) = \frac{\sin(\theta)}{\cos(\theta)} = \frac{CA - B}{A\sqrt{\gamma^2 - C^2}}$$

or

$$\theta = \tan^{-1} \left(\frac{CA - B}{A\sqrt{\gamma^2 - C^2}} \right)$$

Finally we have

$$\begin{aligned}\sqrt{(r \sin(\theta))^2 + (r \cos(\theta))^2} &= r = \sqrt{\left(\frac{CA - B}{\sqrt{\gamma^2 - C^2}} \right)^2 + A^2} \\ &= \sqrt{\frac{A^2 C^2 - 2ACB + B^2 + A^2 \gamma^2 - C^2 A^2}{\gamma^2 - C^2}}\end{aligned}$$

or

$$r = \sqrt{\frac{A^2 \gamma^2 + B^2 - 2ABC}{\gamma^2 - C^2}}$$

In summary we have

$$\begin{aligned}g(n) = r\gamma^n \cos(\beta n + \theta)u(n) &\leftrightarrow G(z) = \frac{Az^2 + Bz}{z^2 + 2Cz + \gamma^2} \\ r &= \sqrt{\frac{A^2 \gamma^2 + B^2 - 2ABC}{\gamma^2 - C^2}} \\ \theta &= \tan^{-1} \left(\frac{CA - B}{A\sqrt{\gamma^2 - C^2}} \right) \\ \beta &= \cos^{-1} \left(\frac{-C}{\gamma} \right)\end{aligned}$$

Now we have all of the parameters we need to solve some examples.

Example. Determine the impulse and unit step responses of the system with transfer function

$$G(z) = \frac{-0.535z}{z^2 - 1.597z + 0.671}$$

To determine the impulse response, we first equate this transfer function with our standard form,

$$G(z) = \frac{-0.535z}{z^2 - 1.597z + 0.671} = \frac{Az^2 + Bz}{z^2 + 2Cz + \gamma^2}$$

From this we can immediately determine $A = 0$, $B = -0.535$, $\gamma = \sqrt{0.671} = 0.819$, $-2C = 1.597$ so $C = -0.7895$. Now we compute the remaining parameters,

$$\begin{aligned}\beta &= \cos^{-1}\left(\frac{-C}{\gamma}\right) = \cos^{-1}\left(\frac{0.7985}{0.819}\right) = 0.224 \\ \theta &= \tan^{-1}\left(\frac{CA - B}{A\sqrt{\gamma^2 - C}}\right) = \tan^{-1}\left(\frac{0.535}{0}\right) = \frac{\pi}{2} = 1.570 \\ r &= \sqrt{\frac{A^2\gamma^2 + B^2 - 2ABC}{\gamma^2 - C^2}} = \sqrt{\frac{0.2862}{0.03340}} = 2.9274\end{aligned}$$

Hence the impulse response of the system is

$$g(n) = 2.927(0.819^n) \cos(0.224n + 1.570)u(n)$$

The initial value of this system is $g(0) = 2.450 \cos(1.570) = 0$, so the system is initially at rest.

To determine the unit step response, we have

$$\begin{aligned}Y(z) &= G(z) \frac{z}{z-1} \\ \frac{Y(z)}{z} &= G(z) \frac{1}{z-1} = \frac{-0.535z}{(z^2 - 1.597z + 0.671)(z-1)} = \frac{Az + B}{z^2 - 1.597z + 0.671} + \frac{D}{z-1}\end{aligned}$$

Using the cover-up method we can easily determine that $D = -7.230$. Multiplying both sides of the equation by z and letting $z \rightarrow \infty$, we get $A = -D = 7.230$. Finally, setting $z = 0$ we get $B = -4.851$. Now we need to recompute our parameters

$$\begin{aligned}\beta &= \cos^{-1}\left(\frac{-C}{\gamma}\right) = 0.224 \\ \theta &= \tan^{-1}\left(\frac{CA - B}{A\sqrt{\gamma^2 - C}}\right) = -0.6110 \\ r &= \sqrt{\frac{A^2\gamma^2 + B^2 - 2ABC}{\gamma^2 - C^2}} = 8.827\end{aligned}$$

The step response is then

$$y(n) = 8.827(0.819^n) \cos(0.224n - 0.6110)u(n) - 7.230u(n)$$

16.8 Solving Difference Equations

Just as we can use Laplace transforms to solve differential equations with initial conditions, in discrete-time we can use z -transforms to solve difference equations with initial conditions. The general idea is to take the z -transform of each term of the difference equation, then solve. For the low order difference equations we will be solving it is useful to summarize the following identities again:

$$\begin{aligned}\mathcal{Z}\{x(k+2)\} &= z^2X(z) - z^2x(0) - zx(1) \\ \mathcal{Z}\{x(k+1)\} &= zX(z) - zx(0) \\ \mathcal{Z}\{x(k)\} &= X(z) \\ \mathcal{Z}\{x(k-1)\} &= z^{-1}X(z) \\ \mathcal{Z}\{x(k-2)\} &= z^{-2}X(z)\end{aligned}$$

It is important to remember that the lower case letters represent the time-domain and the capital letters represent the z -domain. After we have solved the difference equation, it is a good idea to check our answer by using the difference equation to compute the answer and compare this with the answer we get using the z -transform. Finally, sometimes we break the solution into two different parts: the *Zero Input Response (ZIR)* is the response of the system due to initial conditions alone (there is no, or zero input), and the *Zero State Response (ZSR)* is the response of the system due to the input only, assuming all initial conditions (all initial states) are zero. The system transfer function is determined from the ZSR.

Example. Find the ZIR, ZSR, and solve the difference equation

$$y(n+2) - 5y(n+1) + 6y(n) = x(n)$$

with initial conditions $y(0) = 4$ and $y(1) = 3$ and input $x(n) = u(n)$, the unit step function. Taking z -transforms of each term, we have

$$\left[z^2Y(z) - z^2y(0) - zy(1)\right] - 5[zY(z) - zy(0)] + 6Y(z) = X(z)$$

Note that we have not simplified the difference equation yet. Rearranging this equation we get

$$\left[z^2 - 5z + 6\right]Y(z) = X(z) + \left[z^2y(0) + zy(1) - 5zy(0)\right]$$

or

$$Y(z) = \underbrace{\frac{X(z)}{(z-3)(z-2)}}_{ZSR} + \underbrace{\frac{z^2y(0) + zy(1) - 5zy(0)}{(z-3)(z-2)}}_{ZIR}$$

At this point we can determine the transfer function from the ZSR and corresponding impulse response,

$$H(z) = \frac{1}{(z-3)(z-2)}, \quad h(n) = 3^{n-1}u(n-1) - 2^{n-1}u(n-1) = \frac{1}{6}\delta(n) + \frac{1}{3}3^n u(n) - \frac{1}{2}2^n u(n)$$

The ZSR is then given by

$$Y_{ZSR} = \frac{X(z)}{(z-3)(z-2)} = \frac{z}{(z-1)(z-2)(z-3)} = \frac{1}{2} \frac{z}{z-1} - \frac{z}{z-2} + \frac{1}{2} \frac{z}{z-3}$$

or

$$y_{ZSR}(n) = \frac{1}{2}u(n) - 2^n u(n) + \frac{1}{2}3^n u(n)$$

The ZIR is given by

$$Y(z) = \frac{z^2 y(0) + z y(1) - 5z y(0)}{(z-3)(z-2)} = \frac{4z^2 - 17z}{(z-3)(z-2)} = -5 \frac{z}{z-3} + 9 \frac{z}{z-2}$$

or

$$y_{ZIR}(n) = -5(3^n)u(n) + 9(2^n)u(n)$$

Finally the total solution is given by the sum of the ZIR and ZSR,

$$\begin{aligned} y(n) &= y_{ZIR}(n) + y_{ZSR}(n) \\ &= [-5(3^n)u(n) + 9(2^n)u(n)] + \left[\frac{1}{2}u(n) - 2^n u(n) + \frac{1}{2}3^n u(n) \right] \\ &= \left[\left(-5 + \frac{1}{2}\right)3^n + (9-1)2^n + \frac{1}{2} \right] u(n) \\ &= \left[-4.5(3^n) + 8(2^n) + \frac{1}{2} \right] u(n) \end{aligned}$$

Now to check our answer, we know that $y(0) = 4$ and $y(1) = 3$. To generate the other values of $y(n)$, let's rewrite the difference equation as

$$y(n+2) = 5y(n+1) - 6y(n) + x(n)$$

and remember that $x(n) = u(n) = 1$ for $n \geq 0$. Now we can make a table,

<i>DE Prediction</i>	<i>Closed Form</i>
$y(0) = 4$	$y(0) = -4.5(3^0) + 8(2^0) + 0.5 = 4$
$y(1) = 3$	$y(1) = -4.5(3) + 8(2) + 0.5 = 3$
$y(2) = 5y(1) - 6y(0) + 1 = 5(3) - 6(4) + 1 = -8$	$y(2) = -4.5(3^2) + 8(2^2) + 0.5 = -8$
$y(3) = 5y(2) - 6y(1) + 1 = 5(-8) - 6(3) + 1 = -57$	$y(3) = -4.5(3^3) + 8(2^3) + 0.5 = -57$
$y(4) = 5y(3) - 6y(2) + 1 = 5(-57) - 6(-8) + 1 = -236$	$y(4) = -4.5(3^4) + 8(2^4) + 0.5 = -236$

Example. Find the ZIR, ZSR, and solve the difference equation

$$x(n+2) - x(n+1) + 0.25x(n) = f(n-1) - f(n)$$

with initial conditions $x(0) = 0$ and $x(1) = 1$ and input $f(n) = u(n)$, the unit step function. *Note that we must write the difference equation so that the input does not include initial conditions!* We assume all *initial conditions* are associated with the states of the system, in this case the $x(n)$. Taking z -transforms of each term, we have

$$\left[z^2 X(z) - z^2 x(0) - z x(1) \right] - [zX(z) - z x(0)] + 0.25X(z) = z^{-1}F(z) - F(z)$$

Note that we have not simplified the difference equation yet. Rearranging this equation we get

$$\left[z^2 - z + \frac{1}{4} \right] X(z) = [z^{-1} - 1] F(z) + [z^2 x(0) + z x(1) - z x(0)]$$

or

$$X(z) = \underbrace{F(z) \frac{z^{-1} - 1}{(z - \frac{1}{2})^2}}_{ZSR} + \underbrace{\frac{z^2 x(0) + zx(1) - zx(0)}{(z - \frac{1}{2})^2}}_{ZIR}$$

At this point we can determine the transfer function from the ZSR and corresponding impulse response,

$$H(z) = \frac{z^{-1} - 1}{(z - \frac{1}{2})^2} = \frac{1 - z}{z(z - \frac{1}{2})^2}, \quad h(n) = 4\delta(n - 1) - 4\left(\frac{1}{2}\right)^{n-1}u(n - 1) + (n - 1)\left(\frac{1}{2}\right)^{n-2}u(n - 2)$$

The ZSR is then given by

$$X_{ZSR} = \frac{X(z)(z^{-1} - 1)}{(z - \frac{1}{2})^2} = \frac{z(z^{-1} - 1)}{(z - \frac{1}{2})^2} = -\frac{1}{(z - \frac{1}{2})^2} = -(n - 1)\left(\frac{1}{2}\right)^{n-2}u(n - 1)$$

The ZIR is given by

$$X(z) = \frac{z^2 x(0) + zx(1) - zx(0)}{(z - \frac{1}{2})^2} = \frac{z}{(z - \frac{1}{2})^2} = n\left(\frac{1}{2}\right)^{n-1}u(n)$$

Finally the total solution is given by the sum of the ZIR and ZSR,

$$\begin{aligned} y(n) &= y_{ZIR}(n) + y_{ZSR}(n) \\ &= \left[n\left(\frac{1}{2}\right)^{n-1}u(n) \right] - \left[(n - 1)\left(\frac{1}{2}\right)^{n-2}u(n - 1) \right] \\ &= \left(\frac{1}{2}\right)^{n-2} \left[1 - \frac{n}{2} \right] u(n - 2) + \delta(n - 1) \end{aligned}$$

Now to check our answer, we know that $x(0) = 0$ and $x(1) = 1$. To generate the other values of $x(n)$, let's rewrite the difference equation as

$$x(n + 2) = x(n + 1) - 0.25x(n) + f(n - 1) - f(n)$$

and remember that $f(n) = u(n) = 1$ for $n \geq 0$. Now we can make a table,

<i>DE Prediction</i>	<i>Closed Form</i>
$x(0) = 0$	$x(0) = 0$
$x(1) = 1$	$x(1) = 1$
$x(2) = x(1) - \frac{1}{4}x(0) + 0 - 1 = 1 - 0 + 0 - 1 = 0$	$x(2) = \left(\frac{1}{2}\right)^0 \left[1 - \frac{2}{2} \right] = 0$
$x(3) = x(2) - \frac{1}{4}x(1) + 1 - 1 = -\frac{1}{4}$	$x(3) = \left(\frac{1}{2}\right)^1 \left[1 - \frac{3}{2} \right] = -\frac{1}{4}$
$x(4) = x(3) - \frac{1}{4}x(2) + 1 - 1 = -\frac{1}{4}$	$x(4) = \left(\frac{1}{2}\right)^2 \left[1 - \frac{4}{2} \right] = -\frac{1}{4}$
$x(5) = x(4) - \frac{1}{4}x(3) + 1 - 1 = -\frac{1}{4} + \left(\frac{1}{4}\right)^2 = -3\left(\frac{1}{4}\right)^2$	$x(5) = \left(\frac{1}{2}\right)^3 \left[1 - \frac{5}{2} \right] = -3\left(\frac{1}{2}\right)^4$

16.9 Asymptotic Stability

In continuous-time, asymptotic stability is based on the behavior of the impulse response as $t \rightarrow \infty$. There are three cases:

- If $|h(t)| \rightarrow \infty$ as $t \rightarrow \infty$, then the system is *unstable*. This means at least one pole of the system is in the (open) right half plane.
- If $|h(t)| \rightarrow 0$ as $t \rightarrow \infty$, then the system is *stable*. This means all of the poles of the system are in the (open) left half plane.
- If $|h(t)| \leq M$ for some constant M as $t \rightarrow \infty$, then the system is *marginally stable*. This means there are some isolated poles on the $j\omega$ axis and, if there are poles not on the $j\omega$ axis these poles are in the left half plane.

We can easily modify these criteria to discrete-time, and look at the behavior of the impulse response as $k \rightarrow \infty$:

- If $|h(k)| \rightarrow \infty$ as $k \rightarrow \infty$, then the system is *unstable*. This means at least one pole of the system is outside the unit circle (at least one pole has a magnitude larger than one.)
- If $|h(k)| \rightarrow 0$ as $k \rightarrow \infty$, then the system is *stable*. This means all of the poles of the system are inside the unit circle (all poles have a magnitude less than one).
- If $|h(k)| \leq M$ for some constant M as $k \rightarrow \infty$, then the system is *marginally stable*. This means there are some isolated poles on the unit circle and, if there are poles not on the unit circle these poles are inside the unit circle.

16.10 Mapping Poles and Settling Time

Let's assume we have the continuous-time transfer function

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

with corresponding impulse response

$$h(t) = \frac{1}{b-a}e^{-at}u(t) + \frac{1}{a-b}e^{-bt}u(t)$$

Now we will assume we sample this transfer function with sampling interval T , so we have

$$h(kT) = \frac{1}{b-a}e^{-akT}u(kT) + \frac{1}{a-b}e^{-bkT}u(kT)$$

Taking z -transforms of this we get

$$\begin{aligned} H(z) &= \frac{1}{b-a} \frac{z}{z - e^{-aT}} + \frac{1}{a-b} \frac{z}{z - e^{-bT}} \\ &= \frac{1}{b-a} \left[\frac{z}{z - e^{-aT}} - \frac{z}{z - e^{-bT}} \right] \\ &= \frac{1}{b-a} \frac{z(z - e^{-bT}) - z(z - e^{-aT})}{(z - e^{-aT})(z - e^{-bT})} \\ &= \frac{1}{b-a} \frac{z(e^{-aT} - e^{-bT})}{(z - e^{-aT})(z - e^{-bT})} \end{aligned}$$

From this we notice two things:

- The continuous-time poles at $-a$ and $-b$ have been mapped to discrete-time poles at e^{-aT} and e^{-bT} , where T is the sampling interval. Although we have only shown this for a system with two poles, it is true in general.
- Although the continuous-time system has no finite zeros, the discrete-time system has a zero at the origin. Although we know how the poles will map from continuous-time to discrete-time, it is not as obvious how the zeros will map and/or if new zeros will be introduced.
- If the dominant (real) pole in the continuous-time domain is at $-a$ ($|a| < |b|$), the dominant pole in the discrete-time domain will be at e^{-aT} since $e^{-bT} < e^{-aT}$, i.e. the pole at e^{-bT} is closer to the origin than the pole at e^{-aT} .

Next, let's look at the continuous-time transfer function

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

and its discrete-time equivalent

$$H(z) = \frac{z}{z - e^{-aT}}$$

In the continuous-time domain, the 2% settling-time is usually estimated to be four time constants, so we would estimate $T_s \approx \frac{4}{a}$. Hence to achieve a desired settling time we would need $a = \frac{4}{T_s}$. The pole at $-a$ is mapped to a pole at $e^{-aT} = e^{-4T/T_s}$ in discrete-time. Hence to achieve a settling time of T_s , any discrete-time pole p must have a *magnitude* less than or equal to e^{-4T/T_s} , or

$$|p| \leq e^{-4T/T_s}$$

We can rewrite this expression into a slightly more useful form as

$$T_s \approx \frac{-4T}{\ln(|p|)}$$

This form can be used to find the settling time for discrete-time poles.

Example. Assume we have the discrete-time transfer function $H(z) = \frac{z}{z-0.2}$ and we know the sampling interval is $T = 0.3$ sec. What is the estimated (2%) settling time? Here $|p| = 0.2$, so $T_s \approx \frac{-4(0.3)}{\ln(0.2)} = 0.745$ seconds. To check this, we know the impulse response will be $h(k) = (0.2)^k u(k)$. We have then $h(0T) = 1$, $h(1T) = 0.2$, $h(2T) = 0.04$, and $h(3T) = 0.008$. Clearly the settling time is somewhere between $2T = 0.6$ seconds and $3T = 0.9$ seconds.

Example. Assume we have the discrete-time transfer function $H(z) = \frac{z}{z-0.67}$ and we know the sampling interval is $T = 0.1$ seconds. What is the estimated (2%) settling time? Here $|p| = 0.67$, so $T_s \approx \frac{-4(0.1)}{\ln(0.67)} = 0.998$ seconds. To check this we know the impulse response will be $h(k) = (0.67)^k u(k)$. We have then $h(0T) = 1$, $h(1T) = 0.67$, $h(2T) = 0.449$, $h(3T) = 0.301$, $h(4T) = 0.202$, $h(5T) = 0.135$, $h(6T) = 0.090$, $h(7T) = 0.061$, $h(8T) = 0.041$, $h(9T) = 0.027$,

$h(10T) = 0.0182$. The settling time is approximately $10T = 1.0$.

Example. Assume we have the discrete-time transfer function $H(z) = \frac{z}{(z-0.1)(z+0.2+0.1j)(z+0.2-0.1j)}$ and we know the sampling interval is $T = 0.25$ seconds. What is the estimated (2%) settling time? There are three poles in the system, one pole with magnitude 0.1 and two (complex conjugate) poles with magnitude 0.2236. Just as from continuous-time, the *dominant* pole (or poles) is the pole (or poles) with the slowest response time. In discrete-time, the further away from the origin, the slower the response. Hence in this case the dominant poles are the complex conjugate poles with magnitude $|p| = 0.2236$. Then $T_s \approx \frac{-4(0.25)}{\ln(0.2236)} = 0.667$ seconds.

16.11 Sampling Plants with Zero Order Holds

Most of the system (plants) we will be trying to control in the world are inherently continuous-time, and we would like to use discrete-time control. The biggest issue is how we are going to be able to model the continuous-time plant as a discrete-time plant. The most common method is to assume we have discrete-time signals, $u_d(kT)$, which we model as impulse signals which that only exist at the sample times kT . Specifically, we can write $u_d(kT)$ as

$$u_d(kT) = \dots + u_d(-T)\delta(t + T) + u_d(0)\delta(t) + u_d(T)\delta(t - T) + u_d(2T)\delta(t - 2T) + \dots$$

where the delta functions are continuous-time delta functions. This is shown graphically in Figure 65.

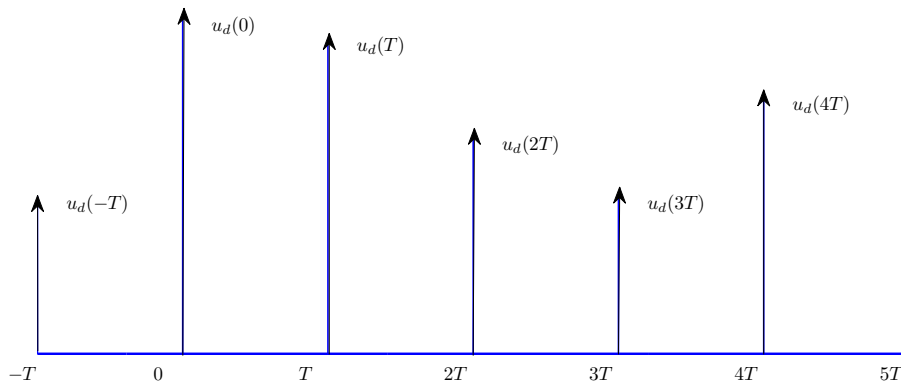


Figure 65: Impulse model of a discrete-time signal.

From these impulse signals we need to generate a continuous-time signal as an input to the plant, $G_p(s)$. The most common way to do this is by the use of a *zero order hold* (zoh), which constructs a continuous-time signal by holding the value of $u(kT)$ constant for time intervals $kT \leq t < (k + 1)T$. The impulse response of the zero order hold is $g_{zoh}(t) = u(t) - u(t - T)$. Let's call the continuous-time output of the zero order hold $u_c(t)$, so we have

$$\begin{aligned} u_c(t) &= g_{zoh}(t) * u_d(kT) \\ &= g_{zoh}(t) * [\dots + u_d(-T)\delta(t + T) + u_d(0)\delta(t) + u_d(T)\delta(t - T) + u_d(2T)\delta(t - 2T) + \dots] \\ &= \dots g_{zoh}(t) * [u_d(-T)\delta(t + T)] + g_{zoh}(t) * [u_d(0)\delta(t)] + g_{zoh}(t) * [u_d(T)\delta(t - T)] + \dots \end{aligned}$$

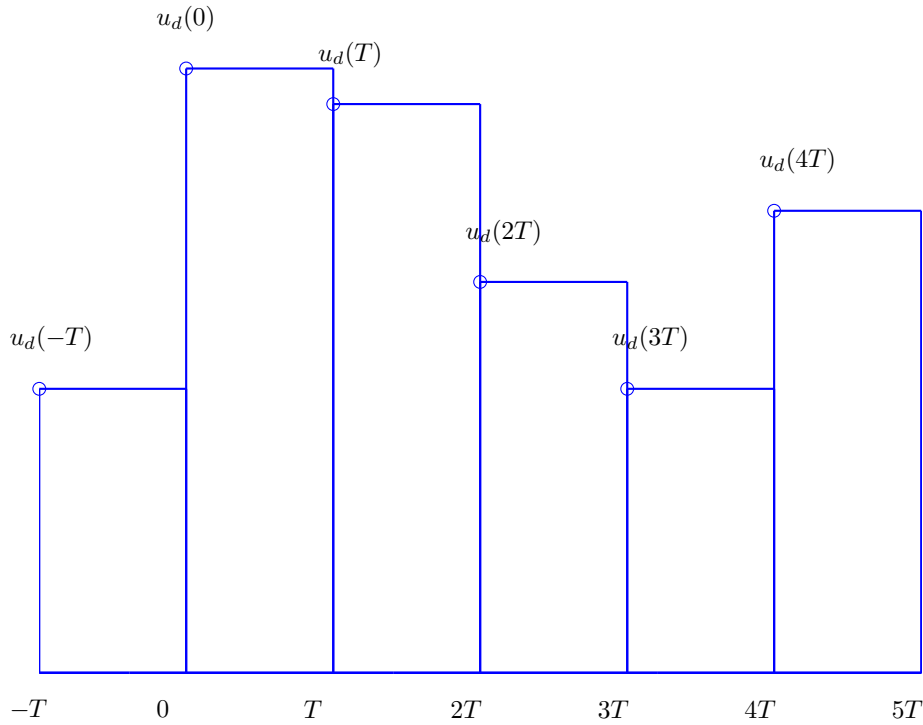


Figure 66: Continuous-time signal $u_c(t)$ as input to the plant. The value of the signal is held constant over the sample interval T .

How let's look at some of these convolutions,

$$\begin{aligned}
 g_{zoh}(t) * [u_d(-T)\delta(t+T)] &= u_d(-T)g_{zoh}(t) * \delta(t+T) = u_d(-T)g_{zoh}(t+T) = u_d(-T)[u(t+T) - u(t)] \\
 g_{zoh}(t) * [u_d(0)\delta(t)] &= u_d(0)g_{zoh}(t) * \delta(t) = u_d(0)g_{zoh}(t) = u_d(0)[u(t) - u(t-T)] \\
 g_{zoh}(t) * [u_d(T)\delta(t-T)] &= u_d(T)g_{zoh}(t) * \delta(t-T) = u_d(T)g_{zoh}(t-T) = u_d(T)[u(t-T) - u(t-2T)]
 \end{aligned}$$

Hence we have the input to the continuous-time plant

$$u_c(t) = \dots + u_d(-T)[u(t+T) - u(t)] + u_d(0)[u(t) - u(t-T)] + u_d(T)[u(t-T) - u(t-2T)] + \dots$$

This is a staircase function, where we have "filled in" the times between samples with the value of the sample at the beginning of the interval. This is shown graphically in Figure 66. It should be pointed out that a zero order hold is not the only method that can be used to convert a discrete-time signal to a continuous-time signal, but it is the most commonly used method, and is the method we will use when we convert continuous-time state equations to discrete-time state equations.

Now that we know how to convert our discrete-time signal to a continuous-time signal for our continuous-time plant, we need to figure out how to model the zero order hold and the plant as an equivalent discrete-time plant. This will allow us to do all of our design in the discrete-time domain. Figure 67 shows a simple single loop system.

What we want is the discrete-time transfer function that corresponds to the product $G_{zoh}(s)G_p(s)$, or

$$G_p(z) = \mathcal{Z} \{G_{zoh}(s)G_p(s)\}$$

Using properties of the Laplace transform we have

$$G_{zoh}(s) = \mathcal{L} \{g_{zoh}(t)\} = \mathcal{L} \{u(t) - u(t - T)\} = \frac{1}{s} - \frac{e^{-sT}}{s}$$

So we have

$$G_p(z) = \mathcal{Z} \left\{ \left[\frac{1}{s} - \frac{e^{-sT}}{s} \right] G_p(s) \right\}$$

We can use this formula directly, but there is a common trick we can use to make things a bit easier. Let's write

$$G_{zoh}(s)G_p(s) = \left[\frac{1}{s} - \frac{e^{-sT}}{s} \right] G_p(s) = [1 - e^{-sT}] G_i(s)$$

where we have defined

$$G_i(s) = \frac{G_p(s)}{s}$$

Now we know from Laplace transforms that e^{-sT} corresponds to a time delay of T , hence we have

$$\mathcal{L}^{-1} \{e^{-sT} G_i(s)\} = g_i(t - T)$$

and then taking z -transforms

$$\mathcal{Z} \{g_i(t - T)\} = \mathcal{Z} \{g_i(kT - T)\} = z^{-1} G_i(z)$$

Where

$$G_i(z) = \mathcal{Z} \{g_i(kT)\}$$

Finally, we will use the Linearity property of the z -transform,

$$G_p(z) = \mathcal{Z} \{G_i(s) - e^{-sT} G_i(s)\} = G_i(z) - z^{-1} G_i(z) = [1 - z^{-1}] G_i(z)$$

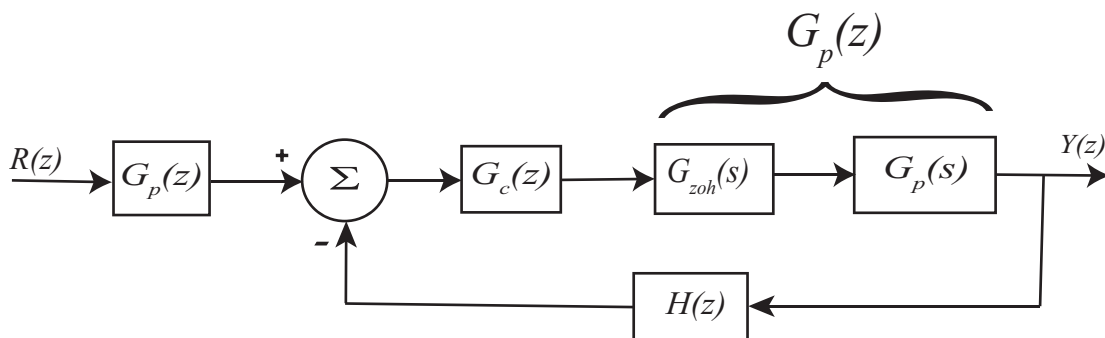


Figure 67: Discrete-time closed loop control system with a zero order hold and a continuous-time plant. We want the equivalent discrete-time plant $G_p(z)$.

which is the answer we want. In summary, the discrete-time transfer function that models the zero order hold and the continuous- time plant is given by

$$G_p(z) = [1 - z^{-1}] \mathcal{Z} \left\{ \frac{G_p(s)}{s} \right\}$$

Example. Consider the continuous-time plant with transfer function $G_p(s) = \frac{1}{s+1}$. What is the equivalent discrete-time transfer function for the plant in series with a zero order hold? We have

$$G_p(z) = [1 - z^{-1}] \mathcal{Z} \left\{ \frac{1}{s(s+1)} \right\}$$

so

$$G_i(s) = \frac{1}{s(s+1)} = \frac{1}{s} - \frac{1}{s+1}$$

and

$$g_i(t) = u(t) - e^{-t}u(t)$$

The sampled version is given as

$$g_i(kT) = u(kT) - e^{-kT}u(kT)$$

with z -transform

$$G_i(z) = \frac{z}{z-1} - \frac{z}{z-e^{-T}} = \frac{z(1-e^{-T})}{(z-1)(z-e^{-T})}$$

Finally, we have

$$G_p(z) = \frac{z-1}{z} G_i(z) = \frac{1-e^{-T}}{z-e^{-T}}$$

Example. Consider the continuous-time plant with transfer function $G_p(s) = \frac{1}{s}$. What is the equivalent discrete-time transfer function for the plant in series with a zero order hold? We have

$$G_p(z) = [1 - z^{-1}] \mathcal{Z} \left\{ \frac{1}{s^2} \right\}$$

so

$$G_i(s) = \frac{1}{s^2}$$

and

$$g_i(t) = tu(t)$$

The sampled version is given as

$$g_i(kT) = kT u(kT)$$

with z -transform

$$G_i(z) = \frac{Tz}{(z-1)^2}$$

Finally, we have

$$G_p(z) = \frac{z-1}{z} G_i(z) = \frac{T}{z-1}$$

16.12 Final Notes

We have focused on using transfer functions in terms of z , such as

$$H(z) = \frac{K(z-a)(z-b)}{(z-c)(z-d)} = \frac{K(z^2 - (a+b)z + ab)}{z^2 - (c+d)z + cd}$$

This form is useful for determining poles and zeros and determining the time-response of the system. However, when implementing controllers (filters) in discrete-time, it is far more convenient to write things in terms of z^{-1} , which represents a delay of one time sample. For implementation purposes, we would write this transfer function as

$$H(z) = \frac{K(1 - (a+b)z^{-1} + abz^{-2})}{1 - (c+d)z^{-1} + cdz^{-2}} = \frac{K(1 - az^{-1})(1 - bz^{-1})}{(1 - cz^{-1})(1 - dz^{-1})}$$

You should be prepared to see and deal with the two equivalent forms of writing the transfer function.

14 Transfer Function Based Discrete-Time Control

Just as with continuous-time control, we can choose to utilize a transfer function approach or a state variable approach to control. In this chapter we will utilize the transfer function approach, while in the remainder of these notes we will utilize the state variable approach. In this brief chapter we first discuss implementing discrete-time transfer functions, then talk about some common conventions used in control systems, and then discuss common control algorithms.

14.1 Implementing Discrete-Time Transfer Functions

Just as in continuous-time we will represent signals in either the time domain or the transform domain, which in this case is the z -transform domain. A simple input-output transfer function block is shown in Figure 68. While in the continuous time-domain it might be somewhat difficult to solve for the output in terms of the input in the time-domain, in the discrete-time domain we can just use the recursive relationship from the transfer function. Specifically, if we have the transfer function $G(z)$, then we can write it as

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_mz^{-m}}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}}$$

where $n \geq m$. Converting this to the time domain, the output $y(k)$ can be written in terms of previous output values and the current and previous input values as

$$y(k) = -a_1y(k-1) - a_2y(k-2) + \dots - a_ny(k-n) + b_0u(k) + b_1u(k-1) + \dots + b_mu(k-m)$$

This is an Infinite Impulse Response (IIR) filter, since the output at any time depends on both previous outputs and previous inputs. If the output depended only on previous inputs it would be a Finite Impulse Response (FIR) filter.

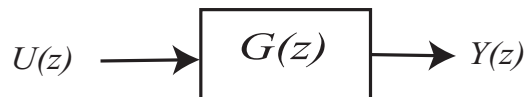


Figure 68: Discrete-time transfer function, $G(z) = \frac{Y(z)}{U(z)}$. Discrete-time transfer functions are often implemented in terms of the difference equation they represent.

14.2 Not Quite Right

At this point we need to stop and think a little bit about what we are doing, and the common conventions used in control systems. Let's assume we have a simplified version of the above expression, which we will write using the explicit dependence on the sample interval T ,

$$y(kT) = -a_1y((k-1)T) + b_0u(kT) + b_1u((k-1)T)$$

Now we have to examine what do we mean by $y(kT)$. Does this mean that this value of y is known at the beginning of the sample interval? This would appear to be correct, but then there is a problem. How can $y(kT)$ be the value of y at the beginning of the sample interval and yet it depends on $u(kT)$, the input at the beginning of the sample interval? Logically, we need to know the value of the input at the beginning of the sample interval $u(kT)$ and then we can compute $y(kT)$. There are two ways around this problem. The first, and cleanest, approach is to assume that $b_0 = 0$. This means the output at time kT only depends on the previous outputs and previous inputs. Alas, in case you have not noticed, people in controls really just want to screw with your mind, so this is not the usual assumption. In controls we usually assume that as long as y is known by the end of the interval (before the next sample time), then it is ok to call it $y(kT)$. Thus, for our example, we might assume $u(kT)$, $u((k-1)T)$, and $y((k-1)T)$ are known at time kT . We then do some calculations to compute y , and if these calculations are finished before the next sample interval, we will call this output value $y(kT)$. When we deal with state variable feedback systems, we will see how we can account for this delay if we want to. Next, we want to look at a basic control loop.

A common configuration for a discrete-time control system is shown in Figure 69. In this simple figure there is a discrete-time prefilter, a discrete-time controller, and a discrete-time representation of a plant. In this figure, we are representing all of the signals in the z -domain, as is the usual custom. In Figure 70 we have represented the signals in the discrete-time domain. If you think about this figure for a while, it should begin to be a little bit disturbing. Why? Well, $e(kT) = G_{pf}r(kT) - y(kT)$, $e(k)$ is the input to the controller which produces the control signal $u(kT)$, which goes into the plant to produce the output $y(kT)$. So now we have $y(kT)$ depends on $e(kT)$ which depends on $y(kT)$. Simulink refers to this as an *algebraic loop*. Often in control systems, if the sampling interval is small enough, then $y(kT)$ is very close to $y((k-1)T)$ and $y((k+1)T)$, so this approximation is not too far off, and the notation is much easier. For some parts of this course, however, we will explicitly insert a delay between the output and the error signal, so the error will be $e(k) = G_{pf}r(k) - y(k-1)$, as is shown in Figure 71. Next we need to discuss some common control configurations.

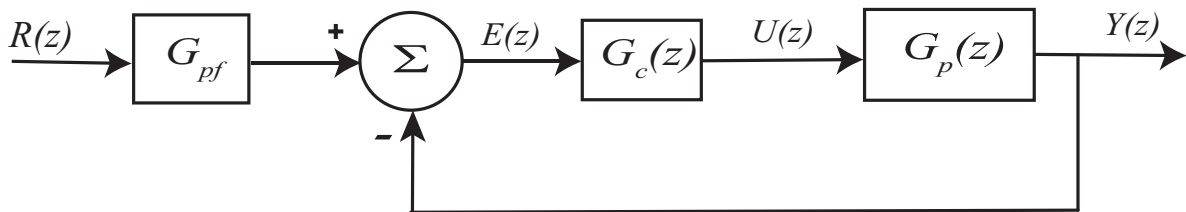


Figure 69: A common discrete-time feedback system written in the z -domain.

14.3 PID and Constant Prefilters

We will first determine how to compute the value of the constant prefilter, though it is generally better to include an integrator in your system than to depend on a prefilter. For the configuration

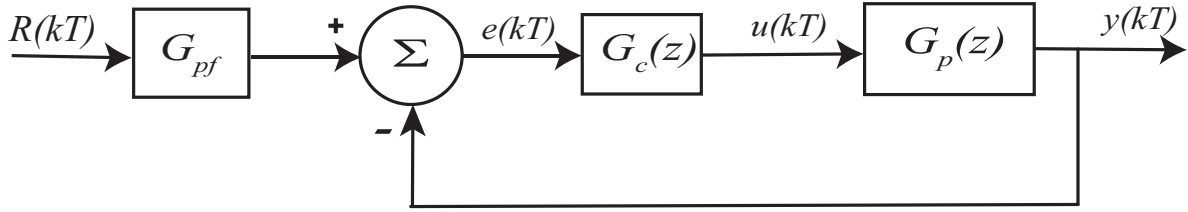


Figure 70: A common discrete-time feedback system written in the time domain.

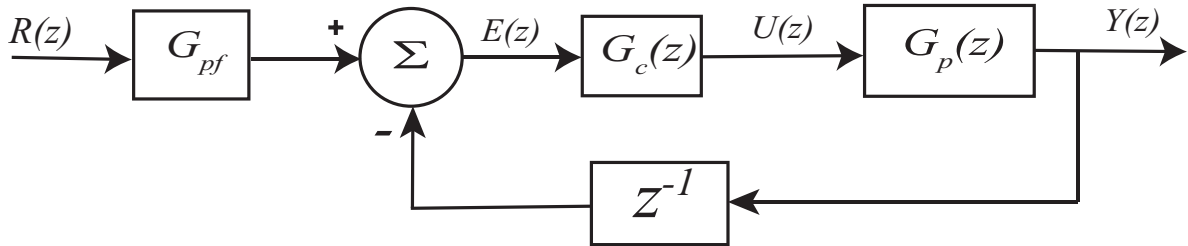


Figure 71: A common discrete-time feedback system written with a delay between the output signal and the error signal.

in Figure 72 we have the closed loop transfer function

$$G_0 = \frac{G_{pf}G_c(z)G_p(z)}{1 + H(z)G_c(z)G_p(z)}$$

Assuming the system is stable, the final value for a unit step input is determined by

$$\begin{aligned} \lim_{t \rightarrow \infty} y(t) &= \lim_{z \rightarrow 1} \frac{z-1}{z} G_0(z) R(z) \\ &= \lim_{z \rightarrow 1} \frac{z-1}{z} G_0(z) \frac{z}{z-1} = G_0(1) \end{aligned}$$

If the input is a unit step, then we want $\lim_{t \rightarrow \infty} y(t) = 1$, so we want $G_0(1) = 1$. Solving for the prefilter we have

$$G_{pf} = \frac{1 + G_c(1)G_p(1)H(1)}{G_c(1)G_p(1)}$$

Now will determine the discrete-time form of a PID controller. For a *proportional controller*, the control signal is directly proportional to the error signal, so we have

$$U(z) = k_p E(z)$$

or

$$G_c(z) = \frac{U(z)}{E(z)} = k_p$$

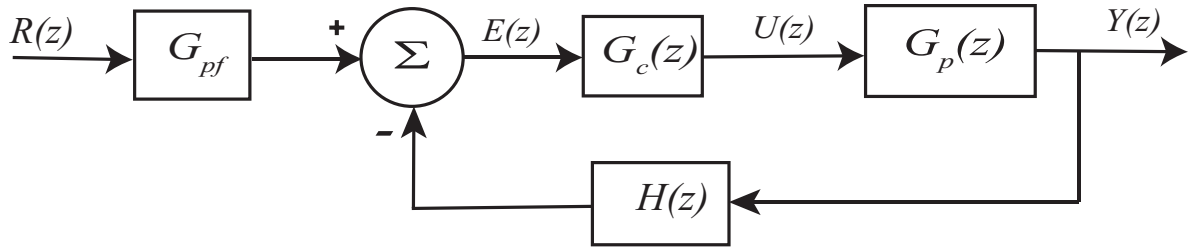


Figure 72: A general discrete-time feedback system written in the z -domain.

which is the same form as we have in continuous time. For an *integral controller* it is probably easiest to start with the continuous-time form,

$$G_c(s) = k_i \frac{1}{s}$$

In the previous chapter we have the relationship

$$\frac{1}{s} \leftrightarrow u(t) = u(kT) \leftrightarrow \frac{z}{z-1} = \frac{1}{1-z^{-1}}$$

So a discrete-time integral controller has the form

$$G_c(z) = k_i \frac{1}{1-z^{-1}}$$

Finally, to get a *derivative controller*, we start with a simple approximation of a derivative, and then take the z -transform,

$$\begin{aligned} u(t) &= k_d \dot{e}(t) \\ &\approx k_d \frac{e(kT) - e((k-1)T)}{T} \end{aligned}$$

or

$$\begin{aligned} U(z) &= \approx k_d \frac{E(z) - z^{-1}E(z)}{T} \\ &= \frac{k_d}{T} (1 - z^{-1})E(z) \end{aligned}$$

It is common to just remove the explicit dependence on the sample interval and just write the derivative controller as

$$G_c(z) = k_d(1 - z^{-1})$$

It should be somewhat comforting that our approximation for the derivative is the inverse (reciprocal) of our expression for the integral. Finally, we can write our discrete-time PID controller as

$$G_c(z) = k_p + k_i \frac{1}{1-z^{-1}} + k_d(1 - z^{-1})$$

Often root locus methods are used for designing closed loop control systems. The construction of the root locus plot is identical to that for continuous-time systems, but the interpretation is different. For continuous-time systems, the system is stable if the poles of the closed loop system are all in the open left-half plane, while for discrete-time systems the system is stable if the poles of the closed loop system are inside the open unit circle. For a continuous-time system, the further the poles are from the $j\omega$ axis, the faster the response is, while for discrete-time systems, the closer to the origin the poles are the faster the response is.

15 Discrete-Time State Equations

In this chapter we develop discrete-time state equations assuming we have a time-invariant continuous-time state variable description of the system. The standard form for a time-invariant continuous time state variable model is

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

Here $x(t)$ is the state vector, $u(t)$ is the input vector to the plant⁹, $y(t)$ is the output vector from the plant, and the A, B, C , and D matrices are constant. If these matrices were not constant this would not be a time-invariant system. A signal flow graph for this continuous-time system is shown in Figure 73. If we assume we sample this system, we will have a discrete-time state variable model of the system which we can write

$$\begin{aligned}x([k+1]T) &= Gx(kT) + Hu(kT) \\ y(kT) &= Cx(kT) + Du(kT)\end{aligned}$$

where we have explicitly shown the dependence on the sampling interval T . A signal flow graph of this discrete-time state variable model is shown in Figure 74. Note that the output equation contains the same C and D matrices as the continuous-time model. In this chapter we will show how to go from the continuous-time to the discrete-time state variable models assuming the sampling is done with a zero order hold (ZOH). We will also show how to derive the discrete-time equations when there may be delays between the input and output. We need to be able to solve the continuous-time state equations, and to do this we first need a bit of background mathematics.

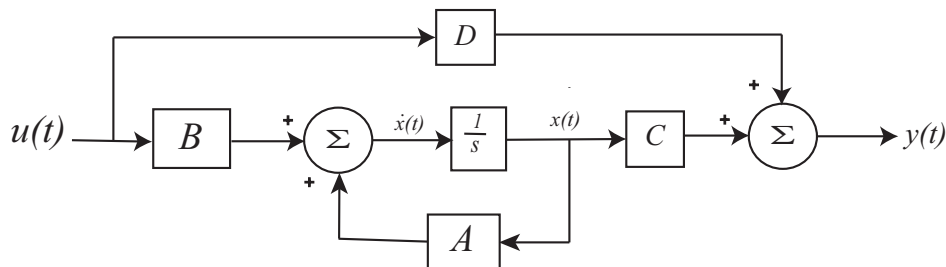


Figure 73: Continuous-time state variable model for a plant.

15.1 The Continuous-Time State Transition Matrix

The function e^{At} , where A is a *square* matrix, is a very important function in continuous-time state variable systems. If A is an $n \times n$ matrix, then as we will see, e^{At} will also be an $n \times n$ matrix. This matrix is called the *state transition matrix*. We will need to be able to compute this function and will need some of its properties in order to derive discrete-time state variable

⁹This is not to be confused with a unit step function. The input may be a unit step function, but it is not necessarily a unit step function. This is common notation and you will need to look at the context to determine if $u(t)$ represents a unit step function or the input to the plant.

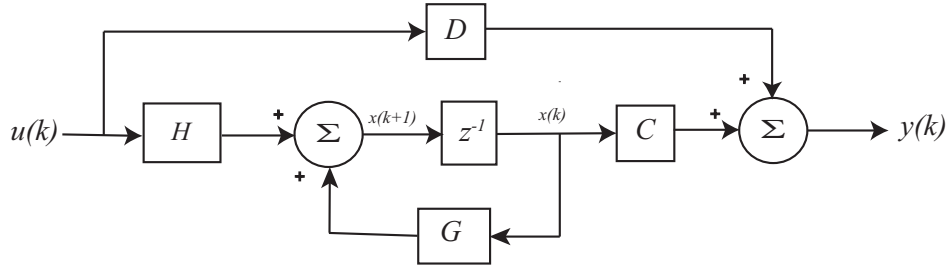


Figure 74: Discrete-time state variable model for a plant.

equations from continuous-time state variable equations.

Consider the function e^{at} for a scalar a . We know that this function has the Taylor series

$$e^{at} = \frac{1}{0!} + \frac{(at)}{1!} + \frac{(at)^2}{2!} + \dots = \sum_{k=0}^{k=\infty} \frac{(at)^k}{k!}$$

Using this motivation, we will define the state transition matrix as

$$e^{At} = \frac{I}{0!} + \frac{At}{1!} + \frac{(At)^2}{2!} + \dots = \sum_{k=0}^{k=\infty} \frac{(At)^k}{k!}$$

where $A^2 = A \times A$, $A^3 = A^2 \times A$, etc. Using this definition we can then compute

$$\begin{aligned} \frac{d}{dt} \{e^{At}\} &= \frac{d}{dt} \left[I + \frac{At}{1!} + \frac{A^2 t^2}{2!} + \frac{A^3 t^3}{3!} + \dots \right] \\ &= \left[0 + A + \frac{A^2 2t}{2!} + \frac{A^3 3t^2}{3!} + \dots \right] \\ &= A \left[I + At + \frac{A^2 t^2}{2!} + \dots \right] \\ &= Ae^{At} = e^{At} A \end{aligned}$$

It can also be shown that $e^{At}e^{As} = e^{A(t+s)}$, for t and s scalars. A special case of this is when $s = -t$. Then we have

$$e^{At}e^{-At} = e^{A0} = I$$

Hence

$$\left[e^{At} \right]^{-1} = e^{-At}$$

so we know what the inverse matrix of e^{At} is. We can use these results to perform integration. For example

$$\int_0^t e^{A\lambda} d\lambda = A^{-1} e^{A\lambda} \Big|_{\lambda=0}^{\lambda=t} = A^{-1} [e^{At} - I]$$

provided A^{-1} exists.

15.2 Solution of the Continuous-Time State Equations

Assume we have the continuous-time state variable model of a plant

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

We can rewrite this system as

$$\dot{x}(t) - Ax(t) = Bu(t)$$

Next we *premultiply* both sides of the equation by e^{-At} , so we have

$$e^{-At}(\dot{x}(t) - Ax(t)) = e^{-At}Bu(t)$$

Now consider

$$\begin{aligned}\frac{d}{dt}(e^{-At}x(t)) &= \left(\frac{d}{dt}e^{-At}\right)x(t) + e^{-At}\left(\frac{d}{dt}x(t)\right) \\ &= -Ae^{-At}x(t) + e^{-At}\dot{x}(t) \\ &= e^{-At}(\dot{x}(t) - Ax(t))\end{aligned}$$

since e^{-At} and A commute. Hence we have

$$\frac{d}{dt}(e^{-At}x(t)) = e^{-At}Bu(t)$$

Finally, we integrate both sides of this equation from an initial time t_0 to the current time t ,

$$\begin{aligned}\int_{t_0}^t \frac{d}{d\lambda}(e^{-A\lambda}x(\lambda))d\lambda &= \int_{t_0}^t e^{-A\lambda}Bu(\lambda)d\lambda \\ e^{-At}x(t) - e^{-At_0}x(t_0) &= \int_{t_0}^t e^{-A\lambda}Bu(\lambda)d\lambda \\ x(t) &= e^{A(t-t_0)}x(t_0) + \int_{t_0}^t e^{A(t-\lambda)}Bu(\lambda)d\lambda\end{aligned}$$

Now that we have solved the state equations, we can convert them to a discrete-time model. However, the model we end up with depends on how we are going to sample the system to get a discrete-time system. We will assume we are using a zero-order-hold (ZOH), which holds the input at a constant value over one sampling interval of length T . Specifically, we assume

$$u(t) = u(kT) \quad kT \leq t < (k+1)T$$

Next we look at an interval from $t_0 = kT$ to $t = (k+1)T$. The solution to the state equations becomes

$$x([k+1]T) = e^{A([k+1]T-kT)}x(kT) + \left\{ \int_{kT}^{[k+1]T} e^{A([k+1]T-\lambda)}d\lambda \right\} Bu(kT)$$

or

$$x([k+1]T) = e^{AT}x(kT) + \left\{ \int_{kT}^{[k+1]T} e^{A([k+1]T-\lambda)} d\lambda \right\} Bu(kT)$$

Notice that the integral is no longer a function of $u(t)$ so we can take both the B and $u(t)$ outside of the integral. Let's change variables, so

$$\sigma = [k+1]T - \lambda$$

With this variable change, the resulting equation is

$$x([k+1]T) = e^{AT}x(kT) + \left\{ \int_0^T e^{A\sigma} d\sigma B \right\} u(kT)$$

For convenience, the dependence on the sampling interval T is suppressed, and we have the resulting discrete-time state equations

$$\begin{aligned} x(k+1) &= Gx(k) + Hu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}$$

where

$$\begin{aligned} G &= e^{AT} \\ H &= \int_0^T e^{A\lambda} d\lambda B \end{aligned}$$

Note that the output (second) equation does not really change when we go from a discrete-time to a continuous time state variable description of a system. The only thing we need to know now is how to compute e^{At} , which is discussed in the next section.

15.3 Computing the State Transition Matrix, e^{At}

In computing the discretized state equations, we need to be able to determine e^{At} (or e^{AT} for sampling interval T). There are three common approaches to this computation: (1) truncating the infinite sum, (2) using the Laplace transform, and (3) matching functions on the eigenvalues. The first two methods are fairly easy to understand, while the third method will just be used without justification.

15.3.1 Truncating the Infinite Sum

Since we know that by definition,

$$e^{At} = \sum_{k=0}^{k=\infty} \frac{(At)^k}{k!}$$

if the sample interval T is small enough ($t = T$) we will get a good approximation using only a few terms in the infinite series. Hence if we were to use only a few terms we would get the

approximations

$$\begin{aligned}
 e^{At} &\approx I + At \quad (\text{two terms}) \\
 &\approx I + At + \frac{1}{2}A^2t^2 \quad (\text{three terms}) \\
 &\approx I + At + \frac{1}{2}A^2t^2 + \frac{1}{6}A^3t^3 \quad (\text{four terms})
 \end{aligned}$$

This technique generally becomes difficult if more than a few terms are needed.

15.3.2 Laplace Transform Method

In our solution to the continuous-time state variable equations, if we assume $t_0 = 0$ (the initial time is zero), $x(0) = 0$ (the initial conditions are zero), and $D = 0$, then we have

$$y(t) = C \int_0^t e^{A(t-\lambda)} B u(\lambda) d\lambda$$

For LTI systems, if we know the input to a system, $u(t)$, and the impulse response of the system, $f(t)$, then the output of the system can be determined by using convolution,

$$y(t) = f(t) \star u(t) = \int_{-\infty}^{\infty} f(t - \lambda) u(\lambda) d\lambda$$

If we know that both the input and the system are causal (i.e., both $u(t)$ and $f(t)$ are zero for $t < 0$), then this simplifies to

$$y(t) = f(t) \star u(t) = \int_0^t f(t - \lambda) u(\lambda) d\lambda$$

If we compare this to our solution to the state equations above, we have

$$f(t) = C e^{At} B$$

Now let's look at solving the state equations using Laplace transforms. Again we assume the initial time is zero, the initial state is zero, and the D matrix is zero. We then have initially

$$\begin{aligned}
 \dot{x}(t) &= Ax(t) + Bu(t) \\
 y(t) &= Cx(t)
 \end{aligned}$$

Taking the Laplace transform we have

$$\begin{aligned}
 sX(s) &= AX(s) + BU(s) \\
 Y(s) &= CX(s)
 \end{aligned}$$

Solving for $X(s)$ and putting this into the $Y(s)$ equation we have

$$\begin{aligned}
 X(s) &= (sI - A)^{-1} BU(s) \\
 Y(s) &= C(sI - A)^{-1} BU(s)
 \end{aligned}$$

Comparing time-domain and s -domain solutions, we have

$$\begin{aligned}y(t) &= f(t) \star u(t) \\ Y(s) &= F(s)U(s)\end{aligned}$$

which means

$$\begin{aligned}F(s) &= C(sI - A)^{-1}B \\ f(t) &= Ce^{At}B\end{aligned}$$

or, finally

$$e^{At} = \mathcal{L}^{-1}\{(sI - A)^{-1}\}$$

In what follows, we will be computing the inverse of a 2×2 matrix, which you are expected to be able to do. If

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

then if A^{-1} exists, it can be computed as

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

A simple way to remember this is (1) divide by the determinant, (2) switch the elements on the major diagonal, and (3) negate the off diagonal terms.

Finally, it should be noted that, as a very simple check, $e^{A0} = e^0 = I$, so if we set $t = 0$ we should get an identity matrix in our computation of e^{At} .

Example. Compute e^{At} for

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$$

First we compute $sI - A$,

$$\begin{aligned}sI - A &= \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} s - 1 & -2 \\ 0 & s - 1 \end{bmatrix}\end{aligned}$$

and then $(sI - A)^{-1}$,

$$(sI - A)^{-1} = \frac{1}{(s - 1)^2} \begin{bmatrix} s - 1 & 2 \\ 0 & s - 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{s - 1} & \frac{2}{(s - 1)^2} \\ 0 & \frac{1}{s - 1} \end{bmatrix}$$

and then find the inverse Laplace transform,

$$e^{At} = \begin{bmatrix} e^t & 2te^t \\ 0 & e^t \end{bmatrix} = \mathcal{L}^{-1} \{ (sI - A)^{-1} \}$$

Example. Compute e^{At} for

$$A = \begin{bmatrix} 2 & -2 \\ 2 & 6 \end{bmatrix}$$

First we compute $sI - A$,

$$\begin{aligned} sI - A &= \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 2 & -2 \\ 2 & 6 \end{bmatrix} \\ &= \begin{bmatrix} s-2 & 2 \\ -2 & s-6 \end{bmatrix} \end{aligned}$$

and then $(sI - A)^{-1}$,

$$(sI - A)^{-1} = \frac{1}{(s-4)^2} \begin{bmatrix} s-6 & -2 \\ 2 & s-2 \end{bmatrix} = \begin{bmatrix} \frac{s-6}{(s-4)^2} & \frac{-2}{(s-4)^2} \\ \frac{2}{(s-4)^2} & \frac{s-2}{(s-4)^2} \end{bmatrix}$$

and then find the inverse Laplace transform,

$$e^{At} = \begin{bmatrix} e^{4t} - 2te^{4t} & -2te^{4t} \\ 2te^{4t} & e^{4t} + 2te^{4t} \end{bmatrix} = \mathcal{L}^{-1} \{ (sI - A)^{-1} \}$$

15.4 Discretization with Delays in the Input

We would now like to extend our development to the case where there are delays in the input. We can model such a system as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t - \tau) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

where τ is the delay. Following the same procedure as before, we will have

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^t e^{A(t-\lambda)}Bu(\lambda - \tau)d\lambda$$

Again assume $t_0 = kT$ and $t = [k+1]T$, so we have

$$x([k+1]T) = e^{AT}x(kT) + \int_{kT}^{[k+1]T} e^{A([k+1]T-\lambda)}Bu(\lambda - \tau)d\lambda$$

Let's assume $0 \leq \tau \leq T$ and look at $u(t)$ and the delayed signal $u(t - \tau)$, shown in Figure 75. In order to evaluate the integral in our expression above we will need to break the integral up into two pieces as follows:

$$\begin{aligned} \int_{kT}^{[k+1]T} e^{A([k+1]T-\lambda)}Bu(\lambda - \tau)d\lambda &= \int_{kT}^{kT+\tau} e^{A([k+1]T-\lambda)}Bu([k-1]T)d\lambda + \int_{kT+\tau}^{[k+1]T} e^{A([k+1]T-\lambda)}Bu(kT)d\lambda \\ &= H_1u([k-1]T) + H_0u(kT) \end{aligned}$$

where

$$H_0 = \int_{kT+\tau}^{[k+1]T} e^{A([k+1]T-\lambda)} d\lambda B = \int_0^{T-\tau} e^{A\sigma} d\sigma B$$

$$H_1 = \int_{kT}^{kT+\tau} e^{A([k+1]T-\lambda)} d\lambda B = e^{A(T-\tau)} \int_0^{\tau} e^{A\sigma} d\sigma B$$

so we have

$$x([k+1]T) = Gx(kT) + H_0u(kT) + H_1u([k-1]T)$$

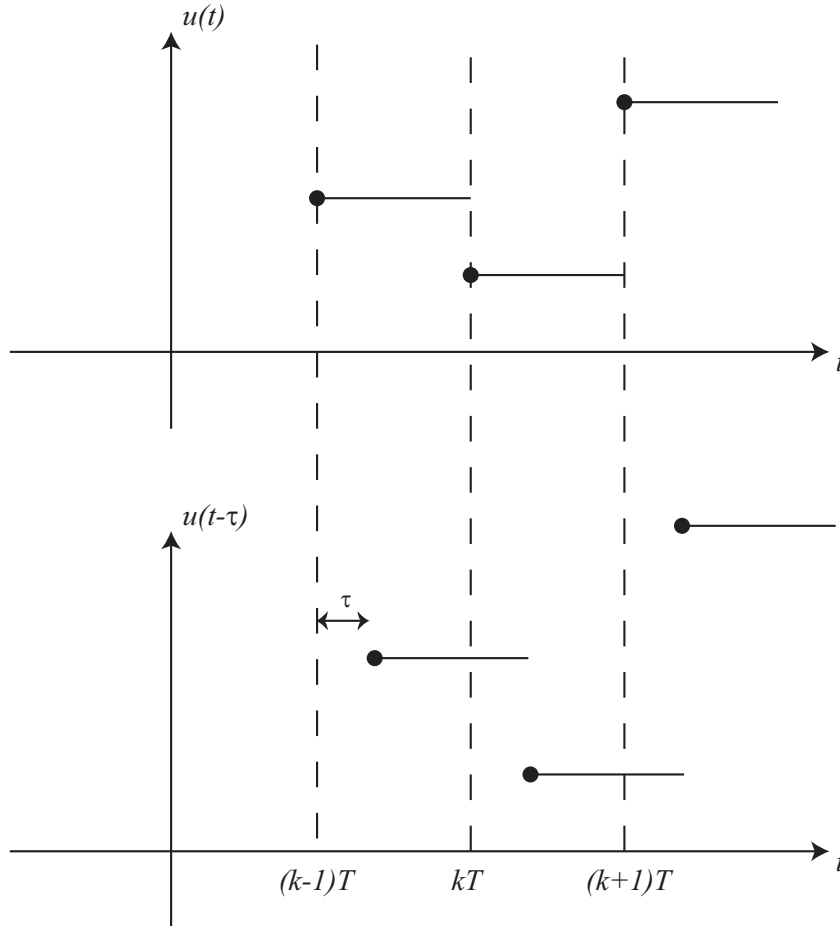


Figure 75: Original and delayed (by an amount τ) input.

Suppressing the dependence on the sampling interval T and writing as a state variable model we have

$$\begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} G & H_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} H_0 \\ I \end{bmatrix} u(k)$$

where

$$G = e^{AT}$$

$$\begin{aligned}
H_0 &= \int_0^{T-\tau} e^{A\sigma} d\sigma B \\
H_1 &= e^{A(T-\tau)} \int_0^{\tau} e^{A\sigma} d\sigma B
\end{aligned}$$

For the special case where $\tau = T$ (a 1 sample delay) we have

$$\begin{aligned}
G &= e^{AT} \\
H_0 &= \int_0^0 e^{A\sigma} d\sigma = 0 \\
H_1 &= \int_0^T e^{A\sigma} d\sigma B = H
\end{aligned}$$

The state equations are then

$$\begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} G & H \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u(k)$$

Example. Consider the continuous-time double integrator system

$$\begin{aligned}
\dot{x}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\
y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)
\end{aligned}$$

A double integrator system with a delay of τ in the input can be described as

$$\begin{aligned}
\dot{x}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t - \tau) \\
y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)
\end{aligned}$$

We need to compute e^{At} , so we'll use the Laplace transform method,

$$\begin{aligned}
sI - A &= \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} s & -1 \\ 0 & s \end{bmatrix} \\
(sI - A)^{-1} &= \frac{1}{s^2} \begin{bmatrix} s & 1 \\ 0 & s \end{bmatrix} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} \\ 0 & \frac{1}{s} \end{bmatrix}
\end{aligned}$$

so

$$e^{At} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}$$

Then

$$\begin{aligned}
G &= e^{AT} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \\
H_0 &= \int_0^{T-\tau} e^{A\sigma} d\sigma B = \int_0^{T-\tau} \begin{bmatrix} 1 & \sigma \\ 0 & 1 \end{bmatrix} d\sigma \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
&= \begin{bmatrix} (T - \tau) & \frac{(T-\tau)^2}{2} \\ 0 & (T - \tau) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{(T-\tau)^2}{2} \\ T - \tau \end{bmatrix} \\
H_1 &= e^{A(T-\tau)} \int_0^\tau e^{A\sigma} d\sigma B = \begin{bmatrix} 1 & T - \tau \\ 0 & 1 \end{bmatrix} \int_0^\tau \begin{bmatrix} 1 & \sigma \\ 0 & 1 \end{bmatrix} d\sigma \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & T - \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tau & \tau^2 \\ 0 & \tau \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & T - \tau \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\tau^2}{2} \\ \tau \end{bmatrix} \\
&= \begin{bmatrix} \frac{\tau^2}{2} + T\tau - \tau^2 \\ \tau \end{bmatrix} = \begin{bmatrix} \tau(T - \frac{\tau}{2}) \\ \tau \end{bmatrix}
\end{aligned}$$

so our new state model of the discretized system is

$$\begin{aligned}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ u(k) \end{bmatrix} &= \begin{bmatrix} 1 & T & \tau(T - \frac{\tau}{2}) \\ 0 & 1 & \tau \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} \frac{(T-\tau)^2}{2} \\ T - \tau \\ 1 \end{bmatrix} u(k) \\
y(k) &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ u(k-1) \end{bmatrix}
\end{aligned}$$

15.5 State Variable to Transfer Function

Just as in continuous-time it is possible to go from a state variable description to a transfer function description of a system. Doing this sometimes helps with different concepts, such as pole placement, steady state error, and the speed of response. Consider the usual state-variable description of a discrete-time system

$$\begin{aligned}
x(k+1) &= Gx(k) + Hu(k) \\
y(k) &= Cx(k) + Du(k)
\end{aligned}$$

Taking the z-transform of the equation and assuming the system is initially at rest ($X(0) = 0$), we get

$$\begin{aligned}
zX(z) &= GX(z) + HU(z) \\
Y(z) &= CX(z) + DU(z)
\end{aligned}$$

Solving for $X(z)$ and putting this into the equation for $Y(z)$ we get

$$\begin{aligned}
X(z) &= (zI - G)^{-1} HU(z) \\
Y(z) &= C(zI - G)^{-1} HU(z)
\end{aligned}$$

Hence the transfer function between the input and output (the plant) is

$$G_p(z) = C(zI - G)^{-1} H$$

Example. Find the equivalent transfer function for the following state variable model

$$\begin{aligned}x(k+1) &= \begin{bmatrix} 0.1 & 0 \\ 0.2 & 0.3 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(k) \\y(k) &= \begin{bmatrix} 0 & 1 \end{bmatrix}\end{aligned}$$

First we compute $zI - G$,

$$\begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} - \begin{bmatrix} 0.1 & 0 \\ 0.2 & 0.3 \end{bmatrix} = \begin{bmatrix} z - 0.1 & 0 \\ -0.2 & z - 0.3 \end{bmatrix}$$

then $(zI - G)^{-1}$,

$$(zI - G)^{-1} = \frac{1}{(z - 0.1)(z - 0.3)} \begin{bmatrix} z - 0.3 & 0 \\ 0.2 & z - 0.1 \end{bmatrix}$$

Finally, $G_p(z) = C(zI - G)^{-1}H$, so

$$\begin{aligned}G_p(z) &= \frac{1}{(z - 0.1)(z - 0.3)} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} z - 0.3 & 0 \\ 0.2 & z - 0.1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ &= \frac{0.2}{(z - 0.1)(z - 0.3)}\end{aligned}$$

15.6 Poles and Eigenvalues

For continuous-time systems we know the poles of the system are given by the determinant of $sI - A$, or the roots of the characteristic equation $\Delta(s) = |sI - A| = 0$. These poles are the eigenvalues of the A matrix. Similarly, for a discrete-time system, the poles of the system are given by the determinant of $zI - G$, or the roots of the characteristic equation $\Delta(z) = |zI - G| = 0$. The poles of the discrete-time system are also the eigenvalues of the G matrix. If all of the poles of a discrete-time system is at the origin, this is referred to as *deadbeat response*. The deadbeat response is the fastest possible response of a discrete-time system. However, the control effort required for deadbeat response may be too large for actual implementation.

16 State Variable Feedback

State variable feedback is a commonly used method in modern control systems. If the system is controllable, then we will be able to use state variable feedback to place the poles of the closed loop system anywhere we want. This is generally not the case for classical transfer function based methods. While we can use state-variable feedback to place the poles of a controllable system where ever we want, we cannot change the location of the system zeros. We can place the poles to try a pole/zero cancelation, but this is usually not desirable since the zeros of the system are just approximations based on our model.

We begin this chapter deriving the equations for state variable feedback, then utilize three different algorithms for determining the state variable feedback gain vector/matrix.

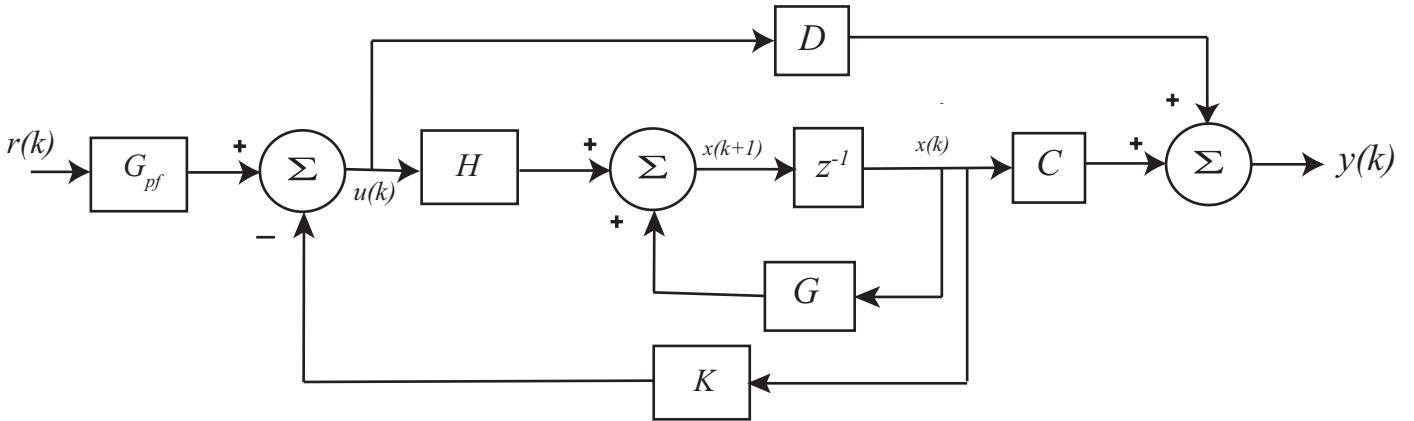


Figure 76: Basic discrete-time state variable feedback system. The feedback gain matrix is K , which is used to place the closed loop poles, and the prefilter gain G_{pf} , which is used to adjust the steady state error for a step input.

Consider the open loop system

$$\begin{aligned}x(k+1) &= Gx(k) + Hu(k) \\ y(k) &= Cx(k) + Du(k)\end{aligned}$$

Assume we have a state variable system as shown in Figure 76, where K is the feedback gain vector/matrix, G_{pf} is the prefilter gain, and $r(k)$ is the new reference input. Hence we have $u(k) = G_{pf}r(k) - Kx(k)$. Using this in our model of the plant we have

$$\begin{aligned}x(k+1) &= Gx(k) + H[G_{pf}r(k) - Kx(k)] = [G - HK]x(k) + [HG_{pf}]r(k) \\ y(k) &= Cx(k) + D[G_{pf}r(k) - Kx(k)] = [C - DK]x(k) + [DG_{pf}]r(k)\end{aligned}$$

The transfer function between the new input $r(k)$ and the output $y(k)$ can then be computed using z transforms as follows:

$$zX(z) = [G - HK]X(z) + [HG_{pf}]R(z)$$

$$\begin{aligned}
[zI - G + HK]X(z) &= [HG_{pf}]R(z) \\
X(z) &= [zI - G + HK]^{-1}[HG_{pf}]R(z) \\
Y(z) &= [C - DK]X(z) + [DG_{pf}]R(z) \\
Y(z) &= \{[C - DK][zI - G + HK]^{-1}[HG_{pf}] + [DG_{pf}]\} R(z)
\end{aligned}$$

so the transfer function/transfer matrix (if there are vector inputs) between the input $R(z)$ and the output $Y(z)$ is

$$F(z) = [C - DK][zI - G + HK]^{-1}[HG_{pf}] + [DG_{pf}]$$

Normally we will assume $D = 0$, and the transfer function reduces to

$$F(z) = C[zI - G + HK]^{-1}[HG_{pf}]$$

Let's assume we have a scalar input and $D = 0$. We know that $Y(z) = F(z)R(z)$, and if the input is a unit step $r(k) = u(k)$, then for a zero steady state error we use the *Final Value Theorem* to make the final final of $y(k) = 1$,

$$\begin{aligned}
\lim_{k \rightarrow \infty} y(k) = 1 &= \lim_{z \rightarrow 1} \frac{z-1}{z} Y(z) = \lim_{z \rightarrow 1} \frac{z-1}{z} F(z)R(z) \\
1 &= \lim_{z \rightarrow 1} \frac{z-1}{z} F(z) \frac{z}{z-1} = \lim_{z \rightarrow 1} F(z) = F(1) = C[I - G + HK]^{-1}[HG_{pf}]
\end{aligned}$$

or

$$G_{pf} = \frac{1}{C[I - G + HK]^{-1}H}$$

This prefilter will produce a steady state error of zero for a unit step input *provided* our system is accurately enough modeled. Note that we need to determine the feedback gain vector K *before* we try to determine the prefilter gain G_{pf} . Since the prefilter is outside of the feedback loop this is not a very robust method of trying to produce a steady state error of zero. A better method would be to make the system a type one system, which will be discussed in the next chapter.

16.1 Pole Placement by Transfer Functions

To determine the state feedback gain vector using the transfer function, we need to realize that the denominator of the transfer function will be determined by

$$\Delta(z) = \det[zI - G + HK]$$

We want the closed loop poles at $\mu_1, \mu_2, \dots, \mu_n$, so the *desired* characteristic equations is

$$\begin{aligned}
\Delta(z) &= (z - \mu_1)(z - \mu_2) \dots (z - \mu_n) \\
&= z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + \dots + \alpha_{n-1} z + \alpha_n
\end{aligned}$$

We then determine K by setting these equal,

$$\Delta(z) = \det[zI - G + HK] = z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + \dots + \alpha_{n-1} z + \alpha_n$$

This is a difficult computation for system with more than a few poles, so it is seldom used except for low order systems.

Example. Assume we have the system

$$G = \begin{bmatrix} 0.1 & 0.2 \\ 0 & 0.2 \end{bmatrix}, \quad H = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}$$

We first check to be sure the system is controllable,

$$\begin{bmatrix} GH & H \end{bmatrix} = \begin{bmatrix} 0.02 & 0 \\ 0.02 & 0.1 \end{bmatrix}$$

Clearly this matrix has rank two, so the system is controllable. Let's find the gain vector K to place the closed loop poles at -0.1 and -0.2. First we compute the desired characteristic polynomial

$$\Delta(z) = [z - (-0.1)][z - (-0.2)] = (z + 0.1)(z + 0.2) = z^2 + 0.3z + 0.02$$

and then compute $\det[zI - G + HK]$,

$$\begin{aligned} zI - G + HK &= \begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} - \begin{bmatrix} 0.1 & 0.2 \\ 0 & 0.2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} \begin{bmatrix} K_1 & K_2 \end{bmatrix} \\ &= \begin{bmatrix} z - 0.1 & -0.2 \\ 0.1K_1 & z - 0.2 + 0.1K_2 \end{bmatrix} \\ \det[zI - G + HK] &= (z - 0.1)(z - 0.2 + 0.1K_2) - (-0.2)(0.1K_1) \\ &= z^2 + (-0.1 - 0.2 + 0.1K_2)z + (0.02 - 0.01K_2 + 0.02K_1) \end{aligned}$$

Equating terms with our desired characteristic polynomial, we get $K_2 = 6$ and then $K_1 = 3$, so

$$K = \begin{bmatrix} 3 & 6 \end{bmatrix}$$

Example. Assume we have the system

$$G = \begin{bmatrix} 0.1 & 0 \\ 0.2 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

We first check to be sure the system is controllable,

$$\begin{bmatrix} GH & H \end{bmatrix} = \begin{bmatrix} 0.1 & 1.0 \\ 0.2 & 0 \end{bmatrix}$$

Clearly this matrix has rank two, so the system is controllable. We know from our previous examples that this system is controllable. Let's find the gain vector K to place the closed loop poles at 0.1 and -0.1. First we compute the desired characteristic polynomial

$$\Delta(z) = [z - (0.1)][z - (-0.1)] = (z - 0.1)(z + 0.1) = z^2 - 0.01$$

and then compute $\det[zI - G + HK]$,

$$\begin{aligned} zI - G + HK &= \begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} - \begin{bmatrix} 0.1 & 0 \\ 0.1 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} K_1 & K_2 \end{bmatrix} \\ &= \begin{bmatrix} z - 0.1 + K_1 & K_2 \\ -0.2 & z \end{bmatrix} \\ \det[zI - G + HK] &= (z - 0.1 + K_1)(z) - (K_2)(-0.2) \\ &= z^2 + (K_1 - 0.1)z + K_2 \cdot 0.2 \end{aligned}$$

Equating terms with our desired characteristic polynomial, we get $K_1 = 0.1$ and then $K_2 = -0.05$, so

$$K = \begin{bmatrix} 0.1 & -0.05 \end{bmatrix}$$

Example. Assume we again have the system

$$G = \begin{bmatrix} 0.1 & 0.2 \\ 0 & 0.2 \end{bmatrix}, \quad H = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}$$

Let's find the gain vector K to place both closed loop poles at 0. This clearly results in *deadbeat control*. First we compute the desired characteristic polynomial

$$\Delta(z) = (z - 0)(z - 0) = z^2$$

and then compute $\det[zI - G + HK]$,

$$\begin{aligned} zI - G + HK &= \begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} - \begin{bmatrix} 0.1 & 0.2 \\ 0 & 0.2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} \begin{bmatrix} K_1 & K_2 \end{bmatrix} \\ &= \begin{bmatrix} z - 0.1 & -0.2 \\ 0.1K_1 & z - 0.2 + 0.1K_2 \end{bmatrix} \\ \det[zI - G + HK] &= (z - 0.1)(z - 0.2 + 0.1K_2) - (-0.2)(0.1K_1) \\ &= z^2 + (-0.1 - 0.2 + 0.1K_2)z + (0.02 - 0.01K_2 + 0.02K_1) \end{aligned}$$

Equating terms with our desired characteristic polynomial, we get $K_2 = 3$ and then $K_1 = 0.5$, so

$$K = \begin{bmatrix} 0.5 & 3 \end{bmatrix}$$

16.2 State Feedback Examples

In this section we examine the response of a two degree of freedom system using state variable feedback. For this system, placing the poles using the transfer function method would be very tedious and so Matlab routines were used which were designed for pole placement of systems described by state equations.

Assume we have the discrete-time state variable model given by

$$\begin{bmatrix} x_1(k) \\ v_1(k) \\ x_2(k) \\ v_2(k) \end{bmatrix} = \begin{bmatrix} 0.4964 & 0.0391 & 0.5153 & 0.0095 \\ -15.5298 & 0.4361 & 14.7498 & 0.5022 \\ 0.4452 & 0.0082 & 0.3068 & 0.0358 \\ 12.7738 & 0.4326 & -21.0039 & 0.2572 \end{bmatrix} \begin{bmatrix} x_1(k) \\ v_1(k) \\ x_2(k) \\ v_2(k) \end{bmatrix} + \begin{bmatrix} 4.0181 \\ 142.8037 \\ 0.3955 \\ 29.8022 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ v_1(k) \\ x_2(k) \\ v_2(k) \end{bmatrix}$$

The sample interval is $T_s = 0.05$ seconds. In this model our output is the first state (x_1). For this plant, the transfer function is

$$G_p(z) = \frac{4.018z^3 + 2.048z^2 + 1.939z + 3.65}{z^4 - 1.496z^3 + 1.492z^2 - 1.392z + 0.8639}$$

with the four poles at $-0.1073 \pm 0.9583j$ and $0.8555 \pm 0.4437j$. This system also has zeros at -0.9732 and $0.2317 \pm 0.9380j$.

The open loop unit step response of this system is shown in Figure 77. We can estimate the settling time using our previous formula as $T_s \approx \frac{-4T}{\ln(|p|)} = \frac{-4(0.05)}{\ln(|-0.1073+0.9583j|)} \approx 5.5$ seconds. This estimated settling time agrees with the results in the figure.

Figure 78 shows the unit step response using state variable feedback to place the closed loop poles at 0.1, 0.2, 0.3 and 0.8. This produces a state feedback gain vector

$$K = \begin{bmatrix} -0.1165 & 0.0044 & 0.1280 & -0.0039 \end{bmatrix}$$

and prefilter gain $G_{pf} = 0.0086$. We can estimate the settling time of this system as $T_s \approx \frac{-4(0.05)}{\ln(0.8)} \approx 0.9$ seconds, which agrees fairly well with the results shown in the figure.

Figure 79 shows the unit step response when the closed loop poles are located at 0.1, 0.2, 0.3 and $-0.3 \pm 0.1j$. This produces a state feedback gain vector

$$K = \begin{bmatrix} 0.1179 & 0.0089 & -0.0796 & 0.0030 \end{bmatrix}$$

and prefilter gain $G_{pf} = 0.1050$. We can estimate the settling time of this system as $T_s \approx \frac{-4(0.05)}{\ln(|-0.3+0.1j|)} \approx 0.17$ seconds, which agrees fairly well with the results shown in the figure.

Figure 80 shows the unit step response when the closed loop poles are located at 0.1, 0.2, 0.3 and $0.3 \pm 0.1j$. This produces a state feedback gain vector

$$K = \begin{bmatrix} -0.0468 & 0.0059 & 0.0566 & -0.0025 \end{bmatrix}$$

and prefilter gain $G_{pf} = 0.0309$. We can estimate the settling time of this system as $T_s \approx \frac{-4(0.05)}{\ln(|0.3+0.1j|)} \approx 0.17$ seconds. This settling time does not agree very well with the response of the system shown in the Figure. This is because our estimate of the settling time is based on a

model having only poles and no zeros. In the previous examples the poles of the system were “far” from the zeros of the system, and had little effect on the step response of the system. In this example, the poles of the system are “near” the zeros of the system, and the zeros are affecting the step response of the system.

Figure 80 shows the unit step response when all the closed loop poles are located at 0. This is *deadbeat response* and produces a state feedback gain vector

$$K = \begin{bmatrix} 0.0765 & 0.0082 & -0.0463 & 0.0014 \end{bmatrix}$$

and prefilter gain $G_{pf} = 0.0858$. Since this is a fourth order system, the settling time is four time steps. Since this is deadbeat response and the zeros of the system do not change, we can determine the system difference equation as follows:

$$\begin{aligned} \frac{Y(z)}{U(z)} &= (0.0858) \frac{4.018z^3 + 2.048z^2 + 1.939z + 3.65}{z^4} \\ &= 0.3447z^{-2} + 0.1758z^{-3} + 0.1663z^{-3} + 0.3132z^{-4} \end{aligned}$$

or

$$y(k) = 0.3447u(k-1) + 0.1758u(k-2) + 0.1663u(k-3) + 0.3132u(k-4)$$

Since we assume the system is initially at rest with no initial conditions (all initial conditions are zero), we have

$$\begin{aligned} y(0) &= 0 \\ y(1) &= 0.3447 \\ y(2) &= 0.3447 + 0.1758 = 0.5205 \\ y(3) &= 0.3447 + 0.1758 + 0.1663 = 0.6868 \\ y(4) &= 0.3447 + 0.1758 + 0.1663 + 0.3132 = 1.0000 \end{aligned}$$

All $y(k) = 1$ for $k \geq 4$. These results match those in the Figure.

16.3 General Guidelines for State Feedback Pole Locations

As a general rule the closer in magnitude the system poles are to the origin, the faster the response of the system. Just as with continuous time systems, if there are multiple poles, the response of the *dominant* poles will dominate the response of the overall system. In this case, the poles with the largest magnitude will dominate the response and system will not reach steady state until the response of these poles have subsided. However, it should also be noted that placing the poles closer to the origin increases the system bandwidth and make the system less robust and more sensitive to noise, external disturbances, and errors in the mathematical model of the system. In addition, poles closer to the origin increases the control effort. Often, the designer needs to use some iterative “guess and check” to see if the system preforms adequately.

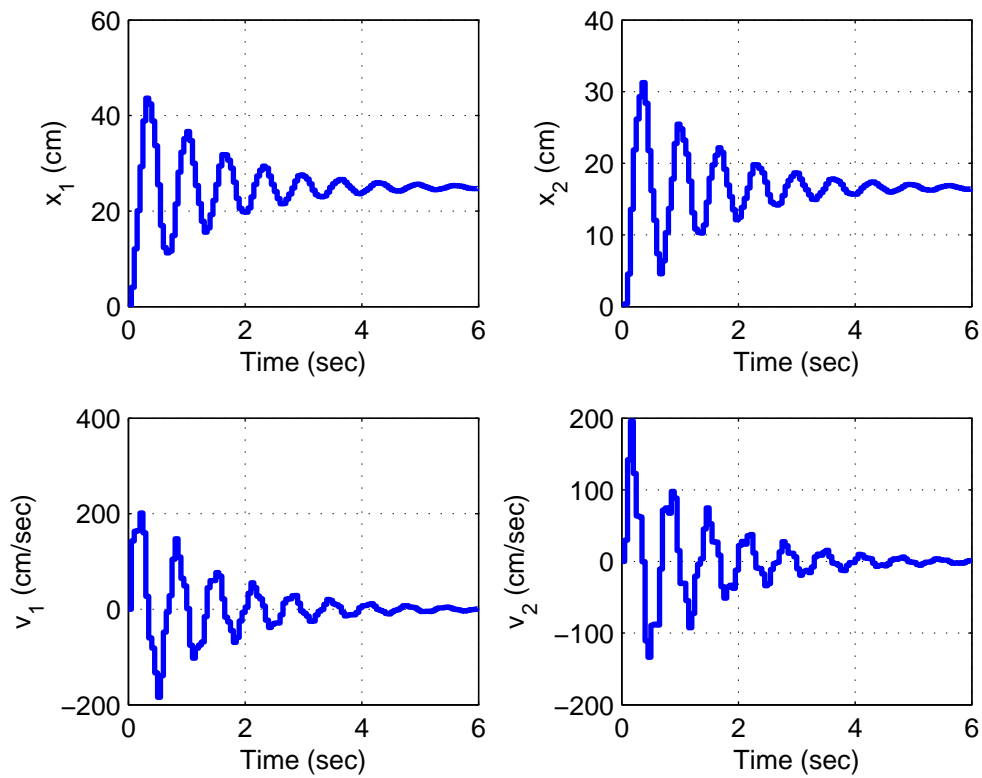


Figure 77: Open loop step response of fourth order system. The poles of the plant are at $-0.1073 \pm 0.9583j$ and $0.8555 \pm 0.4437j$. This system also has zeros at -0.9732 and $0.2317 \pm 0.9380j$.

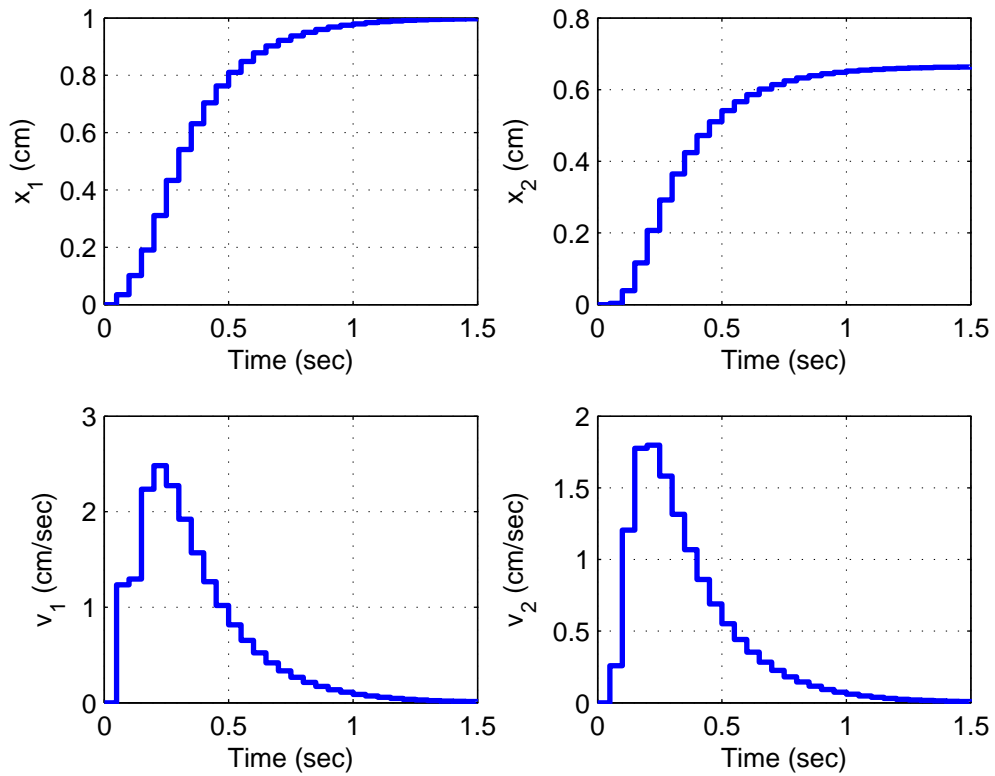


Figure 78: Closed loop step response of fourth order system with state variable feedback. The poles of the closed loop system are at 0.1, 0.2, 0.3 and 0.8. This system zeros remain at -0.9732 and $0.2317 \pm 0.9380j$.

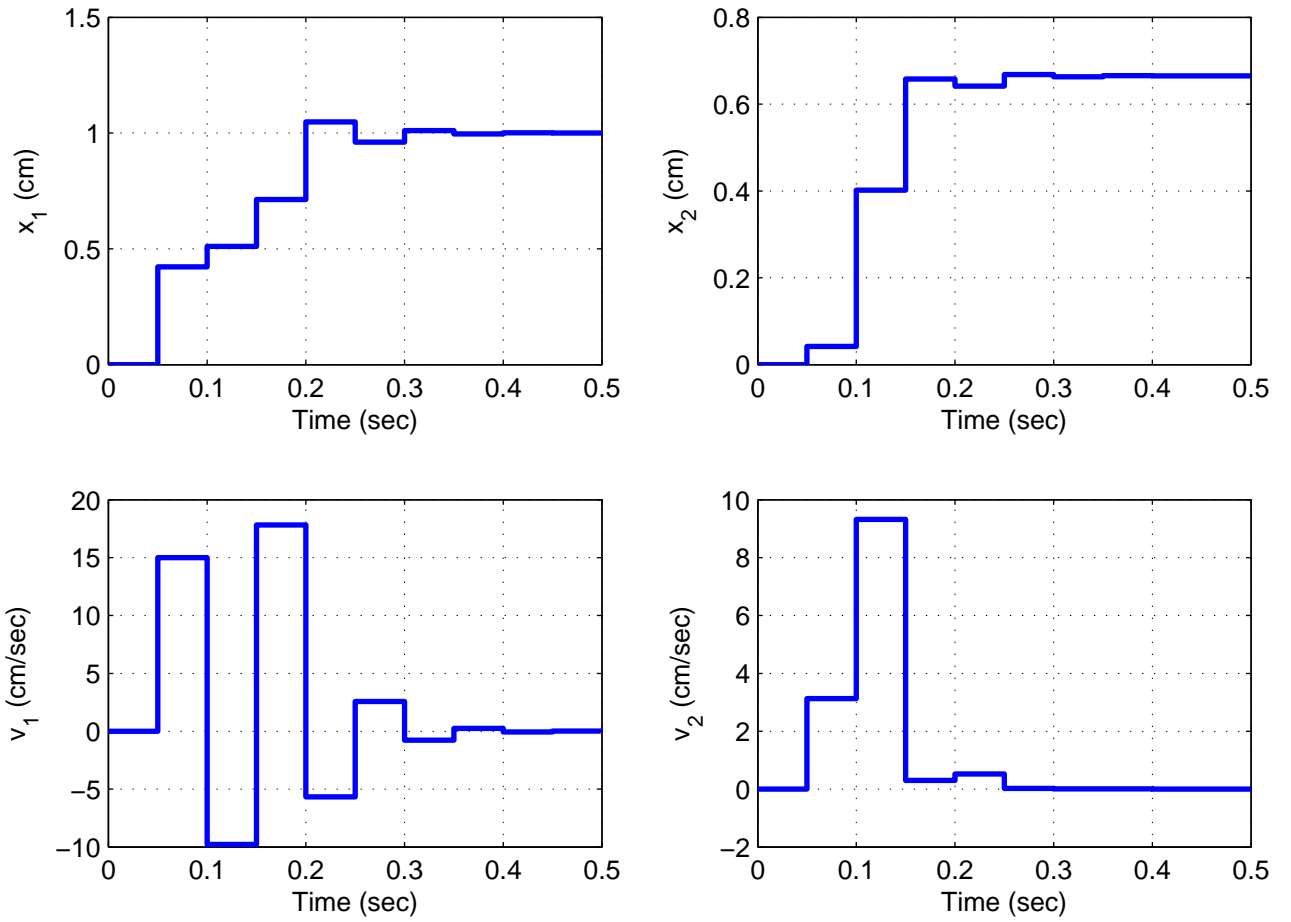


Figure 79: Closed loop step response of fourth order system with state variable feedback. The poles of the closed loop system are at $0.1, 0.2$ and $-0.3 \pm 0.1j$. This system zeros remain at -0.9732 and $0.2317 \pm 0.9380j$.

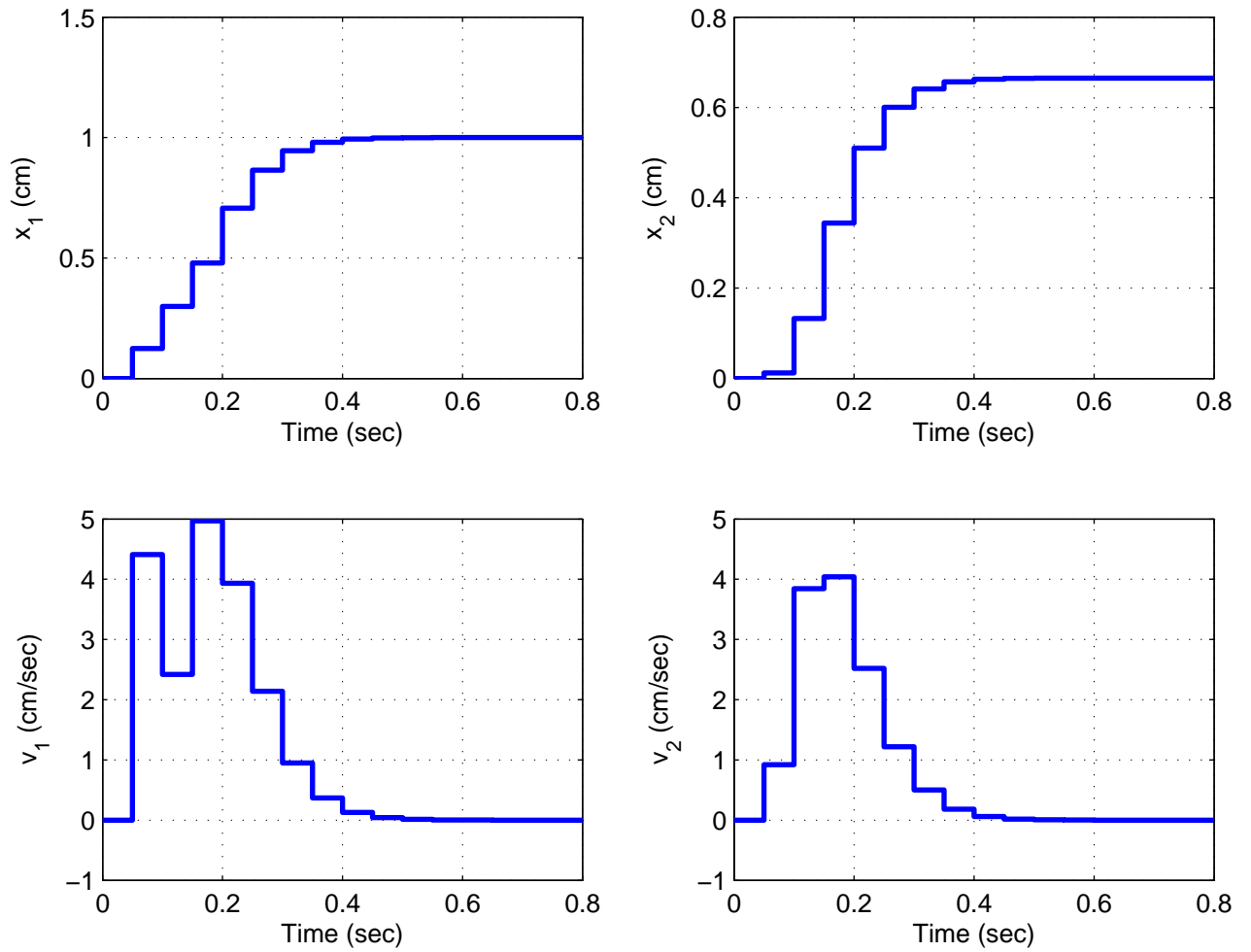


Figure 80: Closed loop step response of fourth order system with state variable feedback. The poles of the closed loop system are at $0.1, 0.2$ and $0.3 \pm 0.1j$. This system zeros remain at -0.9732 and $0.2317 \pm 0.9380j$.

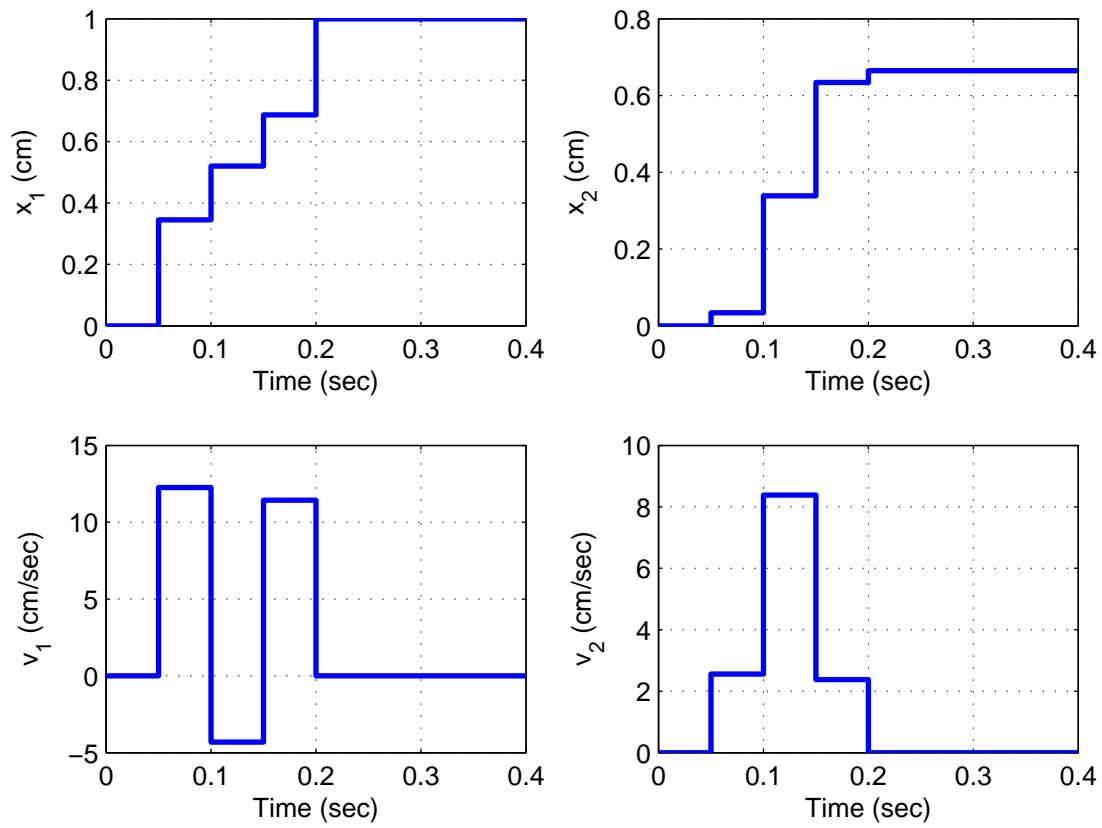


Figure 81: Closed loop step response of fourth order system with state variable feedback. The poles of the closed loop system are at 0. This is *deadbeat response*. This system zeros remain at -0.9732 and $0.2317 \pm 0.9380j$.

17 Frequency Domain Analysis

We have so far been primarily interested in analyzing and designing control systems in the time-domain or in the s -domain. A different approach is to look directly in the frequency domain. This method has the advantage that for many systems the experimental determination of the frequency response is fairly straightforward to determine, while an accurate transfer function may be more difficult to obtain.

The most important property of any control system is that the system remain stable. We will thus begin our analysis by determining how close to being unstable is our current system. This leads to the definitions of the phase and gain margins. Then we will introduce a phase-lead controller that will allow us to increase the phase margin of our system if we want.

17.1 Phase and Gain Margins

Let's consider a closed loop system with transfer function given by

$$G_0(s) = \frac{G(s)}{1 + G(s)H(s)}$$

We know that the system will be asymptotically stable if all of the poles of the closed loop system are in the (open) left half plane (the real parts of the poles are negative). When the system is marginally stable, the poles are on the $j\omega$ axis. We can view this as the point at which the system is about to become unstable. If we have poles on the $j\omega$ axis, then we have

$$1 + H(j\omega)G(j\omega) = 0$$

We can rewrite this as

$$H(j\omega)G(j\omega) = -1$$

We can then break this into two different conditions, a *magnitude condition* and a *phase condition*:

$$\begin{aligned} \text{magnitude condition: } |H(j\omega)G(j\omega)| &= 1 \\ \text{phase condition: } \angle H(j\omega)G(j\omega) &= \pm 180^\circ \end{aligned}$$

When both of these conditions are met, the system is marginally stable. However, we can also use these conditions to determine *how close our system is to being unstable*. The additional amplitude and phase in the product $G(j\omega)H(j\omega)$ before the poles of the closed loop system are on the $j\omega$ axis are measures of the allowable tolerances or changes in $G(j\omega)H(j\omega)$ for system stability. This leads to the ideas of *gain margins* and *phase margins*.

We first define a *phase crossover frequency*, ω_p , to be any frequency at which we meet the phase condition. That is, at a phase crossover frequency we have

$$\angle G(j\omega_p)H(j\omega_p) = \pm 180^\circ$$

We then define the *gain margin* of a feedback system to be the additional amplitude necessary to make $|G(j\omega_p)H(j\omega_p)| = 1$. If there is more than one phase crossover frequency, we use the phase crossover frequency that produces the smallest gain margin. Specifically, we can determine the gain margin using the formula

$$|G(j\omega_p)H(j\omega_p)| \times \text{Gain Margin} = 1$$

which we can rearrange as

$$\text{Gain Margin} = \frac{1}{|G(j\omega_p)H(j\omega_p)|}$$

We usually measure the gain margin in dB, so we have

$$\text{Gain Margin}_{dB} = 20 \log_{10}(1) - 20 \log_{10}(|G(j\omega_p)H(j\omega_p)|) = -20 \log_{10}(|G(j\omega_p)H(j\omega_p)|)$$

or simply

$$\text{Gain Margin}_{dB} = -|G(j\omega_p)H(j\omega_p)|_{dB}$$

Note that if there is no phase crossover frequency the gain margin is infinite. Also, if there is more than one gain margin, then the gain margin for the system is the *smallest* gain margin of the possibilities.

Next we define a *gain crossover frequency*, ω_g , to be any frequency at which we meet the magnitude condition. That is, at a gain crossover frequency we have

$$|G(j\omega_g)H(j\omega_g)| = 1$$

or

$$|G(j\omega_g)H(j\omega_g)|_{dB} = 0$$

The *phase margin* is the additional negative phase necessary to make the phase of $G(j\omega_g)H(j\omega_g)$ equal to $\pm 180^\circ$, whichever is closer. We can write this condition as

$$\angle G(j\omega_g)H(j\omega_g) - \text{Phase Margin} = \pm 180^\circ$$

Note that if there is no gain crossover frequency the phase margin is infinite. If there is more than one phase margin, then the phase margin for a system is the *smallest* phase margin of the possibilities.

Finally, for a *minimum phase system* (all poles and zeros in the left half plane), both the phase and gain margins must be positive for system stability. A reasonable rule of thumb is that the phase margin for a system should be 30 to 60 degrees and the gain margin should be greater than 6 dB.

Example. Consider the feedback system where we have $G(s) = 20/(s+2)$ and $H(s) = 1/(s^2 + 4s + 1)$. The Bode plot of the open loop system $G(s)H(s)$ is displayed in Figure 82. In this

figure, the gain crossover frequency is displayed as a diamond and the phase crossover frequency is displayed as a circle. For this example, we have $\omega_g \approx 1.8$ rad/sec and $\omega_p \approx 3.0$ rad/sec. At the gain crossover frequency the phase of the system is approximately -150° , or

$$\angle G(j\omega_g)H(j\omega_g) \approx -150^\circ$$

We then have

$$\begin{aligned} \angle G(j\omega_g)H(j\omega_g) - \text{Phase Margin} &= -180^\circ \\ -150^\circ - \text{Phase Margin} &= -180^\circ \end{aligned}$$

which gives a phase margin of approximately 30° . Similarly, at the phase crossover frequency we have

$$|G(j\omega_p)H(j\omega_p)| \approx -9dB$$

So the gain margin is approximately 9 dB. Since both the phase and gain margins are positive, and there are no right half plane zeros, this system is stable. In fact, the poles of the closed loop system are at -50.9 and at $-0.46 \pm 2.03j$.

Example. Consider the feedback system where we have $G(s) = 50/(s^2 + s + 100)$ and $H(s) = 1$. The Bode plot of the open loop system $G(s)H(s)$ is displayed in Figure 83. In this figure, there are two gain crossover frequencies (displayed as diamonds) and there is no phase crossover frequency. The first gain crossover frequency is $\omega_g \approx 7.1$ rad/sec. At this gain crossover frequency the phase of the system is approximately -8° , or

$$\angle G(j\omega_g)H(j\omega_g) \approx -8^\circ$$

We then have

$$\begin{aligned} \angle G(j\omega_g)H(j\omega_g) - \text{Phase Margin} &= -180^\circ \\ -8^\circ - \text{Phase Margin} &= -180^\circ \end{aligned}$$

which gives a phase margin of approximately 172° . The second gain crossover frequency is $\omega_g \approx 12.2$ rad/sec, and at this frequency we have

$$\angle G(j\omega_g)H(j\omega_g) \approx -166^\circ$$

We then have

$$\begin{aligned} \angle G(j\omega_g)H(j\omega_g) - \text{Phase Margin} &= -180^\circ \\ -166^\circ - \text{Phase Margin} &= -180^\circ \end{aligned}$$

which gives a phase margin of approximately 14° . The phase margin of the system is the smaller of these two, so the phase margin of the system is 14° . Since both the phase and gain margins of the system are positive, and there are no right hand plane zeros, the system is stable., The poles for this closed loop system are at $-0.50 \pm 12.24j$.

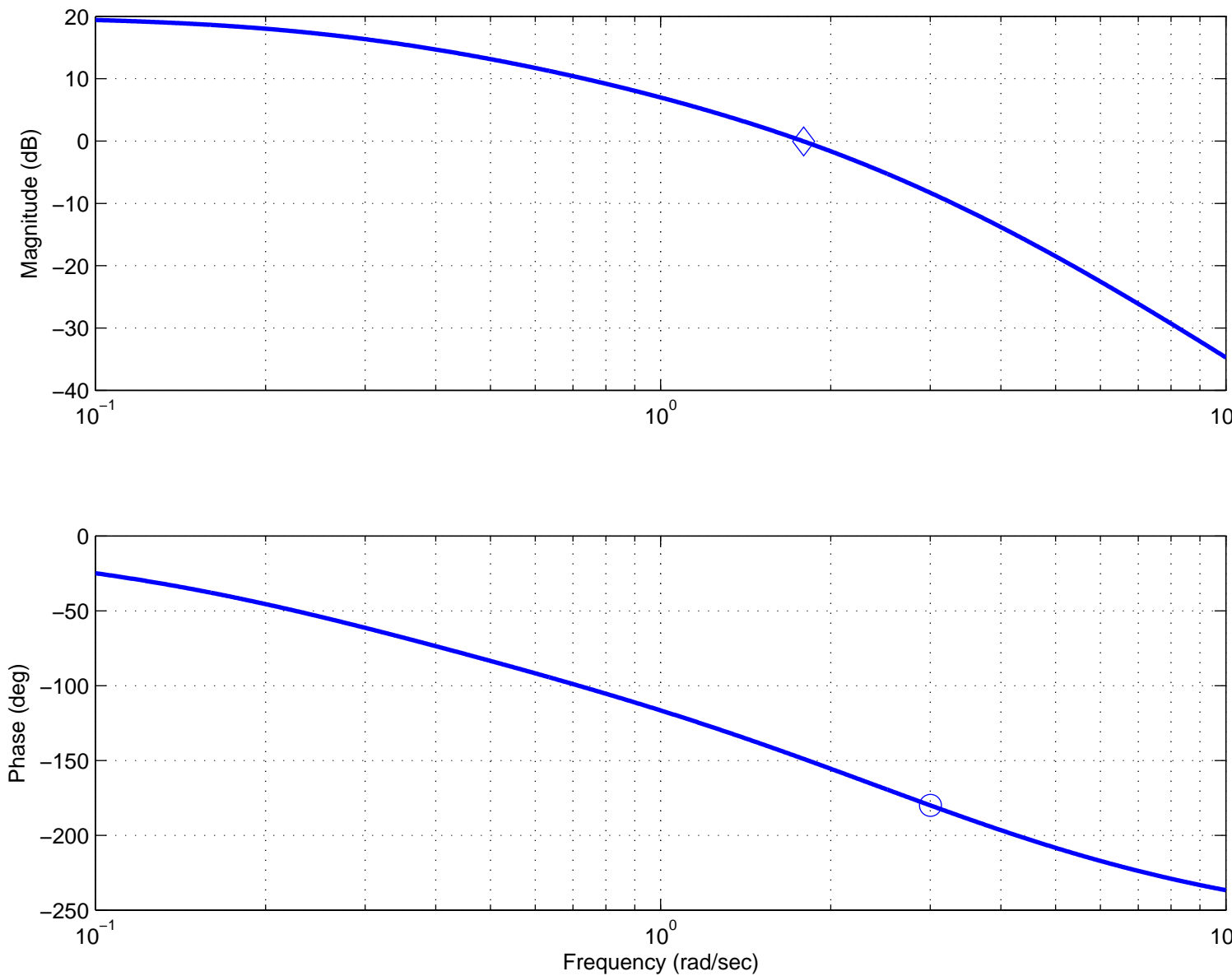


Figure 82: Bode plot of $G(s)H(s) = 20/(s + 2)(s^2 + 4s + 1)$. The gain crossover frequency is shown as a circle, and the phase crossover frequency is shown as a diamond. The gain margin for this system is 9 dB and the phase margin is 30° .

Example. Consider the feedback system where we have $G(s) = (s^2 + 50s + 1000)/(s^3 + s^2 + 100s)$ and $H(s) = 1$. The Bode plot of the open loop system $G(s)H(s)$ is displayed in Figure 84. In this figure, there are two phase crossover frequencies (displayed as circles) and one gain crossover frequency (displayed as a diamond). The gain crossover frequency is $\omega_g \approx 13$ rad/sec. At the gain crossover frequency the phase of the system is approximately -220° , or

$$\angle G(j\omega_g)H(j\omega_g) \approx -220^\circ$$

We then have

$$\begin{aligned} \angle G(j\omega_g)H(j\omega_g) - \text{Phase Margin} &= -180^\circ \\ -220^\circ - \text{Phase Margin} &= -180^\circ \end{aligned}$$

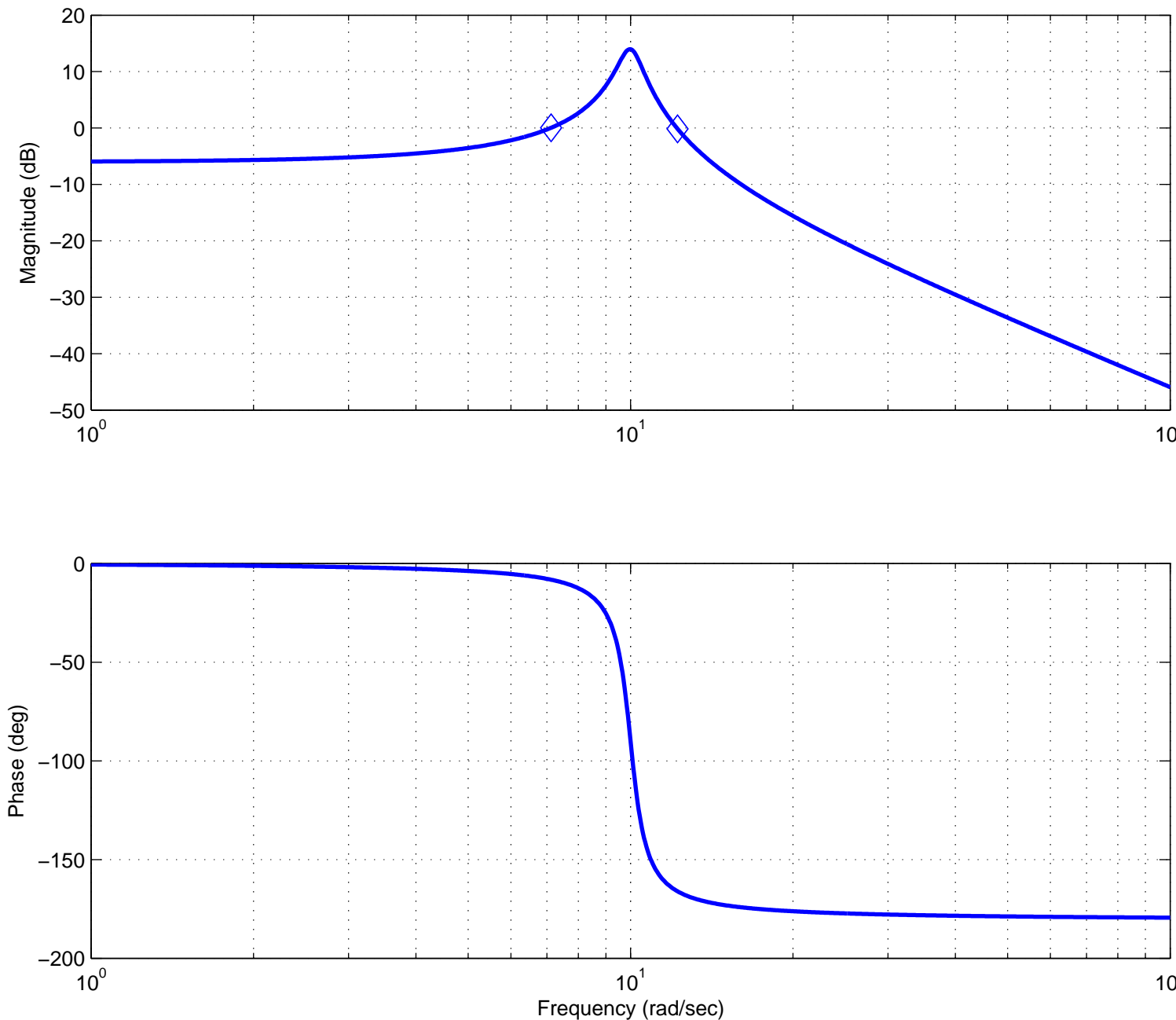


Figure 83: Bode plot of $G(s)H(s) = 50/(s^2 + s + 100)$. The gain crossover frequencies are shown as circles, and there are no phase crossover frequencies. The gain margin for this system is infinite and the phase margin is 14° .

which gives a phase margin of approximately -40° . The two phase crossover frequencies are $\omega_p \approx 10$ rad/sec and $\omega_p \approx 30$ rad/sec. At the first phase crossover frequency we have $|G(j\omega_p)H(j\omega_p)| \approx 20$ dB so this gain margin is approximately -20 dB. At the second phase crossover frequency we have $|G(j\omega_p)H(j\omega_p)| \approx -25$ dB so this gain margin is approximately 25 dB. Hence the gain margin for the system is -20 dB. Since this system has no right half plane zeros and the gain and phase margins are negative, the system is unstable (if would be unstable if *either* the gain or phase margin was negative). The poles of the closed loop system are at

-5.81 and at $1.90 \pm 12.98j$ and the system is clearly unstable.

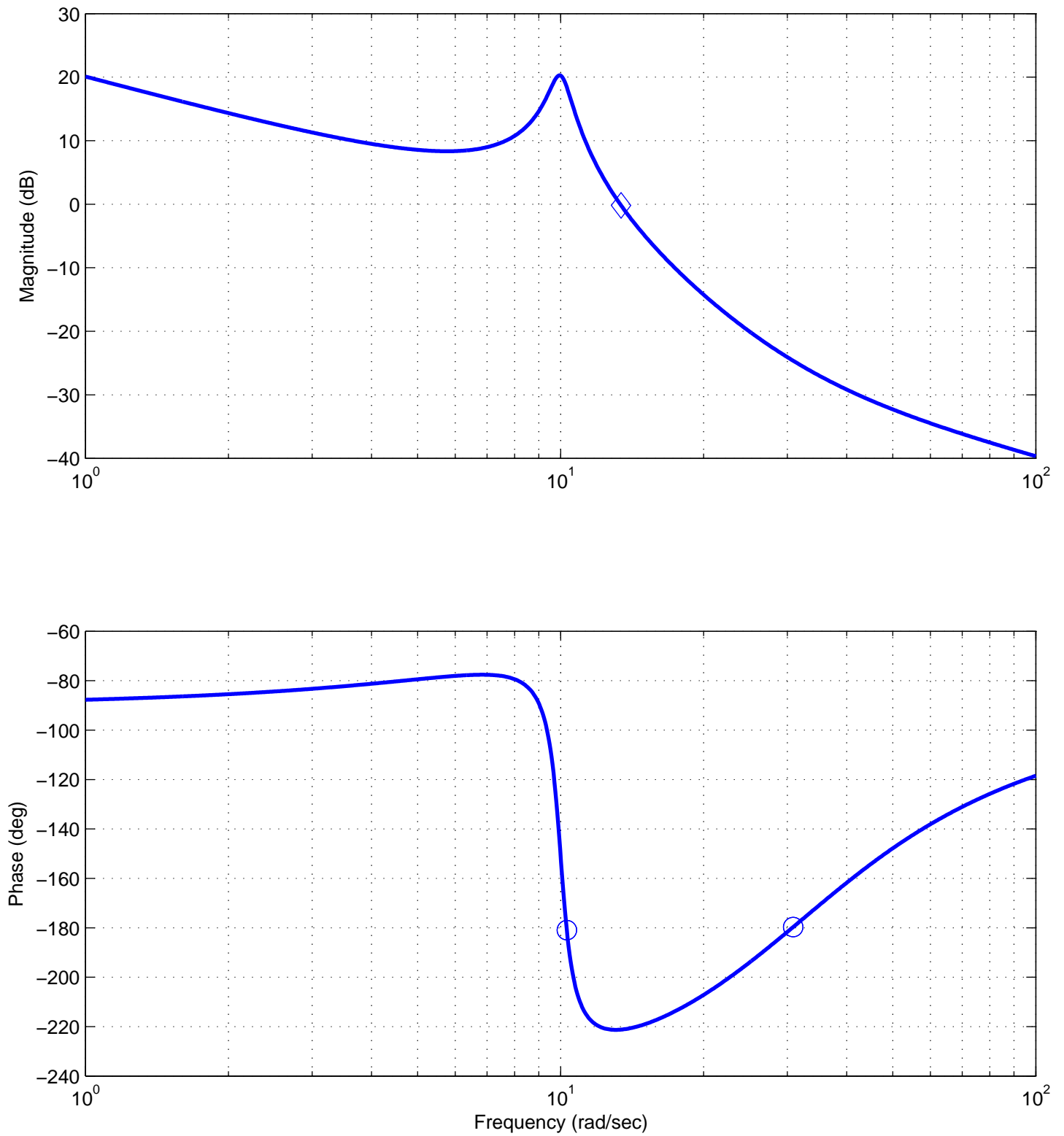


Figure 84: Bode plot of $G(s)H(s) = \frac{s^2 + 50s + 1000}{s^3 + s^2 + 100s}$. The gain crossover frequencies are shown as circles, and the phase crossover frequency is shown as a diamond. The gain margin for this system is -20 dB and the phase margin is -40° .

18 Lead Controllers for Increasing Phase Margin

Generally, if we want to increase the gain margin of a system we just change the gain. To increase the phase margin we usually want to add a controller to add phase to the system. A phase lead controller is the most commonly used type of controller for this purpose. One form of a lead controller is

$$G_c(s) = K \frac{Ts + 1}{\alpha Ts + 1} = K_c \alpha \frac{Ts + 1}{\alpha Ts + 1} = K_c \frac{s + \frac{1}{T}}{s + \frac{1}{\alpha T}}$$

Here α is called the *attenuation factor*, and is mathematically limited to be between 0 and 1, $0 < \alpha < 1$. For practical controllers we usually want $\alpha > 0.05$. Remember that for a *lead* controller the zero is always to the right of the pole.

We want to find an expression for the maximum phase ϕ_m we can get from a phase lead controller. To determine the frequency response, we have

$$\begin{aligned} G_c(j\omega) &= K_c \left(\frac{j\omega + \frac{1}{T}}{j\omega + \frac{1}{\alpha T}} \right) \\ &= K_c \left(\frac{j\omega + \frac{1}{T}}{j\omega + \frac{1}{\alpha T}} \right) \times \left(\frac{-j\omega + \frac{1}{\alpha T}}{-j\omega + \frac{1}{\alpha T}} \right) \\ &= K_c \left(\frac{\omega^2 - j\omega \left(\frac{1}{T} \right) + j\omega \left(\frac{1}{\alpha T} \right) + \frac{1}{\alpha T^2}}{\omega^2 + \left(\frac{1}{\alpha T} \right)^2} \right) \\ &= K_c \frac{\left(\omega^2 + \frac{1}{\alpha T^2} \right) + j\omega \left(\frac{1}{\alpha T} - \frac{1}{T} \right)}{\omega^2 + \left(\frac{1}{\alpha T} \right)^2} \end{aligned}$$

Since we know $K_c > 0$ and $\omega^2 + \left(\frac{1}{\alpha T} \right)^2 > 0$, the entire phase is determined by the numerator. Hence we have

$$\angle G_c(j\omega) = \tan^{-1} \left\{ \frac{\omega \left(\frac{1}{\alpha T} - \frac{1}{T} \right)}{\omega^2 + \frac{1}{\alpha T^2}} \right\}$$

Next we need to find the frequency at which this phase angle reaches its maximum. This means we are going to need to take derivatives of the arctangent function. From calculus we know

$$\frac{d}{d\omega} \tan^{-1}(\theta(\omega)) = \frac{1}{1 + \theta(\omega)^2} \frac{d}{d\omega} \theta(\omega)$$

where for us

$$\theta(\omega) = \frac{\omega \left(\frac{1}{\alpha T} - \frac{1}{T} \right)}{\omega^2 + \frac{1}{\alpha T^2}}$$

Since we are going to take the derivative and set it equal to zero to determine the frequency than maximizes the function, we do not need to worry about the $\frac{1}{1+\theta^2}$ term, since it will always will be positive. We then have

$$\frac{d}{d\omega} \theta(\omega) = \frac{\left(\frac{1}{\alpha T} - \frac{1}{T} \right) \left(\omega^2 + \frac{1}{\alpha T^2} \right) - 2\omega \left(\omega \left[\frac{1}{\alpha T} - \frac{1}{T} \right] \right)}{\left(\omega^2 + \frac{1}{\alpha T^2} \right)^2}$$

The denominator of this function is always positive, and the numerator reduces to

$$\omega^2 + \frac{1}{\alpha T^2} - 2\omega^2 = 0$$

which leads to $\omega = \frac{1}{T\sqrt{\alpha}}$. Thus the frequency at which the maximum phase angle is achieved, ω_m , is

$$\omega_m = \frac{1}{T\sqrt{\alpha}}$$

Now that we know the frequency at which the maximum occurs, we need to find an expression for the maximum phase angle our controller will provide. Hence we have

$$\begin{aligned}\phi_m &= \angle G_c(j\omega_m) \\ &= \tan^{-1} \left(\frac{\frac{1}{T\sqrt{\alpha}} \left(\frac{1}{\alpha T} - \frac{1}{T} \right)}{\frac{1}{\alpha T^2} + \frac{1}{\alpha T^2}} \right) \\ &= \tan^{-1} \left(\frac{\alpha T^2 \left(\frac{1}{T\sqrt{\alpha}} \right) \left(\frac{1}{\alpha T} - \frac{1}{T} \right)}{2} \right) \\ &= \tan^{-1} \left(\frac{\sqrt{\alpha} \left(\frac{1}{\alpha} - 1 \right)}{2} \right) \\ &= \tan^{-1} \left(\frac{\frac{1}{\sqrt{\alpha}} - \sqrt{\alpha}}{2} \right)\end{aligned}$$

For reasons that will become clear when we start to using the lead controller in designing systems, we want to rewrite this expression. We can visualize the above trigonometric relationship using the triangle shown in Figure 85. From this graph, we can compute the length of the hypotenuse as

$$R^2 = \left(\frac{1}{\sqrt{\alpha}} - \sqrt{\alpha} \right)^2 + 2^2 = \frac{1}{\alpha} - 2 + \alpha + 4 = \frac{1}{\alpha} + 2 + \alpha = \left(\frac{1}{\sqrt{\alpha}} + \sqrt{\alpha} \right)^2$$

We can then write

$$\sin(\phi_m) = \frac{\frac{1}{\sqrt{\alpha}} - \sqrt{\alpha}}{R} = \frac{\frac{1}{\sqrt{\alpha}} - \sqrt{\alpha}}{\frac{1}{\sqrt{\alpha}} + \sqrt{\alpha}} = \frac{1 - \alpha}{1 + \alpha}$$

We can rewrite this expression in a more useful form as

$$\begin{aligned}1 - \alpha &= (1 + \alpha) \sin(\phi_m) = \sin(\phi_m) + \alpha \sin(\phi_m) \\ 1 - \sin(\phi_m) &= \alpha + \alpha \sin(\phi_m) = \alpha (1 + \sin(\phi_m))\end{aligned}$$

or

$$\alpha = \frac{1 - \sin(\phi_m)}{1 + \sin(\phi_m)}$$

This equation is very useful for design, since if we know the phase angle ϕ_m we want our controller to add, we now have a way of determining α . Finally we need to determine how much gain the phase lead controller will add to the system. We have

$$G_c(s) = K \frac{Ts + 1}{\alpha Ts + 1}$$

Here the gain K will be used for controlling the steady state error so we do not want to include it in our calculations for the gain contributed by the controller. Hence we want

$$\begin{aligned} \left| \frac{G_c(s)}{K} \right|_{s=j\omega_m} &= \left| \frac{T \left(j \frac{1}{\sqrt{\alpha T}} \right) + 1}{\alpha T \left(j \frac{1}{\sqrt{\alpha T}} \right) + 1} \right| \\ &= \left| \frac{\frac{j}{\sqrt{\alpha}} + 1}{j\sqrt{\alpha} + 1} \right| = \sqrt{\frac{\frac{1}{\alpha} + 1}{\alpha + 1}} = \sqrt{\frac{\frac{1}{\alpha}(1 + \alpha)}{\alpha + 1}} = \frac{1}{\sqrt{\alpha}} \end{aligned}$$

Hence our controller will add $20 \log_{10} \left(\frac{1}{\sqrt{\alpha}} \right) = -20 \log_{10} \sqrt{\alpha}$ dB of gain to the system.

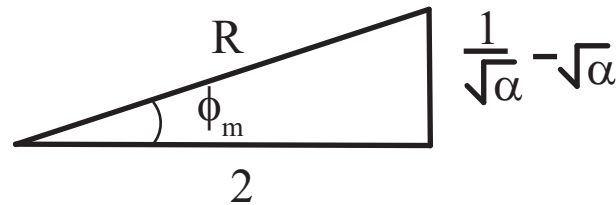


Figure 85: Triangle depicting $\phi_m = \arctan\left(\left(\frac{1}{\sqrt{\alpha}} - \sqrt{\alpha}\right)/2\right)$. From this triangle we can determine that $R = \left(\frac{1}{\sqrt{\alpha}} + \sqrt{\alpha}\right)^2$.

18.1 Algorithm for Controller Design Using Bode Plots

The primary function of the lead compensator is to reshape the frequency response curve by adding phase to the system. The phase lead compensator also adds gain to the system.

[1] Assume the compensator has the form

$$G_c(s) = K_c \frac{s + \frac{1}{T}}{s + \frac{1}{\alpha T}} = K_c \alpha \frac{Ts + 1}{\alpha Ts + 1} = K \frac{Ts + 1}{\alpha Ts + 1}$$

Determine K to satisfy the static error constant requirements.

[2] Using this value of K , draw the Bode diagram of $KG(s)H(s)$. Determine the phase margin.

[3] Determine the necessary phase-lead angle to be added to the system. Add an additional 5° to 12° to the phase lead required, because the phase lead compensator shifts the phase crossover

frequency to the right and decreases the phase margin. ϕ_m is then the total phase our compensator needs to add to the system.

4 Determine α using

$$\alpha = \frac{1 - \sin(\phi_m)}{1 + \sin(\phi_m)}$$

Determine the magnitude where $KG(j\omega)H(j\omega)$ is equal to $-20 \log_{10}(\frac{1}{\sqrt{\alpha}}) = 10 \log_{10}(\alpha)$. This is the new gain crossover frequency $\omega_m = \frac{1}{T\sqrt{\alpha}}$, or $T = \frac{1}{\omega_m\sqrt{\alpha}}$.

Note: If $\alpha < 0.05$, you will probably need two compensators. Choose a phase angle ϕ_m that produces an acceptable α . Finish the design, then treat $KG_c(s)G(s)H(s)$ as the system and go back to step 2.

5 Determine the corner frequencies of the compensator as $z = \frac{1}{T}$ and $p = \frac{1}{\alpha T}$.

6 Determine $K_c = \frac{K}{\alpha}$.

7 Check the gain and phase margins to be sure they are satisfactory.

18.2 Examples

This section includes two examples using the Bode lead controller design algorithm from the previous section.

Example. Assume we have the open loop system

$$G(s) = \frac{1}{s(s+10)(s+50)}$$

and we want a steady state error for a unit ramp of 0.01 and a phase margin of approximately 60 degrees. We will use a phase lead controller of the form

$$G_c(s) = K \frac{Ts + 1}{T\alpha s + 1}$$

to try and accomplish both of these. First of all we take care of the steady state error,

$$K_v = \lim_{s \rightarrow 0} sG_c(s)G(s) = \frac{K}{500}$$

$$e_{ss} = \frac{1}{K_v} = \frac{500}{K} = 0.1$$

which means we need $K = 5000$.

Next we plot the Bode plot of the system using this gain, so in this case we plot the Bode plot of

$$\tilde{G}(s) = \frac{5000}{s(s+10)(s+50)}$$

This plot is shown in Figure 86. This Bode plot indicates a phase margin of approximately 43 degrees, so we need to add at least 17 degrees of phase to the system. Hence we will add 17 degrees plus a *fudge factor* of about 12 degrees. So the total phase we want to add to the system is

$$\phi_m = 17^\circ + 12^\circ = 29^\circ$$

Next we compute the attenuation factor

$$\alpha = \frac{1 - \sin(29^\circ)}{1 + \sin(29^\circ)} \approx 0.347$$

Next we find the place where the magnitude is equal to $20 \log_{10} \sqrt{\alpha}$,

$$20 \log_{10} \sqrt{0.346} = -4.6 \text{ dB}$$

This occurs at $\omega_m \approx 11.1$ rad/sec, this will be our new gain crossover frequency. Finally, we use the relationship $T = \frac{1}{\omega_m \sqrt{\alpha}}$ to determine that $T = 0.153$. Now we have all of the parameters we need for our controller,

$$G_c(s) = K \frac{Ts + 1}{T\alpha s + 1} = 5000 \frac{0.153s + 1}{0.05309s + 1}$$

Note that with this controller we can construct the Bode plot of the open loop system with the phase lead controller, as shown in Figure 87. This graphs shows an actual phase margin of 58.5 degrees. Often we may have to iterate or use something like Matlab's *sisotool* once we get an initial controller.

Example. Consider a system with plant

$$G_p(s) = \frac{10(s+10)}{(s+1)(s+20)^2}$$

and we want a steady state error equal to 0.05 for a unit step and a phase margin of 83 degrees (this is a bit extreme). We will again use a phase lead controller to accomplish this. To determine the required gain, we have

$$K_p = \lim_{s \rightarrow 0} G_c(s)G_p(s) = \frac{100K}{400} = \frac{K}{4}$$

$$e_{ss} = \frac{1}{1 + K_p} = \frac{1}{1 + \frac{K}{4}} = \frac{1}{20}$$

We will use $K = 80$ in this case. Next we plot the Bode plot of

$$G_c(s)G_p(s) = \frac{800(s+10)}{(s+1)(s+20)^2}$$

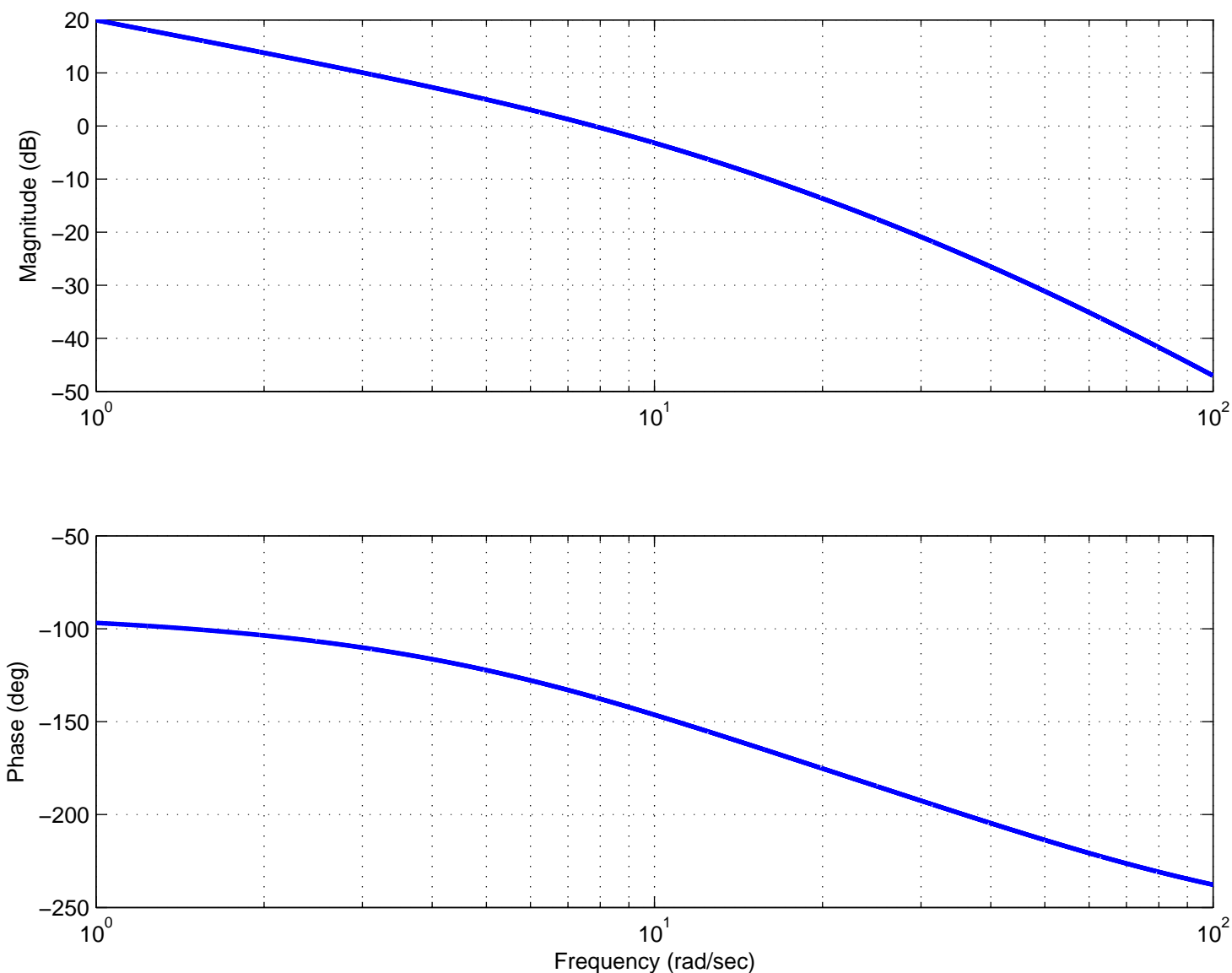


Figure 86: Bode plot of $G(s)H(s) = 5000/(s^3 + 60s^2 + 500s)$. The gain phase margin of this system is approximately 43° .

which is shown in Figure 88. This figure indicates a phase margin of approximately 63 degrees. We need to add approximately 20 degrees of phase plus a fudge factor. We will use a fudge factor of 5 degrees, so

$$\phi_m = 20^\circ + 5^\circ = 25^\circ$$

The attenuation factor is then

$$\alpha = \frac{1 - \sin(25^\circ)}{1 + \sin(25^\circ)} = 0.406$$

Then we need the frequency ω_m where the magnitude is equal to $10 \log_{10} \sqrt{\alpha} \approx -4dB$. This occurs at $\omega_m \approx 30.5$. Finally, using $T = \frac{1}{T\sqrt{\alpha}}$ we get $T \approx 0.051$. The transfer function for the phase lead controller is then

$$G_c(s) = 80 \frac{0.051s + 1}{0.071s + 1}$$

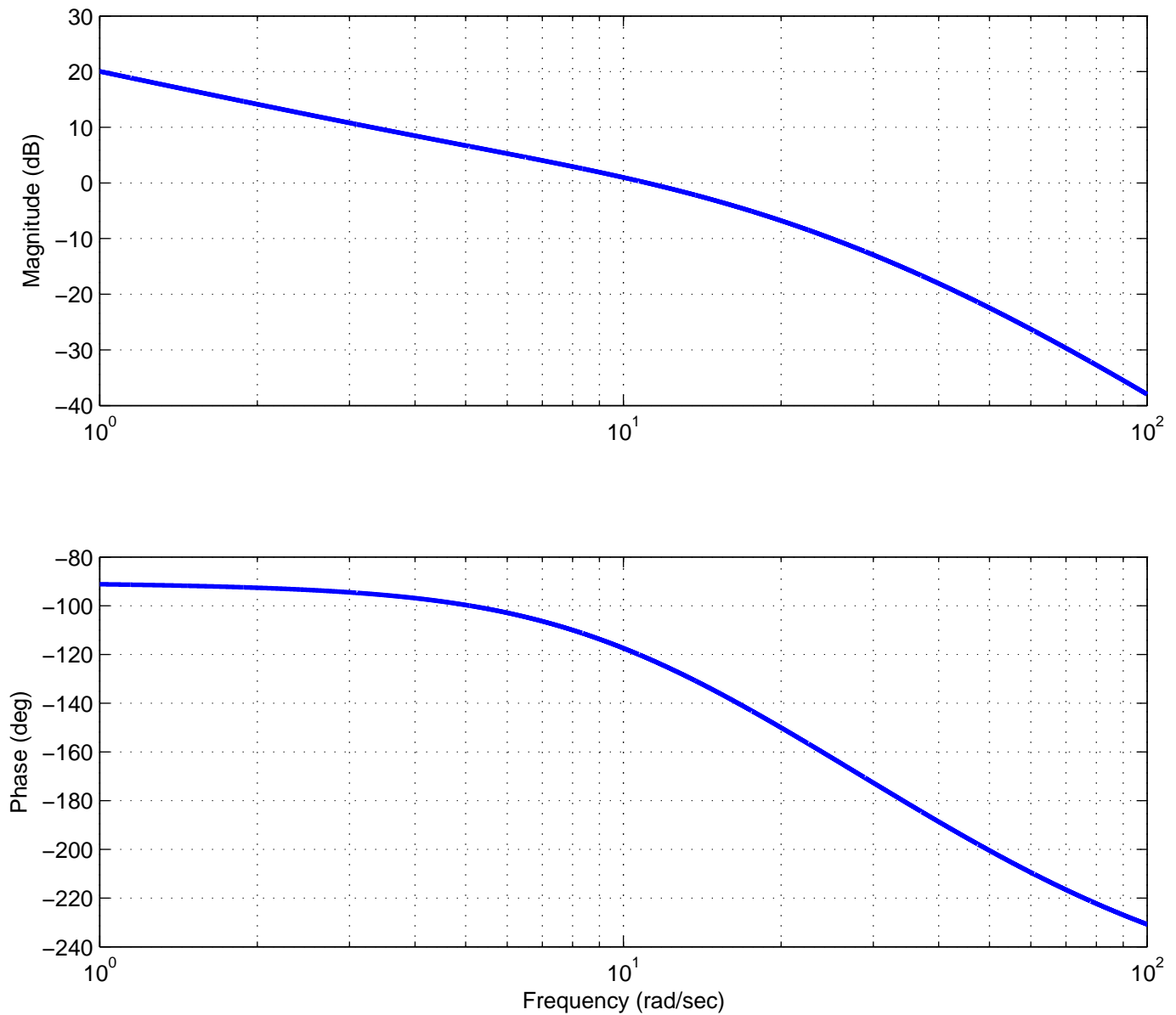


Figure 87: Bode plot of $G(s)H(s) = 1/(s^3 + 60s^2 + 500s)$ with phase lead controller $G_c(s) = 5000(0.153s + 1)/(0.05309s + 1)$. The gain phase margin of this system is now approximately 59° .

Figure 89 shows the Bode plot of the final system, which has a phase margin of approximately 75° . This initial design can now be modified using *sisotool*.

At this point you might be wondering why we do not just use software directly. The reason is that for many systems it is easier to measure the Bode plot directly and then deal directly with the measured Bode plot, rather than trying to fit the Bode plot to a transfer function (which will likely introduce more errors).

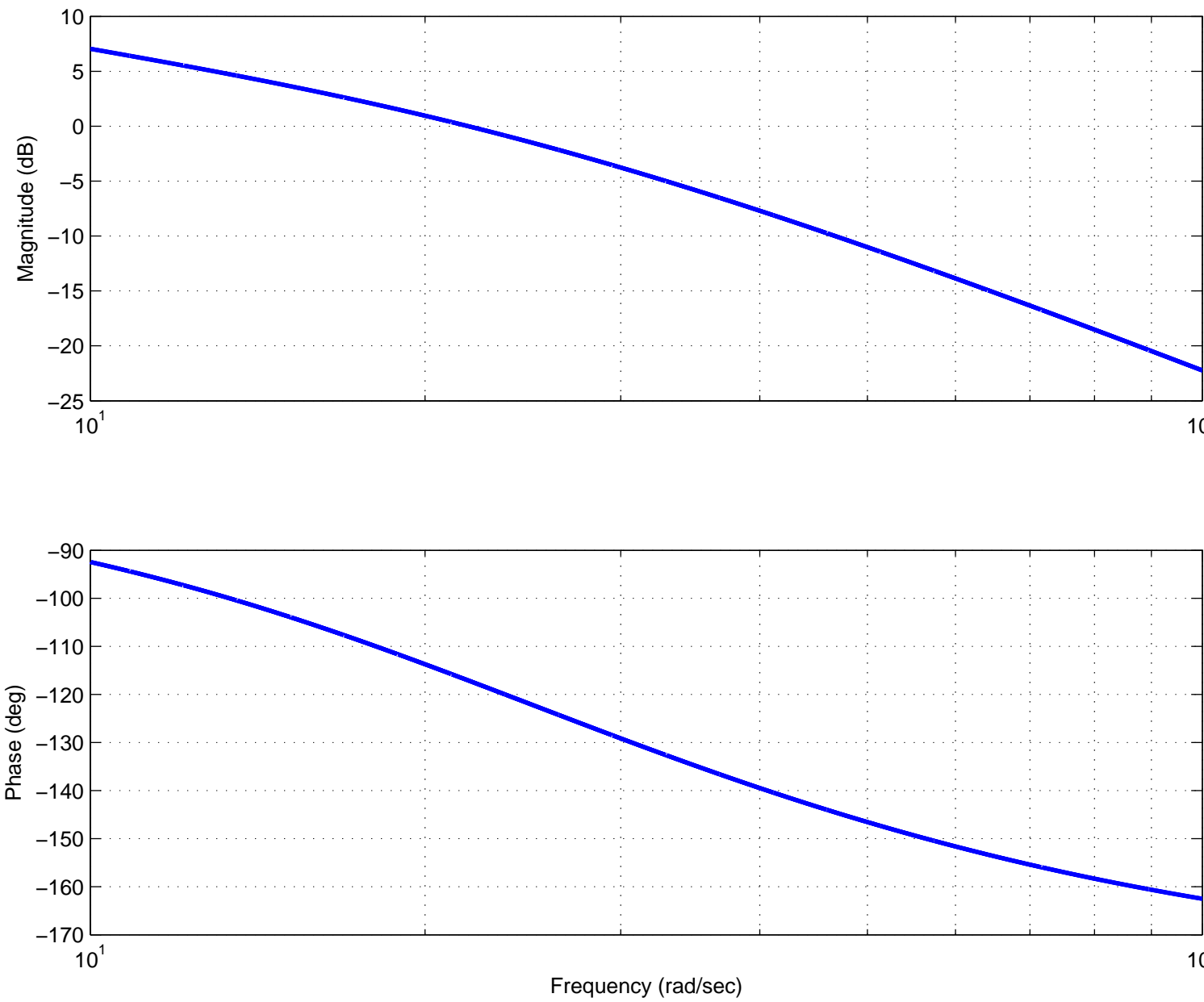


Figure 88: Bode plot of $G(s)H(s) = 800(s + 10)/(s + 1)(s + 20)^2$. The gain phase margin of this system is approximately 63° .

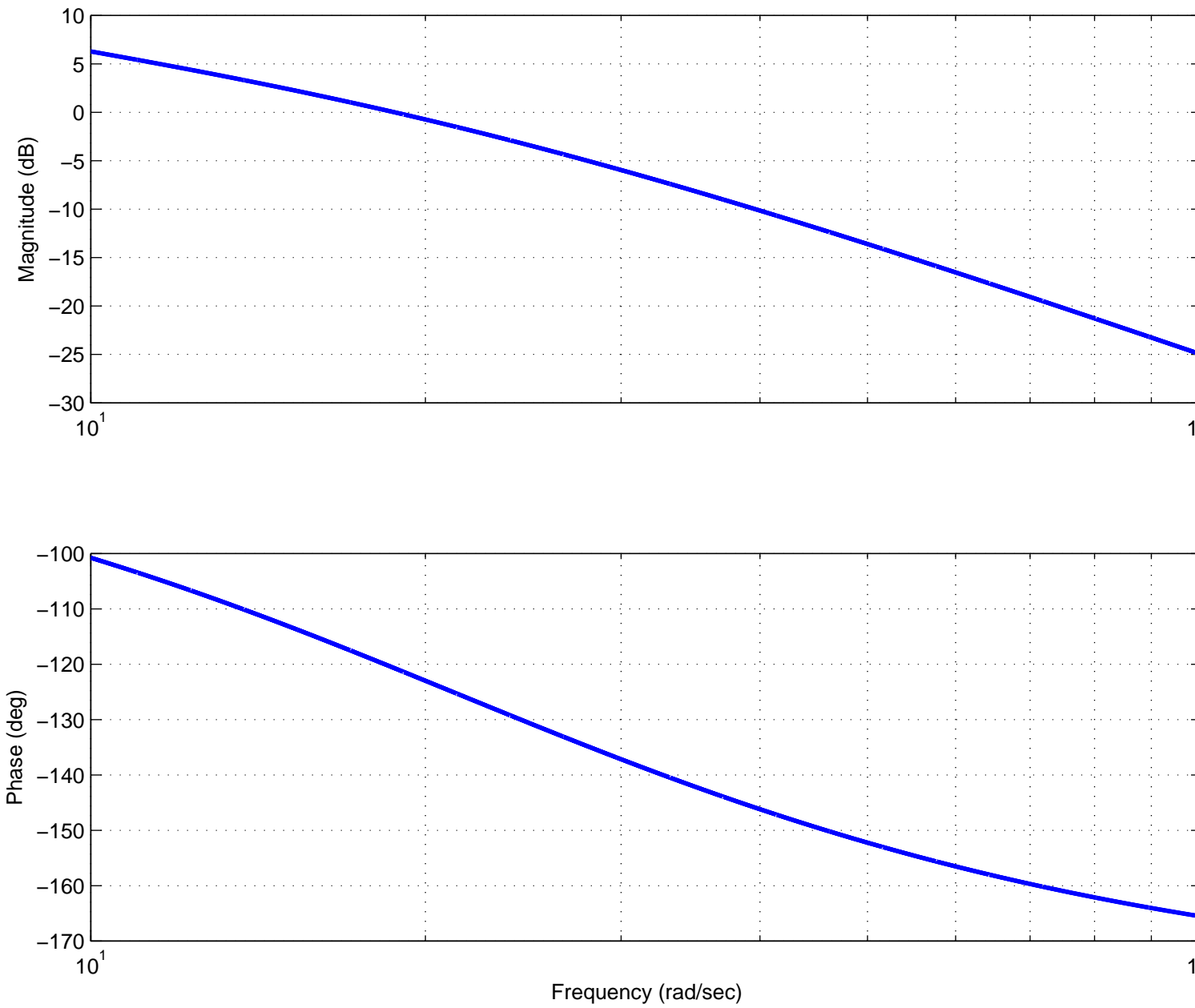


Figure 89: Bode plot of $G(s)H(s) = 10(s + 10)/(s + 1)(s + 20)^2$ with phase lead controller $G_c(s) = 80(0.051s + 1)/(0.071s + 1)$. The gain phase margin of this system is now approximately 59° .

A Matlab Commands

In this section I have listed some common Matlab commands and sections of code that you will be using on the homework problems. You will probably want to use the **help**, **doc**, and **lookfor** commands to learn more about these various functions and commands as you go on through this course. We will only go over some very simple uses of the commands here.

A.1 Figures

The first time you tell Matlab to plot something, it opens a new window and produces a graph. Matlab's default is to plot each graph in the same window, overwriting the previous graph. The **figure** command is given before plotting a new graph to tell Matlab to open a new window for a new graph.

A.2 Transfer Functions

We will make extensive use of transfer functions in this course, so we need to know how to enter them into Matlab. In general, to enter a polynomial such as

$$as^4 + bs^3 + cs^2 + ds + e$$

into Matlab, type

```
poly = [a b c d e];
```

where the powers are implied, only the coefficients are entered. (The semicolon at the end tells Matlab not to regurgitate what you just told it.) Hence, if we have a rational transfer function, such as

$$H(s) = \frac{s^3 + 2s}{s^4 + 3s^3 + s + 5}$$

we can enter the numerator and denominator polynomials separately, as

```
num = [1 0 2 0]; den = [1 3 0 1 5];
```

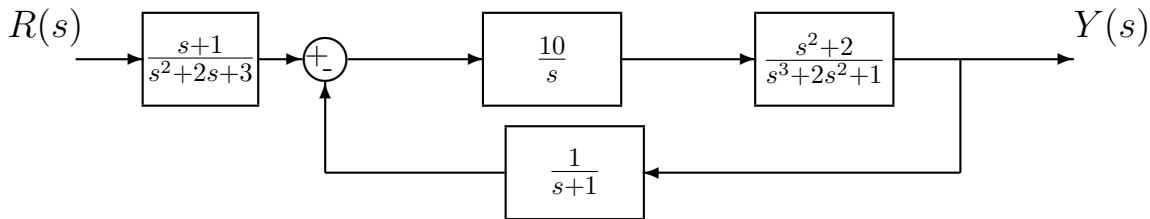
We will usually need to construct the transfer functions explicitly. To do this, type

```
H = tf(num,den)
```

This, without the semicolons, should display the transfer function, so you can check that you entered the correct function. In fact, at any time you can just type H to have Matlab display what the transfer function is.

A.3 Feedback Systems

Let's assume we want to find the closed loop transfer function for the following system using Matlab,



We first need to define all of the transfer functions

```
Gpre = tf([1 1],[1 2 3]);
Gc = tf(10,[1 0]);
Gp = tf([1 0 2],[1 2 0 1]);
H = tf(1,[1 1]);
```

Next, we compute the transfer function for the feedback block using the **feedback** command

```
T = feedback(Gc*Gp,H);
```

Finally we add the prefilter to get the close loop transfer function

```
G0 = Gpre*T;
```

A.4 System Response to Arbitrary Inputs

We will make extensive use both the unit step response and the unit ramp response of a system in this course. For the unit step response, we assume the system is at rest and the input is $u(t) = 1$ (a constant) for all $t \geq 0$, while for the unit ramp response, we assume the system is at rest and the input is $u(t) = t$ for all $t \geq 0$.

The simplest way to determine the step response to a system is

```
step(H);
```

A figure will appear on the screen, with the step response of the system. Note that the system will determine what it thinks are appropriate parameters. Sometimes, we want more control and want different inputs other than a step. In that case we use the command **lsim**. There are many forms for this command. In its basic form, you need to tell it a transfer function, the input function 'u', and the sample times 't'. For example, the following sequence of commands plots the response of the system

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

which is initially at rest (the initial conditions are 0) to an input of $\cos(3t)$ from 0 to 100 seconds in increments of 0.05 seconds and then plots the output.

```
num=[1]; den=[1 2 1];
H = tf(num,den); % get the transfer function
t=[0:0.05:100]; % times from 0 to 100 seconds by increments of 0.05
u = cos(3*t); % input is cos(3t) at the same times
y=lsim(H,u,t); % system output is y
plot(t,y); % plot the output
```

We can (obviously) use the `lsim` command to determine the step response,

```
num=[1]; den=[1 2 1];
H = tf(num,den); % get the transfer function
t=[0:0.05:100]; % times from 0 to 100 seconds by increments of 0.05
nt = length(t); % get the length of the t array
u = ones(1,nt); % input is a sequence of 1's
y=lsim(H,u,t); % system output is y
plot(t,y); % plot the output
```

The following piece of code will plot the step response of system H, showing both the system response and the input (we generally want the system to track the input), with neat labelling.

```
%
% The Step Response
%
t = [0:0.1:10]; % time from 0 to 10 in increments of 0.1
u = ones(1,length(t)); % the input is a sequence of 1's
y = lsim(H,u,t); % simulate the friggin system
figure; % set up a new figure (window)
plot(t,y,'-',t,u,'.-'); % plot the system response/input on one graph
grid; % put on a grid;
title('Step Response of H'); % put on a title
xlabel('Time (Seconds)'); % put on an x axis label
legend('Step Response','Unit Step'); % put on a legend
```

A.5 Changing the Line Thickness

As you hopefully have figured out, Matlab allows you to choose the colors for your graphs. However, sometimes you do not have access to a color printer, or just want to do something different. The following section of code allows you to plot using different line thicknesses.

```
%
% Now do line thickness
%
figure;
hold on % this basically means everything else is on one graph
plot(t,y,'-', 'Linewidth',4); % make the linewidth 4 (really quite large)
```

```

plot(t,u,'-','Linewidth',0.2); % make the linewidth 0.2 (really quite small)
legend('output','input'); grid;
hold off % we are done with this graph
%
```

You should note that even though you are changing the line width, you can still chose both the type of line to draw (dashed, dotted, etc) and the color. Also, this may not look so good on the screen, but usually prints out much better with a reasonable quality printer. Also, sometimes **hold on** and **hold off** can act really weird when you are doing many graphs. This is particularly true if you forgot the **hold off**.

A.6 Poles and Zeros

For any transfer function, the **poles** of the system are the roots of the denominator polynomial, while the **zeros** of the system are the roots of the numerator polynomial. Hence, if we have a transfer function

$$G(s) = \frac{(s+1)(s-1)}{(s+2)^2(s+3)(s+4)}$$

the poles of the system are at -2 (repeated), -3, and -4 while the zeros of the system are at -1, +1 (and ∞ , but we don't usually talk about this). The poles of the transfer function are the same as the eigenvalues of the system. We care about the poles of the system since they indicate how fast the system will respond and the bandwidth of the system. The commands **pole(G)** and **zero(G)** will return the poles and zeros of transfer function G.

A.7 Roots and Polynomials

If we want the roots of a polynomial Q assigned to a variable r , we would use the Matlab command **roots**

```
r = roots(Q);
```

For example, if $Q(s) = s^3 + s + 1$ and we wanted the roots of $Q(s)$, we would type

```
Q = [1 0 1 1];
r = roots(Q);
```

and we would get an array

```
r =
    0.3412 + 1.1615i
    0.3412 - 1.1615i
   -0.6823
```

If we wanted to determine the polynomial with roots at $0.3412 \pm 1.1615j, -0.6823$ we would use the **poly** command

```
Q = poly([0.3412+1.1615*i 0.3412-1.1615*i -0.6823]);
```

or, in our case

```
Q = poly(r);
```

or

```
Q = poly([ r(1) r(2) r(3) ]);
```

If we want to polynomial with roots at $0.3412 \pm 1.1615j, -0.6823$ we can just type

```
Q = poly([ r(1) r(2) ]);
```

A.8 Root Locus Plots

To plot the root locus of a system with open loop transfer function $H(s)$, we use the **rlocus** command,

```
rlocus(H);
```

You will be able to click on a line and determine both the value of the gain K at that point and the corresponding closed loop pole values. If we want to know the values of the closed loop poles at a particular value of K , say $K = 10$, we type

```
r = rlocus(H,10)
```

A.9 Bode Plots, Gain and Phase Margins

To determine the gain and phase margin of a system with open loop transfer function $H(s)$, we use the **margin** command

```
margin(H)
```

To create the bode plot of a system with open loop transfer function $H(s)$, we use the **bode** command

```
bode(H)
```

There are a number of useful variations on the bode command. For example, if we want to view to bode plot over a specified range of frequencies, we type

```
w = logspace(0,2,100);    % create 100 logarithmically spaced points
                          % between 1 (10^0) and 100 (10^2)
bode(H,w);
```

Sometimes we want the magnitude and phase of the transfer function $H(s)$. We can use the command

```
[Mag,Phase,w] = bode(H);
Mag = Mag(:);
Phase = Phase(:);
```

In this command, Matlab returns the magnitude (not in dB), phase, and frequencies the function was evaluated at, but the magnitude and phase are stored in a weird way. The command $Mag = Mag(:)$ forces Matlab to put them in a column. We can also specify which frequencies we want to evaluate the function at

```
[Mag,Phase] = bode(H,w);
```

If we then want to just plot the magnitude of the transfer function we can use

```
Mag = Mag(:);  
Mag_dB = 20*log10(Mag);  
semilogx(w,Mag_dB); grid;
```