

# Notes for ECE-320

Fall 2009

by  
R. Throne

The following pages contain an attempt at writing notes for ECE-320. These notes are not complete, especially the sections on design using Bode plots.

The major sources for these notes are

- *Analog and Digital Control System Design*, by C. T. Chen. Sanders College Publishing, 1993.
- *Linear Control Systems*, by Rohrs, Melsa, and Schulz. McGraw-Hill, 1993.
- *Modern Control Engineering*, by Ogata. Prentice-Hall, 2002.
- *Modern Control Systems*, by Dorf and Bishop. Prentice-Hall, 2005.

# Contents

<b>1</b>	<b>Table of Laplace Transforms</b>	<b>5</b>
<b>2</b>	<b>Laplace Transform Review</b>	<b>6</b>
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 . . . . .	20
2.8	Complex Conjugate Poles-Again . . . . .	22
<b>3</b>	<b>Final Value Theorem and the Static Gain of a System</b>	<b>23</b>
<b>4</b>	<b>Step Response, Ramp Response, and Steady State Errors</b>	<b>25</b>
4.1	Step Response and Steady State Error . . . . .	25
4.2	Ramp Response and Steady State Error . . . . .	29
4.3	Summary . . . . .	32
<b>5</b>	<b>Response of a Ideal Second Order System</b>	<b>34</b>
5.1	Step Response of an Ideal Second Order System . . . . .	34
5.2	Time to Peak, $T_p$ . . . . .	35
5.3	Percent Overshoot, PO . . . . .	36
5.4	Settling Time, $T_s$ . . . . .	37
5.5	Constraint Regions in the $s$ -Plane . . . . .	39
5.6	Summary . . . . .	46
<b>6</b>	<b>Characteristic Polynomial, Modes, and Stability</b>	<b>48</b>
6.1	Characteristic Polynomial, Equation, and Modes . . . . .	48
6.2	Characteristic Mode Reminders . . . . .	49
6.3	Stability . . . . .	50
6.4	Settling Time and Dominant Poles . . . . .	50
<b>7</b>	<b>Time Domain Response and System Bandwidth</b>	<b>52</b>
<b>8</b>	<b>Basic Feedback Configuration</b>	<b>60</b>
<b>9</b>	<b>Model Matching</b>	<b>61</b>
9.1	ITAE Optimal Systems . . . . .	63
9.2	Deadbeat Systems . . . . .	64
9.3	Quadratic Optimal Systems . . . . .	65
9.4	Dynamic Prefilters . . . . .	69
9.5	Type One Systems . . . . .	72
9.6	Frequency Response Matching . . . . .	74
9.7	Summary and Caveates . . . . .	84

<b>10 System Type and Steady State Errors</b>	<b>85</b>
10.1 Review . . . . .	85
10.2 System Type For a Unity Feedback Configuration . . . . .	85
10.3 Steady State Errors for Step and Ramp Inputs . . . . .	86
10.4 Examples . . . . .	87
<b>11 Controller Design Using the Root Locus</b>	<b>91</b>
11.1 Standard Root Locus Form . . . . .	91
11.2 Examples . . . . .	93
11.3 Loci Branches . . . . .	94
11.4 Real Axis Segments . . . . .	95
11.5 Asymptotic Angles and Centroid of the Asymptotes . . . . .	100
11.6 Common Industrial Controller Types . . . . .	110
11.7 Controller and Design Constraint Examples . . . . .	112
11.8 Seemingly Odd Root Locus Behavior . . . . .	136
<b>12 Pole Placement By Matching Coefficients: Diophantine Equations</b>	<b>139</b>
12.1 Theoretical Background . . . . .	140
12.2 Pole Placement with Robust Tracking . . . . .	143
12.3 Summary . . . . .	145
<b>13 System Sensitivity</b>	<b>146</b>
13.1 Sensitivity to Parameter Variations . . . . .	146
13.2 Sensitivity to External Disturbances . . . . .	151
13.3 Summary . . . . .	152
<b>14 The Routh-Hurwitz Stability Criteria</b>	<b>153</b>
14.1 No Element in the First Column is Zero . . . . .	154
14.2 First Column Contains a Zero . . . . .	158
14.3 Entire Row is Zero . . . . .	159
<b>15 State Variables and State Variable Feedback</b>	<b>161</b>
15.1 State Variable to Transfer Function Model . . . . .	163
15.2 State Variable Feedback . . . . .	166
15.3 Controllability for State Variable Systems . . . . .	171
15.4 Summary . . . . .	172
<b>16 Linearization</b>	<b>173</b>
16.1 Linear Systems . . . . .	173
16.2 Taylor Series . . . . .	173
16.3 Linearization Procedure . . . . .	174
<b>A Matlab Commands</b>	<b>i</b>
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)

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

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  $b = \frac{\sqrt{7}}{2}$

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

or

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

**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

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

This almost has the form we want, which is

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

However, to use this form we need  $s+a$  in the numerator, not just  $s$ . To achieve this we will add and subtract  $a = \frac{3}{2}$  in the numerator

$$\begin{aligned} H(s) &= \frac{s + \frac{3}{2} - \frac{3}{2}}{(s + \frac{3}{2})^2 + (\frac{\sqrt{11}}{2})^2} \\ &= \frac{s + \frac{3}{2}}{(s + \frac{3}{2})^2 + (\frac{\sqrt{11}}{2})^2} - \frac{\frac{3}{2}}{(s + \frac{3}{2})^2 + (\frac{\sqrt{11}}{2})^2} \end{aligned}$$

The first term is now what we want, and will produce a term of the form

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

The second term needs some work. It looks like a sine times a decaying exponential, but the scaling is wrong. Again, to put this term in the correct form we will multiply and divide by  $\frac{\sqrt{11}}{2}$

$$H(s) = \frac{s + \frac{3}{2}}{(s + \frac{1}{2})^2 + \left(\frac{\sqrt{11}}{2}\right)^2} - \frac{3}{2} \frac{1}{\frac{\sqrt{11}}{2}} \frac{\frac{\sqrt{11}}{2}}{(s + \frac{1}{2})^2 + \left(\frac{\sqrt{11}}{2}\right)^2}$$

which gives

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

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. Hence, when we have complex poles, we will look for quadratic factors of the general form

$$\frac{cs + d}{s^2 + es + d}$$

Note that there are **two** unknown coefficients in this term. Since we need as many unknowns as the highest power of  $s$  in the denominator, and this term has 2 powers of  $s$ , we need two unknowns. We are now ready to do one more example.

**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

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

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  $a_1$  first using the coverup method

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

To find  $a_2$ , let's use our trick of multiplying by  $s$  and letting  $s \rightarrow \infty$

$$\lim_{s \rightarrow \infty} \left[ s \frac{1}{(s + 2)(s^2 + s + 1)} \right] = \lim_{s \rightarrow \infty} \left[ \frac{1}{3} \frac{s}{s + 2} + \frac{a_2s^2 + a_3s}{s^2 + s + 1} \right]$$

or

$$0 = \frac{1}{3} + a_2$$

so  $a_2 = -\frac{1}{3}$ . Now we have to find  $a_3$  and the only trick we have left is choosing a value of  $s$ . For this particular transfer function,  $s = 0$  is a good choice

$$\lim_{s \rightarrow 0} \left[ \frac{1}{(s+2)(s^2+s+1)} \right] = \lim_{s \rightarrow 0} \left[ \frac{1}{3} \frac{1}{s+2} + \frac{a_2 s + a_3}{s^2+s+1} \right]$$

or

$$\frac{1}{2} = \frac{1}{6} + a_3$$

or  $a_3 = \frac{1}{3}$ . So far we have

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

The first term is easy, now we need to work on the second term. First we complete the square in the denominator

$$s^2 + s + 1 = \left(s + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2$$

so we have

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

The next thing to do is to add and subtract  $\frac{1}{2}$ , so the numerator has the correct form so we have

$$\begin{aligned} H(s) &= \frac{1}{3} \frac{1}{s+2} + \frac{-\frac{1}{3}\left(s + \frac{1}{2} - \frac{1}{2}\right) + \frac{1}{3}}{\left(s + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} \\ &= \frac{1}{3} \frac{1}{s+2} + \frac{-\frac{1}{3}\left(s + \frac{1}{2}\right) + \left(\frac{1}{6} + \frac{1}{3}\right)}{\left(s + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} \\ &= \frac{1}{3} \frac{1}{s+2} + \frac{-\frac{1}{3}\left(s + \frac{1}{2}\right) + \frac{1}{2}}{\left(s + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} \\ &= \frac{1}{3} \frac{1}{s+2} - \frac{1}{3} \frac{\left(s + \frac{1}{2}\right)}{\left(s + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} + \frac{1}{2} \frac{1}{\left(s + \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} \end{aligned}$$

Finally, we have to scale the final term to put it into the correct form

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

So we finally 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

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

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

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

In the time-domain we have then

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

## 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,

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

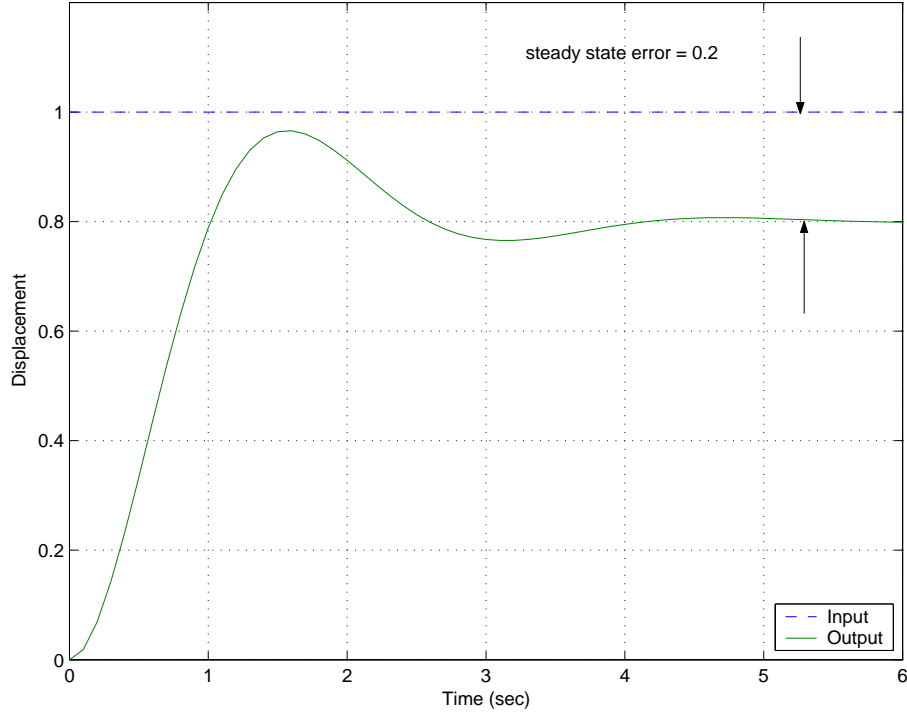


Figure 1: The unit step response and position error for the system in Example 1. This system has a *positive position error*.

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,

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

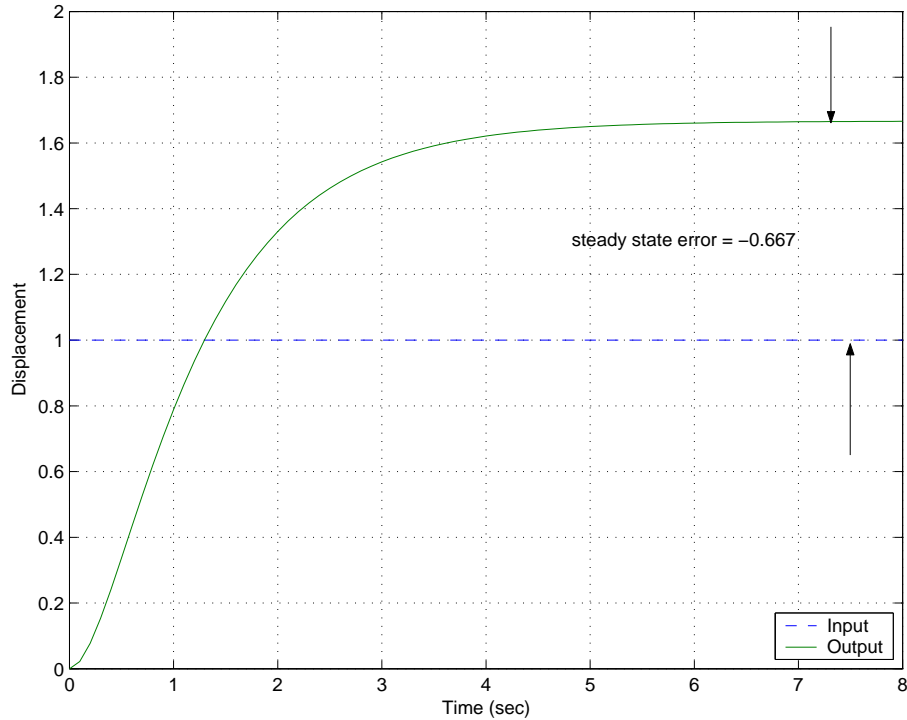


Figure 2: The unit step response and position error for the system in Example 2. This system has a *negative position error*.

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

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

**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

$$\begin{aligned} Y(s) &= \frac{s+2}{s^2+2s+2} \frac{A}{s^2} = a_1 \frac{1}{s} + a_2 \frac{1}{s^2} + \frac{a_3 s + a_4}{s^2+2s+2} \\ &= A \left[ -\frac{1}{2} \frac{1}{s} + \frac{1}{s^2} + \frac{1}{2} \frac{s}{(s+1)^2+1} \right] \\ &= A \left[ -\frac{1}{2} \frac{1}{s} + \frac{1}{s^2} + \frac{1}{2} \frac{s+1}{(s+1)^2+1} - \frac{1}{2} \frac{1}{(s+1)^2+1} \right] \end{aligned}$$

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

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

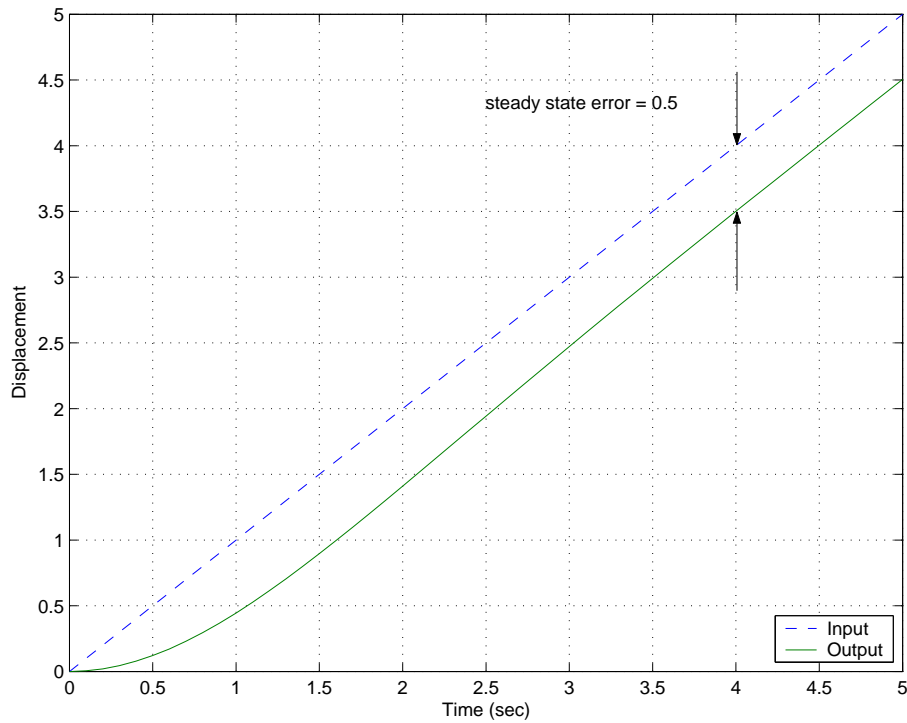


Figure 3: The unit ramp response and steady state error for the system in Example 8. Note that the steady state error is positive in this case, since the steady state value of the output is smaller than the steady state value of the input.

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  $\zeta$  is the damping ratio,  $\omega_n$  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

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

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  $\pi$ . Hence

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

Remember that  $\omega_d$  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  $\tau$  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  $\beta$  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

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

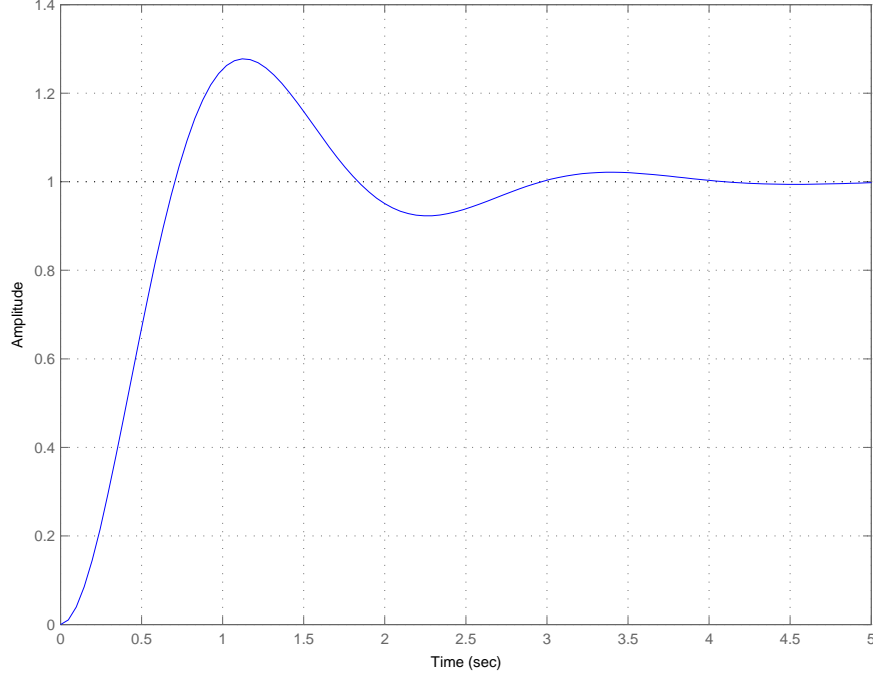


Figure 4: Step response for the system  $H(s) = \frac{9}{s^2 + 2.265s + 9}$ . The settling time should be less than 5 seconds, the time to peak should be less than 1.2 seconds, and the percent overshoot should be 27.8%.

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

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

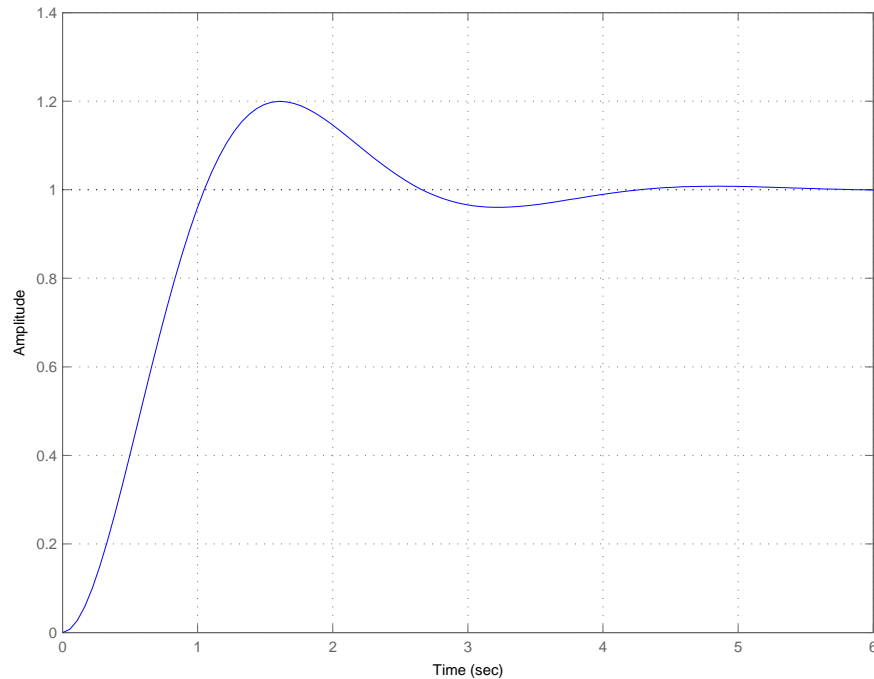


Figure 5: Step response for the system  $H(s) = \frac{K}{s^2+2s+K}$ . The percent overshoot should be less than or equal to 20% and the settling time should be 4 seconds.

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

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.

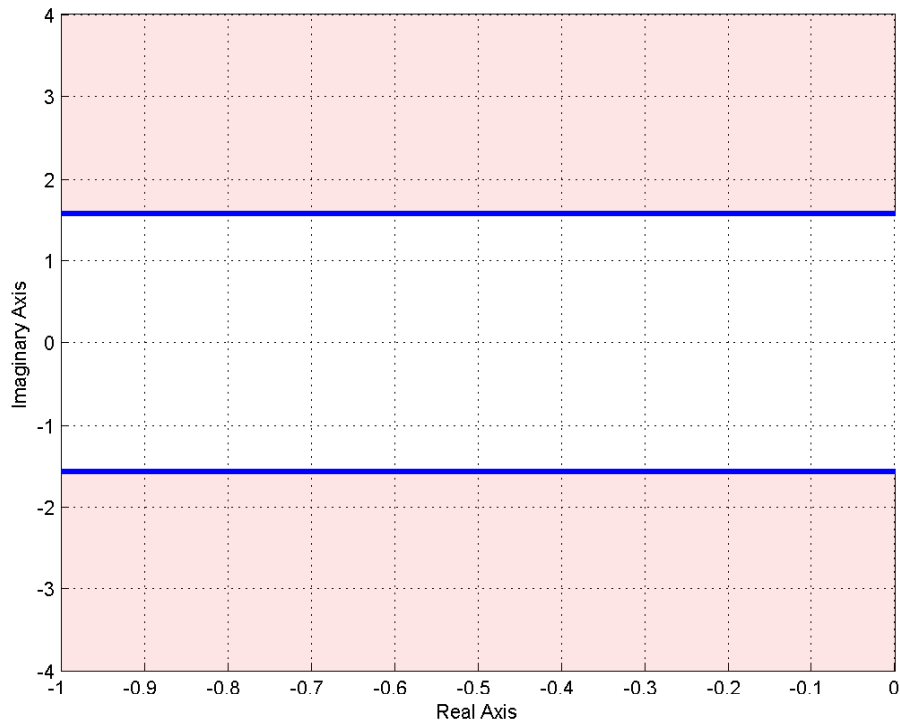


Figure 6: Acceptable pole locations for  $T_p \leq 2$  seconds are shown in the shaded region.



**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  $\zeta$ .

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

Now we use the relationship

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

In summary, we have

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

This angle  $\theta$  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.

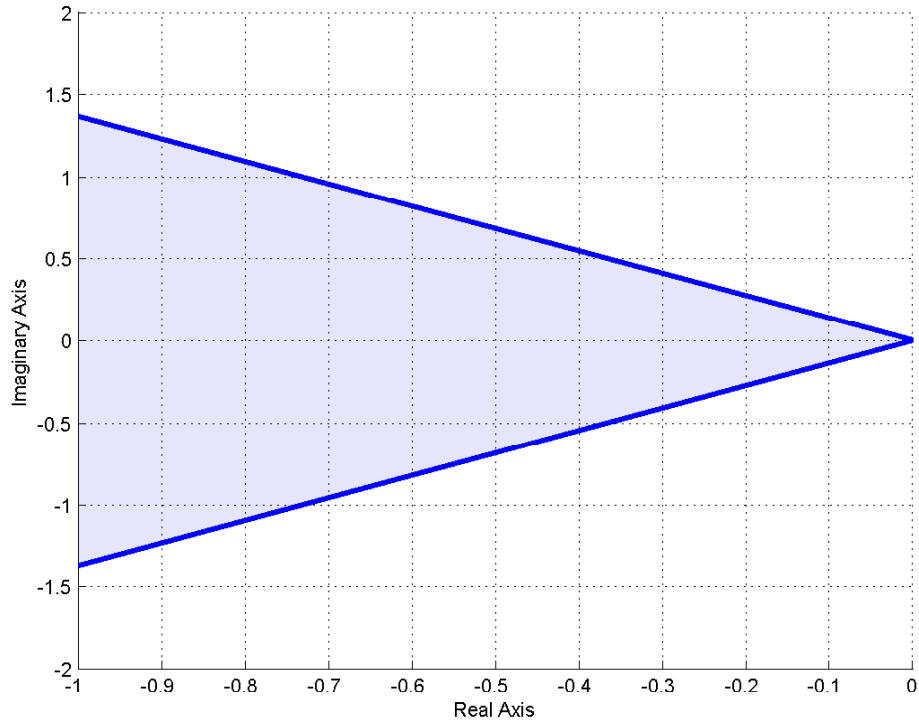


Figure 7: Acceptable pole locations for Percent Overshoot less than or equal to 10%. The acceptable pole locations are shown in the shaded region.

**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.

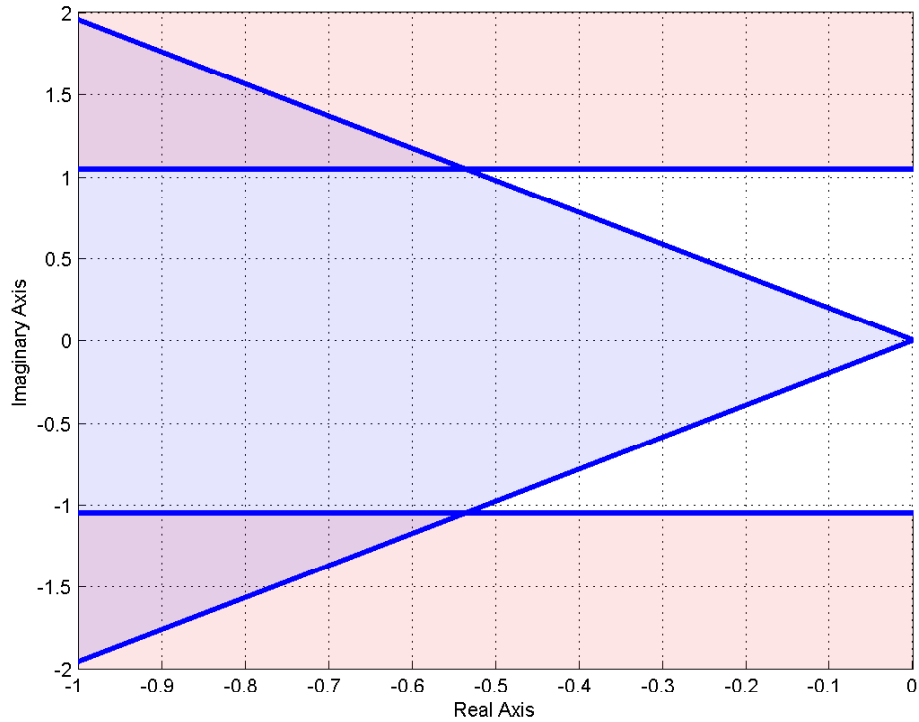


Figure 8: Acceptable pole locations for Percent Overshoot less than or equal to 20% and time to peak less than or equal to 3 seconds. The acceptable pole locations for each constraint are shown in the shaded regions. The overlapping regions are those pole locations that will meet both constraints.

**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.

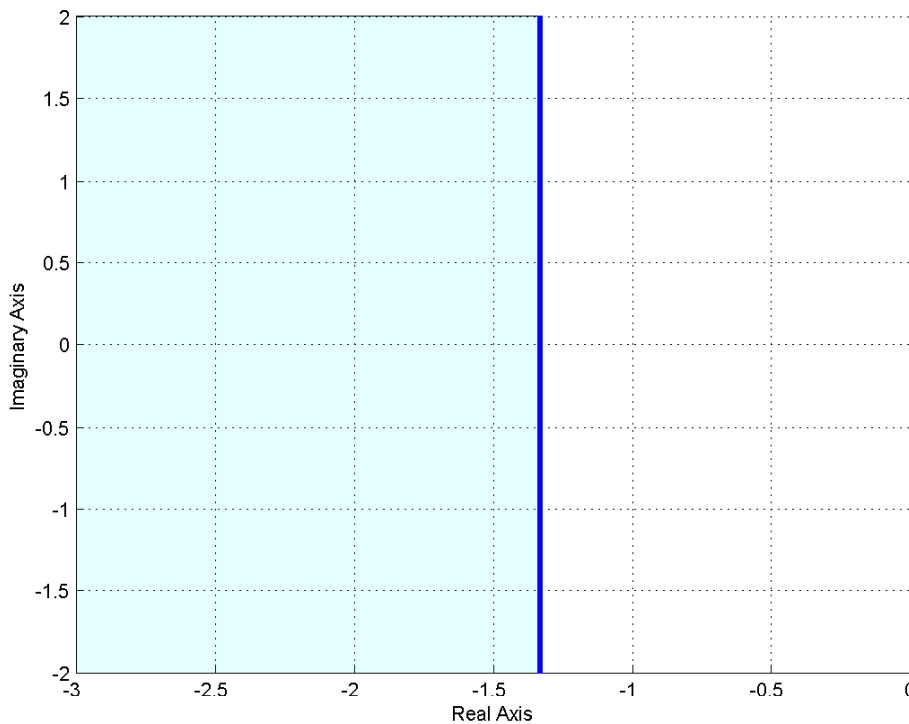


Figure 9: Acceptable pole locations for settling time less than or equal to 3 seconds. The acceptable pole locations are shown in the shaded region.

**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.

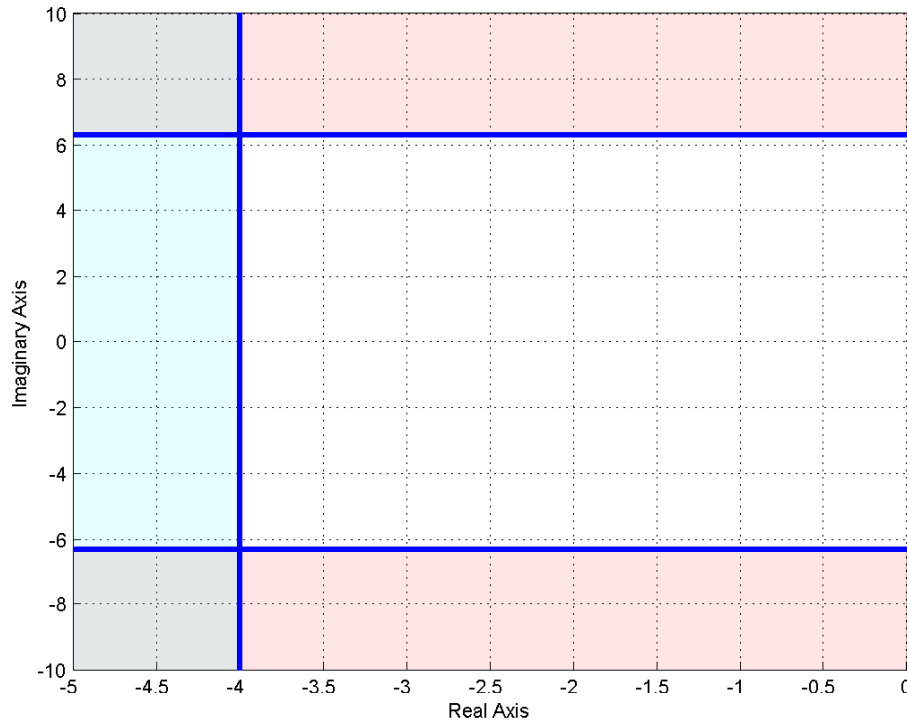


Figure 10: Acceptable pole locations for settling time less than or equal to 1 second and time to peak less than 0.5 seconds. The acceptable pole locations for each constraint are shown in the shaded regions. The overlapping regions are those pole locations that will meet both constraints.

**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.

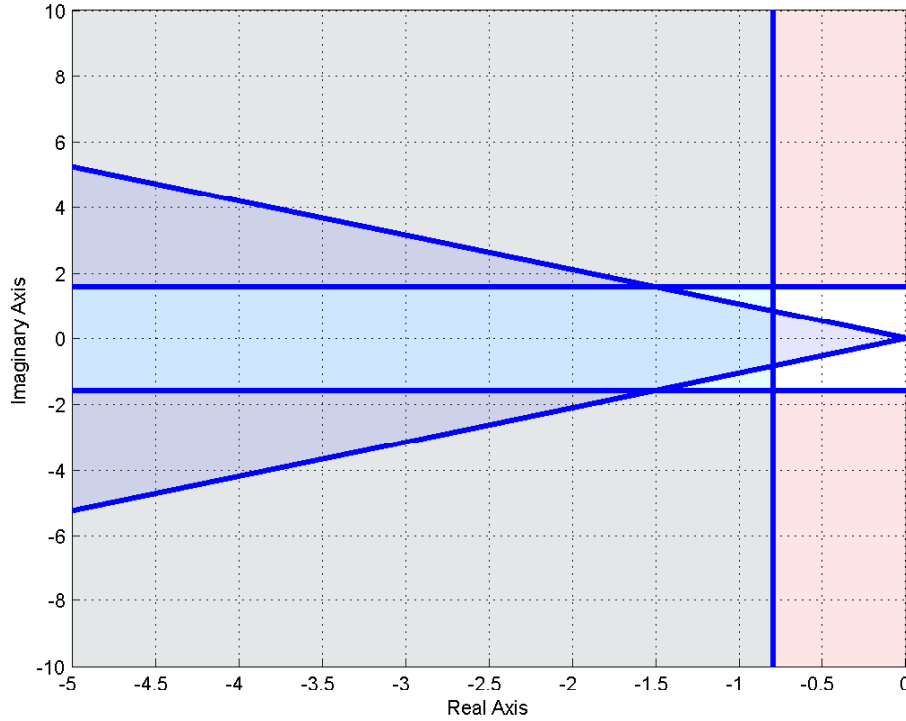


Figure 11: Acceptable pole locations for settling time less than 5 seconds, time to peak less than or equal to 2 seconds, and the percent overshoot less than 5%. The acceptable pole locations for each constraint are shown in the shaded regions. The overlapping regions (two triangular wedges) are those 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  $\pi/T_p^{max}$ , 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  $\theta$  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  $e^{-2t} \sin(3t)$ . 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  $\tau$  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  $\tau$  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  $i^{\text{th}}$  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  $\sigma$ .

*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  $\omega_b$  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  $\tau$  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  $\tau$  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

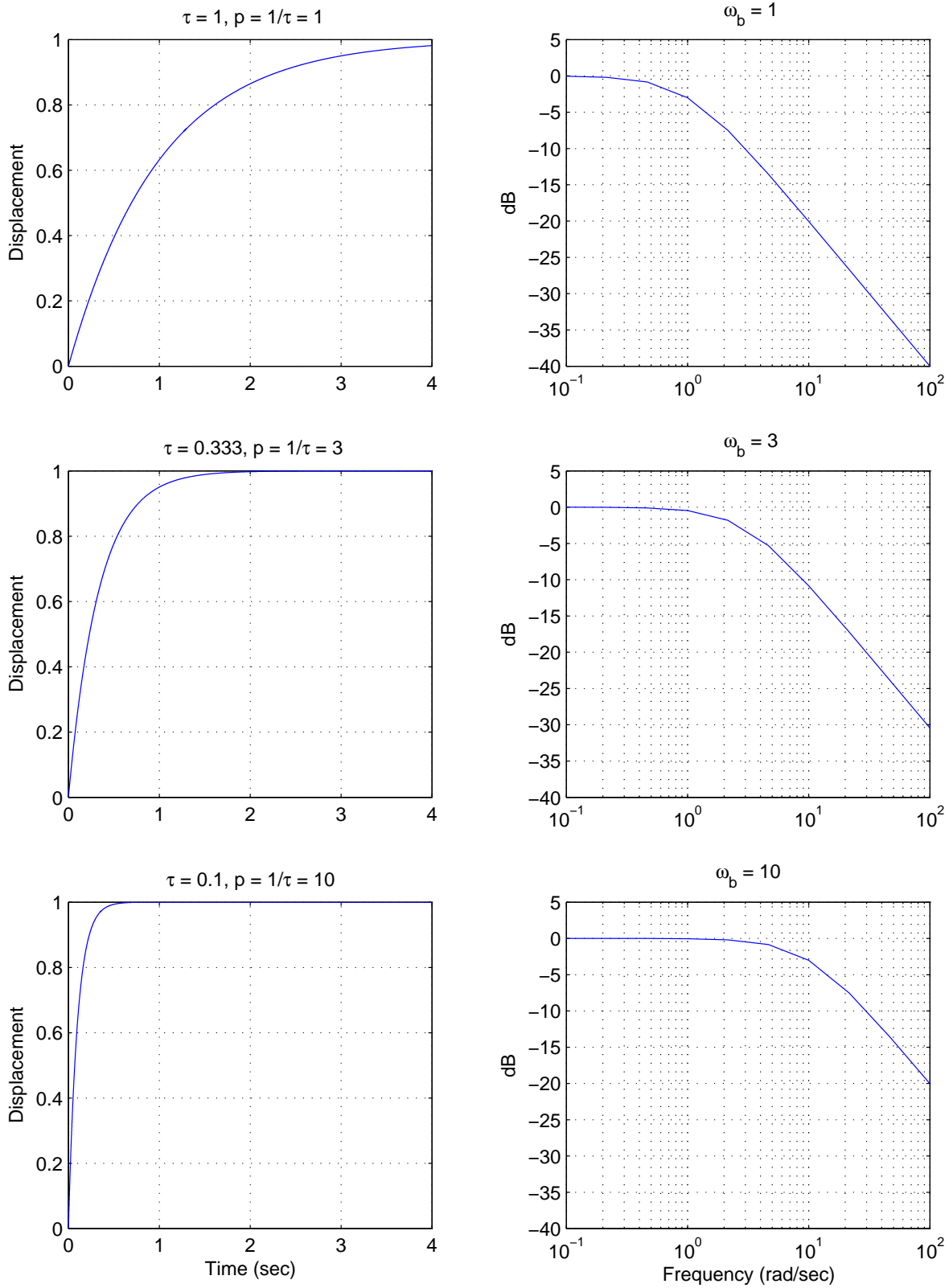


Figure 12: The unit step response and bandwidth for three first order systems. The magnitude of the system pole  $p$  is equal to the bandwidth  $\omega_b$ .

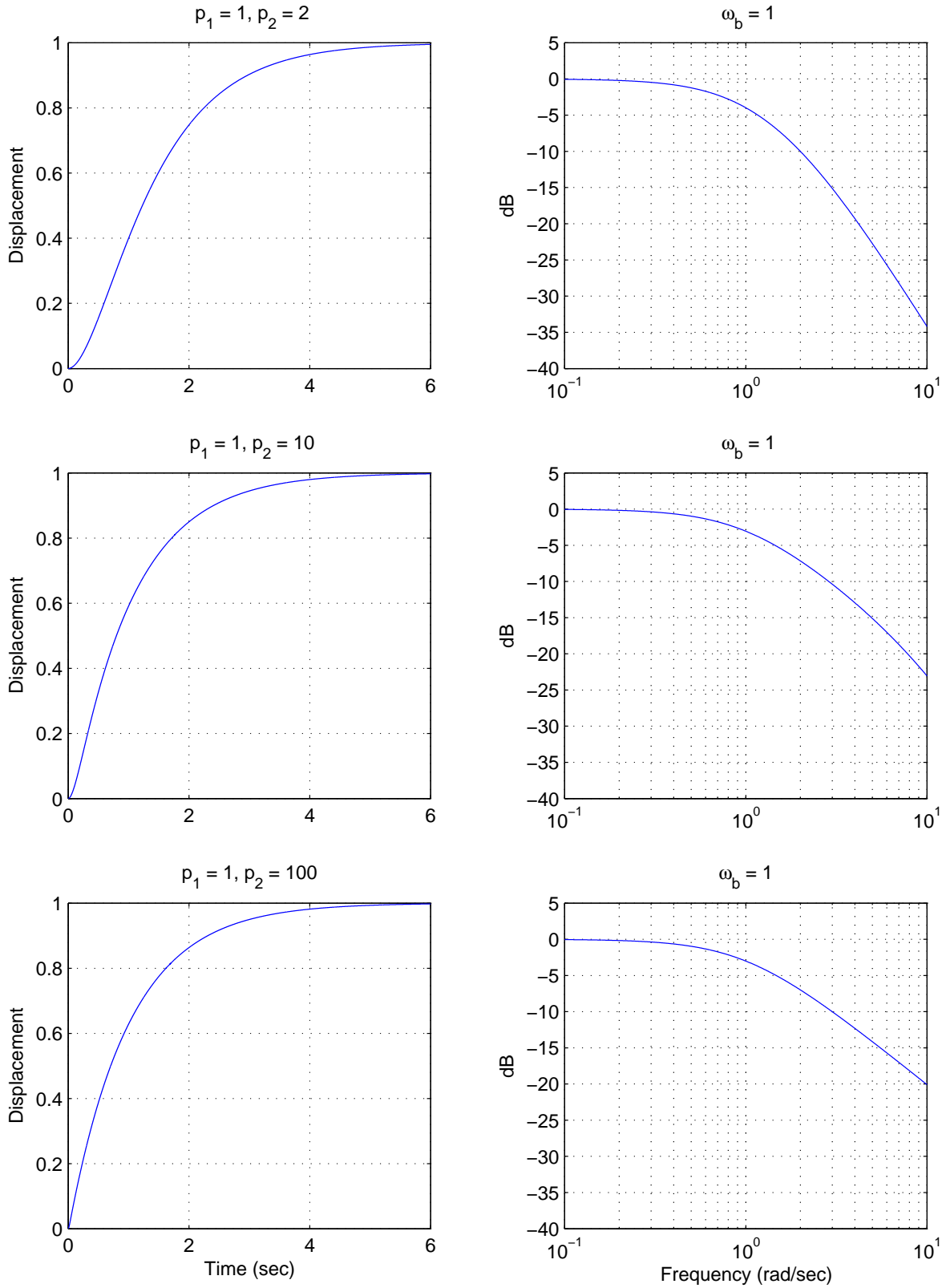


Figure 13: The unit step response and bandwidth for three second order systems with distinct poles. The rate of response is dominated by the *dominant pole* at -1, and the bandwidth (-3 dB point) is determined by this dominant pole.

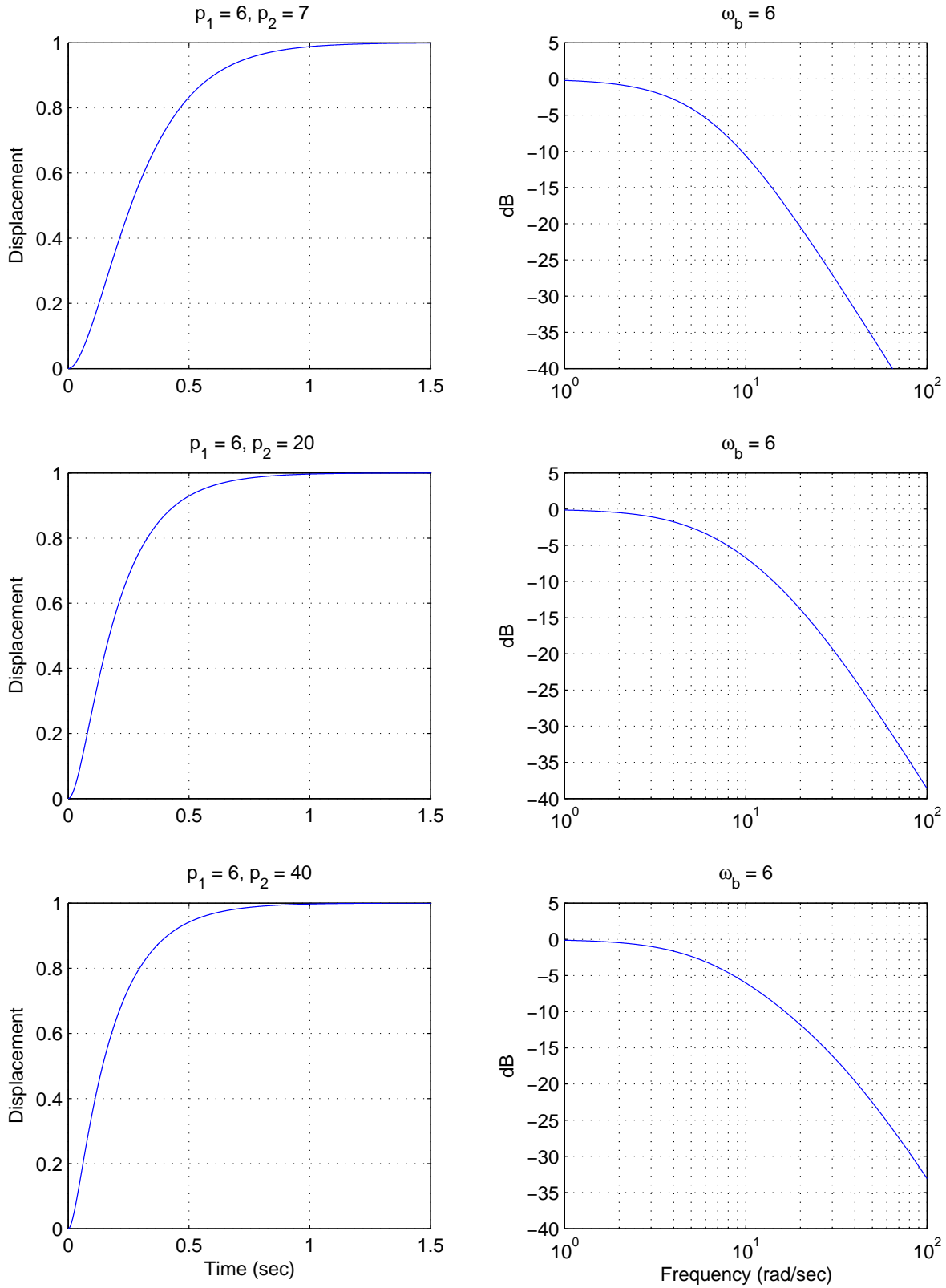


Figure 14: The unit step response and bandwidth for three second order systems with distinct poles. The rate of response is dominated by the *dominant pole* at -6, and the bandwidth (-3 dB point) is determined by this dominant pole.

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  $\sigma$ , 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  $\omega_d$  (the imaginary part of the pole), as  $\sigma$  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  $\omega_b$ , where  $\omega_b$  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

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



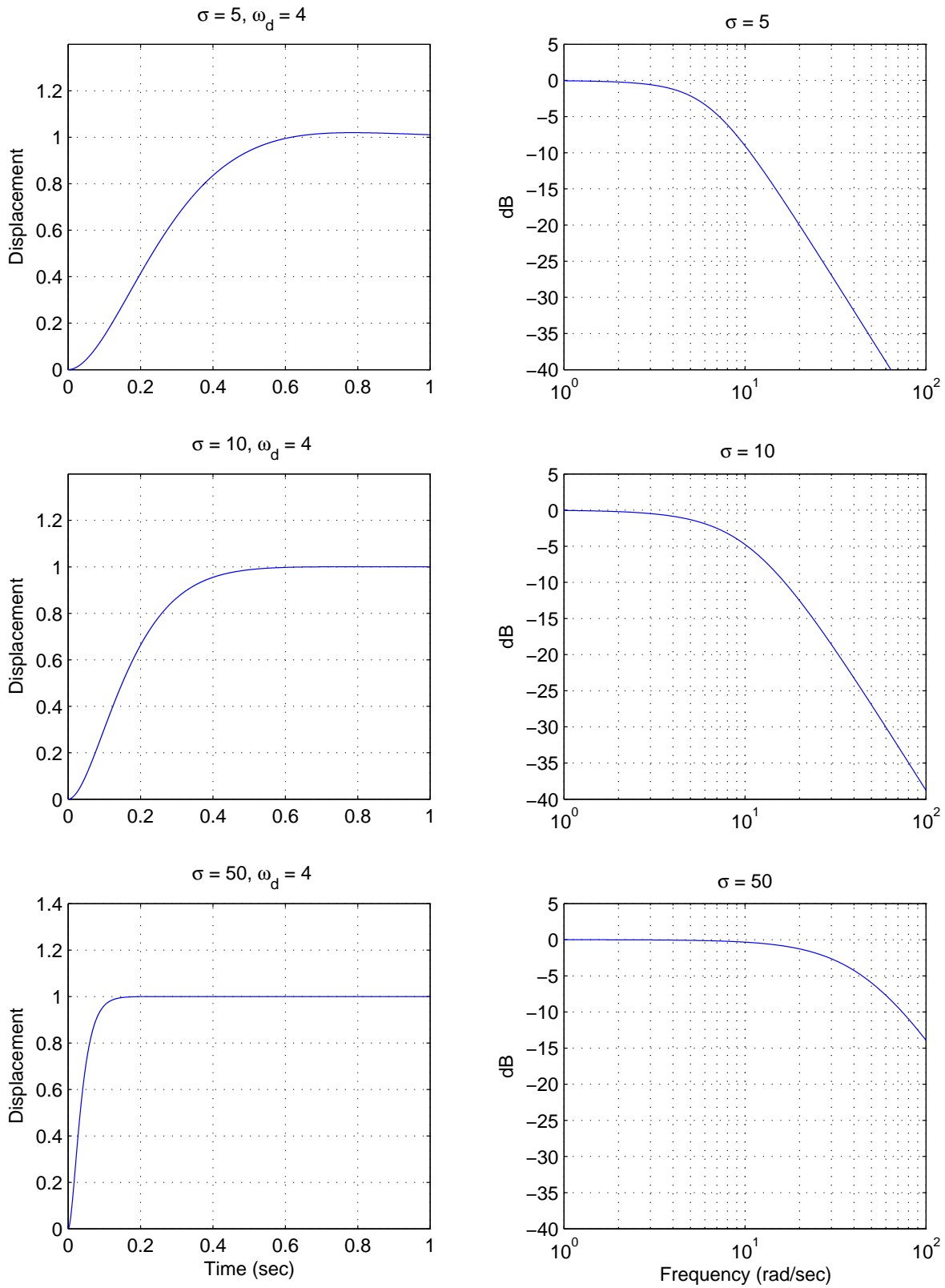


Figure 15: The unit step response and bandwidth for three second order systems with distinct complex conjugate poles. The imaginary parts of the poles are fixed at  $\pm 4$ . Note that the faster the time domain response the larger the system bandwidth.

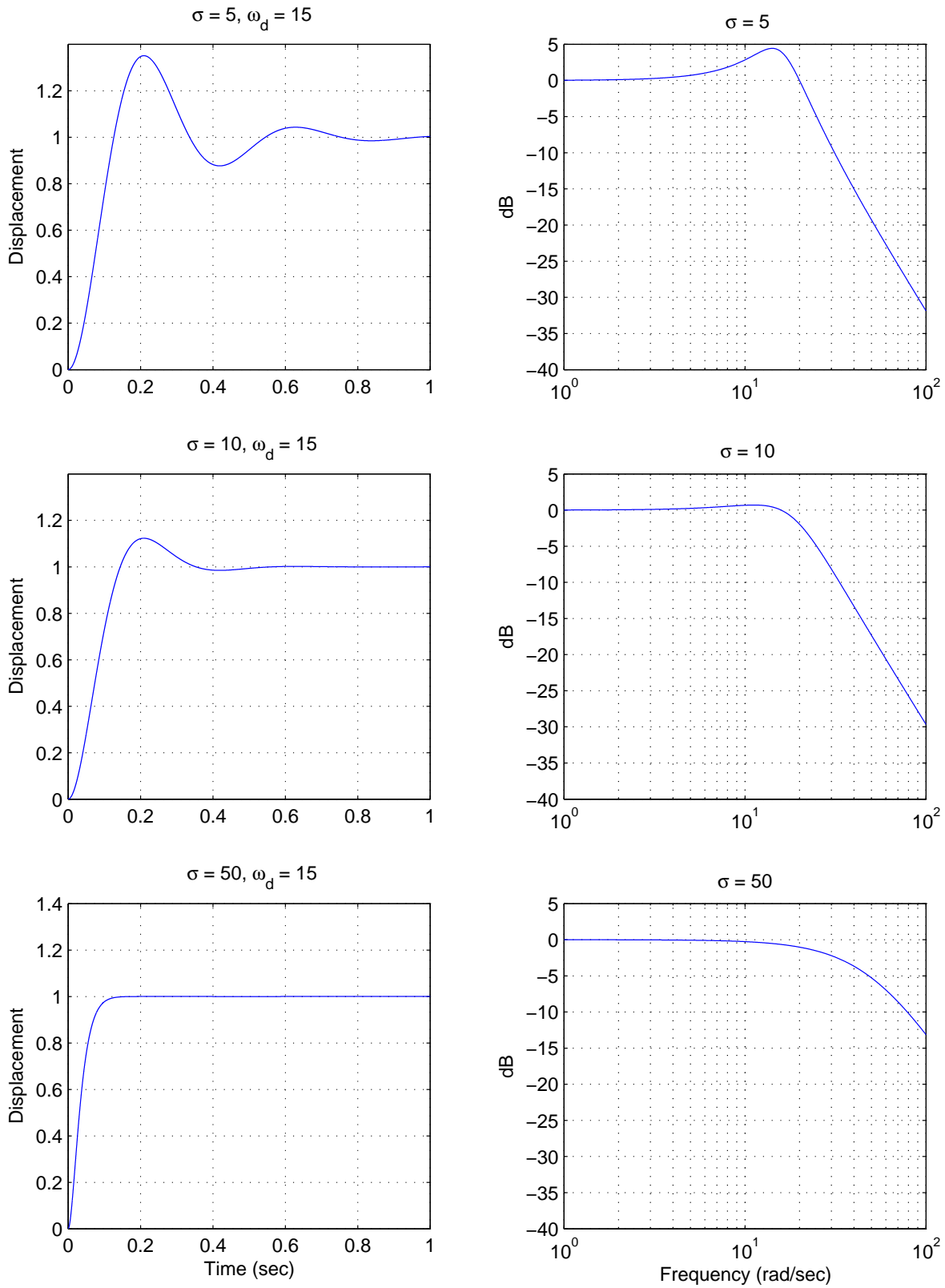


Figure 16: The unit step response and bandwidth for three second order systems with distinct complex conjugate poles. The imaginary parts of the poles are fixed at  $\pm 15$ . Note that the faster the time domain response the larger the system bandwidth.

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  $\omega_b$ , 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,

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

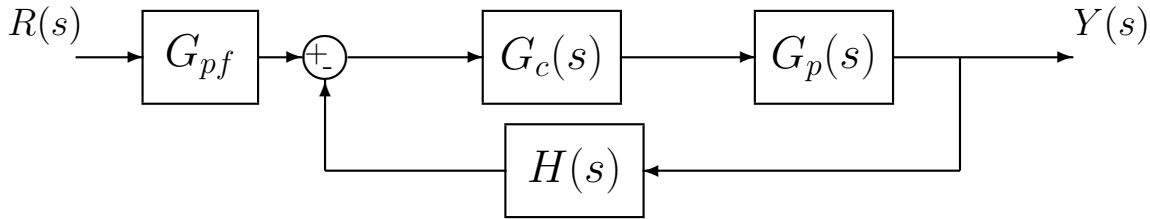
we get the average noise power estimate

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

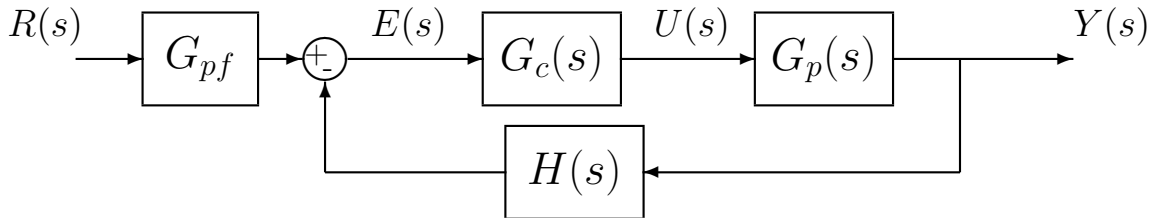
Hence to reduce the average noise power in our system, we want  $\omega_b$  small. However, this usually means a slower response in the time domain. Bummer.

## 8 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

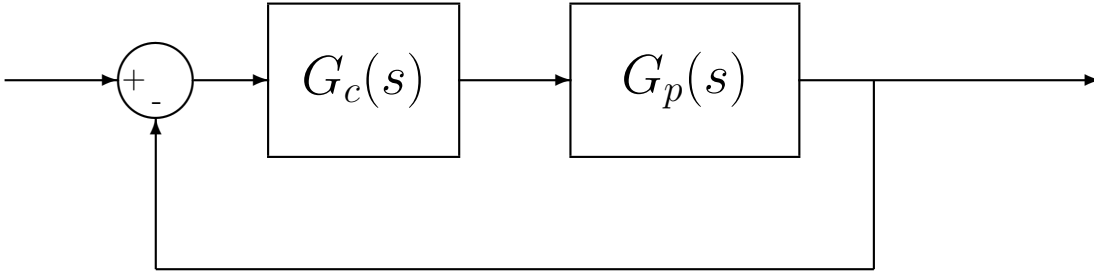
$$\begin{aligned} Y(s) &= G_p(s)G_c(s)[G_{pf}(s)R(s) - H(s)Y(s)] \\ &= G_{pf}(s)G_c(s)G_p(s)R(s) - G_c(s)G_p(s)H(s)Y(s) \\ Y(s) + G_c(s)G_p(s)H(s)Y(s) &= G_{pf}(s)G_c(s)G_p(s)R(s) \\ Y(s)[1 + G_c(s)G_p(s)H(s)] &= G_{pf}(s)G_c(s)G_p(s)R(s) \end{aligned}$$

or the closed-loop transfer function is

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

## 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 if<sup>1</sup>

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.

---

<sup>1</sup>There are other restrictions, but they are not important in this course.

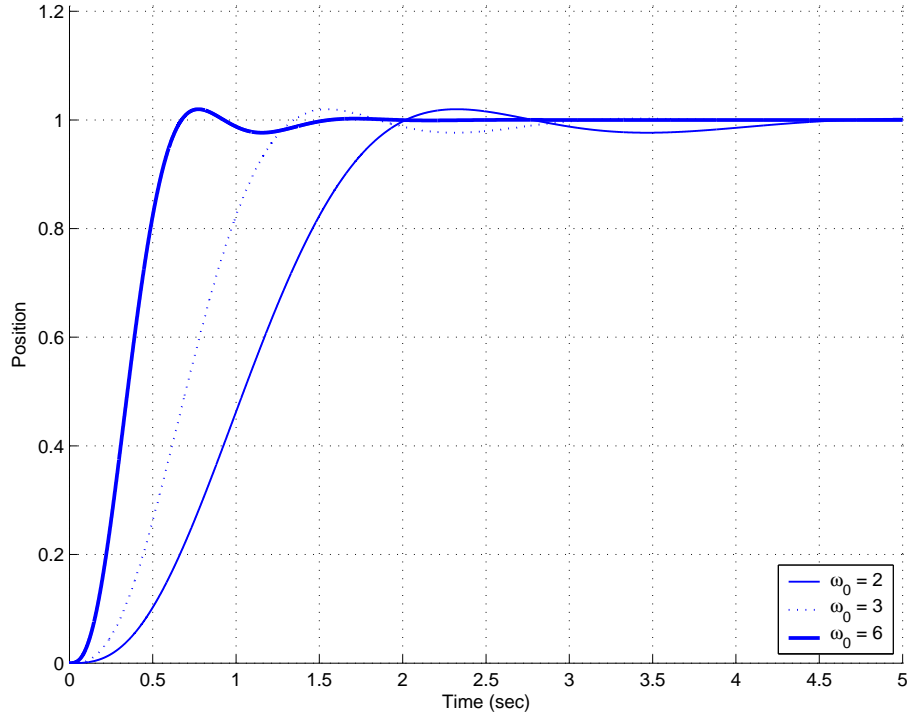


Figure 17: The unit step response for a third order ITAE system with  $\omega_0 = 2, 3$  and  $6$  radians/second. This system will have a steady state error of zero for a step input.

## 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 17 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  $\omega_0$ , 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

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

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

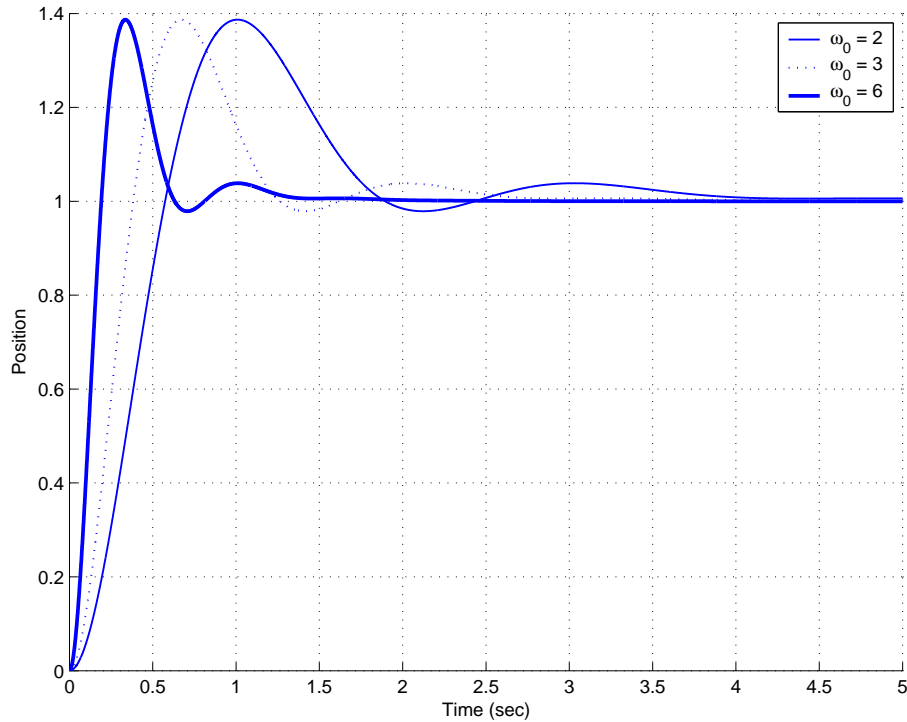


Figure 18: The unit step response for a third order ITAE system that has zero steady state error for both a step and ramp input with  $\omega_0 = 2, 3$  and  $6$  radians/second.

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

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 18 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  $\omega_0$ , the faster the response of the system.

You, the designer, need to choose the value of  $\omega_0$ . The larger the  $\omega_0$ , 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

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

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



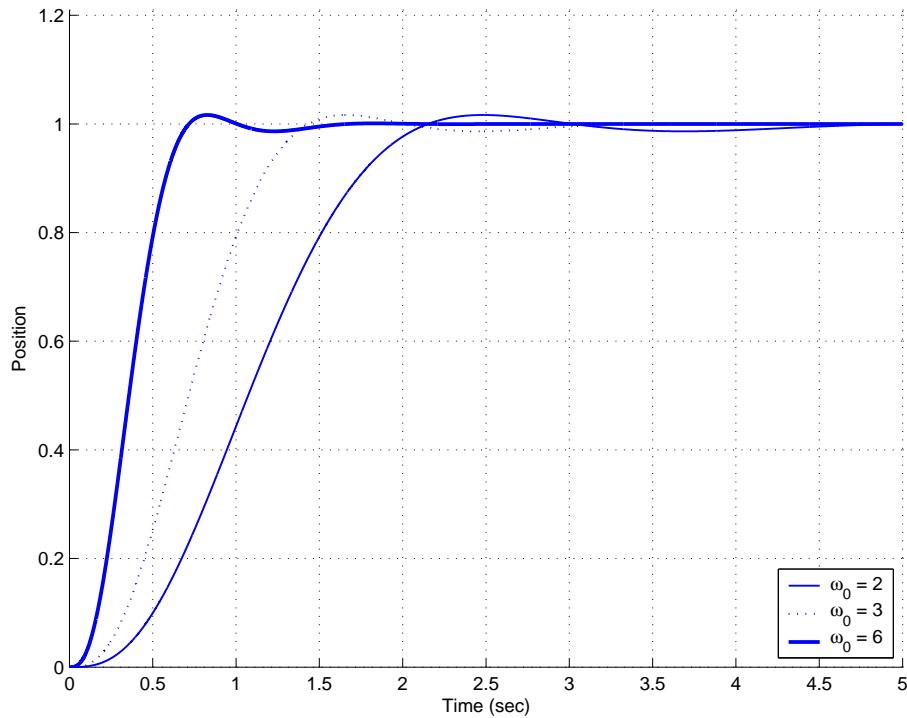


Figure 19: The unit step response for a third order deadbeat system with  $\omega_0 = 2, 3$  and  $6$  radians/second.

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

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 19 shows the step response for a third order deadbeat response for  $\omega_0 = 2, 3$  and  $6$  radians/second. Comparing Figure 19 with Figure 17, 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  $\omega_0$  the faster the response of the system (good) and the more control effort (bad) required.

### 9.3 Quadratic Optimal Systems

For a quadratic optimal system, we want to find the closed loop transfer function  $G_0(s)$  to minimize the quadratic performance index

$$J = \int_0^\infty \{q[y(t) - r(t)]^2 + u^2(t)\} dt$$

where  $y(t)$  is the output of the system,  $r(t)$  is the input to the system,  $q$  is a positive constant that weighs the difference between the input to the system and the output to the system, and  $u(t)$  is the actuating signal (the input to the plant we are trying to control). In general, for this type of controller, we want the output of our system to match (or track) the input to the

system. To determine  $G_0(s)$  to solve this problem, we need to first discuss spectral factorization.

Consider first the plant with proper transfer function  $G_p(s) = \frac{N(s)}{D(s)}$  where  $D(s)$  and  $N(s)$  have no common factors. Next, consider the polynomial

$$Q(s) = D(s)D(-s) + qN(s)N(-s)$$

Clearly  $Q(-s) = Q(s)$ , hence if  $s_1$  is a root of  $Q(s)$ , then so is  $-s_1$ . Since all of the coefficients of  $Q(s)$  are real by assumption (we assume  $G_p(s)$  is real), if  $s_1$  is a root of  $Q(s)$  then so is its complex conjugate  $s_1^*$ . Hence, all of the roots of  $Q(s)$  are symmetric with respect to

- the real axis
- the imaginary axis
- the origin of the  $s$ -plane

Now consider

$$\begin{aligned} Q(j\omega) &= D(j\omega)D(-j\omega) + qN(j\omega)N(-j\omega) \\ &= |D(j\omega)|^2 + q|N(j\omega)|^2 \end{aligned}$$

Since  $D(s)$  and  $N(s)$  have no common factors, there exists no  $\omega_0$  so that both  $D(j\omega_0) = 0$  and  $N(j\omega_0) = 0$ . Since  $q \neq 0$ , there exists no  $\omega_0$  so that  $Q(j\omega_0) = 0$ . Hence  $Q(s)$  has no roots on the  $j\omega$  axis.

Since the roots of  $Q(s)$  are either in the left half plane or the right half plane (none on the  $j\omega$  axis), and since, by symmetry, there will be an equal number in each plane, we will divide up the roots of  $Q(s)$  into those in the open LHP and those in the open RHP. Let's denote the polynomial whose roots are those of the open LHP roots of  $Q(s)$  as  $D_0(s)$ . Then, by symmetry,  $D_0(-s)$  is a polynomial whose roots are the open RHP roots of  $Q(s)$ . Thus,

$$Q(s) = D(s)D(-s) + qN(s)N(-s) = D_0(s)D_0(-s)$$

This is called the spectral factorization of  $Q(s)$ . Now we can give the result.

Consider a plant with proper transfer function  $G_p(s) = N(s)/D(s)$  where

- $N(s)$  and  $D(s)$  have no common factors
- The leading coefficient of  $D(s)$  (the coefficient of the highest power of  $s$  in  $D(s)$ ) is 1. That is,  $D(s)$  is monic. This is important!

An implementable transfer function  $G_0(s)$  that minimizes the performance index

$$J = \int_0^\infty \{q[y(t) - r(t)]^2 + u^2(t)\} dt$$

where  $r(t) = 1$  (a unit step) and  $q > 0$  is given by

$$G_0(s) = \frac{qN(0)N(s)}{D_0(0)D_0(s)}$$

where

$$Q(s) = D(s)D(-s) + qN(s)N(-s) = D_0(s)D_0(-s)$$

Note that we are not guaranteed a zero steady state error with this method. For a zero steady state error for a step input, we should have  $G_0(0) = 1$ . In addition, if  $N(s) = s$  this will not work (since  $N(0) = 0$ ).

**Example 2.** Suppose we have the plant with transfer function

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

and we want to find  $G_0(s)$  to minimize

$$J = \int_0^\infty \{10 [y(t) - r(t)]^2 + u^2(t)\} dt$$

Clearly  $q = 10$ ,  $G_p(s)$  is a proper transfer function,  $N(s)$  and  $D(s)$  have no common factors, and  $D(s)$  is a monic polynomial. Now  $N(s) = 1$ ,  $D(s) = s^2 + 1$ , and  $D(s)$  is monic. So we have

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

and

$$\begin{aligned} Q(s) &= D(s)D(-s) + qN(s)N(-s) \\ &= [s^2 + 1][s^2 + 1] + 10[1][1] \\ &= [s^4 + 2s^2 + 1] + 10[1] \\ &= s^4 + 2s^2 + 11 \end{aligned}$$

Note that  $Q(s)$  is an even function of  $s$ . If it is not, you screwed up! Now, we need to find the roots of  $Q(s)$ . These roots are  $-1.0762 \pm 1.4691j$  and  $1.0762 \pm 1.4691j$ . To construct  $D_0(s)$ , we use only those roots in the LHP, i.e. the roots at  $-1.0762 \pm 1.4691j$ .

$$\begin{aligned} D_0(s) &= (s + 1.0762 - 1.4691j)(s + 1.0762 + 1.4691j) \\ D_0(s) &= s^2 + 2.1525s + 3.3166 \end{aligned}$$

Now, we can compute the optimal  $G_0(s)$  as

$$\begin{aligned} G_0(s) &= \frac{qN(0)N(s)}{D_0(0)D_0(s)} = \frac{(10)(1)(1)}{(3.3166)(s^2 + 2.1525s + 3.3166)} \\ &= \frac{3.01514}{s^2 + 2.1525s + 3.3166} \end{aligned}$$

Note that  $G_0(0) = 0.909$ , which yields a steady state error of  $e_{ss} = 0.091$ . To achieve a steady state error of zero we need a prefilter with a value of  $G_{pf} = 1/0.909 = 1.100$ . Finally, to determine the controller, we use the formula

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

which produces the controller

$$\begin{aligned} G_c(s) &= \frac{1.401s^2 + 1.401}{0.4646s^2 + s + 0.1401} \\ &= \frac{1.401[s^2 + 1]}{0.4646s^2 + s + 0.1401} \end{aligned}$$

Note that the controller has been scaled, and there will be a pole/zero cancellation between the plant and the controller. Since these are marginally stable poles, this may not be a good idea.

**Example 3.** Suppose we have the plant with transfer function

$$G_p(s) = \frac{0.3(s+2)}{0.01s^2 + 0.2s + 1}$$

and we want to find  $G_0(s)$  to minimize

$$J = \int_0^{\infty} \{15 [y(t) - r(t)]^2 + u^2(t)\} dt$$

Clearly  $q = 15$ ,  $G_p(s)$  is a proper transfer function, and  $N(s)$  and  $D(s)$  have no common factors. However, before we use the algorithm, we must be sure  $D(s)$  is a monic polynomial. To do this, we multiply both the top and bottom by 100

$$G_p(s) = \frac{100}{100} \frac{0.3(s+2)}{0.01s^2 + 0.2s + 1} = \frac{30(s+2)}{s^2 + 20s + 100}$$

Now  $N(s) = 30(s+2)$ ,  $D(s) = s^2 + 20s + 100$ , and  $D(s)$  is monic. So we have

$$\begin{aligned} D(s) &= s^2 + 20s + 100 & N(s) &= 30(s+2) \\ D(-s) &= s^2 - 20s + 100 & N(-s) &= 30(-s+2) \end{aligned}$$

and

$$\begin{aligned} Q(s) &= D(s)D(-s) + qN(s)N(-s) \\ &= [s^2 + 20s + 100][s^2 - 20s + 100] + 15[30(s+2)][30(-s+2)] \\ &= [s^4 - 200s^2 + 10000] + 15[-900s^2 + 3600] \\ &= s^4 - 13,700s^2 + 64,000 \end{aligned}$$

Note that  $Q(s)$  is an even function of  $s$ . If it is not, you screwed up! Now we need to find the roots of  $Q(s)$ . These roots are  $\pm 117.027$  and  $\pm -2.161$ . To construct  $D_0(s)$  we use only those roots in the LHP, i.e., the roots at  $-117.027$  and  $-2.161$

$$\begin{aligned} D_0(s) &= (s + 117.027)(s + 2.161) \\ D_0(s) &= s^2 + 119.2s + 252.9 \end{aligned}$$

Now, we can compute the optimal  $G_0(s)$  as

$$\begin{aligned} G_0(s) &= \frac{qN(0)N(s)}{D_0(0)D_0(s)} = \frac{(15)(60)30(s+2)}{(252.9)(s^2 + 119.2s + 252.9)} \\ &= \frac{106.7(s+2)}{s^2 + 119.2s + 252.9} \end{aligned}$$

Note that  $G_0(0) = 0.8438$ , which yields a steady state error for a step input of  $e_{ss} = 0.156$ . For a steady state error of zero we would need a prefilter with a value of  $G_{pf} = 1/0.8438 = 1.185$ . Finally, to determine the controller, we use the formula

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

which produces the controller

$$\begin{aligned} G_c(s) &= \frac{0.09s^2 + 1.8s + 9}{0.0253s^2 + 0.3153s + 1} \\ &= \frac{0.09[s^2 + 20s + 100]}{0.0253s^2 + 0.3153s + 1} \end{aligned}$$

Note that this controller has been scaled and there is a pole/zero cancellation between the controller and the plant. The poles being cancelled are stable poles, so this is probably acceptable.

## 9.4 Dynamic Prefilters

Up to this point, we have used only constant prefilters  $G_{pf}$  to scale the input to achieve the desired steady state error. However, sometimes the plant or the controller introduces zeros into the closed loop transfer function that we do not want. One way to eliminate these zeros is with a *dynamic prefilter*,  $G_{pf}(s)$ . This is best illustrated with an example.

**Example 4.** Suppose we have the plant with transfer function

$$G_p(s) = \frac{(s + 3)}{s^2 + 16s + 68}$$

and we want to find  $G_0(s)$  to minimize

$$J = \int_0^{\infty} \{8[y(t) - r(t)]^2 + u^2(t)\} dt$$

Clearly  $q = 8$ ,  $G_p(s)$  is a proper transfer function,  $N(s)$  and  $D(s)$  have no common factors, and  $D(s)$  is a monic polynomial. Now  $N(s) = (s + 3)$ ,  $D(s) = s^2 + 16s + 68$ . So we have

$$\begin{aligned} D(s) &= s^2 + 16s + 68 & N(s) &= (s + 3) \\ D(-s) &= s^2 - 16s + 68 & N(-s) &= (-s + 3) \end{aligned}$$

and

$$\begin{aligned} Q(s) &= D(s)D(-s) + qN(s)N(-s) \\ &= [s^2 + 16s + 68][s^2 - 16s + 68] + 8[(s + 3)][(-s + 3)] \\ &= s^4 - 128s^2 + 4696 \end{aligned}$$

Note that  $Q(s)$  is an even function of  $s$ . If it is not, you screwed up! Now we need to find the roots of  $Q(s)$ . These roots are  $\pm 8.1403 \pm j1.5046$ . To construct  $D_0(s)$  we use only those roots in the LHP, i.e., the roots at  $-8.1403 + j1.5046$  and  $-8.1403 - j1.5046$

$$\begin{aligned} D_0(s) &= (s + 8.1403 - j1.5046)(s + 8.1403 + j1.5046) \\ D_0(s) &= s^2 + 16.2805s + 68.5274 \end{aligned}$$

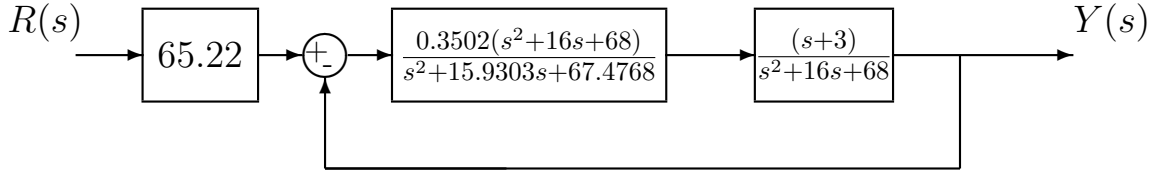
Now, we can compute the optimal  $G_0(s)$  as

$$\begin{aligned} G_0(s) &= \frac{qN(0)N(s)}{D_0(0)D_0(s)} = \frac{(8)(3)(s+3)}{(68.5274)(s^2 + 16.2805s + 68.5274)} \\ &= \frac{0.3502(s+3)}{s^2 + 16.2805s + 68.5274} \end{aligned}$$

Next we adjust the prefilter gain so the steady state error for a step input is zero. Since  $G_0(0) = 0.01533$ , we need to set the prefilter gain to 65.22. Finally, the controller is computed to be

$$G_c(s) = \frac{0.3502(s^2 + 16s + 68)}{s^2 + 15.9303s + 67.4768}$$

The complete feedback control system is shown below.



The step response of this system is shown in Figure 20. As the figure indicates, the steady state error for a step input will eventually be zero, but the transient response is less than satisfactory. At this point, we could go back and increase the value of  $q$  and hope for better response, or we can realize that some of our problem is caused by the zero at  $-3$ . Since there is no pole near  $-3$ , there is nothing to effectively cancel this behavior. Hence we want to insert a *dynamic prefilter* that will do two things: (1) cancel the zero in the closed loop transfer function, and (2) provide a steady state error for a step input of 0. Our new closed loop transfer function will be

$$\begin{aligned} G_0(s) &= \frac{G_{pf}}{(s+3)} \frac{0.3502(s+3)}{s^2 + 16.2805s + 68.5274} \\ &= \frac{G_{pf}0.3502}{s^2 + 16.2805s + 68.5274} \end{aligned}$$

For zero steady state error for a step input we need

$$G_0(0) = \frac{G_{pf}0.3502}{68.5274} = 1$$

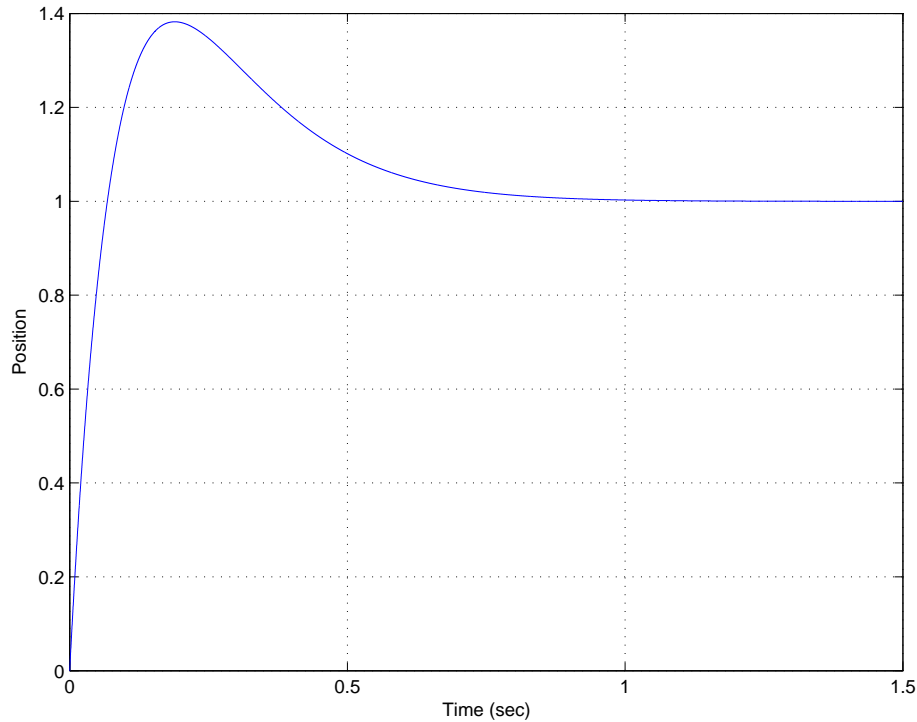


Figure 20: The unit step response of quadratic optimal system from Example 4, utilizing only a constant prefilter  $G_{pf} = 65.22$ .

or  $G_{pf} = 195.68$ . The complete control system with the dynamic prefilter is shown below:

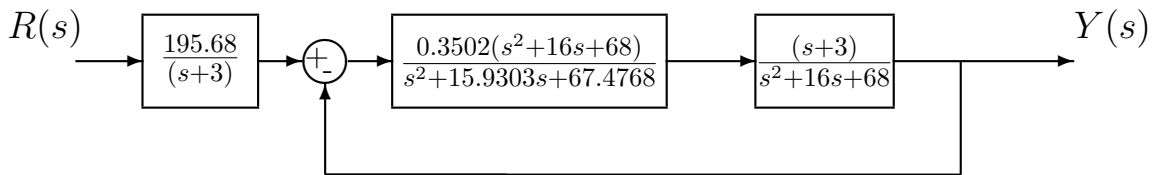


Figure 21 show the step response with the dynamic prefilter. Note the percent overshoot has been essentially eliminated, and the transient behavior is now more like that of an overdamped or critically damped ideal second order system.

While the dynamic prefilter has eliminated the zero from the closed loop transfer function, and improved the transient response, it should be noted that *the prefilter is outside the feedback loop*. In many instances this is not a problem, however, *any value outside the feedback loop is more sensitive to parameter variation and drift than values or transfer functions inside the feedback loop*. We'll have more to say about this when we discuss system sensitivity later.

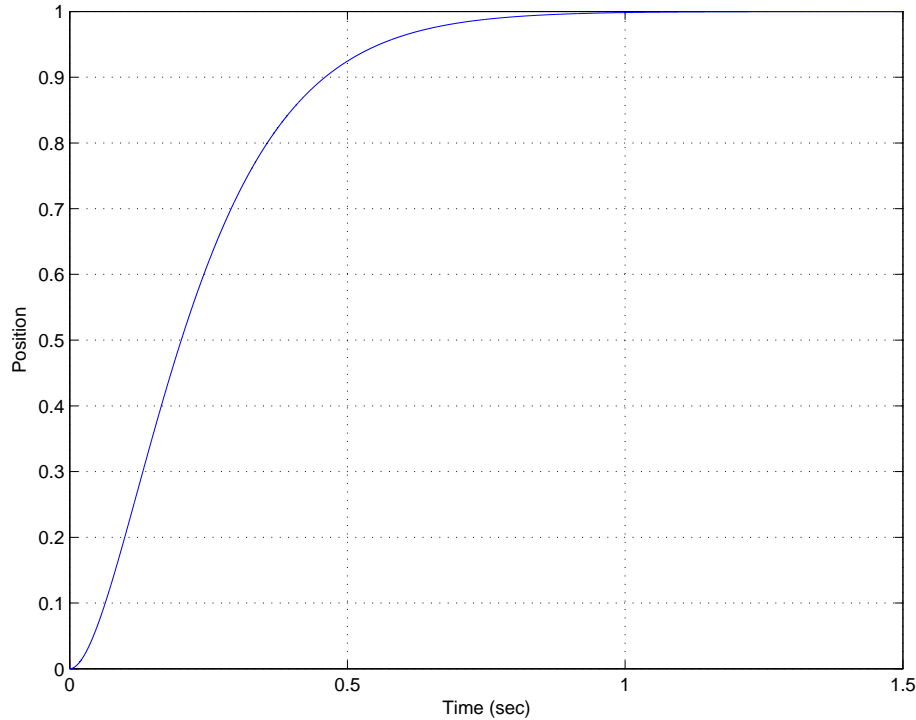


Figure 21: The unit step response of quadratic optimal system from Example 4, utilizing a dynamic prefilter  $G_{pf} = \frac{195.6808}{(s+3)}$ .

## 9.5 Type One Systems

The method we have presented for determining the quadratic optimal controller will always give us a stable controller. However, sometimes we might want to change this method slightly which may either give us an even better controller, or perhaps one which is unstable and therefore unusable. In our next chapter we will start discussing different system types and how they help us determine system parameters such as position and velocity errors very quickly. However, at this point we will revisit two of our previous examples and remove the need for the prefilter. This will mean that we have no blocks outside the feedback loop and, for these examples, produce what are called type 1 systems. The general idea is to take the transfer function  $G_0(s)$  determined by the quadratic optimal formula, and scale it directly so  $G_0(0) = 1$ , thus avoiding the need for a prefilter.

**Example 2 (Again).** Here we have the plant

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

and we determined that for  $q = 10$  we should have the closed loop transfer function

$$G_0(s) = \frac{3.01514}{s^2 + 2.1525s + 3.3166}$$



and the corresponding controller

$$G_c(s) = \frac{1.401[s^2 + 1]}{0.4646s^2 + s + 0.1401}$$

and prefilter

$$G_{pf} = 1.1001$$

Now, instead, let's modify  $G_0(s)$  so that  $G_0(0) = 1$ . The easiest way to do this is to multiply by the prefilter value, so

$$\begin{aligned} G_0(s) &= \frac{1.100 * 3.01514}{s^2 + 2.1525s + 3.3166} \\ &= \frac{3.3166}{s^2 + 2.152s + 3.3166} \end{aligned}$$

Clearly  $G_0(0) = 1$ . Next we compute  $G_c(s)$  as before, to get

$$G_c(s) = \frac{3.015[s^2 + 1]}{s^2 + 2.212s}$$

Note that now we can factor  $\frac{1}{s}$  from the controller, which we could not do before. This is a good thing.

**Example 3 (Again).** Here we have the plant

$$G_p(s) = \frac{30(s + 2)}{s^2 + 20s + 100}$$

and we determined that for  $q = 15$  we should have the closed loop transfer function

$$G_0(s) = \frac{106.7(s + 2)}{s^2 + 119.2s + 252.9}$$

and the corresponding controller

$$G_c(s) = \frac{0.09[s^2 + 20s + 100]}{0.0253s^2 + 0.3153s + 1}$$

and prefilter

$$G_{pf} = 1.1851$$

Now, instead, let's modify  $G_0(s)$  so that  $G_0(0) = 1$ . The easiest way to do this is to multiply by the prefilter value, so

$$\begin{aligned} G_0(s) &= \frac{1.1851 * 106.7[s + 2]}{s^2 + 119.2s + 252.9} \\ &= \frac{126.45[s + 2]}{s^2 + 119.2s + 252.9} \end{aligned}$$

Clearly  $G_0(0) = 1$ . Next we compute  $G_c(s)$  as before, to get

$$G_c(s) = \frac{0.4215[s^2 + 20s + 100]}{s^2 - 7.25s}$$

Note that now we can factor  $\frac{1}{s}$  from the controller, which we could not do before. This is a good thing. However, our controller is no longer stable, which means we would not want to implement it and would use our original controller.

## 9.6 Frequency Response Matching

Although all of the model matching method presented so far have been time domain models, we need to keep in mind the fact that everything we do in the time-domain has an effect in the frequency domain. Thus, when we have designed a closed loop control system with transfer function  $G_0(s)$ , we have to realize that *we have really constructed a filter between the input and output signal!* Hence, if the input to our system is  $u(t) = A \cos(\omega t + \theta)$  the steady state output to our system will be given by

$$y(t) = A|G_0(j\omega)| \cos(\omega t + \theta + \angle G_0(j\omega))$$

**Example 5.** Assume we have designed a controller using model matching techniques so the closed loop transfer function  $G_0(s)$  is that of a third order deadbeat system

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

The step response for this system with  $\omega_0 = 10$  rad/sec is shown in Figure 22 and the Bode plot for this system is shown in Figure 23. For this system the bandwidth is the same as  $\omega_0$ , about 10 rad/sec. Now, in addition to the step response, we may also be interested in how well our system tracks input signals of varying frequencies. Figures 24 and 25 display the output of our control system/filter when the input consists of sine waves with unit amplitude and varying frequencies, from  $\omega = 0.25\omega_0$  to  $\omega = 2\omega_0$ . These figures display the same information that the Bode plot does in a different way. For input frequencies between 0 and  $\omega_0$ , the output signal has a magnitude of 1 (for zero frequency) down to 0.707 (for  $\omega = \omega_0$ ). As the input frequency increases beyond  $\omega_0$ , the amplitude of the output decreases substantially. In addition to the amplitude, we usually need to worry about the phase. In a control system we usually want the output to equal the input, in both magnitude and phase. For this system we see that as the input frequencies increase, the phase difference between the input and output signals increases, which corresponds to a longer delay between the input and output signal.

Although in the majority of this course we are concerned with following time-domain signals such as steps and ramps, it may be important sometimes to be able to design a system that meets certain frequency domain criteria. One way we can approach this is again by using a model matching method. However, this method assumes you can determine a transfer function that meets the required frequency domain requirements and is still implementable.

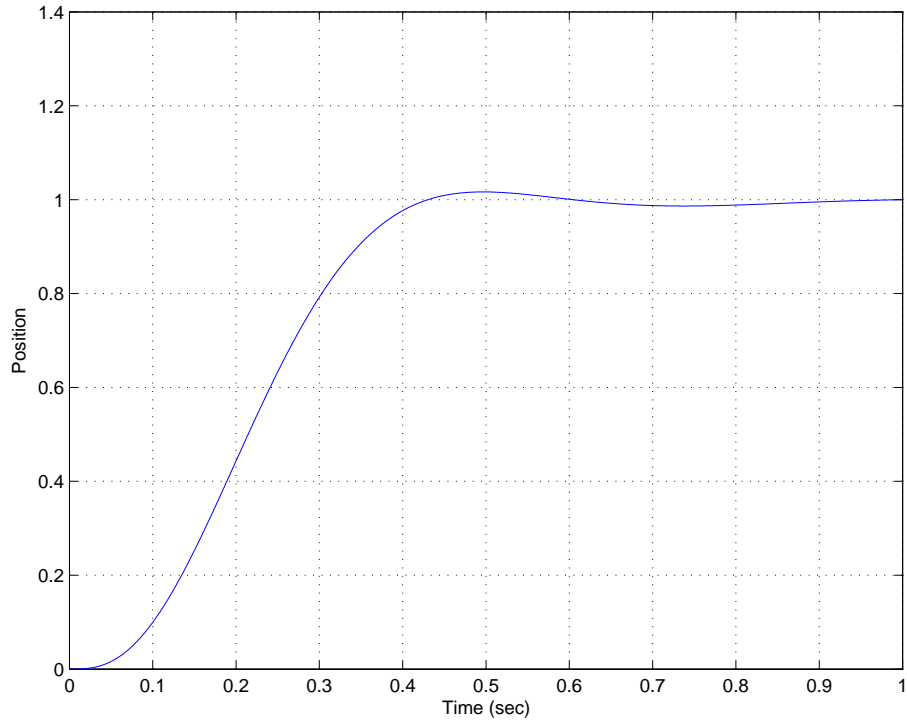


Figure 22: The unit step response of the third order deadbeat system from Example 5 ( $\omega_0 = 10$  rad/sec.)

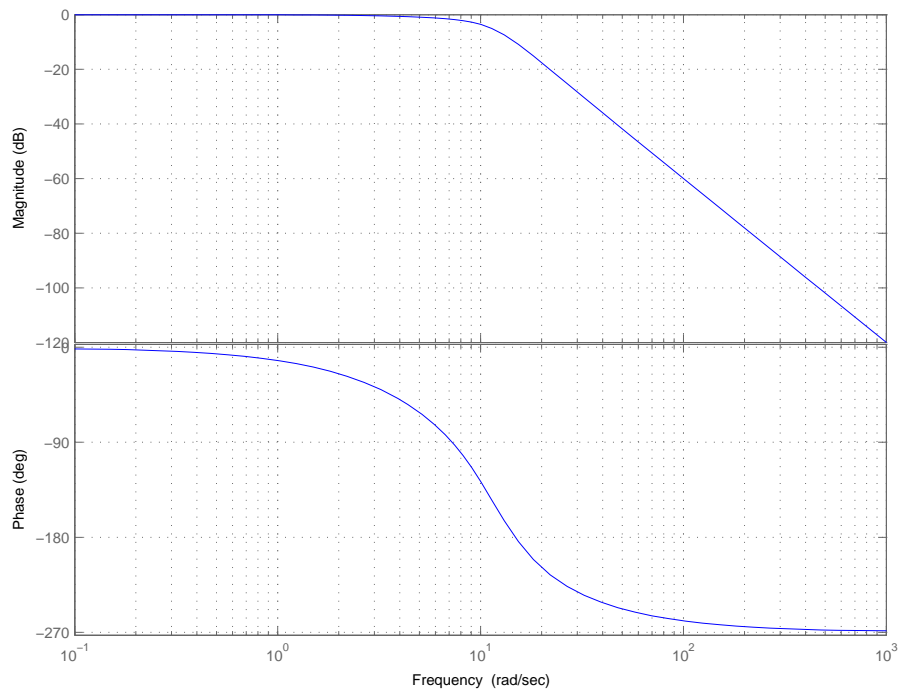


Figure 23: The Bode plot of the third order deadbeat system from Example 5 ( $\omega_0 = 10$  rad/sec.)

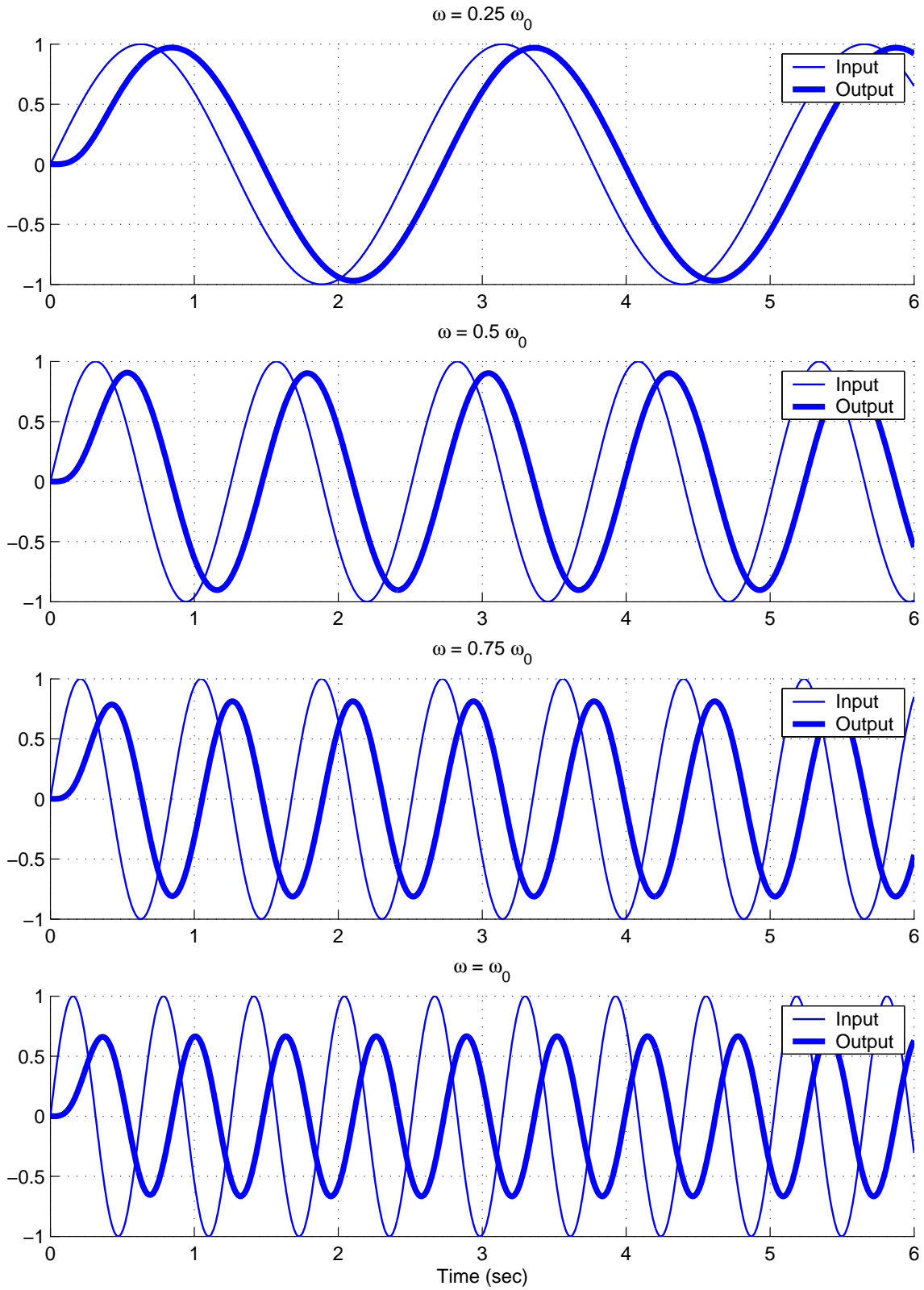


Figure 24: The response of the third order deadbeat system from Example 5 ( $\omega_0 = 10$  rad/sec) to input sine waves with amplitude 1 of varying frequency  $\omega \leq \omega_0$ .

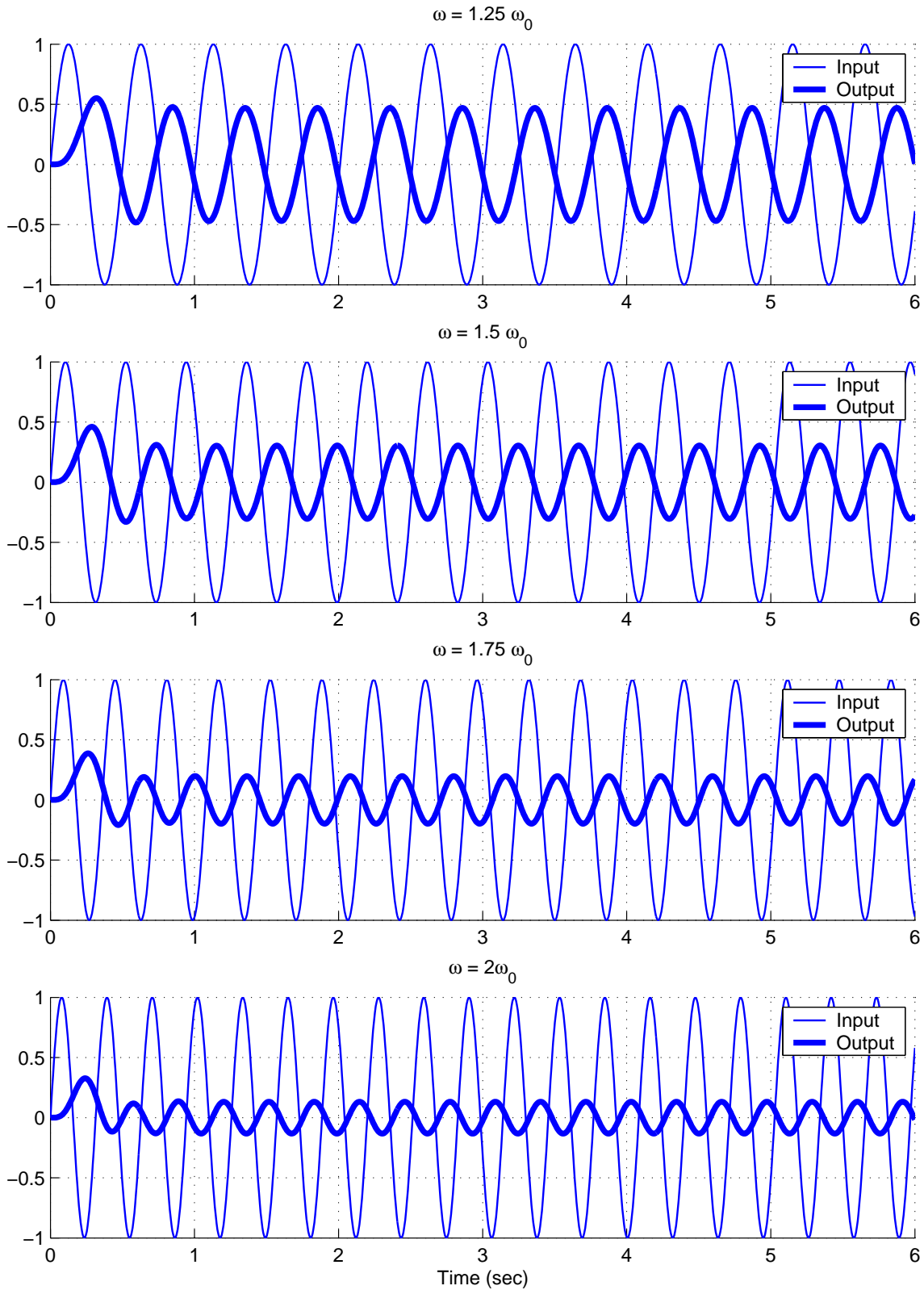


Figure 25: The response of the third order deadbeat system from Example 5 ( $\omega_0 = 10$  rad/sec) to input sine waves with amplitude 1 of varying frequency  $\omega > \omega_0$ .

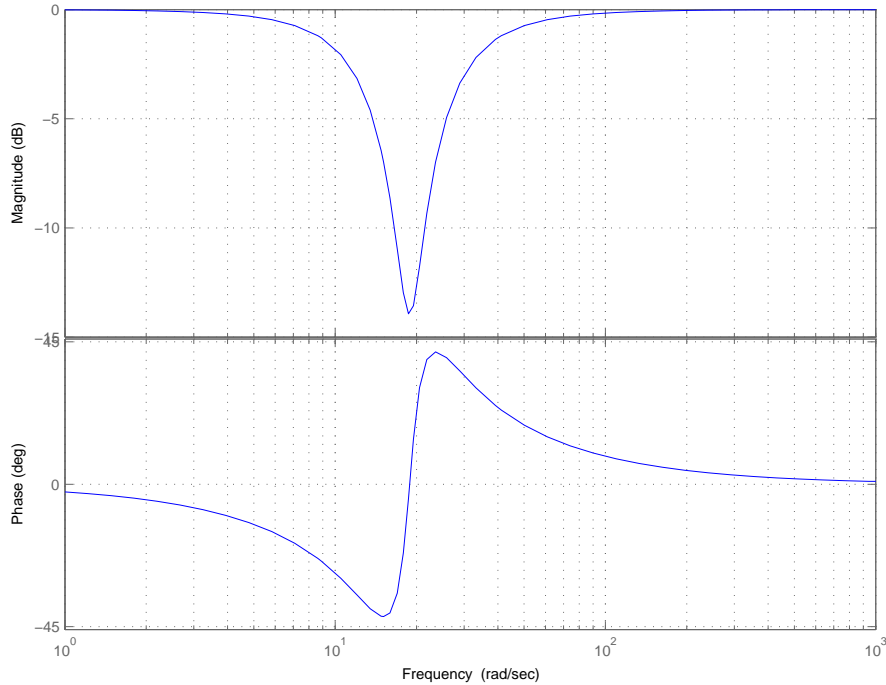


Figure 26: The frequency response of a notch filter with  $\omega_0 = 6\pi$  and  $\zeta_0 = 0.1$ .

**Example 6.** Let's assume that we have a plant with the transfer function

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

In addition, let's assume we want to have zero steady state error for a step input and the output track the input as well as possible. However, any input signal at 3 Hz is to be eliminated since it is likely to be interference from another part of the system. Hence we want our control system to look something like a *notch filter*. One example of a second order notch filter is given by a transfer function of the form

$$G_n(s) = \frac{s^2 + 2\zeta_0\omega_0s + \omega_0^2}{s^2 + \omega_0s + \omega_0^2}$$

Here  $\omega_0$  is the notch frequency and  $\zeta_0$  controls the width of the notch. Since we want to eliminate input signals with frequencies at  $3 \times 2\pi$  radians/sec, we set  $\omega_c = 6\pi = 18.8$  radians/sec. Figures 26, 27 and 28 display the Bode plot of this filter for  $\zeta_0 = 0.1, 0.01,$  and  $0.001,$  respectively. Figures 29 and 30 show the time domain response for input sin waves of various frequencies for the notch filter shown in 28, i.e., with  $\omega_0 = 6\pi$  and  $\zeta_0 = 0.001$ . As these figures demonstrate, as the frequency of the input sine wave increases towards  $\omega_0$ , there is a lag between the input and output signals, while as the frequency of the input sine wave increases beyond  $\omega_0$  the output signal leads the input signal until the input signal is greater than approximately  $2\omega_0$ . These figures also show the *steady state* output of the system is zero when the input sine wave has a frequency equal to  $\omega_0$ . Note that this output is not always zero at this frequency, just during steady state.

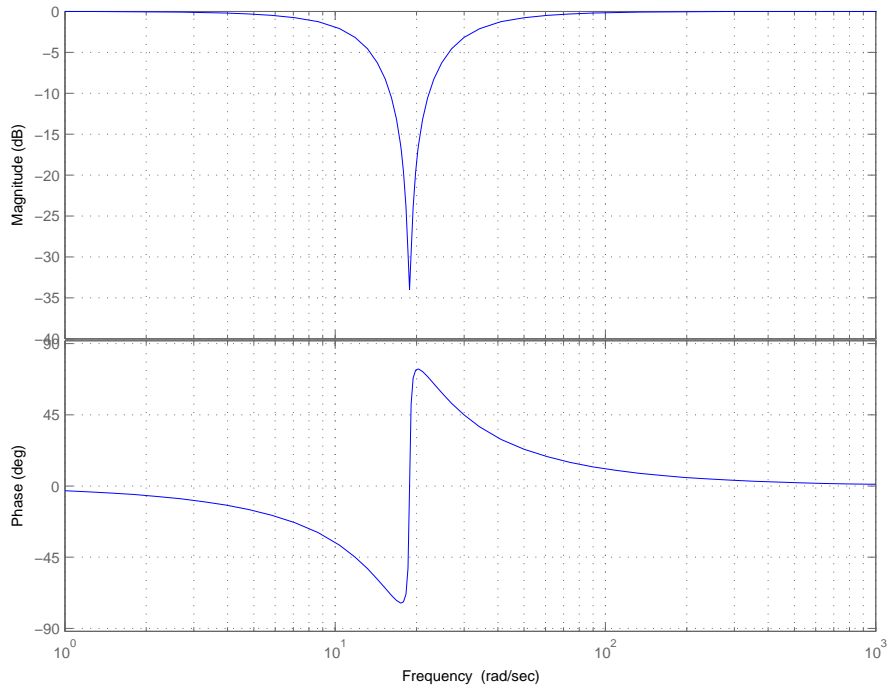


Figure 27: The frequency response of a notch filter with  $\omega_0 = 6\pi$  and  $\zeta_0 = 0.01$ .

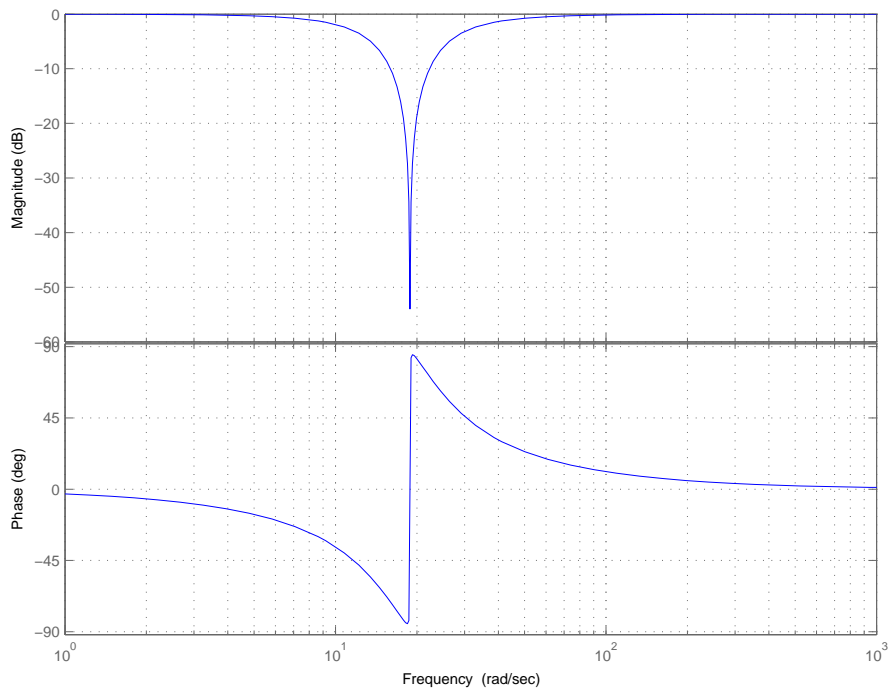


Figure 28: The frequency response of a notch filter with  $\omega_0 = 6\pi$  and  $\zeta_0 = 0.001$ .

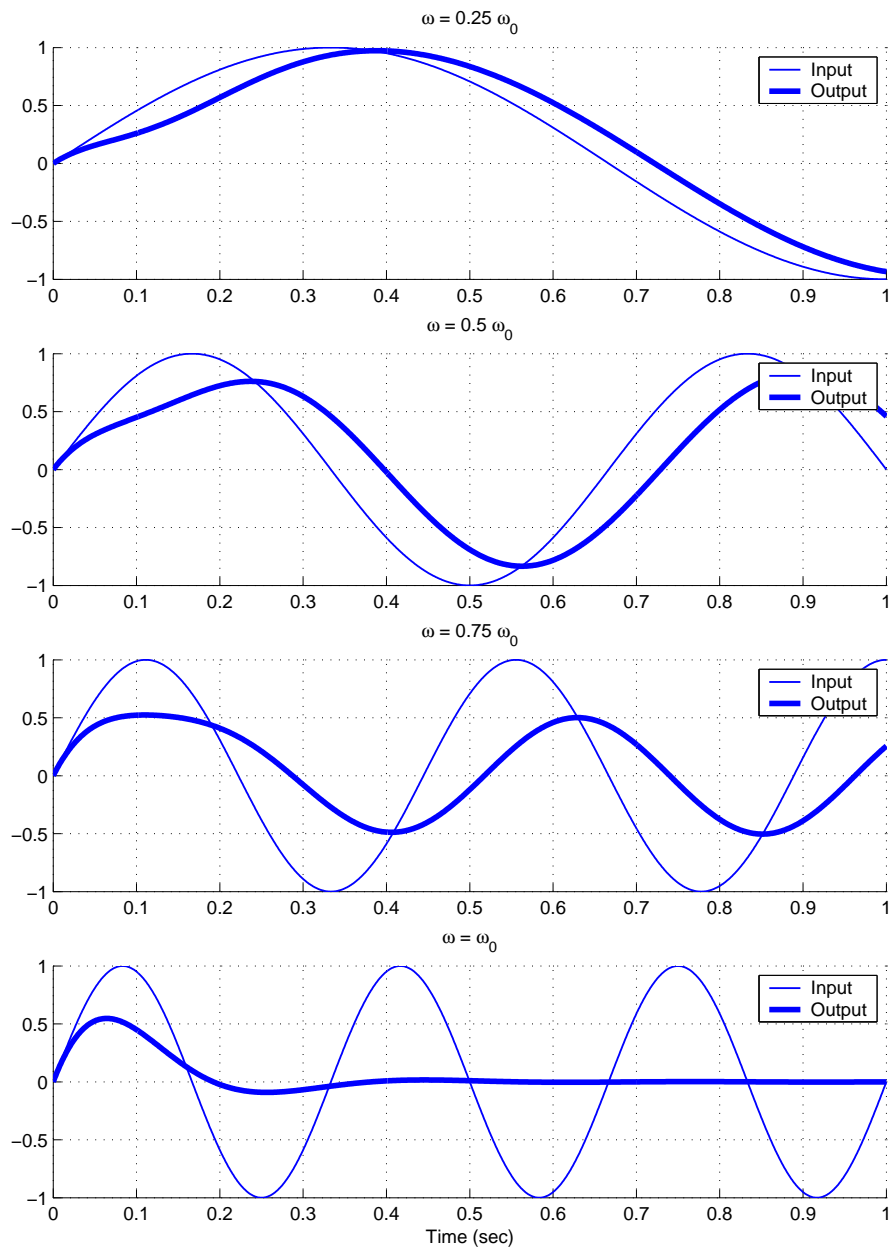


Figure 29: The response of the notch filter from Example 6 ( Figure 28,  $\omega_0 = 6\pi$  rad/sec,  $\zeta_0 = 0.001$ ) to input sine waves with amplitude 1 of varying frequency  $\omega \leq \omega_0$ .



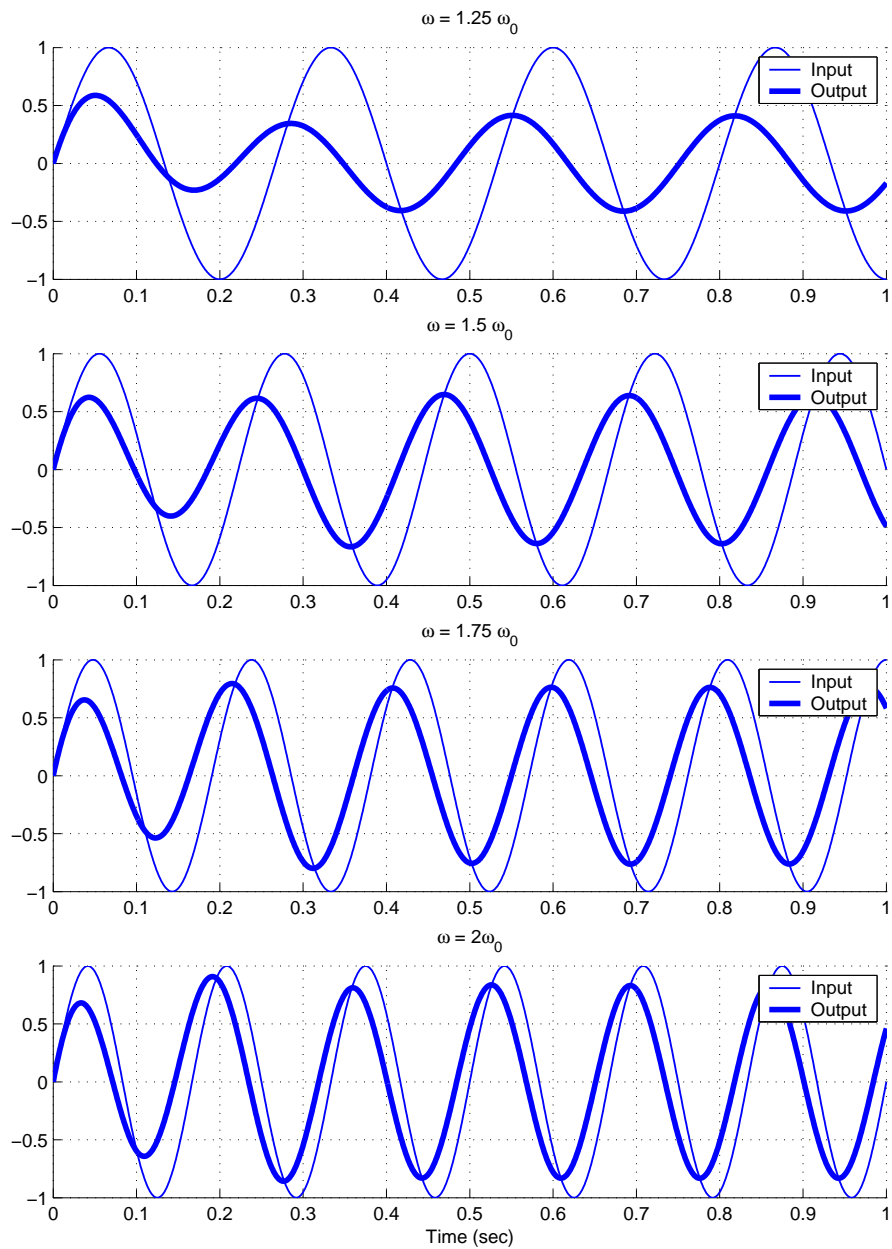


Figure 30: The response of the notch filter from Example 6 ( Figure 28,  $\omega_0 = 6\pi$  rad/sec,  $\zeta_0 = 0.001$ ) to input sine waves with amplitude 1 of varying frequency  $\omega > \omega_0$ .

This transfer function seems to accomplish what we want reasonably well, so we might be tempted to try and determine the controller directly using the formula

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

as before. However, there is a problem. In order for  $G_0(s)$  to be *implementable* we need to match the three conditions in the beginning of this chapter. The only one we will have difficulty with is (1): the degree of  $D_0(s)$  - degree of  $N_0(s) \geq$  degree  $D(s)$  - degree  $N(s)$ . For this notch system we have degree  $D_0(s)$  - degree  $N_0(s) = 0$  and degree  $D(s)$  - degree  $N(s) = 2$ . It looks like we can't use model matching in this case. However, we can if we think a little bit and then modify our desired system.

In any real system all of the input signals we care about will be bandlimited, i.e., they will have not frequency content above a certain frequency. One way to think about this is to think about the Fourier series representation for a periodic signal  $x(t)$ . When we try to reconstruct the original signal by summing its Fourier series, at some point there does not seem to be any improvement. If we stop summing the Fourier series at the  $N^{th}$  term in the series, then the bandwidth of the signal is approximately  $N\omega_f$ , where  $\omega_f$  is the fundamental frequency. In addition to the input signal, we must be aware of the fact that there is always noise in any system. Hence, we want to limit the bandwidth of our system to limit noise in the system while making the system bandwidth large enough for any likely input signal. We have not been given any minimum bandwidth in this problem, so let's make a reasonable assumption that our input signal has no frequency content above 80 Hz. This is a fairly reasonable assumption for many mechanical systems. Hence we want to limit the bandwidth beyond 80 Hz. Since we need two poles for our  $G_o(s)$  to be implementable, let's add two poles at  $80 * 2\pi$ , so the closed loop transfer function looks like

$$G_o(s) = \frac{s^2 + 2\zeta_0\omega_0s + \omega_0^2}{(s^2 + \omega_0s + \omega_0^2)(s + 160\pi)^2}$$

While these poles help with the bandwidth, we should be aware that they also contribute to the phase of the system. Finally, we probably still want  $e_p = 0$ , so we need  $G_0(0) = 1$ , hence our final closed loop transfer function is

$$\begin{aligned} G_0(s) &= \frac{(160\pi)^2(s^2 + 0.0120\pi 36\pi^2)}{(s^2 + 6\pi s + 36\pi^2)(s + 160\pi)^2} \\ &= \frac{252,661.8(s^2 + 0.0377s + 355.3)}{(s^2 + 18.8s + 355.3)(s + 502.6)^2} \end{aligned}$$

The frequency response for this system is displayed in Figure 31. As the figure indicates, the system bandwidth is limited to about  $80 * 2\pi = 502.6$  radians/sec and the notch will eliminate the 3Hz signals. Figure 32 shows the step response for this system. Finally, we can determine the controller as

$$G_c(s) = \frac{25,266.2s^4 + 26,218.7s^3 + 9,028,706.7s^2 + 8,979,126.9s + 17,954,443.7}{s^4 + 1,024.2s^3 + 19,304.9s^2 + 5,110,231.3s}$$

There are some very large numbers in this controller, and we probably need to "balance" the coefficients before we implement this, but that's the subject of another class.

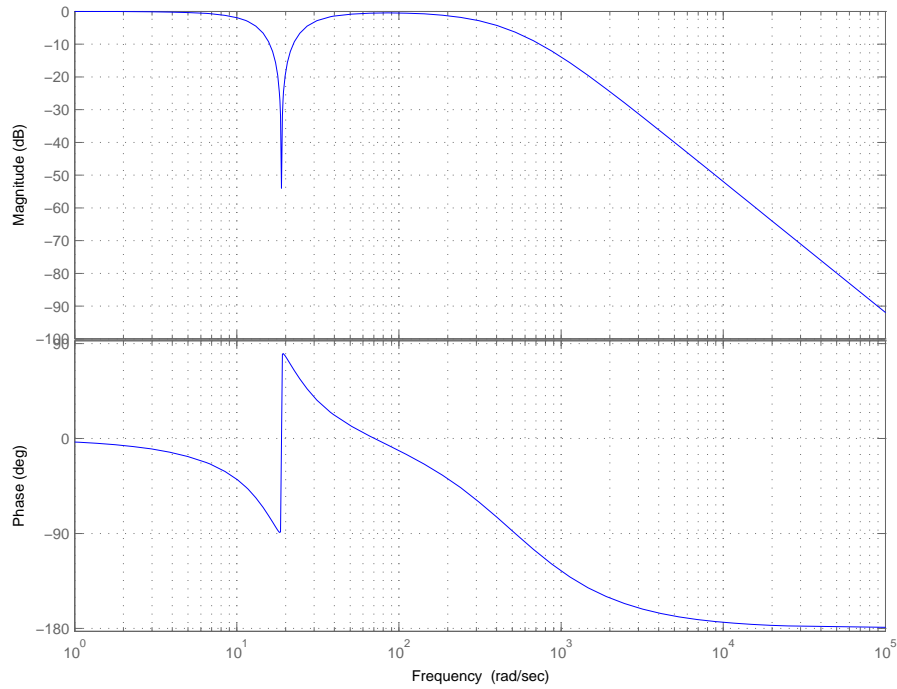


Figure 31: The frequency response of the final notch filter from Example 6. Here  $\omega_0 = 6\pi$  rad/sec,  $\zeta_0 = 0.001$ , and we have added two poles at  $160\pi$  radians/second.

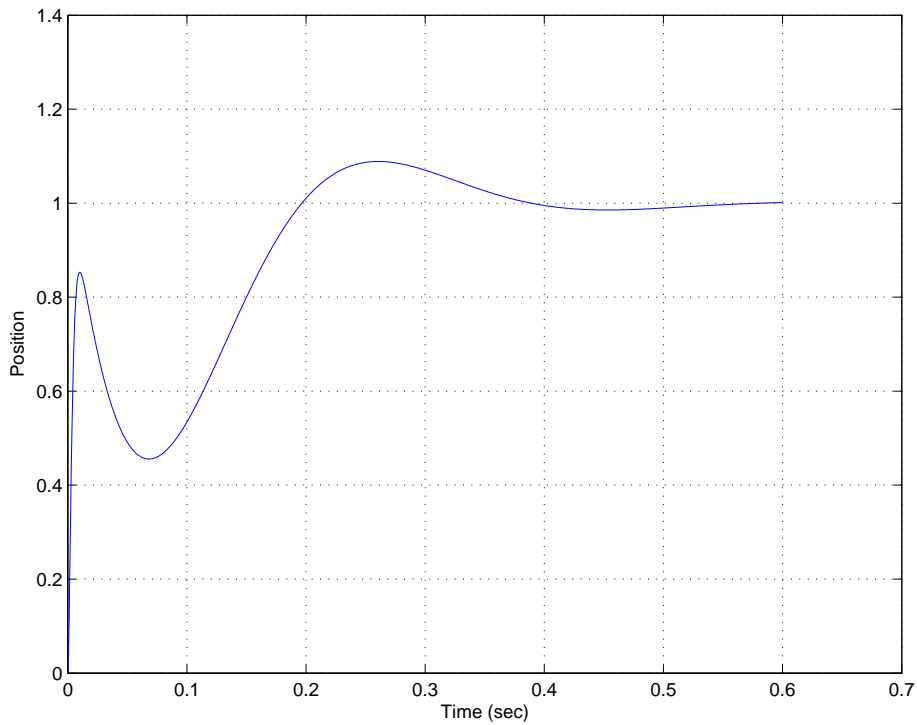


Figure 32: The unit step response of the final notch filter from Example 6. Here  $\omega_0 = 6\pi$  rad/sec,  $\zeta_0 = 0.001$ , and we have added two poles at  $160\pi$  radians/second.

## 9.7 Summary and Caveates

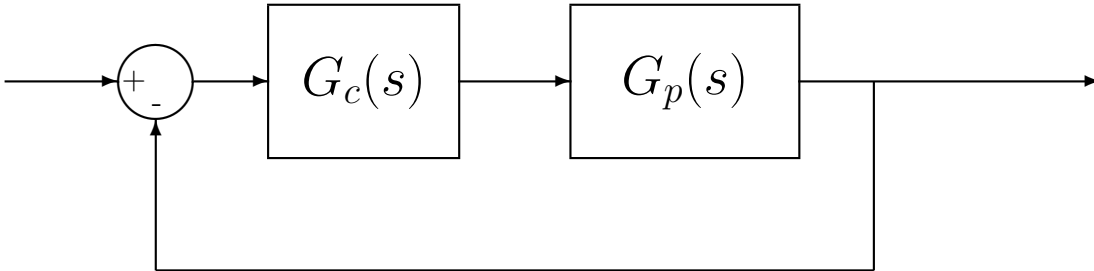
In the first part of this section, the conditions under which it is possible to obtain an implementable closed loop transfer function  $G_0(s)$  have been given, it may not be possible to find such a  $G_0(s)$  using the ITAE, deadbeat, or quadratic optimal criteria. In particular, the ITAE and deadbeat 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  $G(s)$  as<sup>2</sup>

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

---

<sup>2</sup>We 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  $e_{ss} = 0$ .

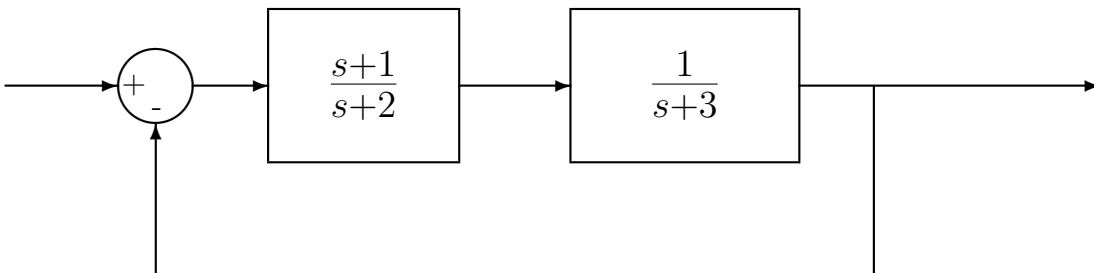
We can summarize these results in Table 1 below.

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

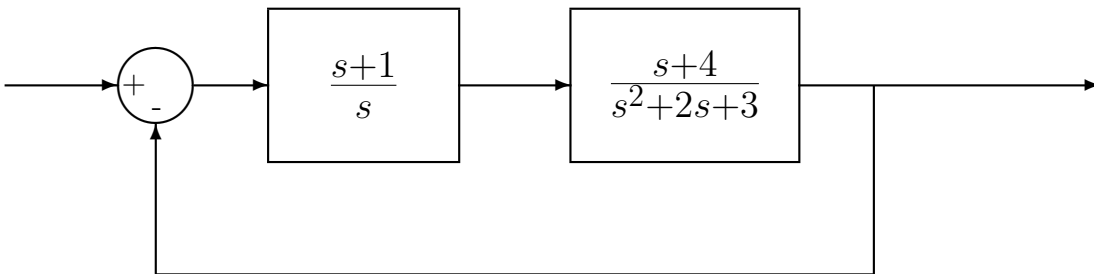
$$\begin{aligned} K_p &= \lim_{s \rightarrow 0} G(s) \\ &= \frac{1}{(2)(3)} \\ &= \frac{1}{6} \end{aligned}$$

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

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

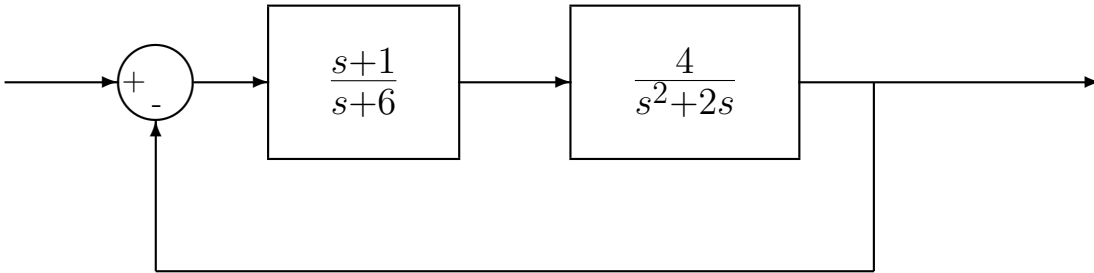
so the steady state error for a ramp input is

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



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

**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

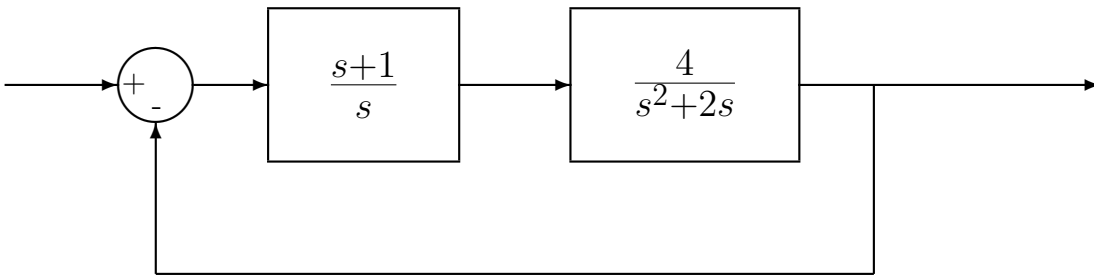
$$\begin{aligned}
 K_v &= \lim_{s \rightarrow 0} sG(s) \\
 &= \frac{(1)(4)}{(6)(2)} \\
 &= \frac{1}{3}
 \end{aligned}$$

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 solely 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.*

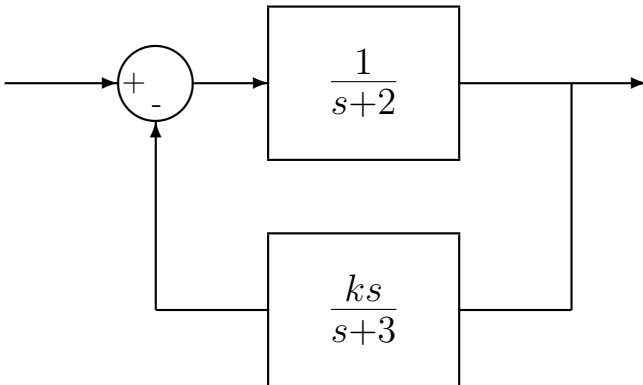
## 11.1 Standard Root Locus Form

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.

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

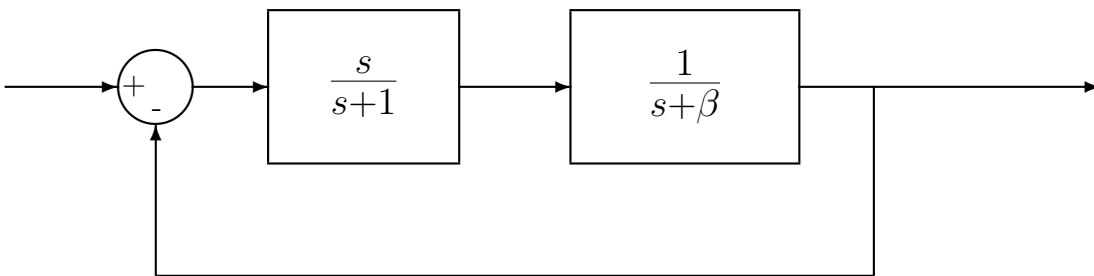
Hence the equation we need is

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

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  $\beta$  varies. To do this, we must first determine the closed loop transfer function.

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

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

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 33.

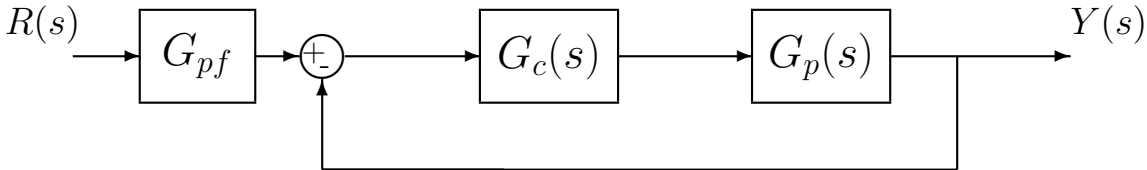


Figure 33: Assumed feedback configuration for root locus analysis in this chapter.

### 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  $G(s)$ .*

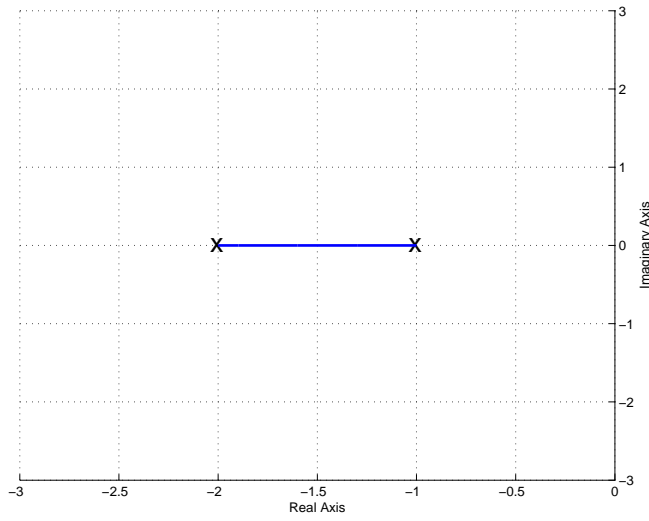
This is the starting point for all root locus construction. If, after identifying all segments of the real axis on the root locus, each pole is matched to a zero by a segment of the root locus, we are done. Note that there may be zeros at infinity. Figures 34, 35, and 36 display the real axis segments 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.

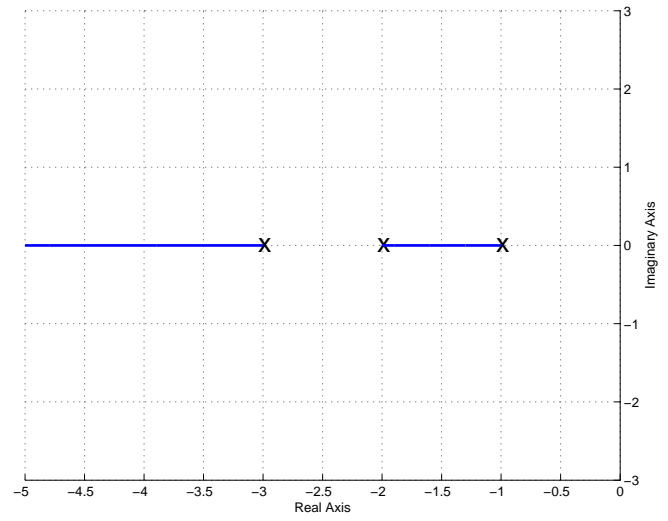
At this point,  $G_4(s)$  and  $G_6(s)$  are the only two completed root locus plots. In  $G_4(s)$  the pole at -3 is paired with a zero at infinity (along the real axis). In  $G_6(s)$  the pole at -1 is paired with the zero at -2, and the pole at -4 is paired with the zero at -3.

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 37 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.

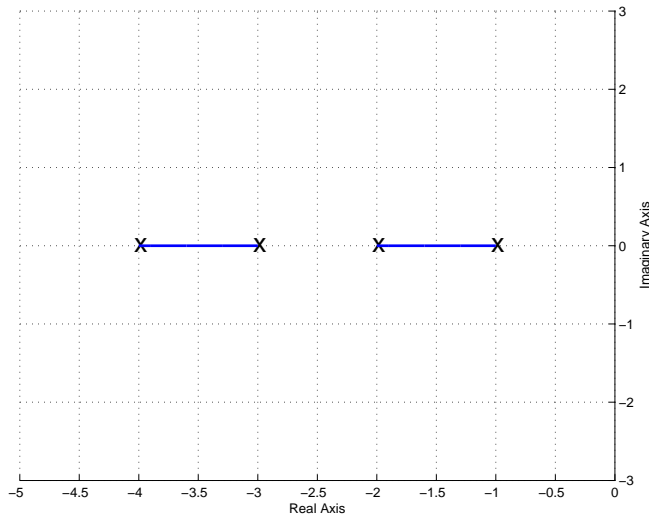
We cannot complete the remaining root locus plots because there are not enough "finite" zeros to pair with each pole. Hence these poles must pair with zeros at "infinity". However, since we are dealing with a two dimensional coordinate system, we need to figure out where (in terms of direction) this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



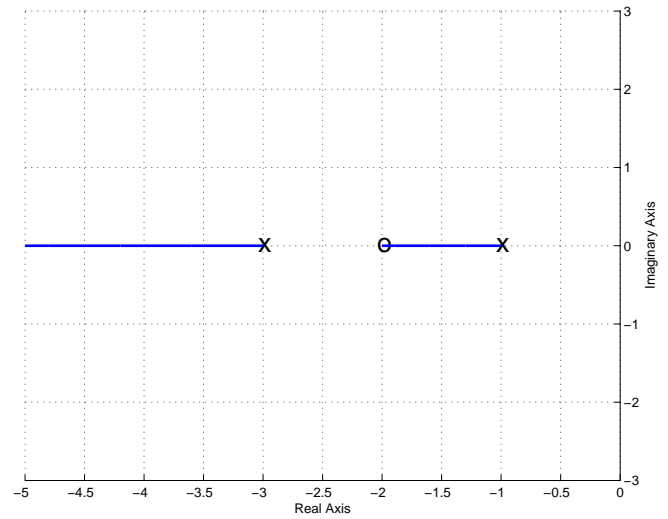
$$G_1(s) = \frac{1}{(s+1)(s+2)}$$



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



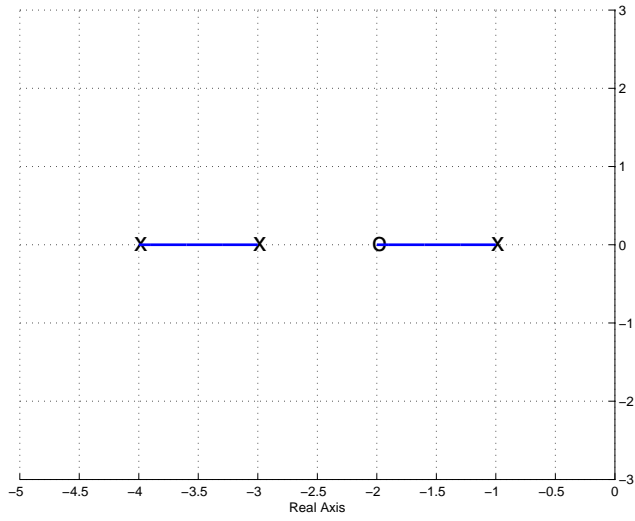
$$G_3(s) = \frac{1}{(s+1)(s+2)(s+3)(s+4)}$$



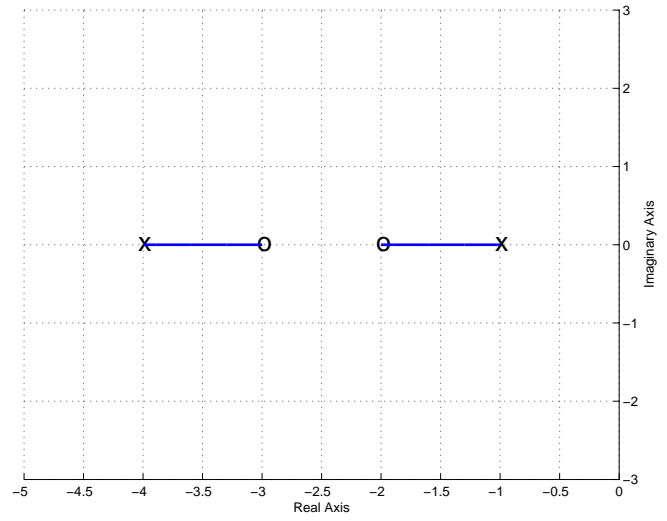
$$G_4(s) = \frac{(s+2)}{(s+1)(s+3)}$$

Figure 34: Root locus real-axis segments for  $G_1(s) - G_4(s)$ .

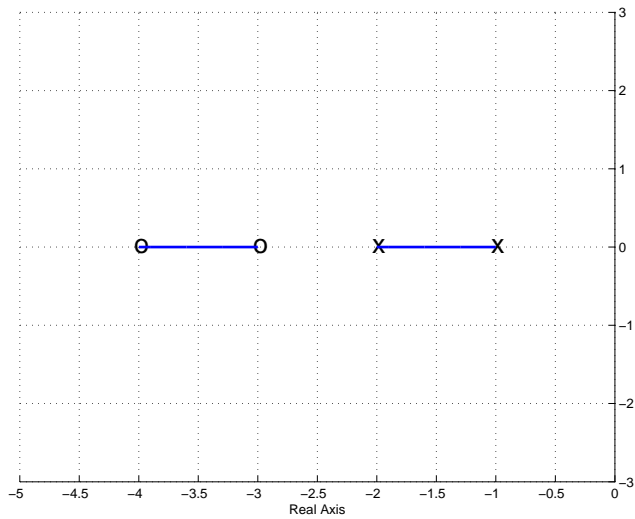




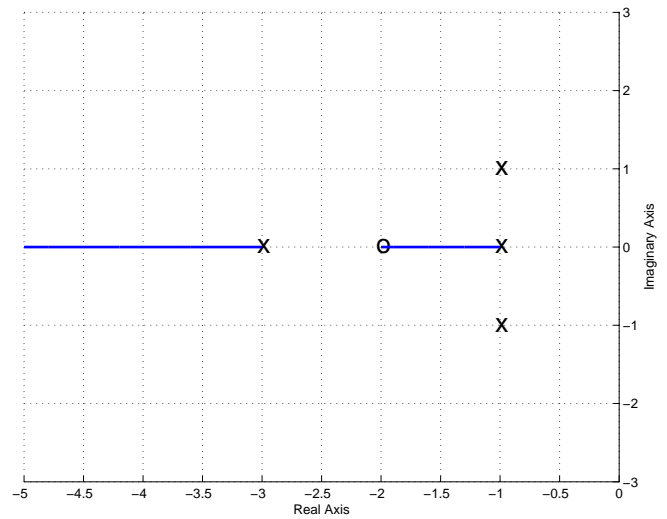
$$G_5(s) = \frac{(s+2)}{(s+1)(s+3)(s+4)}$$



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

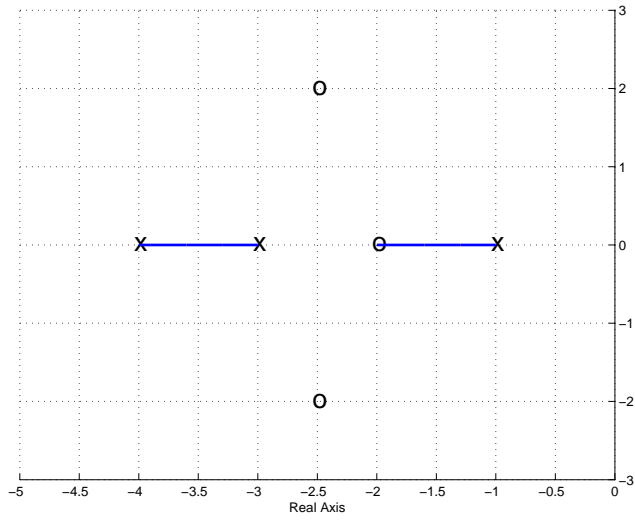


$$G_7(s) = \frac{(s+3)(s+4)}{(s+1)(s+2)}$$

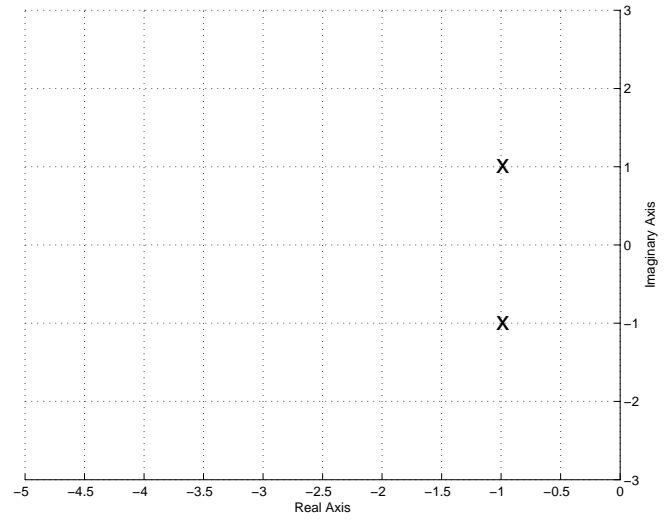


$$G_8(s) = \frac{(s+2)}{(s+1)(s+3)(s^2+2s+2)}$$

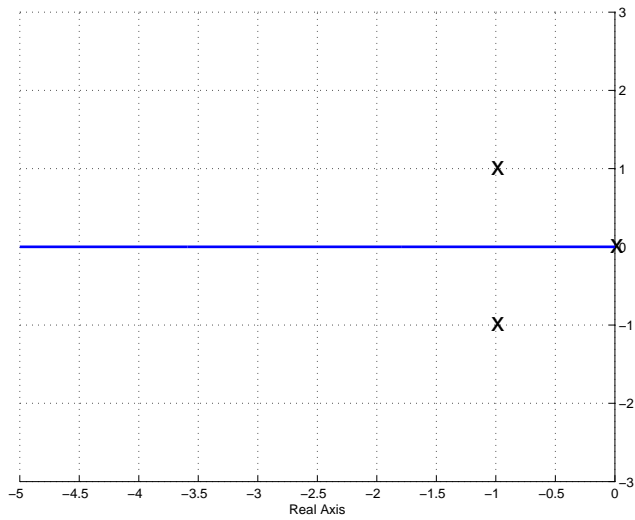
Figure 35: Root locus real-axis segments for  $G_5(s) - G_8(s)$



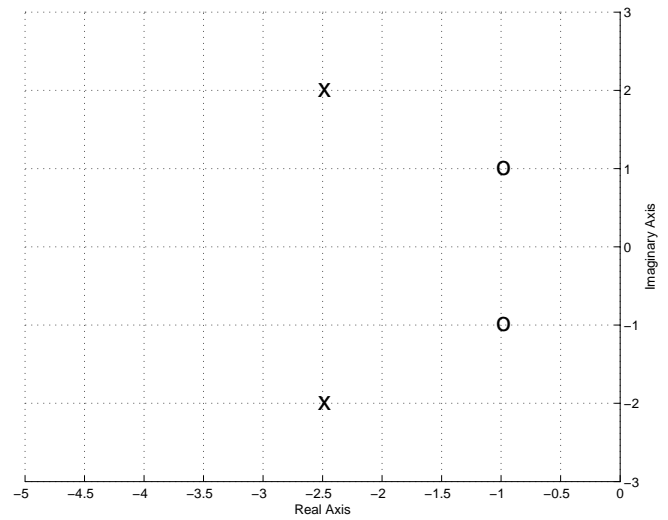
$$G_9(s) = \frac{(s+2)(s^2+5s+10.25)}{(s+1)(s+3)(s+4)}$$



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

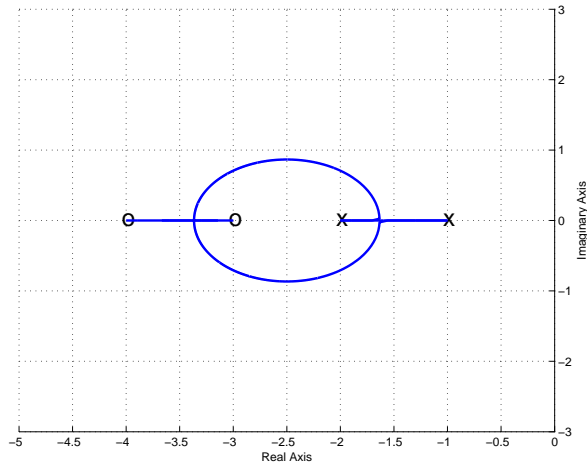


$$G_{11}(s) = \frac{1}{s(s^2+2s+2)}$$

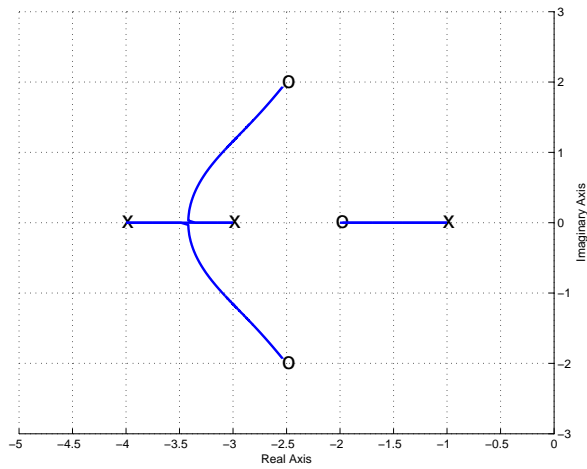


$$G_{12}(s) = \frac{s^2+2s+2}{s^2+5s+10}$$

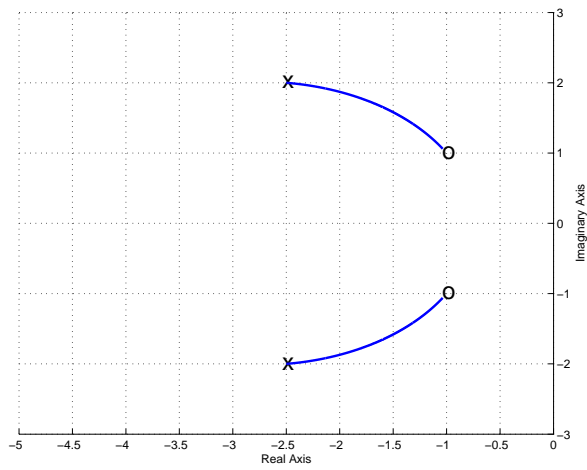
Figure 36: Root locus real-axis segments for  $G_9(s) - G_{12}(s)$ .



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



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



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

Figure 37: 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  $i^{\text{th}}$  pole of  $G(s)$ ,  $z_j$  is the  $j^{\text{th}}$  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  $360^\circ$  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  $-45^\circ$ . 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,  $\sigma_c$ . There are two poles, one at -1 and one at -2, and there are no zeros. Hence  $n = 2$  and  $m = 0$ . To determine  $\sigma_c$ ,

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

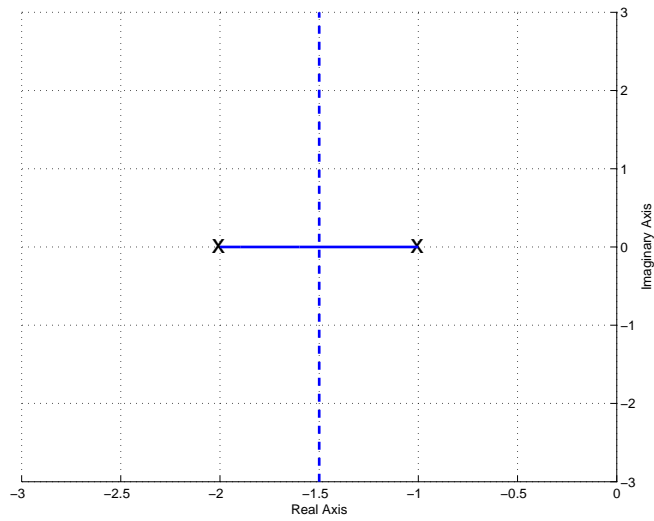
Hence the asymptotes radiate from this point. Next we compute the asymptote angles

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

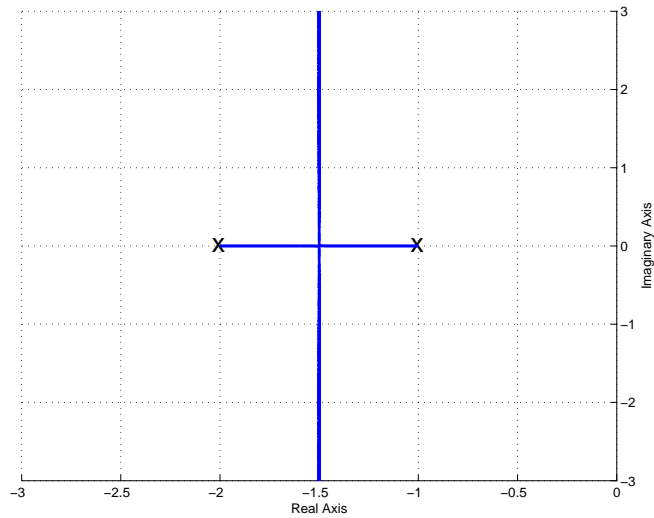
Since the root locus is symmetric, we know we must also have  $-90^\circ$ . We are looking for zeros for two poles and we have two angles, so we are done. Figure 38 shows both the real axis segment and the asymptotes for  $G_1(s)$  and then the completed root locus. Again, we don't know where the root 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,  $\sigma_c$ . 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  $\sigma_c$ ,

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



Real axis segment and the asymptotes



Completed root locus plot

Figure 38: 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  $\pm 90$  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  $-60^\circ$ . We are looking for zeros for three poles and we have three angles, so we are done. Figure 39 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,  $\sigma_c$ . 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  $\sigma_c$ ,

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

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  $-45^\circ$  and  $-135^\circ$ . We are looking for zeros for four poles and we have four angles, so we are done. Figure 40 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.

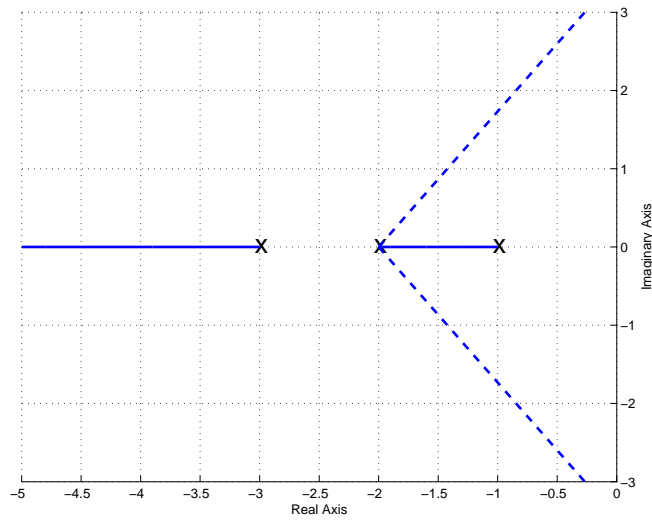
$G_5(s)$  First we determine the centroid of the asymptotes,  $\sigma_c$ . There are three poles, one at -1, one at -3, and one at -4, and one zero at -2. Hence  $n = 3$  and  $m = 1$ . To determine  $\sigma_c$ ,

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

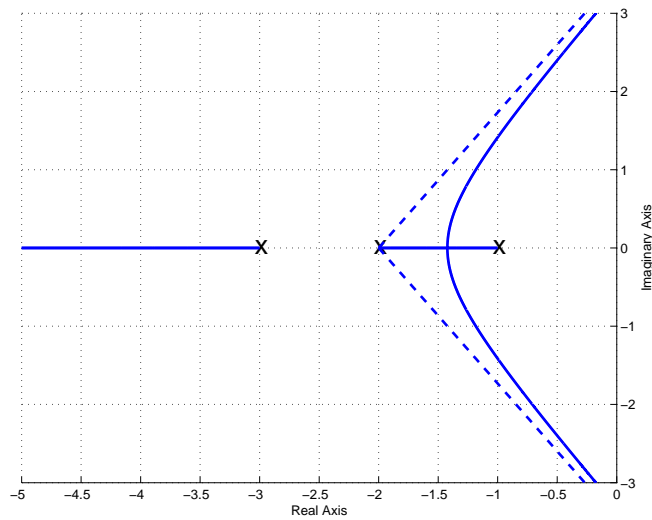
Hence the asymptotes radiate from this point. Next we compute the asymptote angles

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

Since the root locus is symmetric, we know we must also have  $-90^\circ$ . We are looking for zeros for two poles and we have two angles, so we are done. Figure 41 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

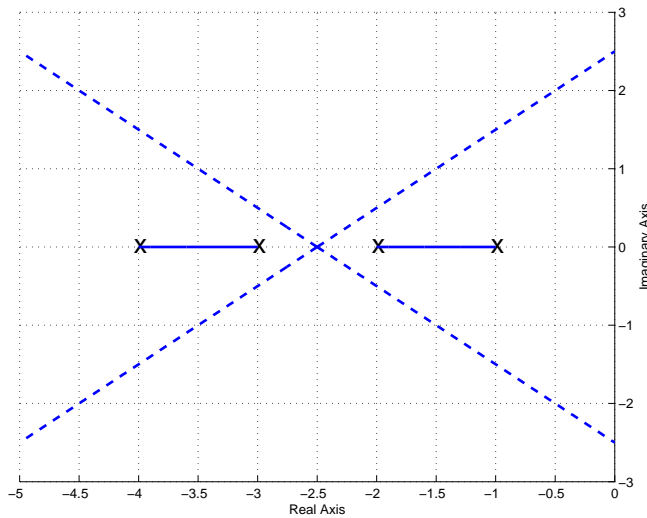


Real axis segment and the asymptotes

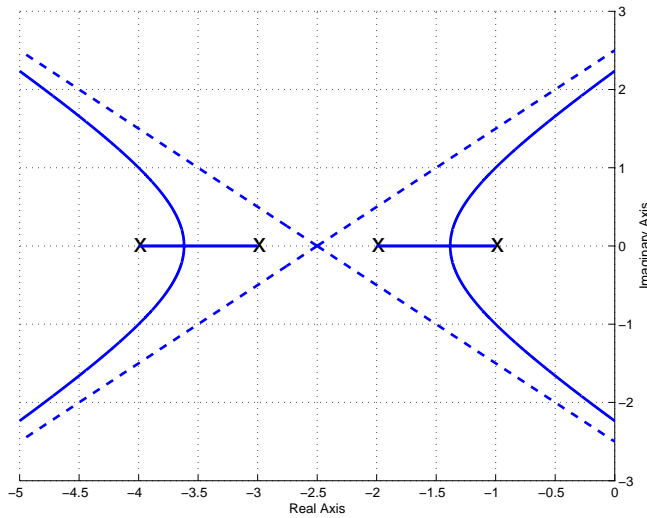


Completed root locus plot (with the asymptotes)

Figure 39: 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  $\pm 60$  degrees.



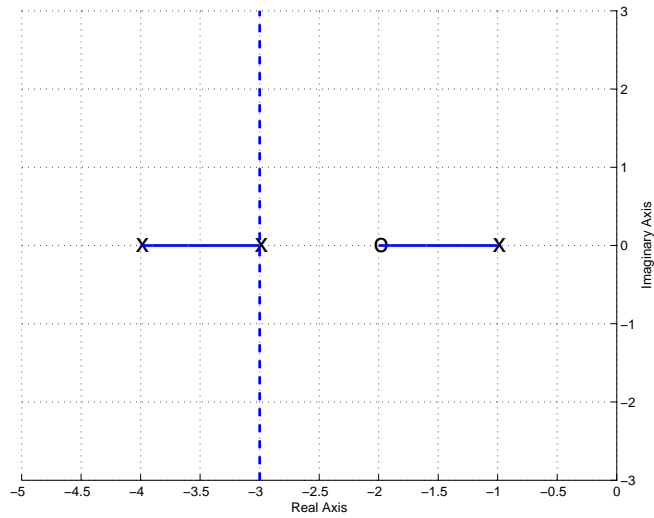
Real axis segment and the asymptotes



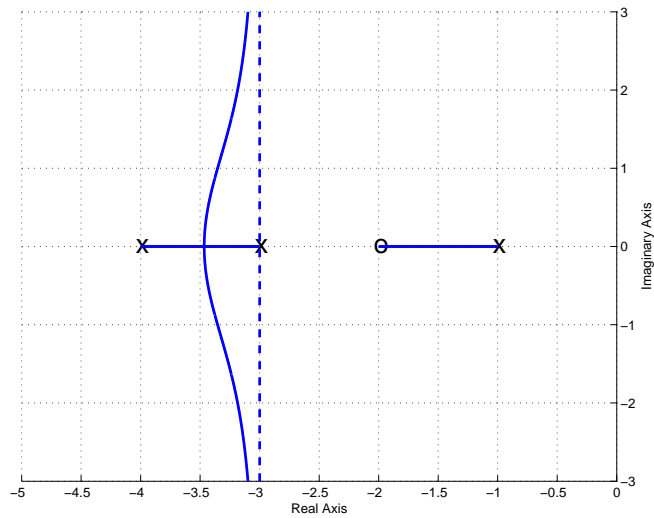
Completed root locus plot (with the asymptotes)

Figure 40: 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  $\pm 45$  and  $\pm 135$  degrees.





Real axis segment and the asymptotes



Completed root locus plot (with the asymptotes)

Figure 41: 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  $\pm 90$  degrees.

real axis and is symmetric.

$G_8(s)$  First we determine the centroid of the asymptotes,  $\sigma_c$ . 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  $\sigma_c$ ,

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

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  $-60^\circ$ . We are looking for zeros for three poles and we have three angles, so we are done. Figure 42 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,  $\sigma_c$ . There are two poles, one at -1+j, one at -1-j, and no zeros. Hence  $n = 2$  and  $m = 0$ . To determine  $\sigma_c$ ,

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

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  $-90^\circ$ . We are looking for zeros for two poles and we have two angles, so we are done. Figure 43 shows both the real axis segment and the asymptotes for  $G_{10}(s)$  and then the completed root locus.

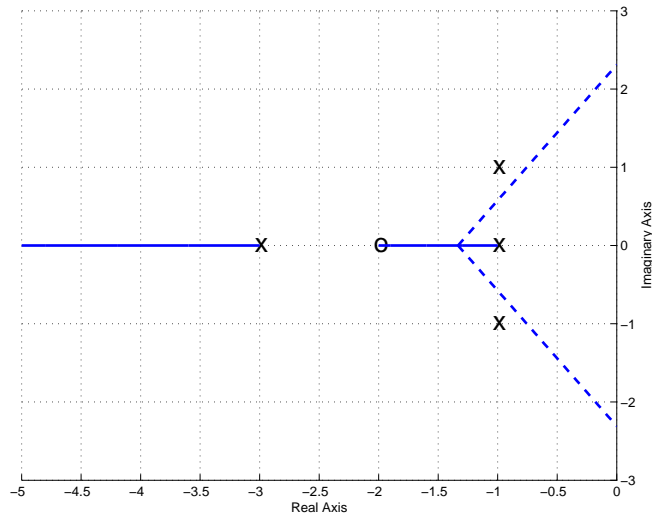
$G_{11}(s)$  First we determine the centroid of the asymptotes,  $\sigma_c$ . There are three poles, one at -1+j, one at -1-j, one at 0, and no zeros. Hence  $n = 3$  and  $m = 0$ . To determine  $\sigma_c$ ,

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

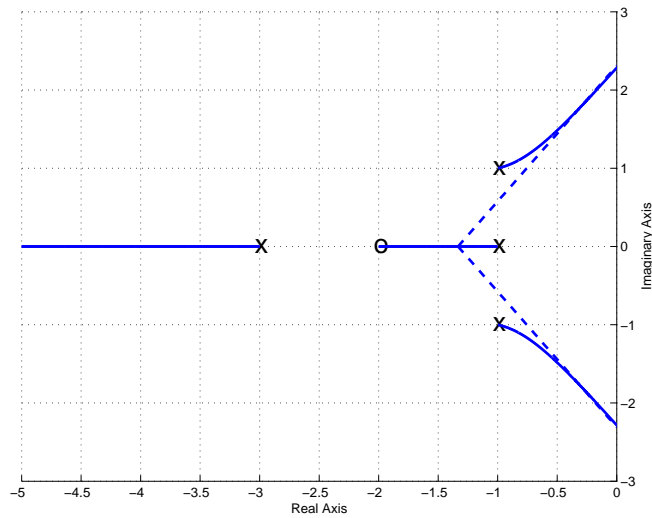
Hence the asymptotes radiate from this point. Next we compute the asymptote angles

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

Since the root locus is symmetric, we know we must also have  $-60^\circ$ . We are looking for zeros for three poles and we have three angles, so we are done. Figure 44 shows both the real axis segment and the asymptotes for  $G_{11}(s)$  and then the completed root locus.

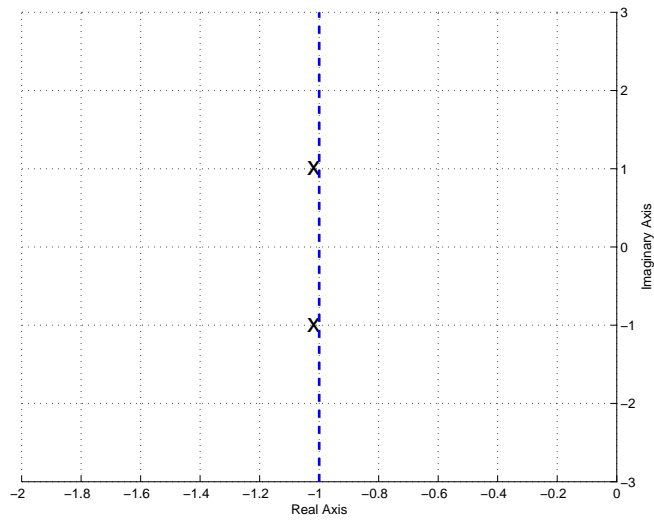


Real axis segment and the asymptotes

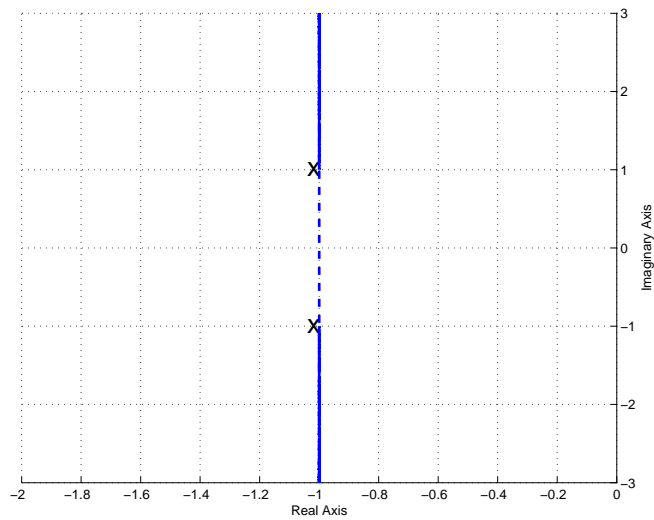


Completed root locus plot (with the asymptotes)

Figure 42: 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  $\pm 60$  degrees.

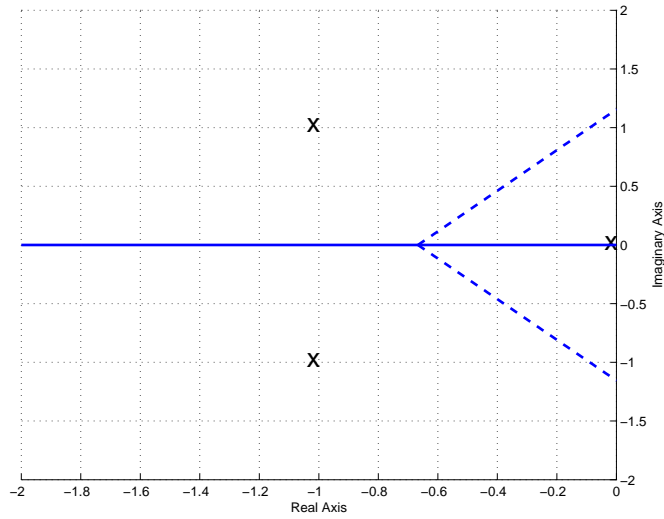


Real axis segment and the asymptotes

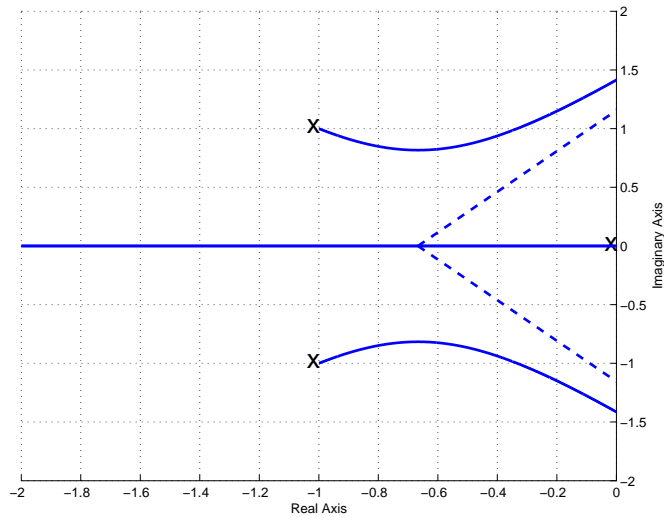


Completed root locus plot (with the asymptotes)

Figure 43: 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  $\pm 90$  degrees.



Real axis segment and the asymptotes



Completed root locus plot (with the asymptotes)

Figure 44: 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  $\pm 60$  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

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

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  $z$ <sup>3</sup>. 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 33 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  $\sigma$  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:

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

---

<sup>3</sup>If  $p$  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.



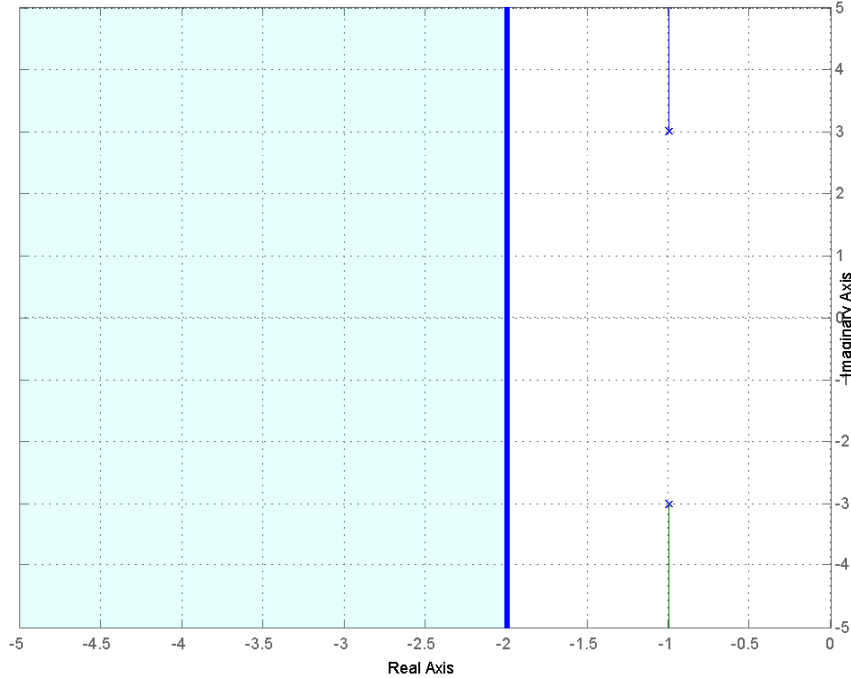


Figure 45: Root locus plot for  $G_p(s) = \frac{2}{s^2+2s+10}$  with the proportional (P) controller. To match the settling time all poles must be to the left of -2.

The root locus for the system with the proportional controller is shown in Figure 45, 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.

---

<sup>4</sup>Here  $k = k_p$ .

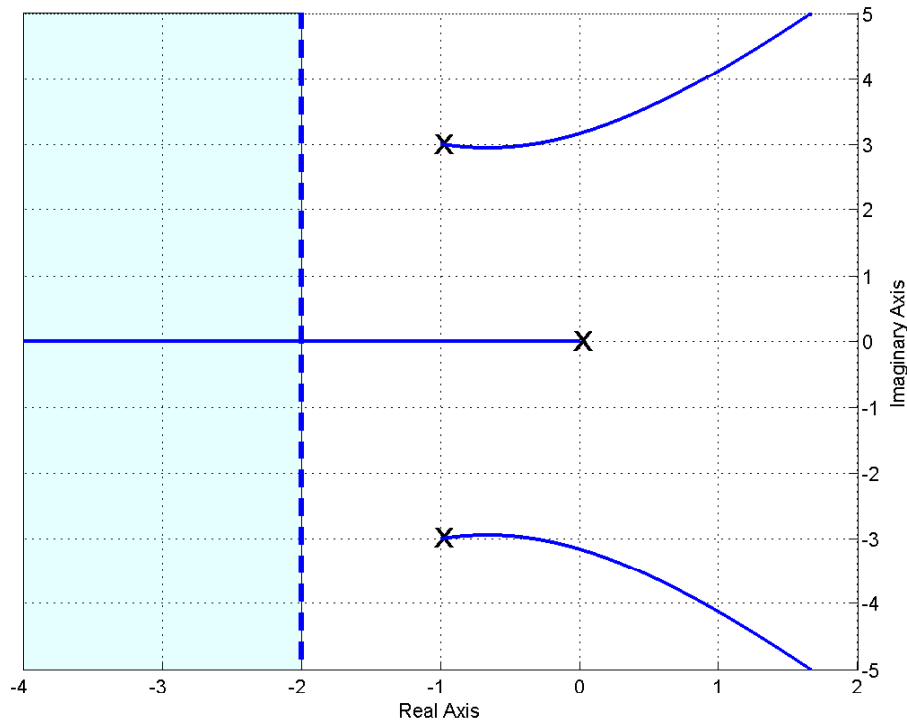


Figure 46: Root locus plot for  $G_p(s) = \frac{2}{s^2+2s+10}$  with the integral (I) controller. To match the settling time all poles must be to the left of -2.

**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 46, 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  $k$ <sup>5</sup>, 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:

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

<sup>5</sup>Here  $k = k_i$ .

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

Here the centroid of the asymptotes depends on where we locate the zero of the system. Figure 47 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 48 shows the root locus for this system with  $z = 1$ ,  $z = 2$ , and  $z = 3$ . Note 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 49 displays

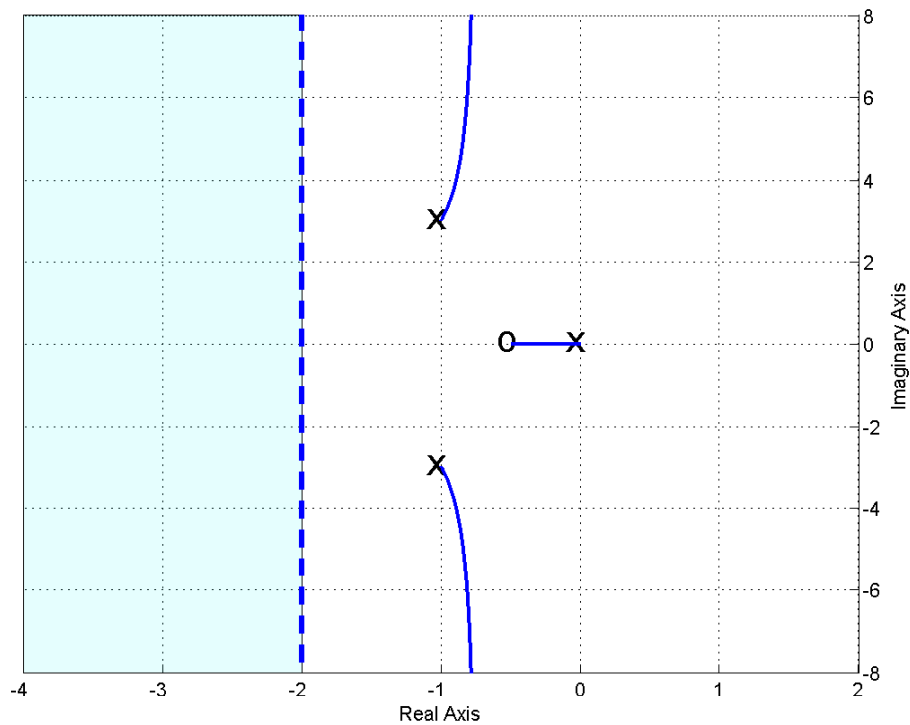
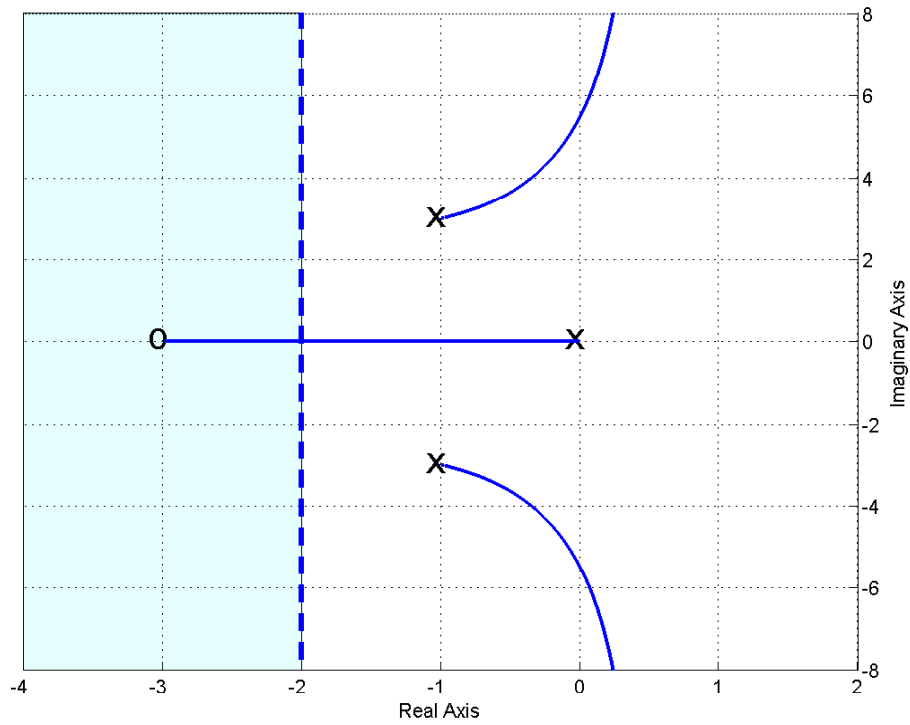


Figure 47: Root locus plots for  $G_p(s) = \frac{2}{s^2+2s+10}$  with the proportional+integral (PI) controllers with the controller zero at -3 (top) and -0.5 (bottom). To match the settling time *all* poles must be to the left of -2.

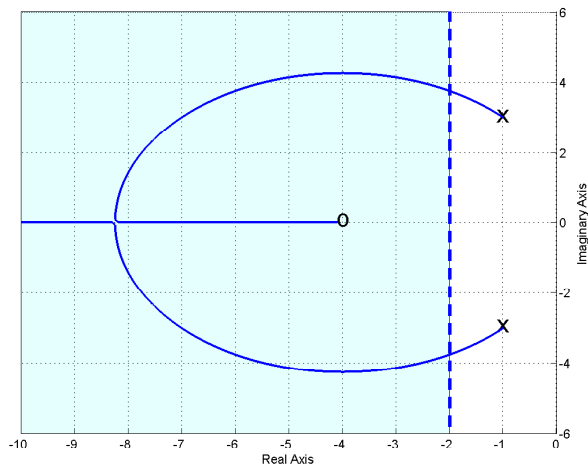
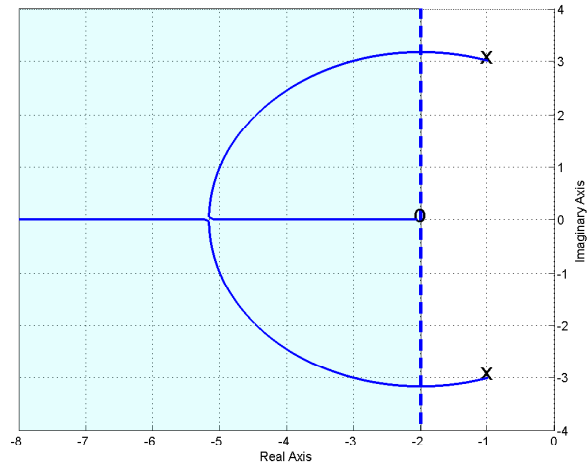
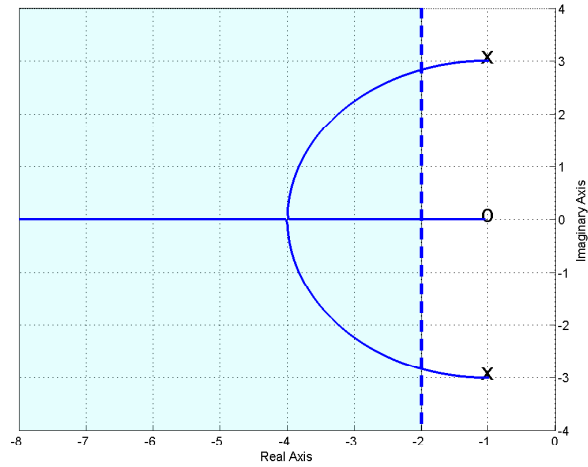


Figure 48: Root locus plots for  $G_p(s) = \frac{2}{s^2+2s+10}$  with the proportional+derivative (PD) controllers with the controller zero at -1 (top), -2 (middle), and -4 (bottom). To match the settling time *all* poles must be to the left of -2.

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 50 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 51 displays the root locus for  $l = 2$  for various combinations of  $z$  and  $p$ . Figure 52 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 53 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.

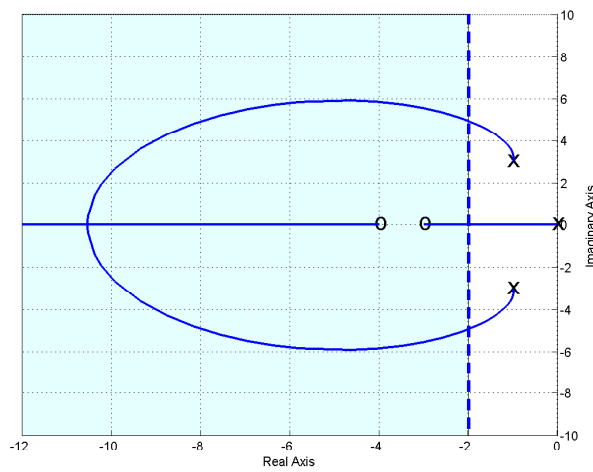
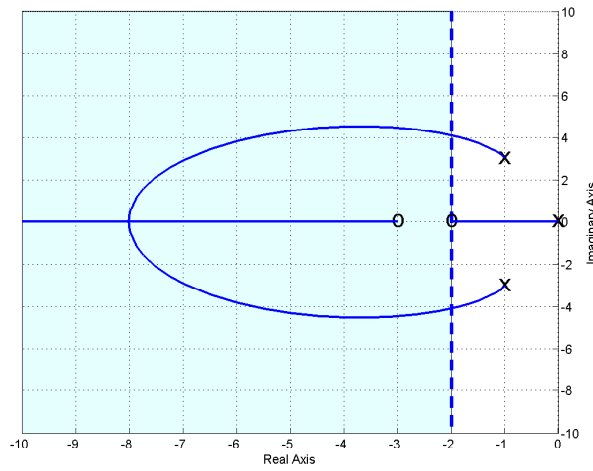
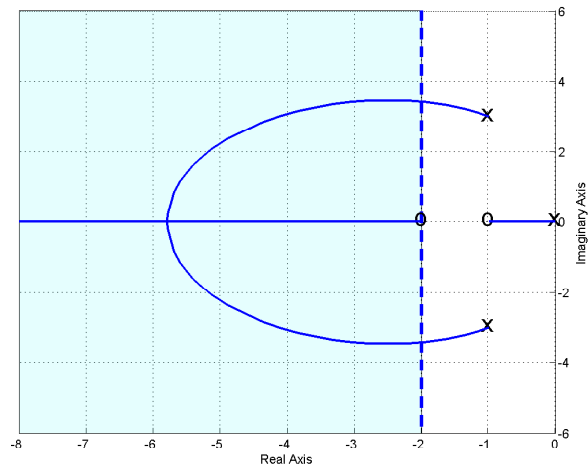


Figure 49: Root locus plots for  $G_p(s) = \frac{2}{s^2+2s+10}$  with proportional+integral+derivative (PID) controllers with real zeros. To match the settling time *all* poles must be to the left of -2.

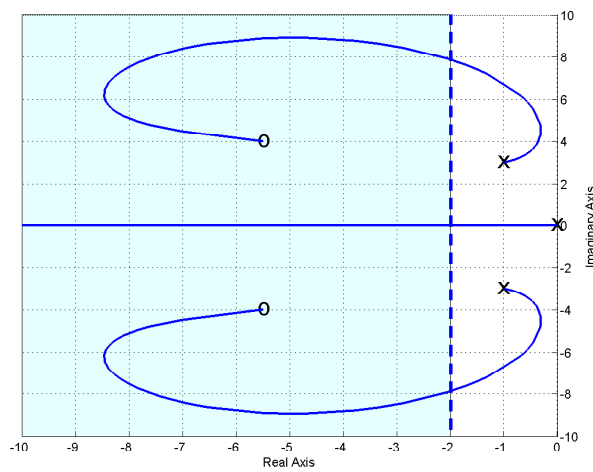
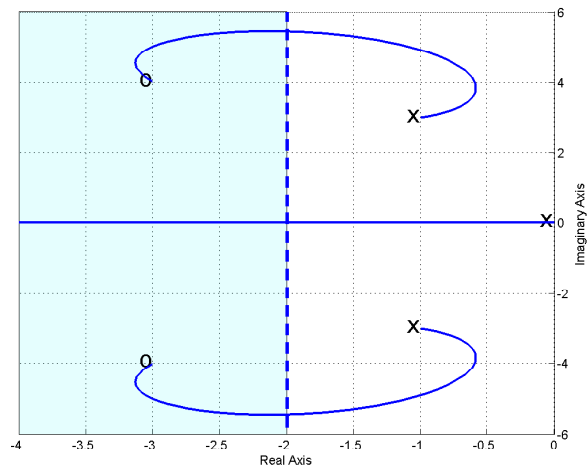
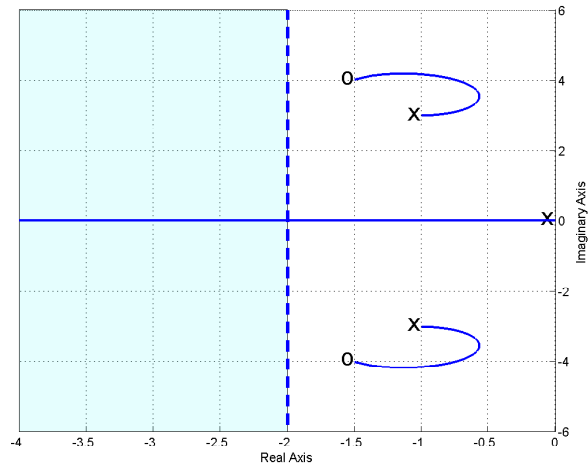


Figure 50: Root locus plots for  $G_p(s) = \frac{2}{s^2+2s+10}$  with proportional+integral+derivative (PID) controllers with complex conjugate zeros. To match the settling time *all* poles must be to the left of -2.



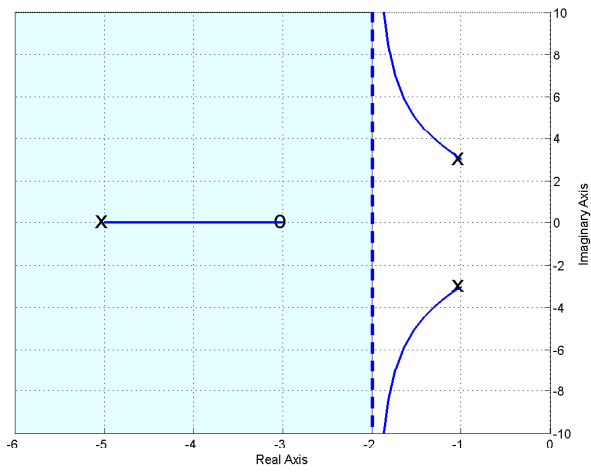
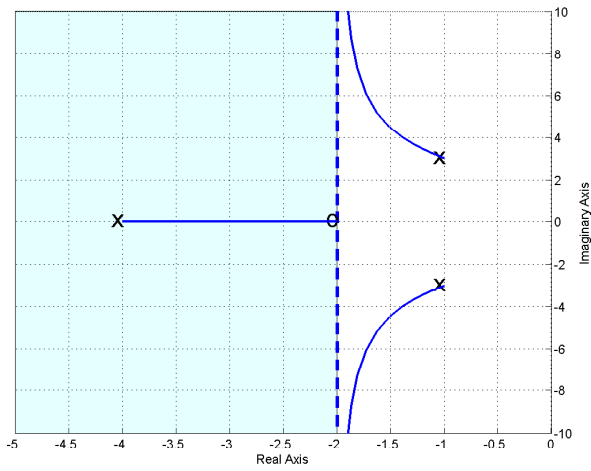
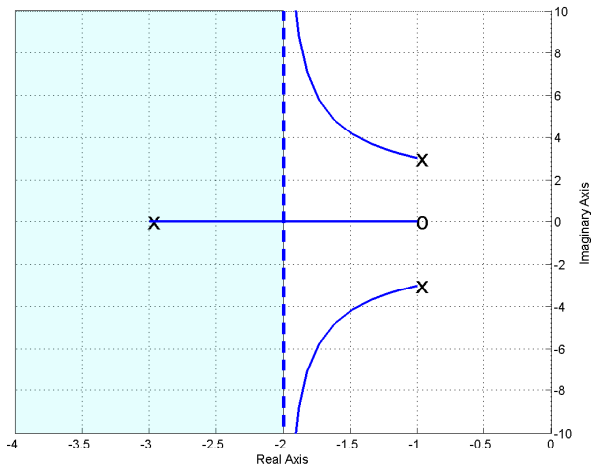


Figure 51: Root locus plots for  $G_p(s) = \frac{2}{s^2+2s+10}$  with lead controllers for  $l = 2$  and asymptote  $\sigma_c = -2$ . To match the settling time *all* poles must be to the left of -2.

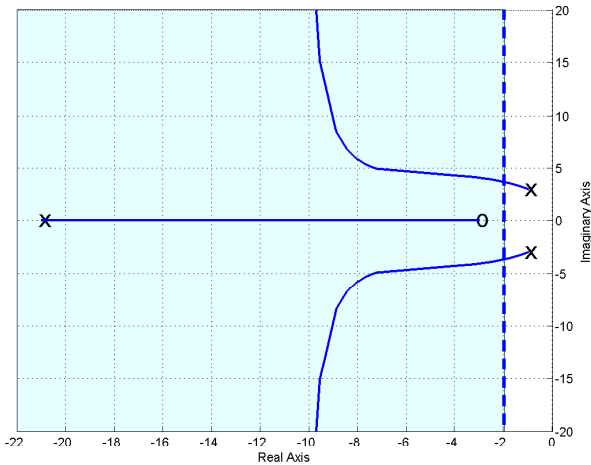
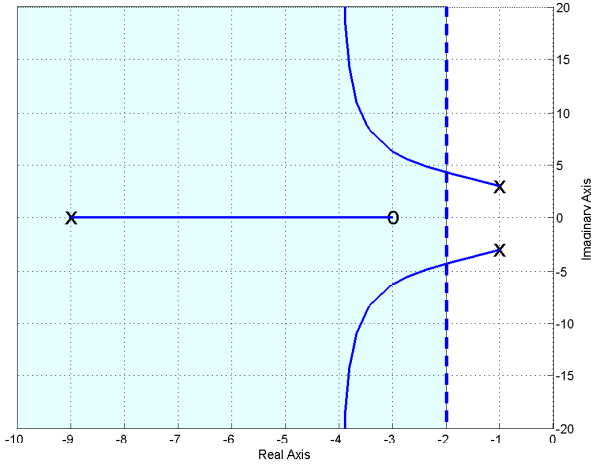
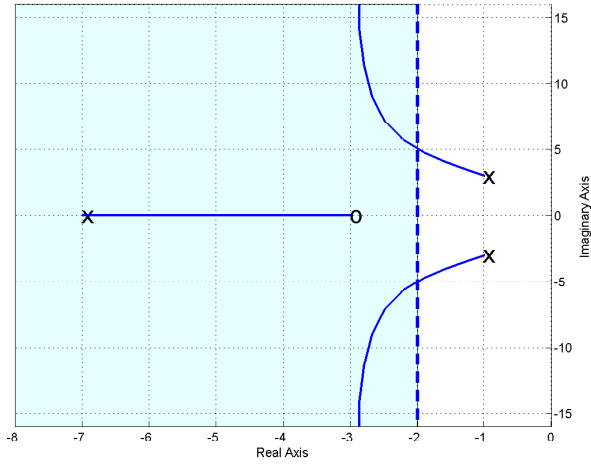


Figure 52: Root locus plots for  $G_p(s) = \frac{2}{s^2+2s+10}$  with lead controllers for  $l = 4$  and asymptote  $\sigma_c = -3$  (top),  $l = 6$  and asymptote  $\sigma_c = -4$  (middle), and  $l = 18$  and asymptote  $\sigma_c = -10$  (bottom). To match the settling time *all* poles must be to the left of  $-2$ .

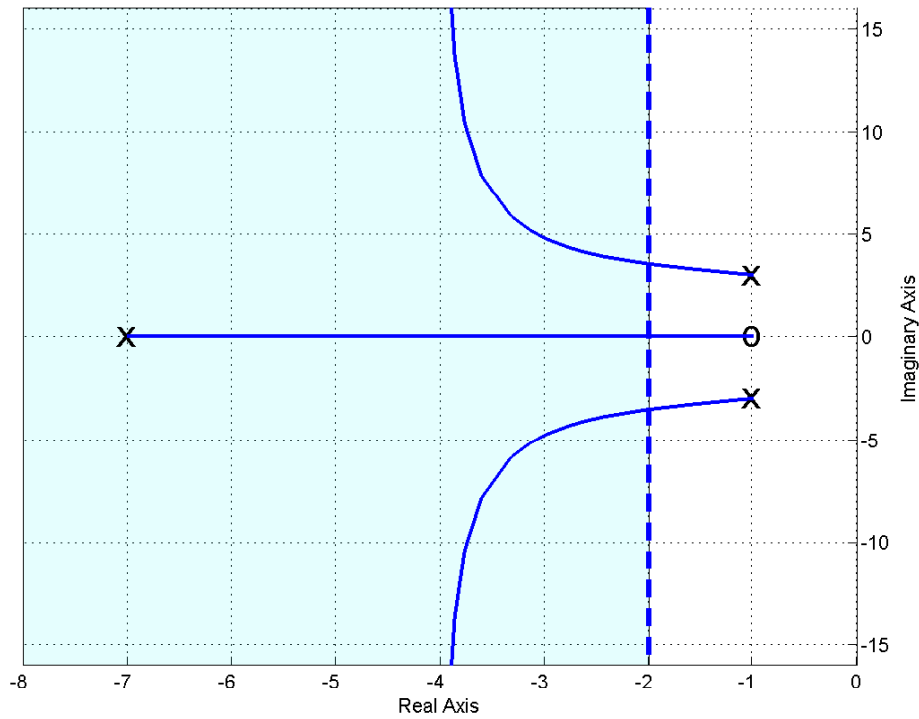


Figure 53: Root locus plot for  $G_p(s) = \frac{2}{s^2+2s+10}$  with the lead controller for  $l = 6$  and asymptote  $\sigma_c = -4$ . As the complex conjugate poles move to the left, the pole on the real axis moves to the right. It may not be possible to find a value (or range of values) for  $k$  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 33 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  $\sigma$  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  $62.8^\circ$  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 54. 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 55. 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

---

<sup>6</sup>Here  $k = k_p$ .

<sup>7</sup>Here  $k = k_i$ .

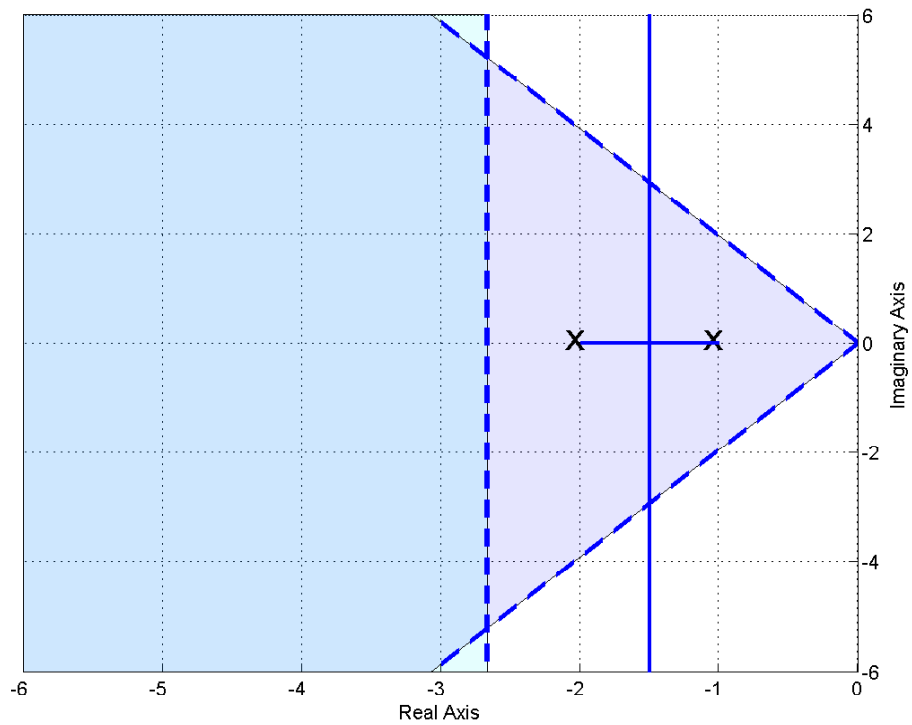


Figure 54: Root locus plot for  $G_p(s) = \frac{2}{s^2+3s+2}$  with the proportional (P) controller. This system will never meet the settling time criteria, but it will meet the percent overshoot criteria as long as  $k$  is not too large.

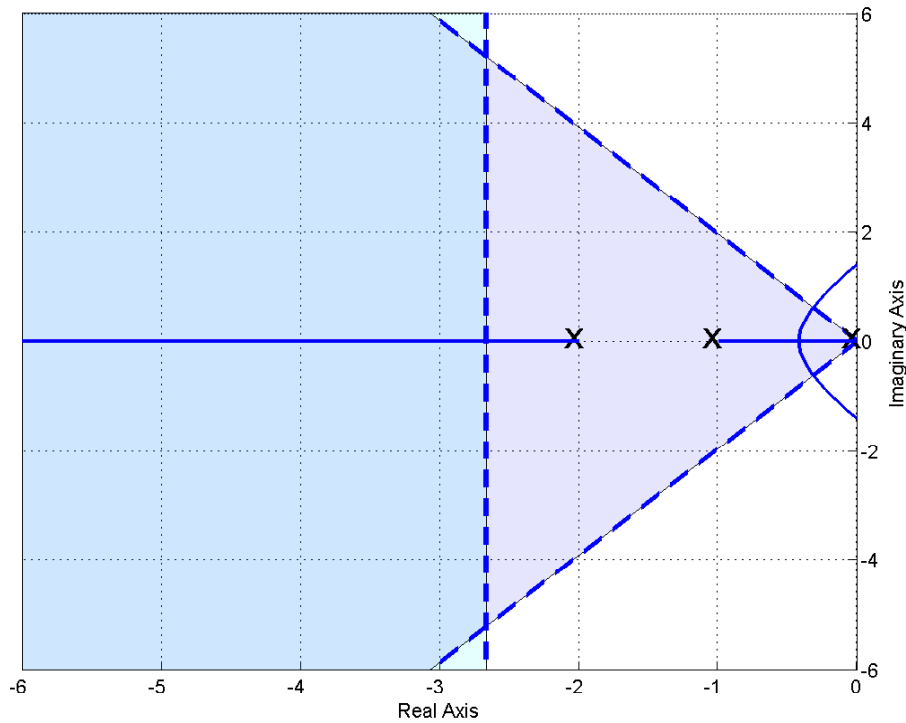


Figure 55: Root locus plot for  $G_p(s) = \frac{2}{s^2+3s+2}$  with the integral (I) controller. This system will never meet the settling time criteria, but it will meet the percent overshoot criteria as long as  $k$  is not too large. This system will become unstable for  $k$  sufficiently large.

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 56. 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 57. 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

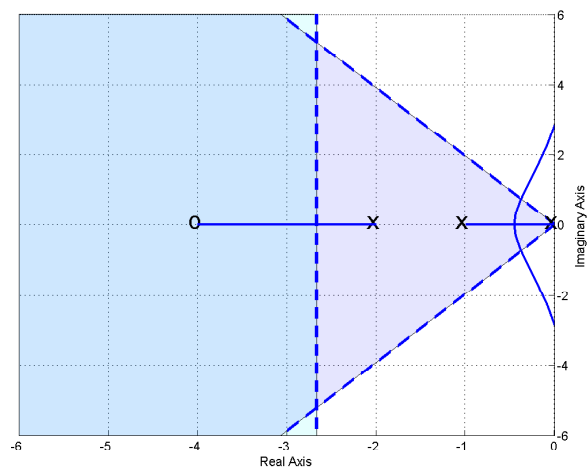
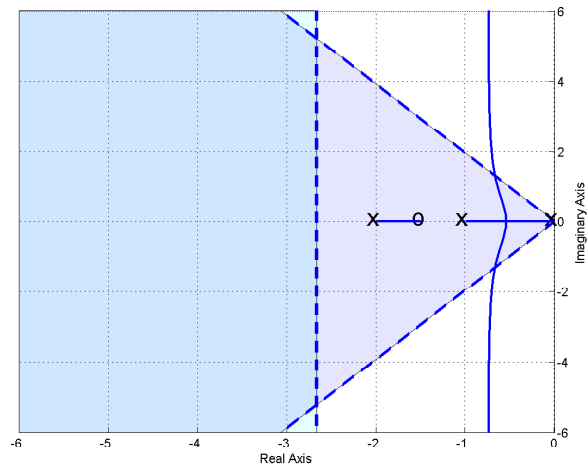
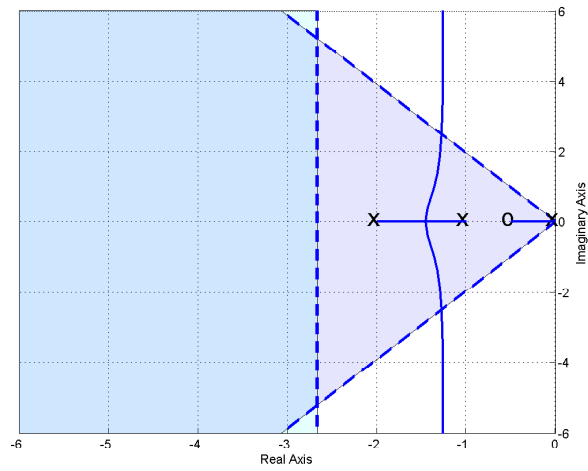


Figure 56: Root locus plot for  $G_p(s) = \frac{2}{s^2+3s+2}$  with three proportional+integral (PI) controllers. None of these systems will ever meet the settling time criteria. All three will meet the percent overshoot criteria if  $k$  is small enough. The third system will become unstable for  $k$  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 58 and 59 displays different examples of PID controllers with real zeros. The first two systems in Figure 59 show poor choices for the locations of the controller zeros. The results in the second panel of Figure 59 shows that for some configurations, it is possible for the system to become unstable if  $k$  is chosen incorrectly.

Complex Conjugate Zeros Figure 60 and 61 displays different examples of PID controllers with complex conjugate zeros. The first two systems in Figure 60 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 61 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 62 and 63. Figure 62 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 63 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  $\frac{kz}{p} \geq 6$ .



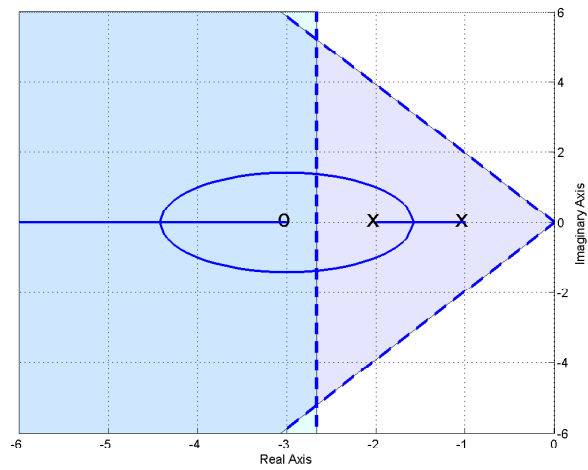
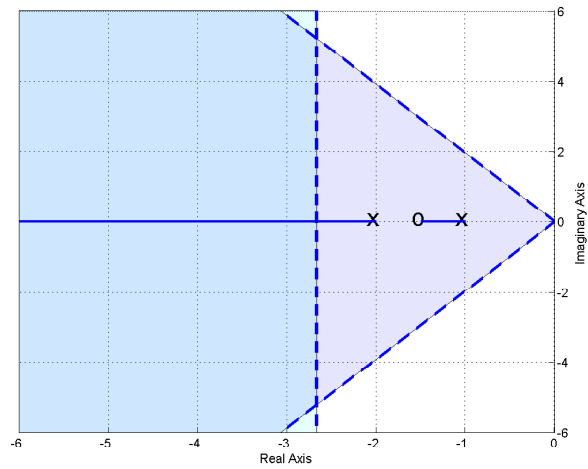
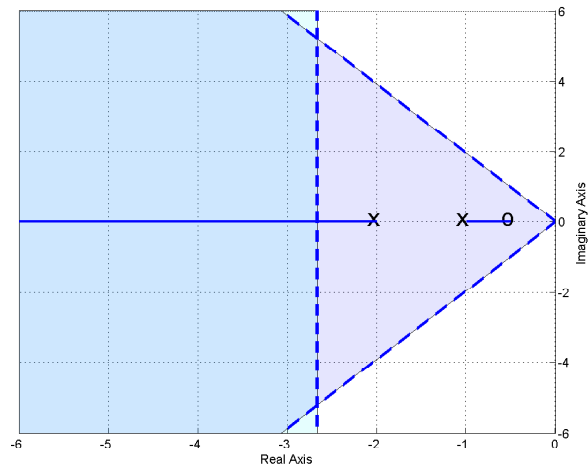


Figure 57: Root locus plots for  $G_p(s) = \frac{2}{s^2+3s+2}$  with three proportional+derivative (PD) controllers. Only the third system meets the settling time criteria (for  $k$  large enough). All three will meet the percent overshoot criteria.

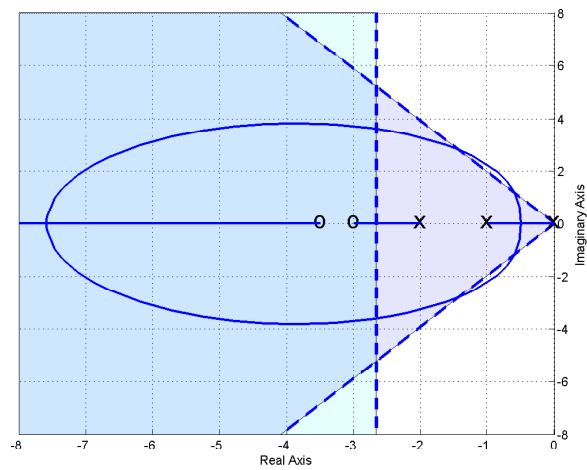
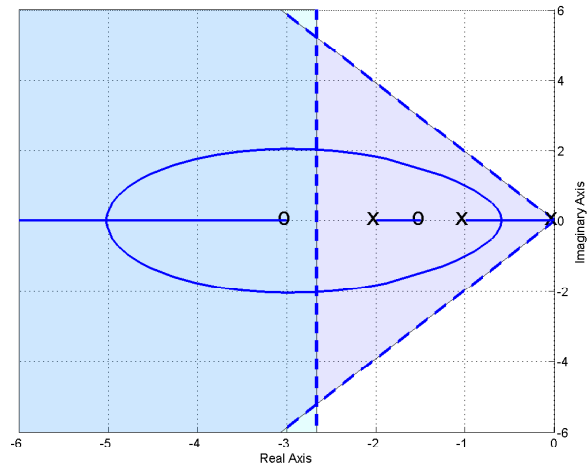
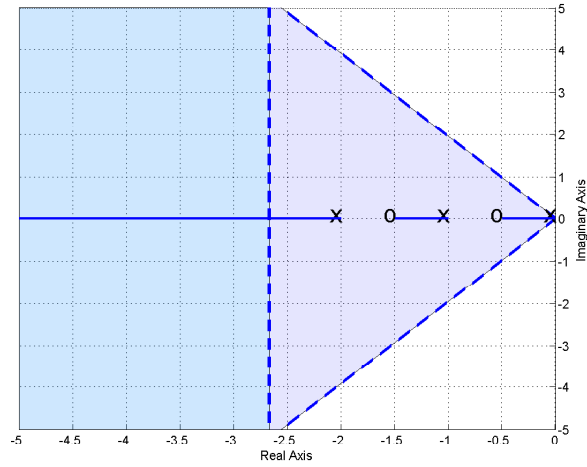


Figure 58: Root locus plots for  $G_p(s) = \frac{2}{s^2+3s+2}$  with proportional+integral+derivative (PID) controllers with real. The systems in the top two panels demonstrate bad choices for the location of the controller zeros.

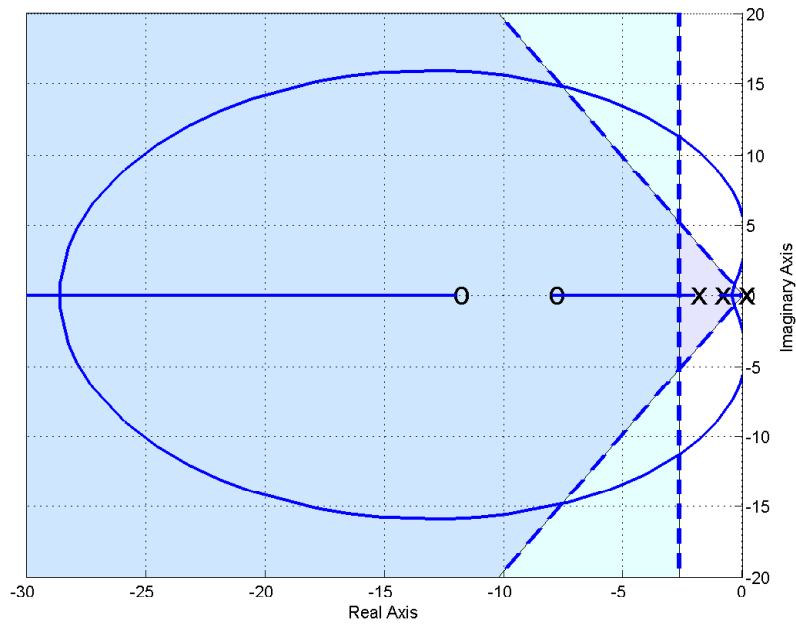
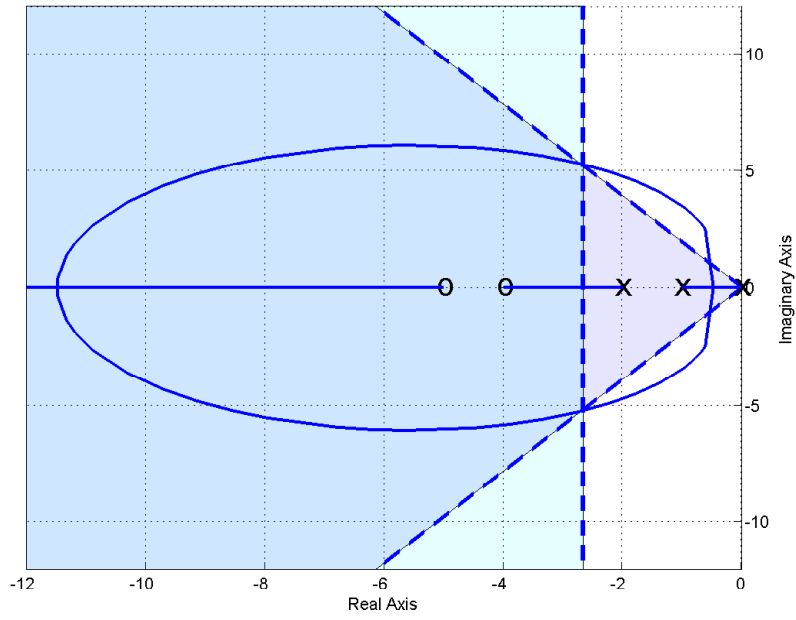


Figure 59: Root locus plots for  $G_p(s) = \frac{2}{s^2+3s+2}$  with proportional+integral+derivative (PID) controllers with real zeros.

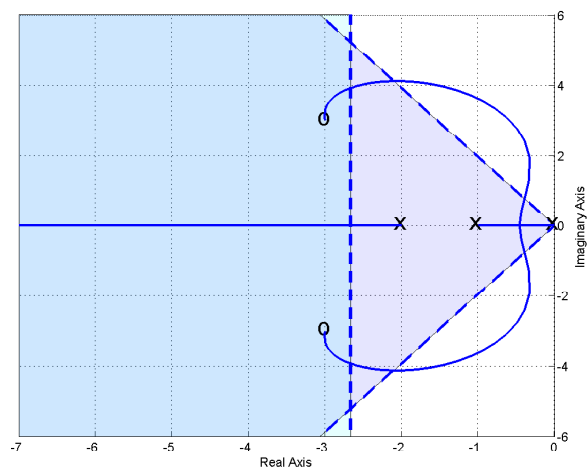
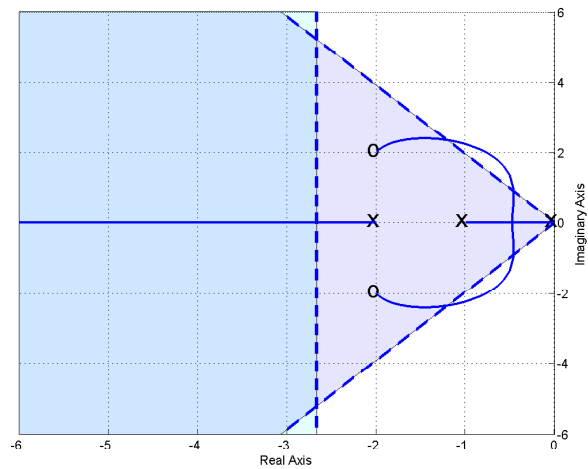
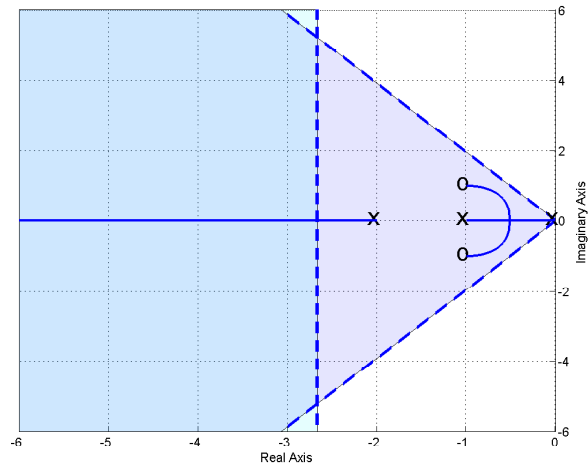


Figure 60: Root locus plots for  $G_p(s) = \frac{2}{s^2+3s+2}$  with proportional+integral+derivative (PID) controllers with complex conjugate zeros. The systems in the top two panels demonstrate bad choices for the location of the controller zeros.

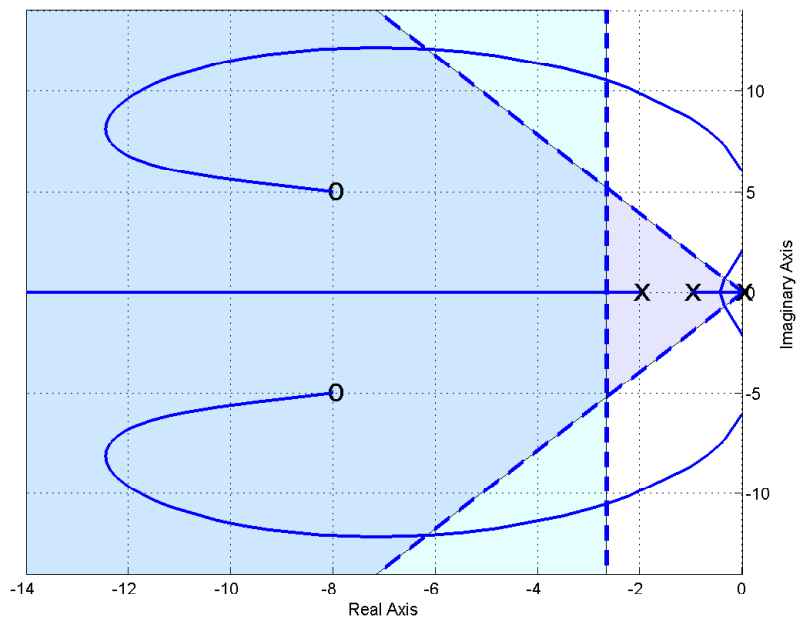
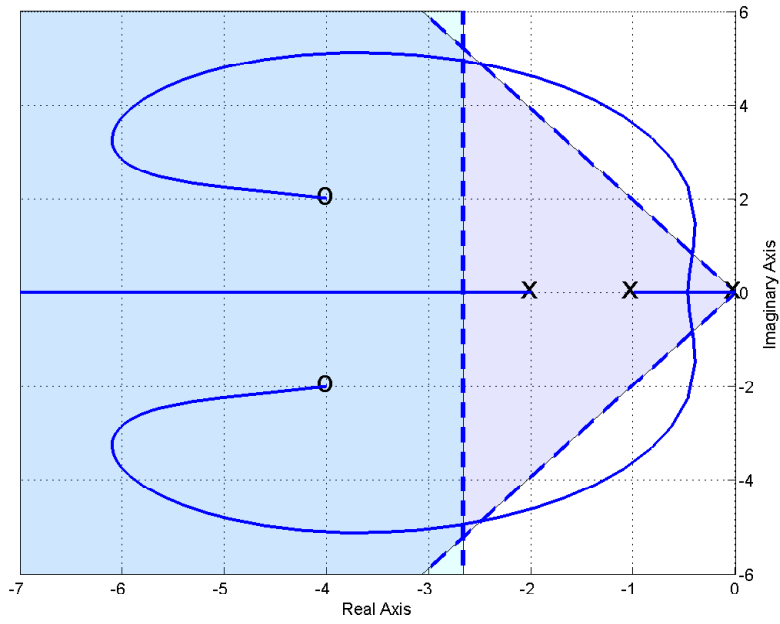


Figure 61: Root locus plots for  $G_p(s) = \frac{2}{s^2 + 3s + 2}$  with proportional+integral+derivative (PID) controllers with complex conjugate zeros.

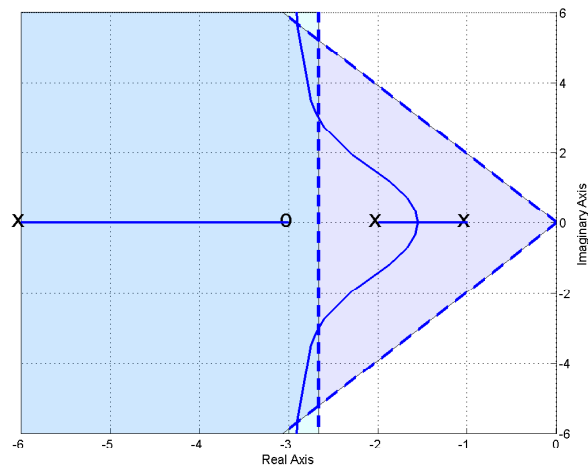
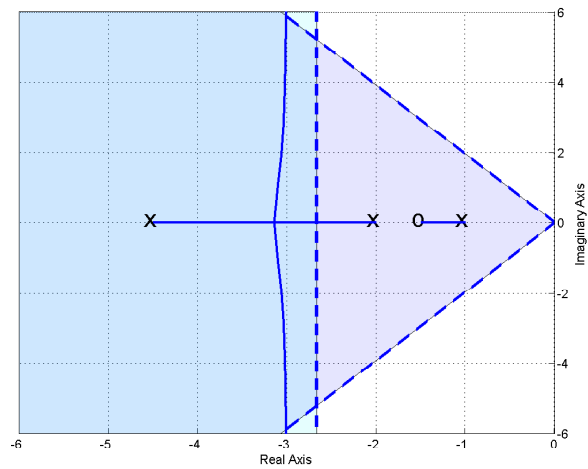
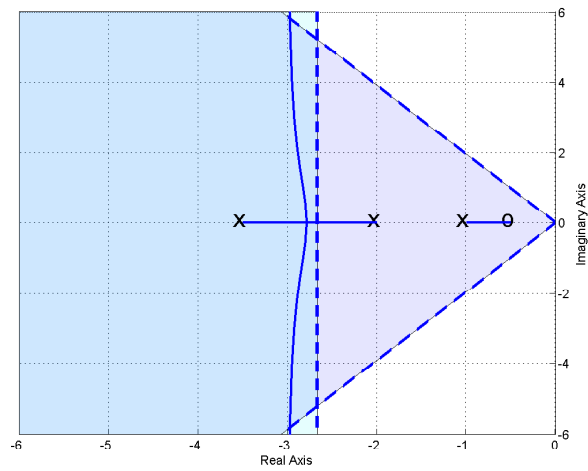


Figure 62: Root locus plots for  $G_p(s) = \frac{2}{s^2+3s+2}$  with lead controllers. Here  $l = 3$  for all three controllers. However, only the third system is capable of meeting both the settling time and percent overshoot criteria. If  $k$  becomes too large in these systems the percent overshoot criteria will not be met.

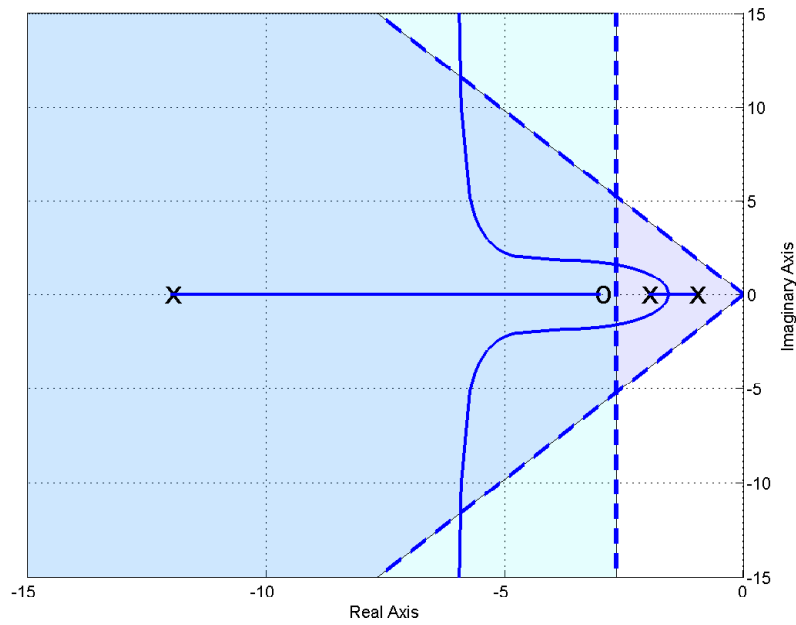
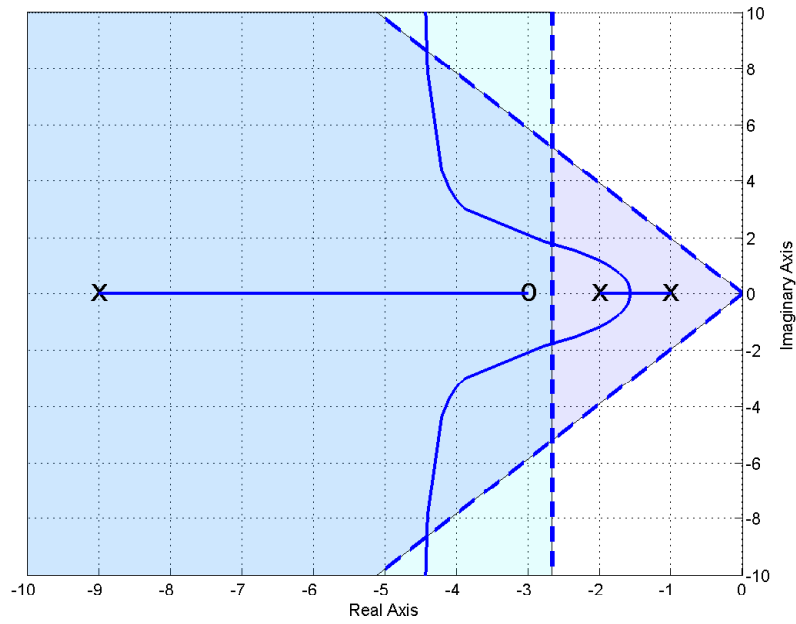


Figure 63: Root locus plots for  $G_p(s) = \frac{2}{s^2+3s+2}$  with lead controllers. Here  $l = 6$  (top panel) and  $l = 9$  (bottom panel). If  $k$  becomes too large in these systems the percent overshoot criteria will not be met.

## 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 64 displays this behavior as  $l$  is varied from 23 to 25.

Figure 65 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.



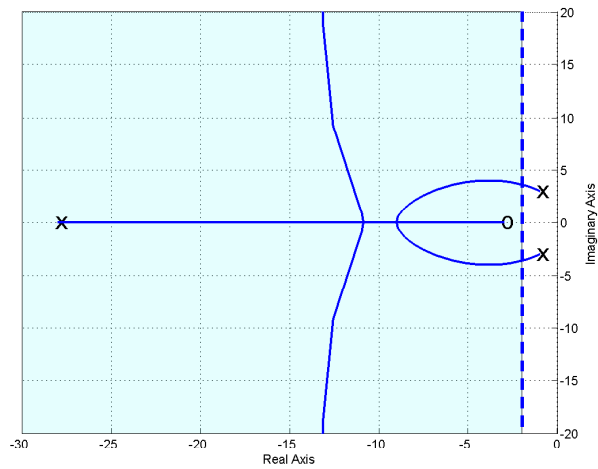
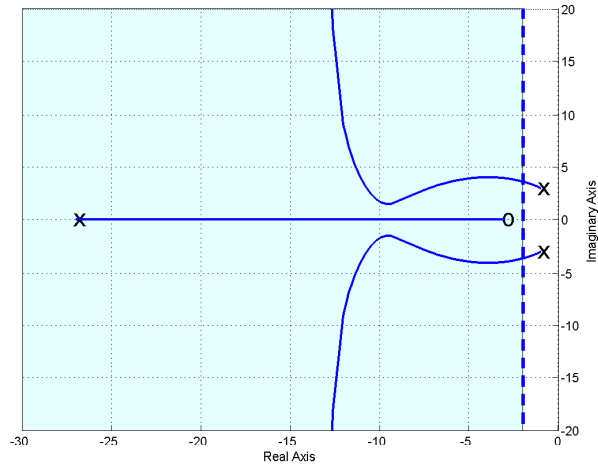
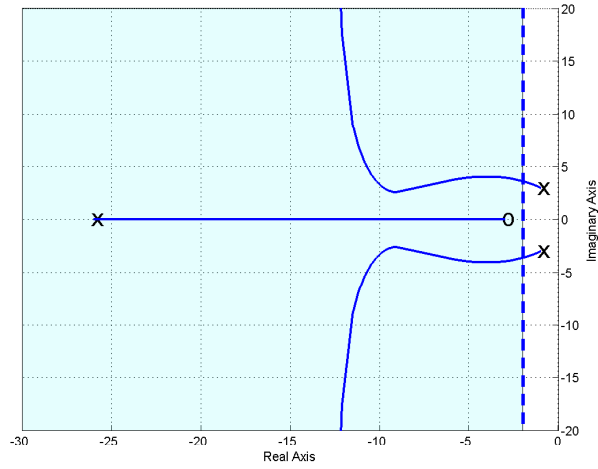


Figure 64: Root locus plot for  $G_p(s) = \frac{2}{s^2+2s+10}$  with the lead controller for  $l = 23, 24,$  and  $25$ . This graph exemplifies some of the possible root locus behavior we have not seen yet. To match the settling time *all* poles must be to the left of  $-2$ .

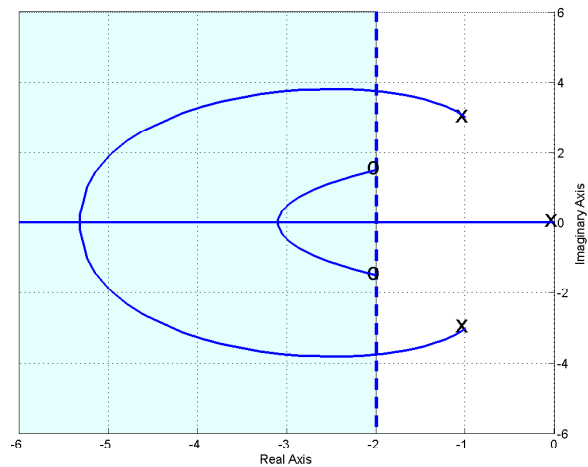
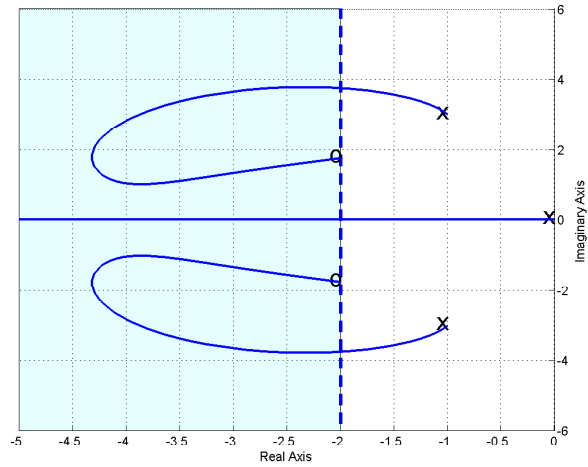
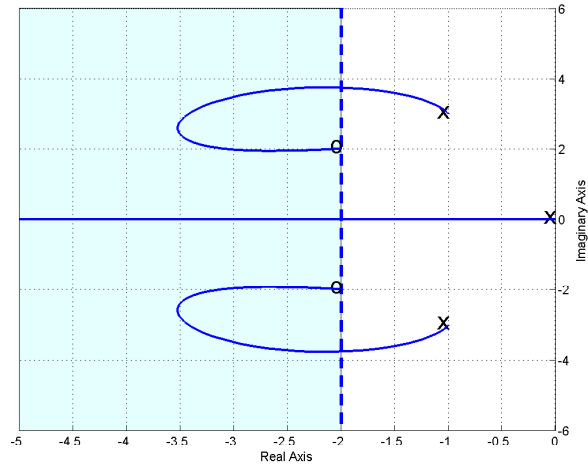
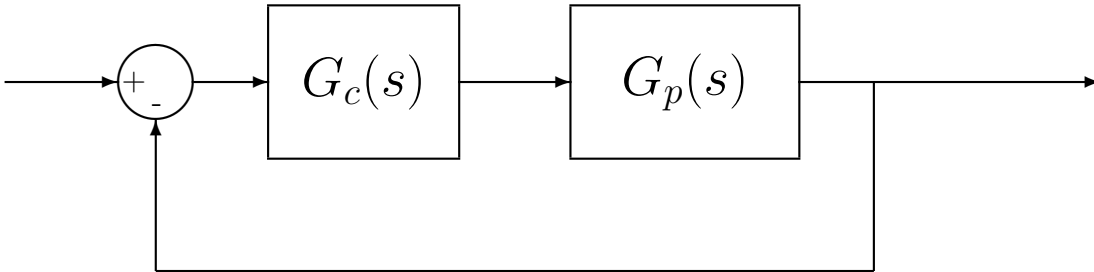


Figure 65: Root locus plot for  $G_p(s) = \frac{2}{s^2+2s+10}$  with the pid controller as the imaginary part of the zeros move closer to the real axis. This graph exemplifies some of the possible root locus behavior we have not seen yet. To match the settling time *all* poles must be to the left of -2.

## 12 Pole Placement By Matching Coefficients: Diophantine Equations

An alternative approach to controller design is to use a controller  $G_c(s)$  to put the closed-loop poles of a system in desired locations. We will start this section with an example, then explain the conditions under which this approach will work, and then do some more examples.

Consider the following unity feedback system



with plant

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

Assume we want to place the closed-loop poles at  $-2 \pm j$  and  $-8$ , so we want the denominator of the closed-loop system to be

$$\begin{aligned} D_0(s) &= (s + 2 + j)(s + 2 - j)(s + 8) \\ &= s^3 + 12s^2 + 37s + 40 \end{aligned}$$

Let's assume the controller has the form

$$G_c(s) = \frac{B_0 + B_1s}{A_0 + A_1s}$$

where  $A_1 \neq 0$  (so the controller is proper). Now, the closed-loop transfer function  $G_0(s)$  is given by

$$\begin{aligned} G_0(s) &= \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \\ &= \frac{\left(\frac{B_0+B_1s}{A_0+A_1s}\right)\left(\frac{s+1}{s^2+s+1}\right)}{1 + \left(\frac{B_0+B_1s}{A_0+A_1s}\right)\left(\frac{s+1}{s^2+s+1}\right)} \\ &= \frac{(B_0 + B_1s)(s + 1)}{(A_0 + A_1s)(s^2 + s + 1) + (B_0 + B_1s)(s + 1)} \end{aligned}$$

Since we know where we want the closed-loop poles, we equate denominators:

$$D_0(s) = s^3 + 12s^2 + 37s + 40 = (A_0 + A_1s)(s^2 + s + 1) + (B_0 + B_1s)(s + 1)$$

and then equate powers of  $s$ :

$$\begin{aligned} s^3 : \quad & 1 = A_1 \\ s^2 : \quad & 12 = A_1 + A_0 + B_1 \\ s^1 : \quad & 37 = A_0 + A_1 + B_0 + B_1 \\ s^0 : \quad & 40 = A_0 + B_0 \end{aligned}$$

We then have the system of equations

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_0 \\ B_0 \\ A_1 \\ B_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 12 \\ 37 \\ 40 \end{bmatrix}$$

The solution to this system of equations is  $A_0 = 15$ ,  $B_0 = 25$ ,  $A_1 = 1$ , and  $B_1 = -4$ . The controller is then

$$G_c(s) = \frac{25 - 4s}{15 + s}$$

and the closed-loop transfer function is

$$\begin{aligned} G_0(s) &= \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \\ &= \frac{\left(\frac{25-4s}{15+s}\right)\left(\frac{s+1}{s^2+s+1}\right)}{1 + \left(\frac{25-4s}{15+s}\right)\left(\frac{s+1}{s^2+s+1}\right)} \\ &= \frac{(25 - 4s)(s + 1)}{(15 + s)(s^2 + s + 1) + (25 - 4s)(s + 1)} \\ &= \frac{(25 - 4s)(s + 1)}{s^3 + 12s^2 + 39s + 40} \end{aligned}$$

We have achieved the desired closed-loop poles. However, we have introduced a new zero into the system at  $\frac{25}{4}$ . As you will see, this is the major drawback to this kind of controller. While we can force the closed-loop poles to be anything we want, we will be introducing zeros into the system. If these zeros are acceptable, then we are done. If they are not acceptable, then we need to try and do something. One option is to try changing where we want the closed-loop poles to be. A second option is to try and eliminate any introduced zeros with a dynamic prefilter, however we cannot cancel zeros in the right half plane, since we would need our prefilter to have poles in the right half plane. An alternative is to try a different type of controller. For this example, the steady state error for a unit step input is  $e_p = 1 - \frac{5}{8} = \frac{3}{8}$ . One method of obtaining a steady state error of zero for a unit step input is with a prefilter (with gain  $\frac{8}{5}$ ). A better way is to design the controller so that the resulting system is a type 1 system. We will show how to do the latter in a subsequent section.

## 12.1 Theoretical Background

The results we need to know are stated in the following theorem. There are two parts to the theorem. The first part states the results for a strictly proper plant, while the second part states

the results for a plant where the numerator and denominator polynomials have the same degree. The important information from the theorem is knowing the minimum order of the required controller  $m$  and the order of the closed-loop transfer function  $n + m$ .

**Theorem** *Strictly Proper Plant* Assume we have a strictly proper  $n^{\text{th}}$  order plant transfer function,  $G_p(s) = N(s)/D(s)$ . Since  $G_p(s)$  is strictly proper we have the degree of  $N(s) <$  the degree of  $D(s)$ . Since  $G_p(s)$  is  $n^{\text{th}}$  order the degree of  $D(s) = n$ . Assume also that  $N(s)$  and  $D(s)$  have no common factors. Then for any polynomial  $D_0(s)$  of degree  $n + m$  a proper controller  $G_c(s) = B(s)/A(s)$  of degree  $m$  exists so that the characteristic equation of the resulting closed-loop system is equal to  $D_0(s)$ . If  $m = n - 1$ , the controller is unique. If  $m \geq n$ , the controller is not unique and some of the coefficients can be used to achieve other design objectives.

**Theorem** *Special case: degree  $N(s) =$  degree  $D(s)$ .* Assume we have a proper  $n^{\text{th}}$  order plant transfer function,  $G_p(s) = N(s)/D(s)$ , where the degree of  $D(s) =$  degree  $N(s) = n$ . Assume also that  $N(s)$  and  $D(s)$  have no common factors. Then for any polynomial  $D_0(s)$  of degree  $n + m$  a proper controller  $G_c(s) = B(s)/A(s)$  of degree  $m$  exists so that the characteristic equation of the resulting closed-loop system is equal to  $D_0(s)$ . If  $m = n$ , and the controller is chosen to be strictly proper, the controller is unique. If  $m \geq n + 1$ , the controller is not unique and some of the coefficients can be used to achieve other design objectives.

How do we do this? For plant  $G_p(s) = N(s)/D(s)$ , controller  $G_c(s) = B(s)/A(s)$ , and desired characteristic equation  $D_0(s)$  we will have to solve the equation

$$A(s)D(s) + B(s)N(s) = D_0(s)$$

This is called the *Diophantine* equation. We solve this equation by equating powers of  $s$ , setting up a system of equations, and then solving. The closed-loop transfer function will be

$$G_0(s) = \frac{B(s)N(s)}{D_0(s)}$$

where  $B(s)$  contains the zeros we have added to the system.

**Example 1.** Assume we are trying to control the plant

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

Since  $n = 2$  we need the order of the controller  $m \geq n - 1$  or  $m \geq 1$ . We'll choose  $m = 1$ . Hence we will be looking at a controller of the form

$$G_c(s) = \frac{B_0 + B_1s}{A_0 + A_1s}$$

where  $A_1 \neq 0$  (we need a proper controller transfer function). Next, we need to know the desired characteristic equation,  $D_0(s)$ . We need to have  $n + m = 3$  poles. Let's assume we want the closed-loop poles to be at  $-10 \pm 5j$  and  $-20$ . Then

$$\begin{aligned} D_0(s) &= (s + 10 + 5j)(s + 10 - 5j)(s + 20) \\ &= s^3 + 40s^2 + 525s + 2500 \end{aligned}$$

Now we need to solve the Diophantine equations

$$\begin{aligned} A(s)D(s) + B(s)N(s) &= D_0(s) \\ (A_0 + A_1s)(s^2 + 1) + (B_0 + B_1s)(10) &= s^3 + 40s^2 + 525s + 2500 \end{aligned}$$

Now we equate powers of  $s$

$$\begin{aligned} s^3 : \quad 1 &= A_1 \\ s^2 : \quad 40 &= A_0 \\ s^1 : \quad 525 &= A_1 + 10B_1 \\ s^0 : \quad 2500 &= A_0 + 10B_0 \end{aligned}$$

In this case we can solve directly to get  $A_0 = 40$ ,  $B_0 = 246$ ,  $A_1 = 1$  and  $B_1 = 52.4$ . Hence our controller is

$$G_c(s) = \frac{246 + 52.4s}{40 + s}$$

and the closed-loop transfer function is

$$G_0(s) = \frac{(246 + 52.4s)10}{s^3 + 40s^2 + 525s + 2500}$$

We have introduced a zero at -4.69. The steady state error for a step input is  $e_{ss} = 1 - G_0(0) = 0.016$ .

**Example 2.** Assume we are trying to control the plant

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

Since  $n = 3$  we need the order of the controller  $m \geq n - 1$  or  $m \geq 2$ . We'll choose  $m = 2$ . Hence we will be looking at a controller of the form

$$G_c(s) = \frac{B_0 + B_1s + B_2s^2}{A_0 + A_1s + A_2s^2}$$

where  $A_2 \neq 0$  (we need a proper controller transfer function). Next, we need to know the desired characteristic equation,  $D_0(s)$ . We need to have  $n + m = 5$  poles. Let's assume we want all the closed-loop poles to be at -5. Then

$$\begin{aligned} D_0(s) &= (s + 5)^5 \\ &= s^5 + 25s^4 + 250s^3 + 1250s^2 + 3125s + 3125 \end{aligned}$$

Now we need to solve the Diophantine equations

$$\begin{aligned} A(s)D(s) + B(s)N(s) &= D_0(s) \\ (A_0 + A_1s + A_2s^2)(s^3 + 4s^2 + 3s + 6) + (B_0 + B_1s + B_2s^2)(s + 1) &= D_0(s) \end{aligned}$$

Now we equate powers of  $s$

$$\begin{aligned}
 s^5 : \quad & 1 = A_2 \\
 s^4 : \quad & 25 = A_1 + 4A_2 \\
 s^3 : \quad & 250 = A_0 + 4A_1 + 3A_2 + B_2 \\
 s^2 : \quad & 1250 = 4A_0 + 3A_1 + 6A_2 + B_1 + B_2 \\
 s^1 : \quad & 3125 = 3A_0 + 6A_1 + B_0 + B_1 \\
 s^0 : \quad & 3125 = 6A_0 + B_0
 \end{aligned}$$

Hence we have to solve the system of equations

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & 0 & 0 \\ 1 & 4 & 3 & 0 & 0 & 1 \\ 4 & 3 & 6 & 0 & 1 & 1 \\ 3 & 6 & 0 & 1 & 1 & 0 \\ 6 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ B_0 \\ B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 25 \\ 250 \\ 1250 \\ 3125 \\ 3125 \end{bmatrix}$$

Solving this system we get  $A_0 = 190.6$ ,  $B_0 = 1981.0$ ,  $A_1 = 21.0$ ,  $B_1 = 446.0$ ,  $A_2 = 1$ , and  $B_2 = -27.7$ . Hence our controller is

$$G_c(s) = \frac{1981.0 + 446.0s - 27.7s^2}{190.6 + 21.0s + s^2}$$

and the closed-loop transfer function is

$$G_0(s) = \frac{(1981.0 + 446.0s - 27.7s^2)(s + 1)}{s^5 + 25s^4 + 250s^3 + 1250s^2 + 3125s + 3125}$$

We have introduced two zeros at 19.7 and -3.62. The steady state error for a unit step input is  $e_{ss} = 1 - G_0(0) = 0.366$ , which is quite poor. In this case, we would probably use a prefilter with amplitude  $\frac{1}{G_0(0)} = 1.581$ .

## 12.2 Pole Placement with Robust Tracking

We would like to avoid the prefilter approach to achieving a steady state error of zero, since the system may change over time. If we can make the plant-controller combination a type 1 system, then the closed-loop system will have a steady state error of zero for a step input even if the plant changes over time (or our model is not exact). To do this, we will insert an integrator in the controller. To do this, we increase the degree of the controller we need by 1, and use the extra parameter to create a type 1 system. To create the type one system, we will increase the order of the controller by one and set  $A_0 = 0$ .

**Example 3.** Assume we are trying to control the plant

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

Since  $n = 2$  we need the order of the controller  $m \geq n - 1$  or  $m \geq 1$ . We'll choose  $m = 1$ . Hence we will be looking at a controller of the form

$$G_c(s) = \frac{B_0 + B_1s}{A_0 + A_1s}$$

where  $A_1 \neq 0$  (we need a proper controller transfer function). Next, we need to know the desired characteristic equation,  $D_0(s)$ . We need to have  $n + m = 3$  poles. Let's assume we want the closed-loop poles to be at  $-5 \pm j$  and  $-20$ . Then

$$\begin{aligned} D_0(s) &= (s + 5 + j)(s + 5 - j)(s + 20) \\ &= s^3 + 30s^2 + 226s + 520 \end{aligned}$$

Now we need to solve the Diophantine equations

$$\begin{aligned} A(s)D(s) + B(s)N(s) &= D_0(s) \\ (A_0 + A_1s)(s^2 + 3s + 2) + (B_0 + B_1s)(3) &= s^3 + 30s^2 + 226s + 520 \end{aligned}$$

Now we equate powers of  $s$

$$\begin{aligned} s^3 : \quad 1 &= A_1 \\ s^2 : \quad 30 &= A_0 + 3A_1 \\ s^1 : \quad 226 &= 3A_0 + 2A_1 + 3B_1 \\ s^0 : \quad 520 &= 2A_0 + 3B_0 \end{aligned}$$

In this case, we can solve directly to get  $A_0 = 27.00$ ,  $B_0 = 155.33$ ,  $A_1 = 1.00$  and  $B_1 = 47.67$ . Hence, our controller is

$$G_c(s) = \frac{155.33 + 47.67s}{27.00 + s}$$

and the closed-loop transfer function is

$$G_0(s) = \frac{3(155.33 + 47.67s)}{s^3 + 30s^2 + 226s + 520}$$

We have introduced a zero at  $-3.26$ . The steady state error for a unit step input is  $e_{ss} = 1 - G_0(0) = 0.104$ .

Now let's assume we want a steady state error of zero, but don't want to use a prefilter. To do this, we increase the order of the controller by one, (so  $m = 2$ ) and, to be sure we have a type one system, we set  $A_0 = 0$ . Hence, we assume a controller of the form

$$G_c(s) = \frac{B_0 + B_1s + B_2s^2}{A_1s + A_2s^2}$$

where  $A_2 \neq 0$ . We now need a characteristic polynomial with  $n + m = 4$  roots, so there are four closed-loop poles to assign. Let's assume we want to keep the poles we have, and put the new pole at  $-30$ . Hence, the closed-loop poles are at  $-5 \pm j$ ,  $-20$ , and  $-30$ . Then

$$\begin{aligned} D_0(s) &= (s + 5 + j)(s + 5 - j)(s + 20)(s + 30) \\ &= s^4 + 60s^3 + 1126s^2 + 7300s + 15600 \end{aligned}$$



Now we need to solve the Diophantine equations

$$A(s)D(s) + B(s)N(s) = D_0(s)$$

$$(A_1s + A_2s^2)(s^2 + 3s + 2) + (B_0 + B_1s + B_2s^2)(3) = s^4 + 60s^3 + 1126s^2 + 7300s + 15600$$

Now we equate powers of  $s$

$$\begin{aligned} s^4 : \quad & 1 = A_2 \\ s^3 : \quad & 60 = A_1 + 3A_2 \\ s^2 : \quad & 1126 = 3A_1 + 2A_2 + 3B_2 \\ s^1 : \quad & 7300 = 2A_1 + 3B_1 \\ s^0 : \quad & 15600 = 3B_0 \end{aligned}$$

We can easily solve these equations to give  $B_0 = 5200$ ,  $A_1 = 57.0$ ,  $B_1 = 2395.3$ ,  $A_2 = 1$ , and  $B_2 = 317.6$ .

$$G_c(s) = \frac{5200 + 2395.3s + 317.7s^2}{57s + s^2}$$

and the closed-loop transfer function is

$$G_0(s) = \frac{3(5200 + 2395.3s + 317.7s^2)}{s^4 + 60s^3 + 1126s^2 + 7300s + 15600}$$

We have introduced zeros at  $-3.7 \pm 1.49j$ . Since we have a type one system, the steady state error for a step input is zero.

### 12.3 Summary

We have shown that, by utilizing the Diophantine equations, we can place the closed-loop poles wherever we want. In addition, by choosing the order of the controller larger than is necessary to place the poles, we can also force the system to be a type 1 system (or even a type two system). However, in utilizing this method, we introduce zeros into the system. The only way to determine if the added zeros are detrimental to acceptable transient behavior is to simulate the system. By appropriate choice of desired closed-loop poles, we can sometimes change the locations of these zeros so that the system response is acceptable.

## 13 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.

### 13.1 Sensitivity to Parameter Variations

The system's sensitivity to changes in a parameter  $\alpha$  is defined as the ratio of the percentage change in the system transfer function  $G_0(s)$  to the percentage change in the parameter  $\alpha$  to its nominal value  $\alpha_0$ . Note that  $\alpha$  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  $\alpha$  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

$$\begin{aligned} S_{\alpha}^{G_0}(s) &= \left. \frac{\alpha}{G_0(s)} \frac{\partial G_0(s)}{\partial \alpha} \right|_{\alpha_0} \\ &= \left. \frac{\alpha}{G_0(s)} G_0(s) \left( \frac{1}{N_0(s)} \frac{\partial N_0(s)}{\partial \alpha} - \frac{1}{D_0(s)} \frac{\partial D_0(s)}{\partial \alpha} \right) \right|_{\alpha_0} \end{aligned}$$

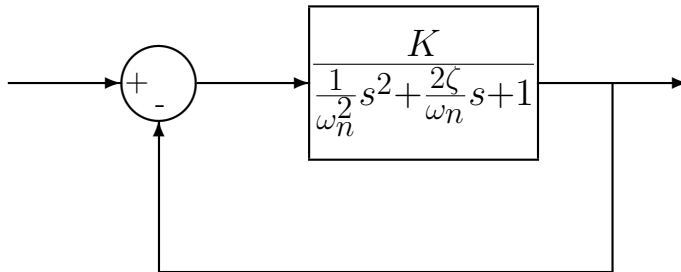
or

$$S_{\alpha}^{G_0}(s) = \left( \frac{\alpha}{N_0(s)} \frac{\partial N_0(s)}{\partial \alpha} - \frac{\alpha}{D_0(s)} \frac{\partial D_0(s)}{\partial \alpha} \right) \Big|_{\alpha_0}$$

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  $\alpha_0$

**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  $\omega_n$  (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 66 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  $\omega_n$  until  $\omega$  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

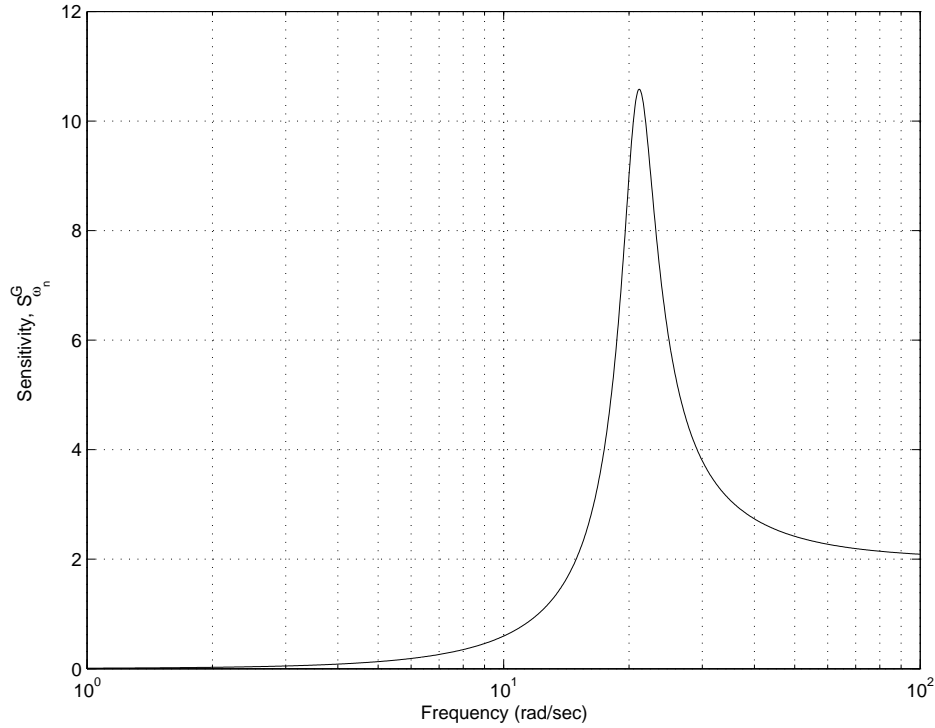
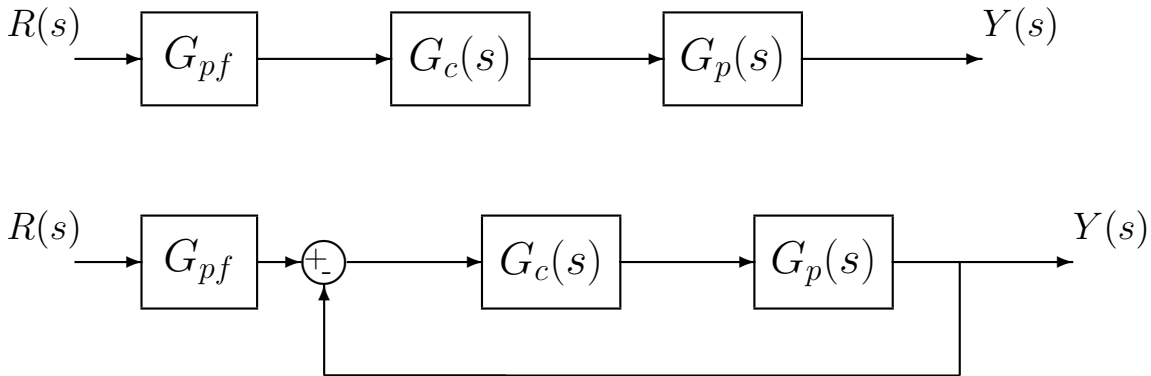


Figure 66: The sensitivity function of Example 1,  $S_{\omega_n}^{G_0}(j\omega)$ , as a function of frequency for the nominal values  $K = 0.1$ ,  $\omega_n = 20$ , and  $\zeta = 0.1$ .

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

$$\begin{aligned} S_{G_p}^{G_0^{open}} &= \frac{G_p(s)}{N_0(s)} \frac{\partial N_0(s)}{\partial G_p(s)} - \frac{G_p(s)}{D_0(s)} \frac{\partial D_0(s)}{\partial G_p(s)} \\ &= \frac{G_p(s)}{G_{pf}(s)G_c(s)G_p(s)} G_{pf}(s)G_c(s) - 0 \\ &= 1 \end{aligned}$$

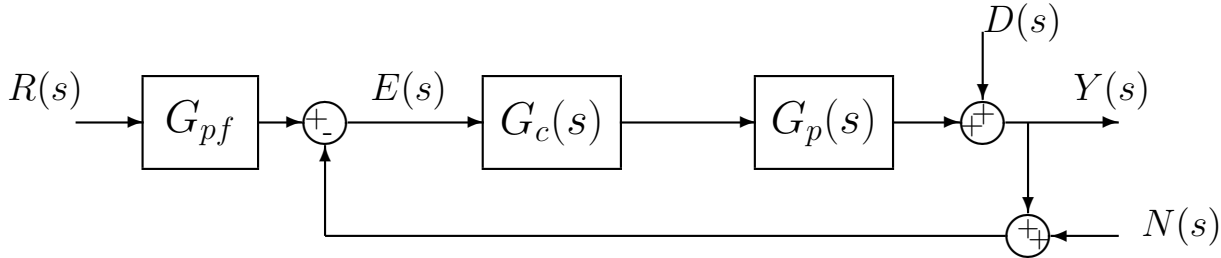
For the close loop system

$$\begin{aligned} S_{G_p}^{G_0^{closed}} &= \frac{G_p(s)}{N_0(s)} \frac{\partial N_0(s)}{\partial G_p(s)} - \frac{G_p(s)}{D_0(s)} \frac{\partial D_0(s)}{\partial G_p(s)} \\ &= \frac{G_p(s)}{G_{pf}(s)G_c(s)G_p(s)} G_{pf}(s)G_c(s) - \frac{G_p(s)}{1 + G_c(s)G_p(s)} G_c(s) \\ &= 1 - \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \\ &= \frac{[1 + G_c(s)G_p(s)] - [G_c(s)G_p(s)]}{1 + G_c(s)G_p(s)} \\ &= \frac{1}{1 + G_c(s)G_p(s)} \end{aligned}$$

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!*

## 13.2 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.

### 13.3 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,  $\omega$ ,  $|S_{\alpha_0}^{G_0}(j\omega)|$ . 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.



## 14 The Routh-Hurwitz Stability Criteria

The Routh-Hurwitz criteria is a method of determining if a system is stable. For example, for a simple unity feedback system with proportional control, the Routh-Hurwitz criteria can be used to determine if the system will become unstable for any range of gain  $k_p$ . Alternatively, if a system has a variable (or unknown) parameter, the Routh-Hurwitz criteria can be used to determine the range of values of that parameter before the system becomes unstable. The Routh-Hurwitz criteria is a necessary and sufficient criteria for the stability of linear systems.

The Routh-Hurwitz criteria is based on the characteristic equation of the transfer function between the input and the output,

$$\Delta(s) = a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + a_1 s + a_0 = 0$$

From this information we construct the Routh array as follows:

$$\begin{array}{c|cccc} s^n & a_n & a_{n-2} & a_{n-4} & \dots \\ s^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & \dots \\ s^{n-2} & b_{n-1} & b_{n-3} & b_{n-5} & \dots \\ s^{n-3} & c_{n-1} & c_{n-3} & c_{n-5} & \dots \\ \vdots & \vdots & & & \\ s^0 & h_{n-1} & & & \end{array}$$

Whenever you construct the Routh array, you initially

- write down the powers of  $s$  from the highest to  $s^0$ . It is sometimes difficult to determine when to stop the array, and if you have written down the powers of  $s$  this becomes much easier.
- starting with the coefficient of the highest power of  $s$  in  $\Delta(s)$ , write down every other coefficient in the first row. Then write down the odd coefficients in the second row. The rows may not be the same size. If they are not, append as many zeros to the second row as necessary to make it as long as the first row.

Next we need to compute all of the remaining rows. To fill in the third row, use the formulas

$$\begin{aligned} b_{n-1} &= \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}} \\ b_{n-3} &= \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}} \\ b_{n-5} &= \frac{a_{n-1}a_{n-6} - a_n a_{n-7}}{a_{n-1}} \\ &\vdots \end{aligned}$$

If you need a coefficient to use the formulas, and there is no such coefficient in the array, *assume the coefficient is zero*. Rather than memorizing the formula, look at the pattern! It's very similar to finding the determinant of a matrix.

To fill in the fourth row, use the formulas

$$\begin{aligned} c_{n-1} &= \frac{b_{n-1}a_{n-3} - a_{n-1}b_{n-3}}{b_{n-1}} \\ c_{n-3} &= \frac{b_{n-1}a_{n-5} - a_{n-1}b_{n-5}}{b_{n-1}} \\ c_{n-5} &= \frac{b_{n-1}a_{n-7} - a_{n-1}b_{n-7}}{b_{n-1}} \\ &\vdots \end{aligned}$$

You continue in this way until all rows are filled. Once the Routh array has been constructed, we use the Routh-Hurwitz criteria:

*The number of roots of  $\Delta(s)$  with positive real parts is equal to the number of sign changes in the first column of the Routh array.*

There are three different conditions for the Routh array we will consider

- No element in the first column is zero
- There is a zero in the first column, but some elements in the row containing the zero are nonzero
- There is a zero in the first column, and the entire row containing that zero is all zeros.

We will consider each of these in the next three sections.

## 14.1 No Element in the First Column is Zero

**Example 1.** Assume we have the characteristic polynomial  $\Delta(s) = s^2 + ps + 2$ , and we want to know the values of  $p$  for which the system is stable. We construct the initial Routh array as follows:

$$\begin{array}{c|cc} s^2 & 1 & 2 \\ s^1 & p & 0 \\ s^0 & b_1 & \end{array}$$

Note that we have appended a zero to the end of the second row, so it is the same size as the first row. The final entry in the array,  $b_1$ , is then computed as

$$b_1 = \frac{(p)(2) - (1)(0)}{p} = 2$$

Hence the final array looks like

$$\begin{array}{c|cc} s^2 & 1 & 2 \\ s^1 & p & 0 \\ s^0 & & 2 \end{array}$$

Now we need to count the sign changes going from top to bottom. All of the elements in the first column will be positive if  $p > 0$ , and there will be no sign changes, and no roots of  $\Delta(s)$  with positive real parts. Hence if  $p > 0$ , the system will always be stable. For this simple example we can use the quadratic formula to find the roots of  $\Delta(s)$  as

$$\frac{-p \pm \sqrt{p^2 - 8}}{2}$$

which shows the roots will always have negative real parts if  $p > 0$ . While we can use the quadratic formula in this case, when the order of the characteristic equation becomes larger this becomes more difficult.

**Example 2.** Consider the following characteristic equation,  $\Delta(s) = s^3 + 0.5s + s + 1 = 0$ . Is the system stable?

We construct the initial Routh array as

$$\begin{array}{c|cc} s^3 & 1 & 1 \\ s^2 & 0.5 & 1 \\ s^1 & b_1 & \\ s^0 & c_1 & \end{array}$$

To compute  $b_1$  we have

$$b_1 = \frac{(0.5)(1) - (1)(1)}{0.5} = -1$$

so the array looks like

$$\begin{array}{c|cc} s^3 & 1 & 1 \\ s^2 & 0.5 & 1 \\ s^1 & -1 & \\ s^0 & c_1 & \end{array}$$

In order to compute  $c_1$  we need another entry in the  $s^1$  row, hence this entry is a zero

$$\begin{array}{c|cc} s^3 & 1 & 1 \\ s^2 & 0.5 & 1 \\ s^1 & -1 & 0 \\ s^0 & c_1 & \end{array}$$

Computing  $c_1$  we have

$$c_1 = \frac{(-1)(1) - (0.5)(0)}{-1} = 1$$

Hence the final array looks like

$$\begin{array}{l|ll} s^3 & 1 & 1 \\ s^2 & 0.5 & 1 \\ s^1 & -1 & 0 \\ s^0 & 1 & \end{array}$$

There are *two* sign changes in the first column (from + to - and then from - to +), hence there are two roots of the characteristic polynomial that are in the right half plane.

**Example 3.** Assume we have a plant modeled as  $G_p(s) = \frac{2}{s^2+s+1}$ , and we try and control the system using an integral controller  $G_c(s) = \frac{k_i}{s}$  in a unity feedback configuration. The closed loop transfer function is then

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

Hence the characteristic polynomial is  $\Delta(s) = s^3 + s^2 + s + 2k_i$ , and we want to know if there is a range of  $k_i$  for which the system is stable. We construct the initial Routh array as follows:

$$\begin{array}{l|ll} s^3 & 1 & 1 \\ s^2 & 1 & 2k_i \\ s^1 & b_1 & \\ s^0 & c_1 & \end{array}$$

To compute  $b_1$  we have

$$b_1 = \frac{(1)(1) - (1)(2k_i)}{1} = 1 - 2k_i$$

so the array looks like

$$\begin{array}{l|ll} s^3 & 1 & 1 \\ s^2 & 1 & 2k_i \\ s^1 & 1 - 2k_i & \\ s^0 & c_1 & \end{array}$$

In order to compute  $c_1$  we need another entry in the  $s^1$  row, hence this entry is a zero

$$\begin{array}{l|ll} s^3 & 1 & 1 \\ s^2 & 1 & 2k_i \\ s^1 & 1 - 2k_i & 0 \\ s^0 & c_1 & \end{array}$$

Computing  $c_1$  we have

$$c_1 = \frac{(1 - 2k_i)(2k_i) - (1)(0)}{1 - 2k_i} = 2k_i$$

Hence the final array looks like

$$\begin{array}{l|ll} s^3 & 1 & 1 \\ s^2 & 1 & 2k_i \\ s^1 & 1 - 2k_i & 0 \\ s^0 & 2k_i & \end{array}$$

Now we need to count the sign changes going from top to bottom. The first two elements in the row are positive, hence for no sign changes we need all elements in the row to be positive, hence we need

$$\begin{aligned} 1 - 2k_i &> 0 \\ 2k_i &> 0 \end{aligned}$$

or  $0 < k_i < 0.5$  for stability.

**Example 4.** Consider the characteristic polynomial  $\Delta(s) = s^4 + 2s^3 + ps^2 + s + 3$ . Are there any values of  $p$  for which the system is stable?

We form the initial Routh array

$$\begin{array}{l|lll} s^4 & 1 & p & 3 \\ s^3 & 2 & 1 & 0 \\ s^2 & b_1 & b_2 & \\ s^1 & c_1 & & \\ s^0 & d_1 & & \end{array}$$

where we have appended a 0 to the end of the second column so it is the same length as the first column. To compute  $b_1$  and  $b_2$  we have

$$\begin{aligned} b_1 &= \frac{(2)(p) - (1)(1)}{2} = \frac{2p - 1}{2} = p - 0.5 \\ b_2 &= \frac{(2)(3) - (1)(0)}{2} = 3 \end{aligned}$$

The updated array looks like

$$\begin{array}{l|lll} s^4 & 1 & p & 3 \\ s^3 & 2 & 1 & 0 \\ s^2 & p - 0.5 & 3 & \\ s^1 & c_1 & & \\ s^0 & d_1 & & \end{array}$$

We can compute  $c_1$  as

$$c_1 = \frac{(p - 0.5)(1) - (2)(3)}{p - 0.5} = \frac{p - 6.5}{p - 0.5}$$

In order to compute  $d_1$  we also need to append a zero to the fourth row, so the updated array looks like

$$\begin{array}{l|ll}
s^4 & 1 & p & 3 \\
s^3 & 2 & 1 & 0 \\
s^2 & p - 0.5 & 3 & \\
s^1 & \frac{p-6.5}{p-0.5} & 0 & \\
s^0 & d_1 & & 
\end{array}$$

Finally, we compute

$$d_1 = 3$$

and the final array looks like

$$\begin{array}{l|ll}
s^4 & 1 & p & 3 \\
s^3 & 2 & 1 & 0 \\
s^2 & p - 0.5 & 3 & \\
s^1 & \frac{p-6.5}{p-0.5} & 0 & \\
s^0 & 3 & & 
\end{array}$$

In order for the system to be stable, there can be no sign changes in the first row, which in this case means all entries must be positive. Hence we need

$$\begin{aligned}
p - 0.5 &> 0 \\
\frac{p - 6.5}{p - 0.5} &> 0
\end{aligned}$$

which reduces to the condition  $p > 6.5$  for stability.

## 14.2 First Column Contains a Zero

If the first column of the Routh array contains a zero there are going to be a problem, since in computing the entry in the next column you will be dividing by zero, a very bad thing to do. To get around this problem, we assume the zero is instead a small *positive* number  $\epsilon$ , and continue computing the rest of the array. When we are all done we let  $\epsilon \rightarrow 0$  while *remaining positive*.

**Example 5.** Consider the characteristic polynomial  $\Delta(s) = s^4 + s^3 + s^2 + s + 3$ . The completed Routh array looks like

$$\begin{array}{l|ll}
s^4 & 1 & 1 & 3 \\
s^3 & 1 & 1 & 0 \\
s^2 & \epsilon & 3 & \\
s^1 & \frac{\epsilon-3}{\epsilon} & 0 & \\
s^0 & 3 & & 
\end{array}$$

In the third row, as  $\epsilon \rightarrow 0$ ,  $\epsilon$  is still positive, so the first entry in the row is positive. In the fourth row, as  $\epsilon \rightarrow 0$ , we have  $1 - \frac{3}{\epsilon}$  which becomes large and, more importantly, *negative* as  $\epsilon \rightarrow 0$ , since  $\epsilon > 0$ . Hence there are two sign changes, and thus there are two poles in the right half plane.

**Example 6.** Consider the characteristic polynomial  $\Delta(s) = s^4 + s^3 + s^2 + s + K$ . Determine if there are any values of  $K$  for which the system is stable. The completed Routh array look like

$$\begin{array}{c|ccc} s^4 & 1 & 1 & K \\ s^3 & 1 & 1 & 0 \\ s^2 & \epsilon & K & \\ s^1 & \frac{\epsilon-K}{\epsilon} & 0 & \\ s^0 & K & & \end{array}$$

Since  $\epsilon$  is positive, the first entry in row 3 is positive. In order for the first entry in row 5 to be positive, we must have  $K > 0$ . Looking now at the first entry of the fourth row, if both  $K > 0$  and  $\epsilon > 0$ , the entry will be negative. Hence there are no values of  $K$  for which the system is stable.

### 14.3 Entire Row is Zero

Sometimes an entire row of the Routh array is zero. This means we have found a factor of the characteristic equation. In order to continue with the Routh array when an entire row is zero, you form what is called an auxiliary polynomial (the polynomial that factors the characteristic polynomial), take it's derivative, and use those coefficients in the row which originally contained all zeros. This is easier explained with a few examples.

**Example 7.** Consider the characteristic polynomial  $\Delta(s) = s^7 + 3s^6 + 3s^5 + s^4 + s^3 + 3s^2 + 3s + 1$ . The initial Routh array looks like

$$\begin{array}{c|cccc} s^7 & 1 & 3 & 1 & 3 \\ s^6 & 3 & 1 & 3 & 1 \\ s^5 & \frac{8}{3} & 0 & \frac{8}{3} & 0 \\ s^4 & 1 & 0 & 1 & \\ s^3 & 0 & 0 & & \end{array}$$

At this point we have an entire row that is zero. We back up one row to find the auxiliary polynomial  $s^4 + 1$  (recall each row contains every other power). At this point we know that  $s^4 + 1$  is a factor of  $\Delta(s)$ , using long division we have

$$\Delta(s) = (s^4 + 1)(s^3 + 3s^2 + 3s + 1)$$

Now we take the derivative of the auxiliary polynomial and use that for the entries in the row with all zeros. In this case we compute

$$\frac{d}{ds}(s^4 + 1) = 4s^3$$

and use these coefficients in the fifth row. The completed Routh array looks like

$$\begin{array}{l|llll}
s^7 & 1 & 3 & 1 & 3 \\
s^6 & 3 & 1 & 3 & 1 \\
s^5 & \frac{8}{3} & 0 & \frac{8}{3} & 0 \\
s^4 & 1 & 0 & 1 & \\
s^3 & 4 & 0 & & \\
s^2 & \epsilon & 1 & & \\
s^1 & -\frac{4}{\epsilon} & & & \\
s^0 & 1 & & & 
\end{array}$$

From the Routh array we see there are two sign changes, so there are two poles in the right half plane. In this case the two right half plane poles come from two of the roots of  $s^4 + 1$ .

**Example 8.** Consider the characteristic polynomial  $\Delta(s) = s^4 + s^3 - s - 1$ . Use the Routh array to determine the number of right half plane poles (if any). The initial Routh array looks like

$$\begin{array}{l|lll}
s^4 & 1 & 0 & -1 \\
s^3 & 1 & -1 & 0 \\
s^2 & 1 & -1 & \\
s^1 & 0 & 0 & 
\end{array}$$

The entire fourth row is zero, so the row above determines the auxiliary polynomial, which in this case is  $s^2 - 1$ . Using long division we can factor the characteristic polynomial as

$$\Delta(s) = (s^2 - 1)(s^2 + s + 1)$$

Taking the derivative of the auxiliary polynomial we get

$$\frac{d}{ds}(s^2 - 1) = 2s$$

The finished Routh array is

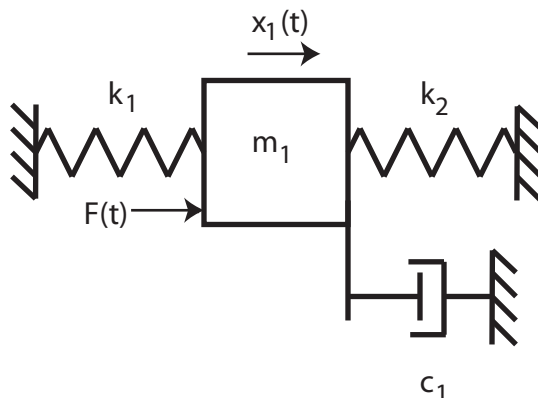
$$\begin{array}{l|lll}
s^4 & 1 & 0 & -1 \\
s^3 & 1 & -1 & 0 \\
s^2 & 1 & -1 & \\
s^1 & 2 & 0 & \\
s^0 & -1 & & 
\end{array}$$

There is one sign change in the first column, so there is one pole in the right half plane. This pole is one of the roots of the equation  $s^2 - 1$ .



## 15 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

$$\begin{aligned} \dot{q}_1(t) &= q_2(t) \\ \dot{q}_2(t) &= -2\zeta\omega_n \dot{x}(t) - \omega_n^2 x(t) + \omega_n^2 K_{static} u(t) \\ &= -2\zeta\omega_n q_2(t) - \omega_n^2 q_1(t) + \omega_n^2 K_{static} u(t) \\ &= -\omega_n^2 q_1(t) - 2\zeta\omega_n q_2(t) + \omega_n^2 K_{static} u(t) \end{aligned}$$

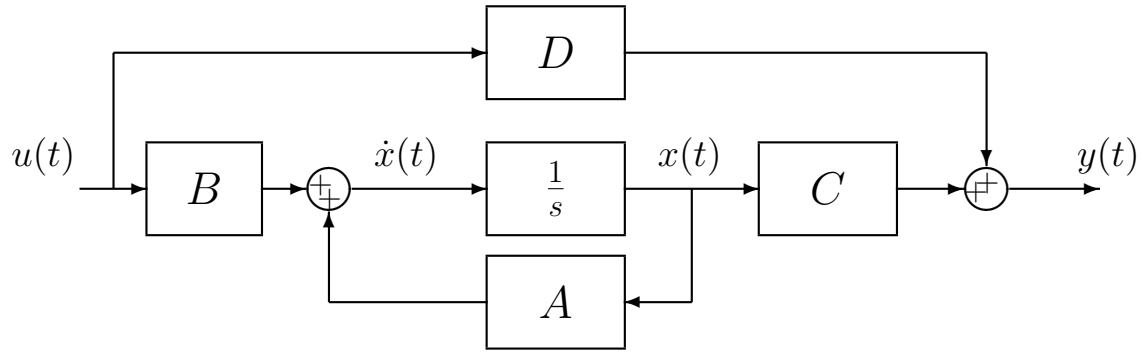


Figure 67: General state variable form for an open-loop plant

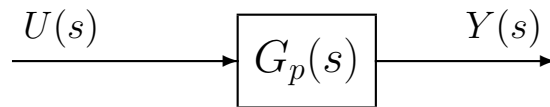


Figure 68: General transfer function form for an open-loop plant

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 67 shows the general form for an open-loop state variable model of a plant, while Figure 68 shows the equivalent transfer function form.

## 15.1 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

$$\begin{aligned}
 G(s) &= C(sI - A)^{-1}B \\
 &= \frac{1}{s(s-1)} [s \quad 2s-2] \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
 &= \frac{1}{s(s-1)} [s + (2s-2)] \\
 &= \frac{3s-2}{s(s-1)}
 \end{aligned}$$

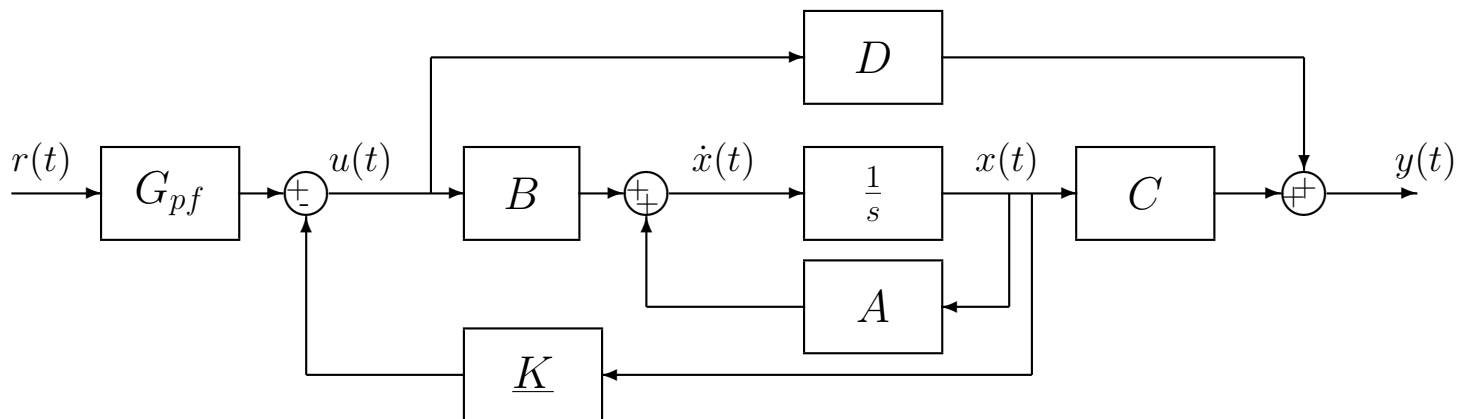


Figure 69: State variable model of a plant with state variable feedback.

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.

## 15.2 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 69 displays a state variable model of a plant with state variable feedback.

With the state variable feedback the state equations become

$$\begin{aligned} \dot{\underline{x}}(t) &= \underline{A}\underline{x}(t) + \underline{B}u(t) \\ &= \underline{A}\underline{x}(t) + \underline{B}[G_{pf}r(t) - \underline{K}\underline{x}(t)] \\ &= [\underline{A} - \underline{B}\underline{K}]\underline{x}(t) + [\underline{B}G_{pf}]r(t) \\ &= \tilde{\underline{A}}\underline{x}(t) + \tilde{\underline{B}}r(t) \end{aligned}$$

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

$$\begin{aligned} \dot{\underline{x}} &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \underline{x} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u \\ y &= [1 \ 2] \underline{x} \end{aligned}$$



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.

### 15.3 Controllability for State Variable Systems

A single-input single-output state variable system is said to be *controllable*<sup>8</sup> 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

$$\begin{aligned} 5K_1 - 4 &= a_1 \\ 5K_1 &= a_1 + 4 \\ K_1 &= \frac{a_1 + 4}{5} \end{aligned}$$

---

<sup>8</sup>This 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.

## 15.4 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 Linearization

Up to this point we have assumed that we have a transfer function model of the system we are trying to control. However, a transfer function model only exists if the system has a linear model. If a model is not linear, then we need to determine a linear model of the system in order to use the techniques we have developed in this class. However, this model is likely to be valid only over a limited range of values. Before we go into how to get a linear model, we need to be clear on what we mean by a linear system and review Taylor series.

### 16.1 Linear Systems

In general, if we have input  $u(t)$  and output  $y(t)$  we can represent the input output relationship of a system, whether it is linear or not, as

$$u(t) \rightarrow y(t)$$

Assume input  $u_1(t)$  produces output  $y_1(t)$  and input  $u_2(t)$  produces  $y_2(t)$ ,

$$\begin{aligned}u_1(t) &\rightarrow y_1(t) \\u_2(t) &\rightarrow y_2(t)\end{aligned}$$

The system is said to be *linear* if and only if

$$\alpha_1 u_1(t) + \alpha_2 u_2(t) \rightarrow \alpha_1 y_1(t) + \alpha_2 y_2(t)$$

for all  $\alpha_1$ ,  $\alpha_2$ ,  $u_1(t)$ , and  $u_2(t)$ . If a system is not linear, we cannot take its Laplace transform, and thus cannot use transfer functions. However, we can often produce a linear model of a system if we assume it does not deviate too much from a fixed (nominal) value. Hence we are looking for a linear model near a fixed point. Usually we will assume the fixed point is an equilibrium point. This is very similar to first biasing a transistor circuit, and then using small signal analysis about this biasing point.

### 16.2 Taylor Series

Assume we have a function  $f(z)$  and we want to approximate the function near  $z = 0$ . The Taylor series approximation near  $z = 0$  is

$$f(z) \approx f(0) + f'(0)z + \text{higher order terms}$$

You should be able to derive all of the entries in Table 16.2. This approximation is only valid for  $z$  near 0. The further away from zero we go, the worse the approximation is likely to be.

$f(z)$	Linear Approximation
$(1+z)^a$	$1+az$
$e^{az}$	$1+az$
$\cos(az)$	$1$
$\sin(az)$	$az$
$\ln(1+z)$	$z$
$\cos(\alpha+z)$	$\cos(\alpha) - z \sin(\alpha)$
$\sin(\alpha+z)$	$\sin(\alpha) + z \cos(\alpha)$

Table 2: Functions and their linear approximation near  $z = 0$ .

### 16.3 Linearization Procedure

Our goal here is to find a linear model that we can use to determine the transfer function of a system. The procedure we will go through is listed below, and will be followed with a few examples.

**Step 1** Determine the nominal *operating point* of the system and the equation that these operating points solve. We will assume the operating points are the static equilibrium points. At the static equilibrium points, all derivatives are zero. For the linearization to be valid, the system must not stray very far from this operating point. Label these points  $x_0, y_0, u_0$ , etc. *These points are assumed to be constants.*

**Step 2** Look at variations from these operation points. For example, we assume

$$\begin{aligned}x(t) &= x_0 + \Delta x(t) \\y(t) &= y_0 + \Delta y(t) \\u(t) &= u_0 + \Delta u(t)\end{aligned}$$

Note that only  $\Delta x(t), \Delta y(t)$ , etc. vary with time.  $x_0, y_0$ , etc. are constants. Now we have two cases to consider:

**Step 2a** If our functions are *arguments* to other standard functions, we leave this approximation as it is. For example,  $\cos(x(t))$  would be rewritten  $\cos(x_0 + \Delta x(t))$ . Similarly for all other trigonometric functions and exponentials.

**Step 2b** If our functions are not arguments to standard functions, we rewrite the functions as

$$\begin{aligned}x(t) &= x_0 + \Delta x(t) = x_0 \left(1 + \frac{\Delta x(t)}{x_0}\right) \\y(t) &= y_0 + \Delta y(t) = y_0 \left(1 + \frac{\Delta y(t)}{y_0}\right) \\u(t) &= u_0 + \Delta u(t) = u_0 \left(1 + \frac{\Delta u(t)}{u_0}\right)\end{aligned}$$

We rewrite the functions in this way because this is the form we will use the Taylor series on. Here our *small*  $z$  will be  $\frac{\Delta x(t)}{x_0}, \frac{\Delta y(t)}{y_0}$ , etc.

**Step 3** Substitute our expressions for  $x(t)$ ,  $y(t)$ , etc., into the dynamics, and simplify where possible.

**Step 4** Using Taylor series, expand out all nonlinear terms.

**Step 5** Put the Taylor series expansion into the defining differential equation and multiply out all terms.

**Step 6** Drop all second order (or higher) terms. Thus terms of the form  $\left(\frac{\Delta x(t)}{x_0}\right)^2$ ,  $\left(\frac{\Delta x(t)}{x_0}\right)\left(\frac{\Delta y(t)}{y_0}\right)$ , etc. will be dropped.

**Step 7** Using the relationships found in step 1, try and remove all constant terms in the model. If there are any constant terms left over, you have made an error. All of the remaining terms should be  $\Delta$  terms.

**Step 8** Find the resulting transfer function.

**Example 1.** Assume we have the model of a system with input  $u(t)$  and output  $x(t)$

$$\dot{x}(t) + 3x^2(t) = u(t) + 3$$

and we want to find a linearized model about the static equilibrium point.

*Step 1* At equilibrium we have the equation  $3x_0^2 = u_0 + 3$ .

*Step 2* Assume  $x(t) = x_0 \left(1 + \frac{\Delta x(t)}{x_0}\right)$  and  $u(t) = \left(1 + \frac{\Delta u(t)}{u_0}\right)$

*Step 3* Now substitute into the dynamics and do some simplification

$$\begin{aligned} \frac{d}{dt} \left[ x_0 \left(1 + \frac{\Delta x(t)}{x_0}\right) \right] + 3 \left[ x_0 \left(1 + \frac{\Delta x(t)}{x_0}\right) \right]^2 &= u_0 \left(1 + \frac{\Delta u(t)}{u_0}\right) + 3 \\ \Delta \dot{x}(t) + 3x_0^2 \left(1 + \frac{\Delta x(t)}{x_0}\right)^2 &= u_0 + \Delta u(t) + 3 \end{aligned}$$

*Step 4* Expand out the only nonlinear term we have

$$\left(1 + \frac{\Delta x(t)}{x_0}\right)^2 \approx 1 + 2\frac{\Delta x(t)}{x_0}$$

*Step 5* We now substitute the expanded term into the equation, and simplify as much as possible

$$\begin{aligned} \Delta \dot{x}(t) + 3x_0^2 \left[1 + 2\frac{\Delta x(t)}{x_0}\right] &\approx u_0 + \Delta u(t) + 3 \\ \Delta \dot{x}(t) + 3x_0^2 + 6x_0\Delta x(t) &\approx u_0 + \Delta u(t) + 3 \end{aligned}$$

*Step 6* We have no higher order terms.

*Step 7* From step 1, we have  $3x_0^2 = u_0 + 3$ . Substituting this into our equation from step 5 we have

$$\begin{aligned}\Delta\dot{x}(t) + [u_0 + 3] + 6x_0\Delta x(t) &\approx u_0 + \Delta u(t) + 3 \\ \Delta\dot{x}(t) + 6x_0\Delta x(t) &\approx \Delta u(t)\end{aligned}$$

*Step 8* Taking Laplace transforms we have

$$s\Delta X(s) + 6x_0\Delta X(s) \approx \Delta U(s)$$

or

$$\frac{\Delta X(s)}{\Delta U(s)} \approx \frac{1}{s + 6x_0}$$

**Example 2.** Assume we have the model of a system with input  $u(t)$  and output  $x(t)$

$$2\dot{x}(t) + \sqrt{x(t)} = \cos(u(t))$$

and we want to find a linearized model about the static equilibrium point.

*Step 1* At equilibrium we have the equation  $\sqrt{x_0} = \cos(u_0)$ .

*Step 2* For the square root term we will assume the form  $x(t) = x_0 \left(1 + \frac{\Delta x(t)}{x_0}\right)$  while for the cosine term we will assume the form  $u(t) = u_0 + \Delta u(t)$

*Step 3* Now substitute into the dynamics and do some simplification

$$\begin{aligned}2\frac{d}{dt} \left[ x_0 \left(1 + \frac{\Delta x(t)}{x_0}\right) \right] + \sqrt{x_0 \left(1 + \frac{\Delta x(t)}{x_0}\right)} &= \cos(u_0 + \Delta u(t)) \\ 2\Delta\dot{x}(t) + \sqrt{x_0} \sqrt{1 + \frac{\Delta x(t)}{x_0}} &= \cos(u_0 + \Delta u(t))\end{aligned}$$

*Step 4* Expanding out the nonlinear terms we have

$$\begin{aligned}\sqrt{1 + \frac{\Delta x(t)}{x_0}} &= \left(1 + \frac{\Delta x(t)}{x_0}\right)^{\frac{1}{2}} \approx 1 + \frac{1}{2} \frac{\Delta x(t)}{x_0} \\ \cos(u_0 + \Delta u(t)) &\approx \cos(u_0) - \Delta u(t) \sin(u_0)\end{aligned}$$

*Step 5* We now substitute the expanded terms into the equation, and simplify as much as possible

$$\begin{aligned}2\Delta\dot{x}(t) + \sqrt{x_0} \left[1 + \frac{1}{2} \frac{\Delta x(t)}{x_0}\right] &\approx \cos(u_0) - \Delta u(t) \sin(u_0) \\ 2\Delta\dot{x}(t) + \sqrt{x_0} + \frac{1}{2\sqrt{x_0}} \Delta x(t) &\approx \cos(u_0) - \Delta u(t) \sin(u_0)\end{aligned}$$

*Step 6* We have no higher order terms.



*Step 7* From step 1, we have  $\sqrt{x_0} = \cos(u_0)$ . Substituting this into our equation from step 5 we have

$$2\Delta\dot{x}(t) + [\cos(u_0)] + \frac{1}{2\sqrt{x_0}}\Delta x(t) \approx \cos(u_0) - \Delta u(t) \sin(u_0)$$

$$2\Delta\dot{x}(t) + \frac{1}{2\sqrt{x_0}}\Delta x(t) \approx -\Delta u(t) \sin(u_0)$$

*Step 8* Taking Laplace transforms we have

$$2s\Delta X(s) + \frac{1}{2\sqrt{x_0}}\Delta X(s) \approx -\Delta U(s) \sin(u_0)$$

or

$$\frac{\Delta X(s)}{\Delta U(s)} \approx \frac{-\sin(u_0)}{2s + \frac{1}{2\sqrt{x_0}}}$$

**Example 3.** Assume we have the model of a system with input  $u(t)$  and output  $x(t)$

$$\dot{x}(t) + \frac{1}{\sqrt{x(t)}}e^{-au(t)} = 1$$

and we want to find a linearized model about the static equilibrium point.

*Step 1* At equilibrium we have the equation  $\frac{1}{\sqrt{x_0}}e^{-au_0} = 1$ .

*Step 2* For the square root term we will assume the form  $x(t) = x_0 \left(1 + \frac{\Delta x(t)}{x_0}\right)$  while for the exponential term we will assume the form  $u(t) = u_0 + \Delta u(t)$

*Step 3* Now substitute into the dynamics and do some simplification

$$\frac{d}{dt} \left[ x_0 \left(1 + \frac{\Delta x(t)}{x_0}\right) \right] + \frac{1}{\sqrt{x_0 \left(1 + \frac{\Delta x(t)}{x_0}\right)}} e^{-au_0 - a\Delta u(t)} = 1$$

$$\Delta\dot{x}(t) + \frac{e^{-au_0}}{\sqrt{x_0}} \frac{e^{-a\Delta u(t)}}{\sqrt{1 + \frac{\Delta x(t)}{x_0}}} = 1$$

*Step 4* Expanding out the nonlinear terms we have

$$\frac{1}{\sqrt{1 + \frac{\Delta x(t)}{x_0}}} = \left(1 + \frac{\Delta x(t)}{x_0}\right)^{-\frac{1}{2}} \approx 1 - \frac{1}{2} \frac{\Delta x(t)}{x_0}$$

$$e^{-a\Delta u(t)} \approx 1 - a\Delta u(t)$$

*Step 5* We now substitute the expanded terms into the equation, and simplify as much as possible

$$\Delta\dot{x}(t) + \frac{e^{-au_0}}{\sqrt{x_0}} \left(1 - \frac{1}{2} \frac{\Delta x(t)}{x_0}\right) (1 - a\Delta u(t)) \approx 1$$

$$\Delta\dot{x}(t) + \frac{e^{-au_0}}{\sqrt{x_0}} \left(1 - \frac{1}{2} \frac{\Delta x(t)}{x_0} - a\Delta u(t) + \frac{a}{2} \frac{\Delta x(t)}{x_0} \Delta u(t)\right) \approx 1$$

*Step 6* We drop the product  $\Delta x(t)\Delta u(t)$  (i.e., we assume it is zero), so we have

$$\begin{aligned}\Delta \dot{x}(t) + \frac{e^{-au_0}}{\sqrt{x_0}} \left( 1 - \frac{1}{2} \frac{\Delta x(t)}{x_0} - a\Delta u(t) \right) &\approx 1 \\ \Delta \dot{x}(t) + \frac{e^{-au_0}}{\sqrt{x_0}} - \frac{e^{-au_0}}{\sqrt{x_0}} \frac{1}{2} \frac{\Delta x(t)}{x_0} - \frac{e^{-au_0}}{\sqrt{x_0}} a\Delta u(t) &\approx 1\end{aligned}$$

*Step 7* From step 1, we have  $\frac{e^{-au_0}}{\sqrt{x_0}} = 1$ . Substituting this into our equation from step 6 we have

$$\Delta \dot{x}(t) - \frac{1}{2} \frac{\Delta x(t)}{x_0} - a\Delta u(t) \approx 0$$

*Step 8* Taking Laplace transforms we have

$$s\Delta X(s) - \frac{1}{2x_0}\Delta X(s) - a\Delta U(s) \approx 0$$

or

$$\frac{\Delta X(s)}{\Delta U(s)} \approx \frac{a}{s - \frac{1}{2x_0}}$$

# 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 tell 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];
```

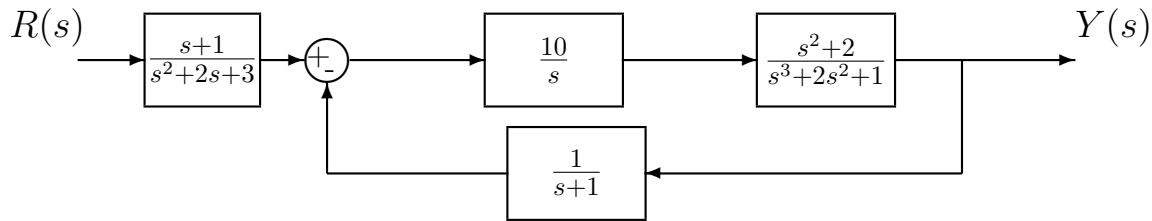
We will usually need to construct the transfer functions explicitly. To do this, type

```
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  $\infty$ , 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 (100) and 100 (102)
```

```
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;
```