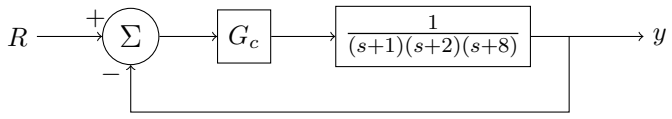
Lecture 12

Last lecture: Control system design via root locus for proportional control.
Today: Continue discussion of control system design

- Integral control
- Derivative control

Last Lecture

Recall the system from the previous lecture:



We designed a proportional controller $G_c = K_p$ so that a unit step input at $r(t)$ lead to a $y(t)$ response that met the following requirements:

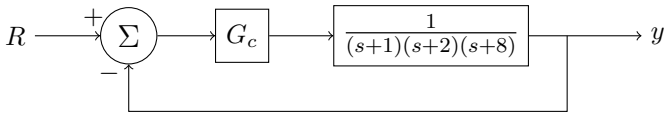
1. overshoot $\leq 20\%$
2. peak time ≤ 2 seconds
3. steady-state error ≤ 0.4

Our design process was as follows:

1. Sketch the root locus
2. From the overshoot requirement, find the minimum damping: $\zeta \geq 0.46$
3. From the peak time requirement, find the minimum natural frequency: $\omega_n > 1.76$
4. Plot the circle for $\omega_n = 1.76$ and the line for $\zeta = 0.46$ and find where the intercept the root locus
5. Use the magnitude criterion to find the limiting gains: $\zeta \geq 0.46 \Rightarrow K_p \leq 38$, $\omega_n > 1.76 \Rightarrow K_p \geq 9$.
6. From the the steady-state error requirement, find $K_p \geq 24$. Therefore, $24 \leq K_p \leq 38$.
7. Pick a K_p in that range.
8. Verify the second-order approximation.
9. Validate with simulation.

Integral Control – Example 2

Let us consider the same system, but with new performance requriements.



1. overshoot $\leq 20\%$
2. peak time ≤ 2 seconds
3. steady-state error ≤ 0.15 (previously 0.4)

Let's start by trying a proportional controller $G_c = K_p$ again. Following the same procedure as last time, we have $9 \leq K_p \leq 38$ from the overshoot and peak time requirements, and will now look at steady-state error. For a step input:

$$e_{ss} = \frac{1}{1 + \lim_{s\to 0} G}$$

$$e_{ss} = \frac{1}{1 + \lim_{s\to 0} \frac{K_p}{(s+1)(s+2)(s+8)}} = \frac{1}{1 + \frac{K_p}{16}} = \frac{16}{16 + K_p}$$

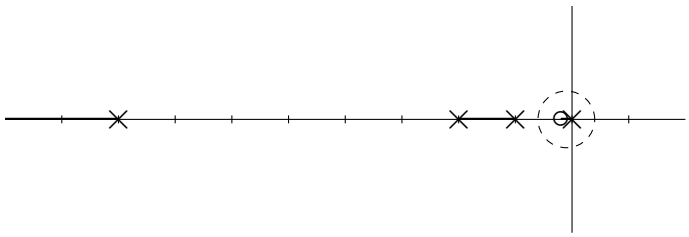
$$e_{ss} \leq 0.15 \quad \Rightarrow \quad K_p \geq 90.7$$

This means we must have $K_p \leq 38$ to meet the overshoot requirement but $K_p \geq 90.7$ to meet the steady-state error. Proportional control will not work! We need a more complicated controller.

How can we reduce e_{ss}?

- If our controller has an integrator $(1/s)$, we will have a Type-1 system, therefore $e_{ss} = 0$ for a step input.
- $G_c = K_p/s$ will eliminated steady-state error but will introduce its own problems
  - Changes shape of root locus.
  - In this example, it would make meeting the peak time and overshoot requirements impossible.
- Can we add something else to G_c that will minimize the effect of this pole on transient behavior?
- We could add a zero just to the left of the origin. Then, the root locus would be mostly unaffected.

$$\text{PI Control: } G_c = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} = \frac{K(s + z_c)}{s}, \quad z \text{ is near } 0$$



Let's take a closed look at poles and zeros in PI control: We have an open-loop pole at $s = 0$ and an open-loop zero at $s = -z_c$, where z_c is near zero.

- If there is no other pole or zero between $s = 0$ and $s = z_c$, then they will cancel each other out with respect to the root locus.

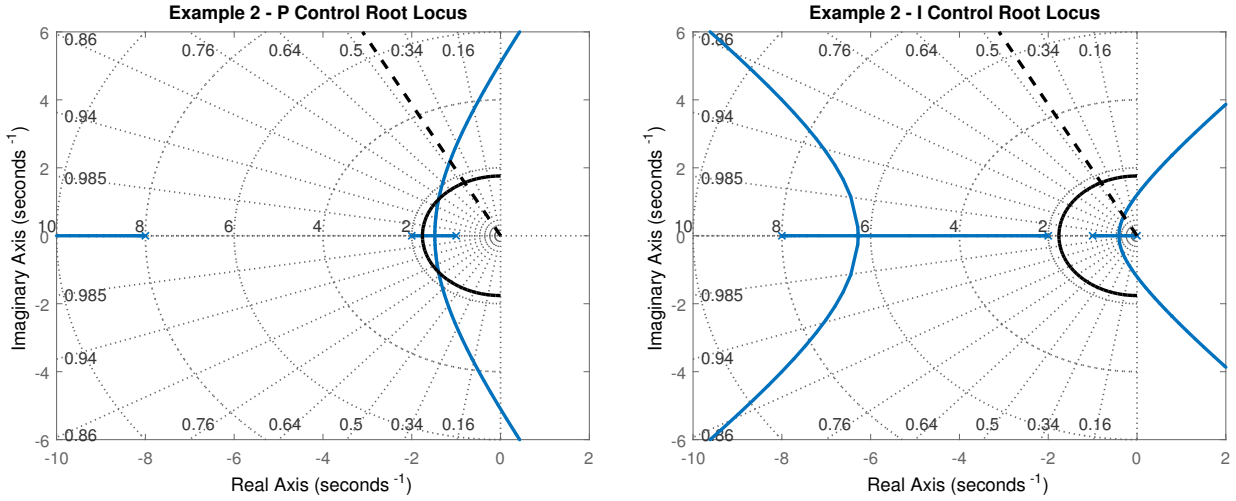
- The root locus shape (angle criterion) and gains (magnitude criterion) remain the same as for proportional control

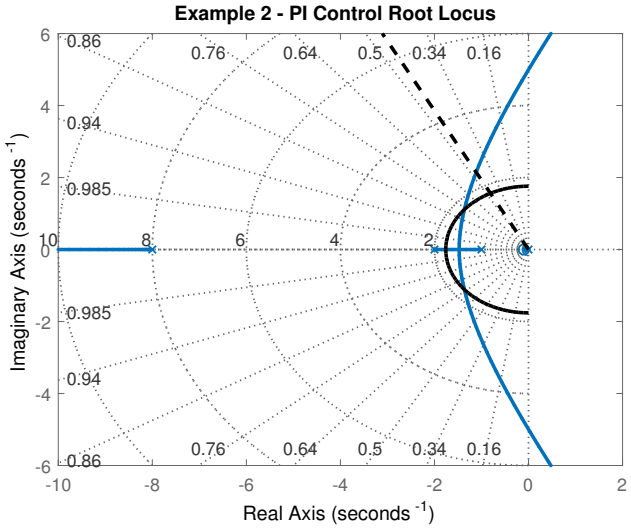
For our plant $\frac{1}{(s+1)(s+2)(s+8)}$, this means we could keep the controller gain the same as in proportional control ($24 \leq K \leq 38$, let's say $K = 30$) while eliminating steady-state error with a integrator/zero pair (for example, we will place the pole at $s = -0.1$). Then,

$$G_c = \frac{30(s + 0.1)}{s}$$

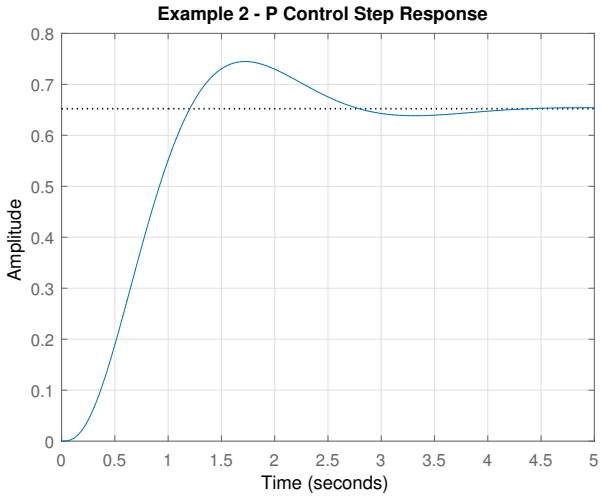
will meet the design requirements. This is called **PI Control**.

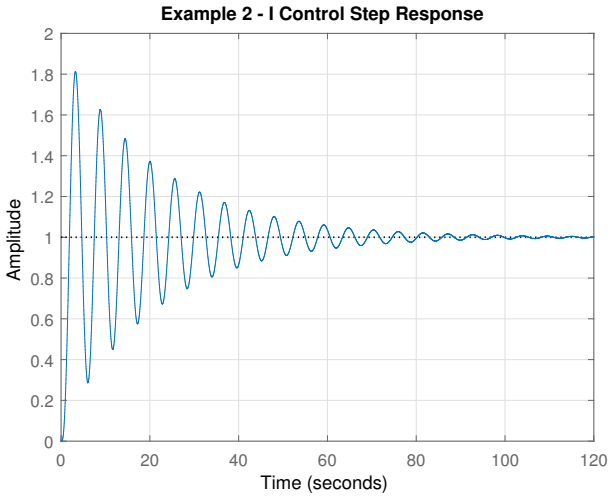
Now, we will use Matlab to compare the root locus of P, I, and PI controllers, and plot a step response for $K = 30$.

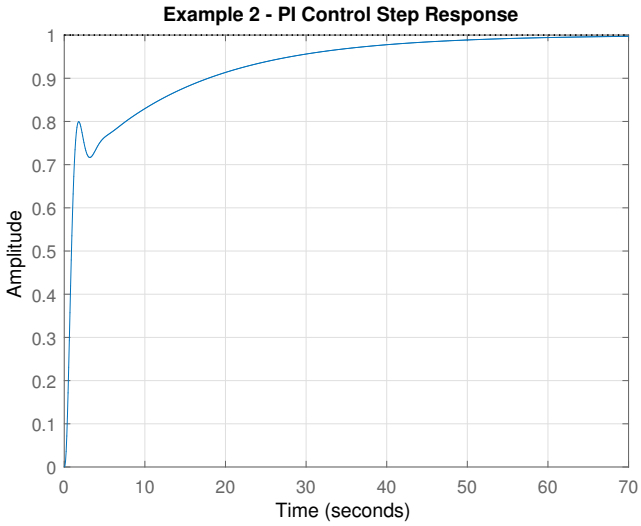




The proportional control root locus is the same as we found last lecture. Integral control is unable to meet the design requirements: gains that meet the peak time requirements must be outside the $\omega_n > 1.76$ circle — this never occurs while the closed-loop system is stable. PI control preserves the root locus of the proportional control system, but also eliminates steady-state error.

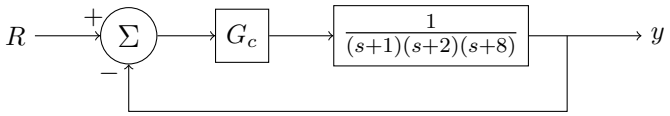

Example 2 - P Control Step Response
Amplitude
Time (seconds)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Example 2 - I Control Step Response
Amplitude
Time (seconds)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
0
20
40
60
80
100
120


Example 2 - PI Control Step Response
Amplitude
Time (seconds)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70

Derivative Control – Example 3

Let us consider the same system, again with new performance requirements.



1. overshoot $\leq 20\%$
2. peak time ≤ 1.3 seconds (previously 2)
3. steady-state error ≤ 0.4

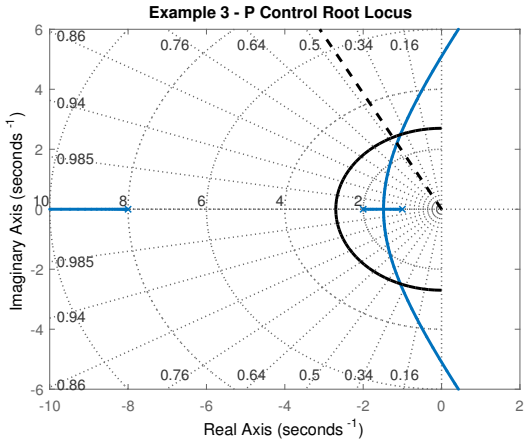
We will attempt to find a controller using proportional control.

From the first example:

$$O.S. \leq 20\% \Rightarrow \zeta \geq 0.46$$

$$t_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \Rightarrow \omega_n \geq 2.7$$

We can plot the root locus and graphically find the gain requirements.



We need to find a point on the root locus that is outside the $\omega_n \geq 2.7$ circle and below the $\zeta = 0.46$ line. Clearly, this is not possible. Proportional control will not work for these requirements.

What other options do we have? Let's start with proportional control and add a zero to change the shape of the root locus.

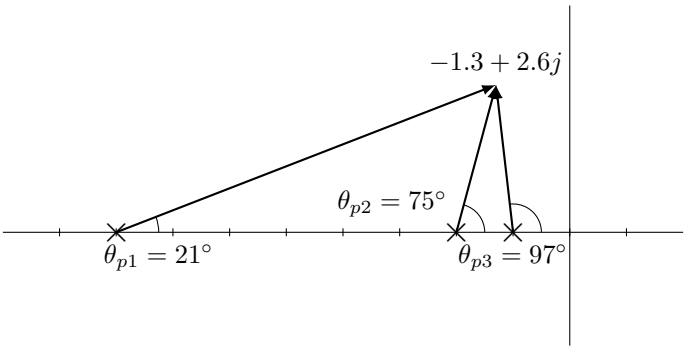
$$\text{PD Control: } G_c = K_p + K_d s = K_d \left(s + \frac{K_p}{K_d} \right) = K_d(s + z_c$$

Note that this transfer function is improper — we would need to implement a fast pole to make it proper, but at this point don't worry about it.

The method for determining the PD zero location is as follows:

- Choose a point you want the root locus to pass though.
- Use the angle criterion to find the zero location that makes it possible.

For this example, we will find a zero location that let's us place a closed-loop pole at $s = -1.3 \pm 2.6j$. This would give us $\zeta = 0.46$ and $\omega_n = 2.9$, meeting our design requirements. (dobule check all math)



Angle criterion:

$$\sum \theta_{zi} - \sum \theta_{pi} = 180° \pm \ell 360°$$
$$\theta_z - 21° - 75° - 97° = 180° \pm \ell 360°$$
$$\theta_z = 373° \pm \ell 360° = 13°$$

So, if $\theta_z = 13°$, then

$$\tan(\theta_z) = \tan(13°) = \frac{2.6}{z - 1.3}$$

(Reminder: tangent = opposite over adjacent.)

Therefore, $z = 12.5$; a zero is at $s = -12.5$ will make the root locus pass through $1.3 + 2.6j$. In addition, our 2nd-order approximation will still be valid.

Next, we can use the magnitude criterion to find the appropriate gain.

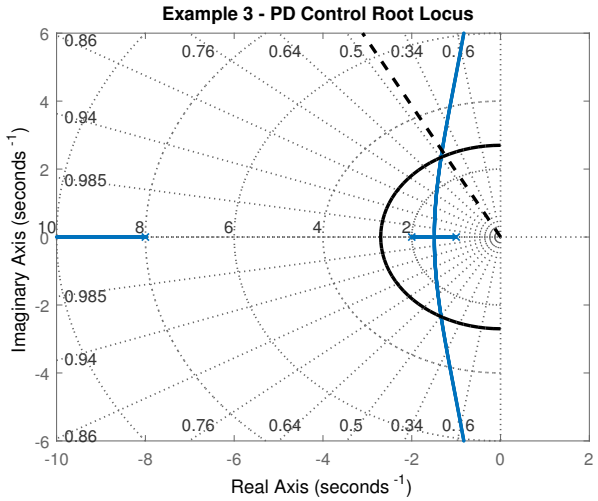
$$K = \frac{M_{p1} M_{p2} M_{p3}}{M_z} = \frac{(7.2)(2.7)(2.6)}{11.6} = 4.4$$
$$\Rightarrow \quad G_c = 4.4(s + 12.5)$$

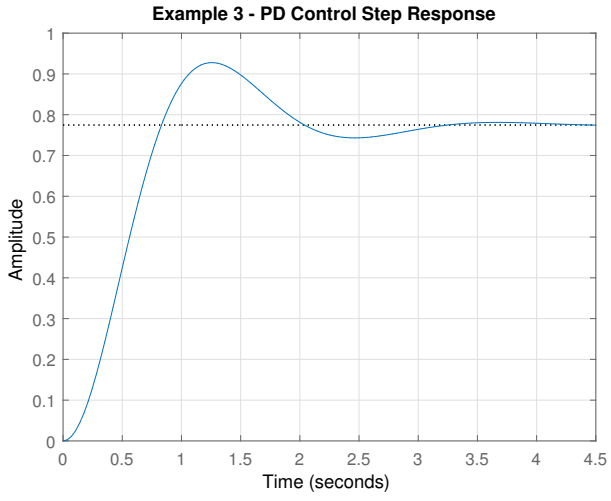
We can now verify the steady-state error requirement:

$$e_{ss} = \frac{1}{1 + \lim_{s \to 0} \frac{4.4(s+12.5)}{(s+1)(s+2)(s+8)}} = \frac{1}{1 + \frac{55}{16}}$$
$$e_{ss} = 0.23$$

We can then validate with simulation (plots on next page).

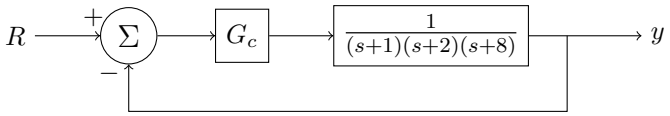
Key Point: Derivative control is used to modify the root locus. Integral control does not modify the root locus, it only decreases the error.





PID Control – Example 4

Let us consider the same system, again with even more restrictive performance requirements.



1. overshoot $\leq 20\%$
2. peak time ≤ 1.3 seconds (previously 2)
3. steady-state error ≤ 0.05 (previously 0.4)

To meet these requirements, we are going to take our PD controller from the last example ad add integral control. Then,

$$\text{PID Control: } G_c = K_p + \frac{K_i}{s} + K_d s = \frac{K_d\left(s^2 + \frac{K_p}{K_d}s + \frac{K_i}{K_d}\right)}{s} = \frac{K_d(s+z_1)(s+z_2)}{s}$$

z_1 will be chosen to shape the root locus (as in PD control), while z_2 will be placed near zero to cancel the effects of the integrator on the root locus (as in PI control). Then, we can tune the gain K_d.

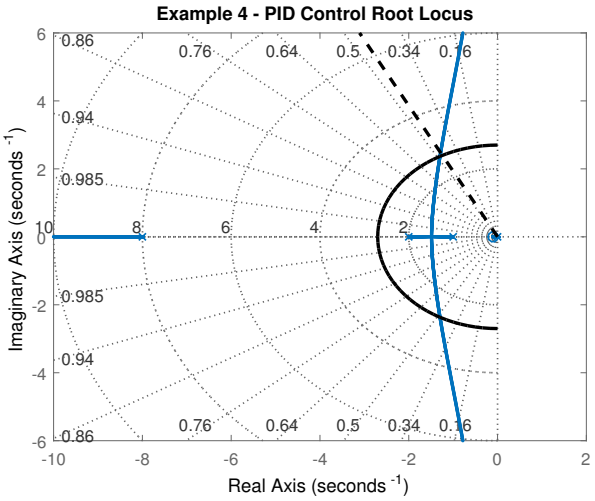
Using the gains and zero locations from the previous examples, we have:

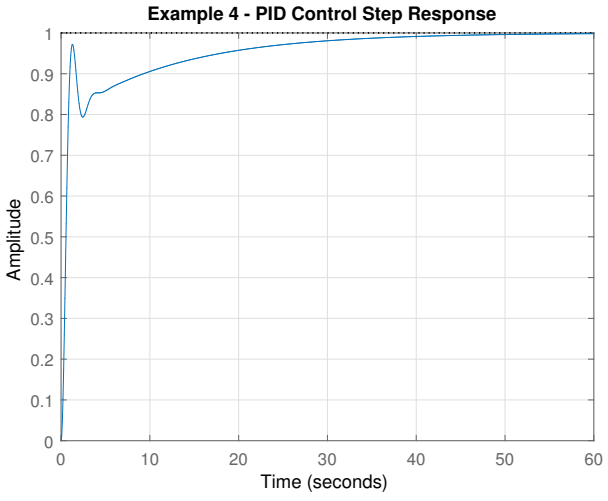
$$G_c = \frac{4.4(s+12.5)(s+0.1)}{s}$$

Is the second-order approximation valid here?

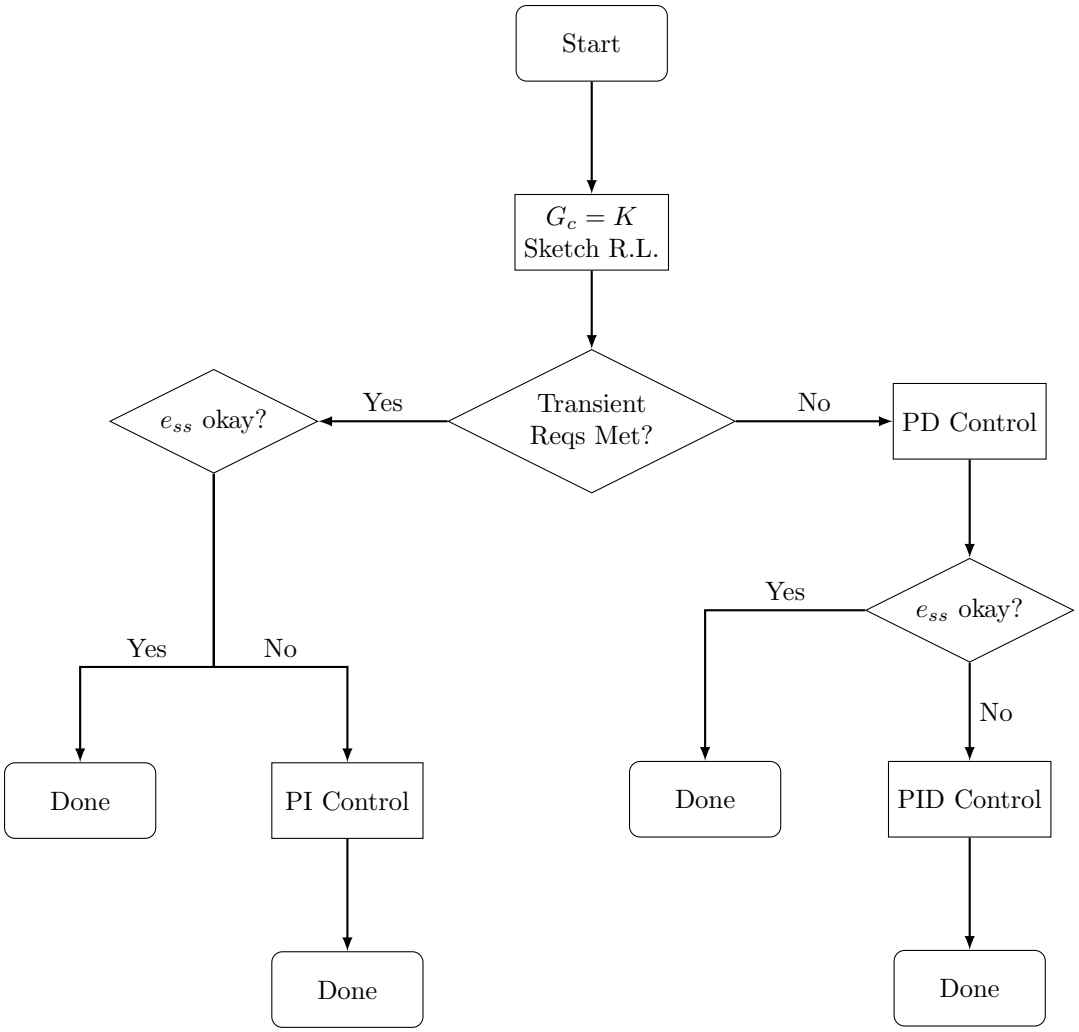
• Yes – the zero at -12.5 is far enough left to be non-dominant, and the pole and zero near the origin cancel each other.

The root locus and step response of this system are shown below.





The following procedure can be used for root locus controller design:

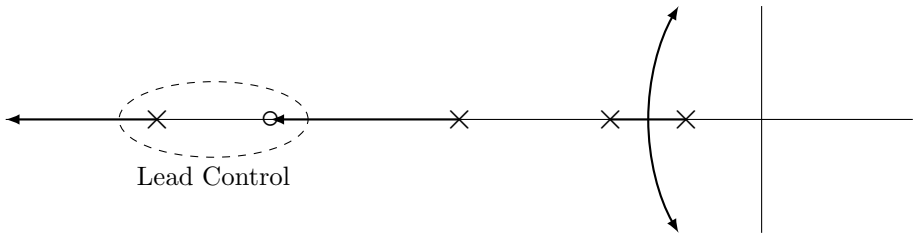


Lead/Lag Control

We will briefly talk about some alternatives to PI, PD, and PID control: Lead, Lag, and Lead/Lag control.

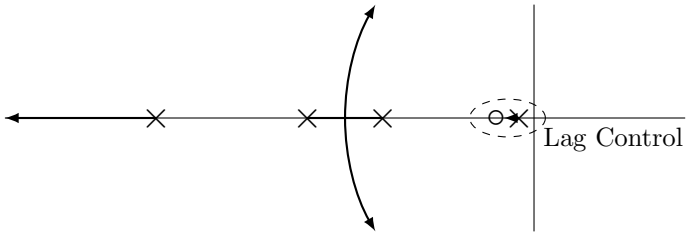
Lead Control

- $\frac{s+z_c}{s+p_c}$, $z_c < p_c$
- Lead control is kind of like derivative control.
- Place a zero and a pole to the left of the zero to shape the root locus.
- Pulls the root locus to the left, but not as much as just a zero at $s = -z_c$.
- PD control reduces the number of asymptotes by 1. Lead control keeps them the same.



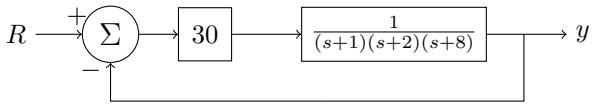
Lag Control

- $\frac{s+z_c}{s+p_c}$, $z_c > p_c$
- Kind of like integral control.
- Place a pole just left of the origin (instead of *at* the origin), and a zero to the left of the pole.
- The pole and zero would still cancel each other out with respect to transient behavior.
- Does not effect root locus (pole and zero cancel).
- Reduces e_{ss}, but does not increase system type (i.e., can't reduce e_{ss} to zero).
- The state-state error will be less than for proportional control, but this may be okay based on design requirements.
- A large ratio of z_c/p_c will reduce error the most, which is accomplished by moving p_c near the origin (while keeping z_c close).

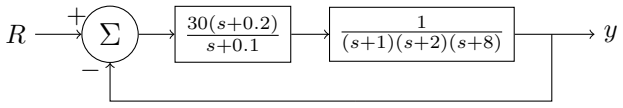


Example

Compare the steady-state error for a step input for the two systems:



Proportional



Lag Control

For a Type-0 system, $e_{ss} = \frac{1}{1+\lim_{s\to 0} G}$.

$$\text{Proportional: } e_{ss} = \frac{1}{1+\lim_{s\to 0} \frac{30}{(s+1)(s+2)(s+8)}} = \frac{1}{1+\frac{30}{16}} = 0.35$$

$$\text{Lag Control: } e_{ss} = \frac{1}{1+\lim_{s\to 0} \frac{30(s+0.2)}{(s+0.1)(s+1)(s+2)(s+8)}} = \frac{1}{1+\frac{6}{1.6}} = 0.21$$

So, lag control reduces the steady-state error (but does not bring it to zero).

Lead/Lag Control

We can combine lead control and lag control to get "lead/lag" control, similar to how we combined PD and PI control to get PID control.

Final Comments on Controller Design

- P, PI, PD, and PID controllers work well for many applications and are widely used.
- There are not the only types of controller possible.
- If you have a system where the 2nd order approximation is **not** valid, our design process won't work right.
- But, our knowledge of what closed-loop pole locations mean is still valid:
  - All poles in LHP $\rightarrow$ stable system
  - Poles on real axis $\rightarrow$ no overshoot
  - Closed-loop zero near closed-loop pole $\rightarrow$ cancellation
  - Poles farther left $\rightarrow$ faster transient behavior