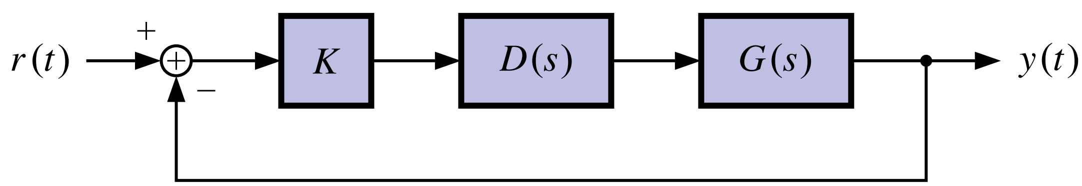
ROOT-LOCUS CONTROLLER DESIGN

7.1: Using root-locus ideas to design controller

- We have seen how to draw a root locus for given plant dynamics.
- We include a variable gain K in a unity-feedback configuration—we know this as proportional control.
- Sometimes, proportional control with a carefully chosen value of K is sufficient for the closed-loop system to meet specifications.
- But, what if the set of closed-loop pole location does not simultaneously satisfy the geometry that defines the specifications?
- We need to modify the locus itself by adding extra dynamics—a compensator or controller $D(s)$:



- We redraw the locus and pick K in order to put the poles where we want them. HOW?

$$T(s) = \frac{K D(s)G(s)}{1 + K D(s)G(s)}. \qquad \text{Now, let } \widetilde{G}(s) = D(s)G(s)$$

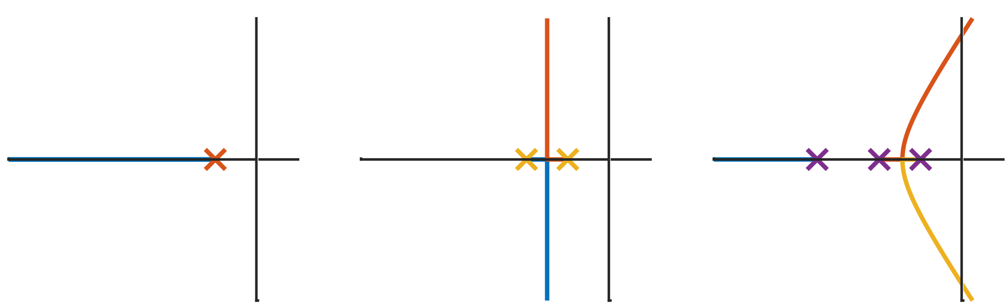
$$= \frac{K\widetilde{G}(s)}{1 + K\widetilde{G}(s)}$$

➠ We know how to draw this locus!

- Adding a compensator effectively adds dynamics to the plant.

Adding a left-half-plane pole or zero

- Two questions: (1) What types of compensation should we use, and (2) how do we figure out where to put the additional dynamics?
- In ECE4510/5510, the methods we discuss are "science-inspired art."
  - We need to get a "feel" for how the root locus changes when poles and zeros are added, to understand what dynamics to use for $D(s)$.
- In more advanced courses, we learn more powerful methods:
  - In ECE5520, we learn how to put all closed-loop poles exactly where we want them (but, where do we want them?)
  - In ECE5530, we learn how to find the optimal set of pole locations.
- But, for us to get started, speaking in generalities, adding a left-half-plane pole pulls the root locus to the right.
  - This tends to lower the system's relative stability and slow down the settling of the response.
  - But, providing that the closed-loop system is stable, the pole can also decrease steady-state errors.

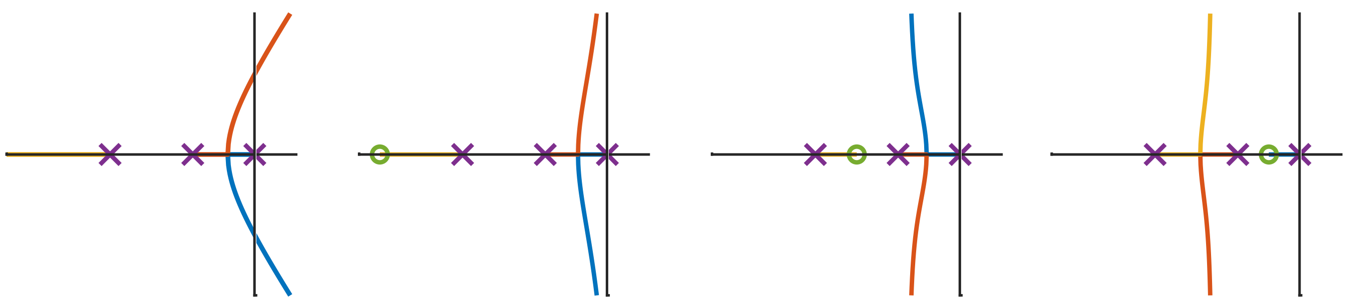


  - In first plot: The system is stable for all K, responses are smooth.
  - In second plot: System also stable for all K, but when poles become complex, response shows overshoot and oscillations.

- In third plot: The system is stable only for small K, and oscillations increase as the poles approach the imaginary axis.
- But, steady-state error improves from left to right (assuming the closed-loop system is stable).

■ Again, generally speaking, adding a left-half-plane zero pulls the root locus to the left.

- This tends to make the system more stable, and speed up the settling of the response.
- Physically, a zero adds derivative control to the system, introducing anticipation into the system, speeding up transient response.
- However, steady-state errors can get worse.



- In first plot: System is stable only for small K, and oscillates as poles approach imaginary axis.
- In second plot: System is stable for all K, but still oscillates.
- In third and fourth plots: More stable, less oscillation.
- But, steady-state error degrades from left to right.

■ Can't physically add a zero without a pole: Must put pole very far left in s-plane so we don't deteriorate desired impact of zero.

7.2: Reducing steady-state error

- We have a number of options available to us if we wish to reduce steady-state error.

1) Proportional feedback

$$D(s) = 1. \qquad u(t) = Ke(t)$$

$$T(s) = \frac{KG(s)}{1 + KG(s)}.$$

- Same as what we have already looked at.
- Controller consists of only a "gain knob."
  - Increasing gain K often reduces steady-state error, but can degrade transient response.
  - We have to take the locus "as given" since we have no extra dynamics to modify it.
  - Can't independently choose steady-state error and transient response. Can design for one or other, not both.
- Usually a very limited approach, but a good place to start.

2) Integral feedback

$$D(s) = \frac{1}{T_I s} \qquad u(t) = \frac{K}{T_I}\int_0^t e(\tau)\,\mathrm{d}\tau$$

$$T(s) = \frac{\frac{K}{T_I}\frac{G(s)}{s}}{1 + \frac{K}{T_I}\frac{G(s)}{s}}.$$

- Usually used to reduce/eliminate steady-state error. *i.e.,* if $e(t)$ constant, $u(t)$ will become very large and hopefully correct the error.
- Ideally, we would like no error, $e_{ss} = 0$. (Maybe 1 % to 2 % in reality)

ANALYSIS: For a unity-feedback control system, the steady-state error to a unit-step input is:

$$e_{ss} = \frac{1}{1 + KD(0)G(0)}.$$

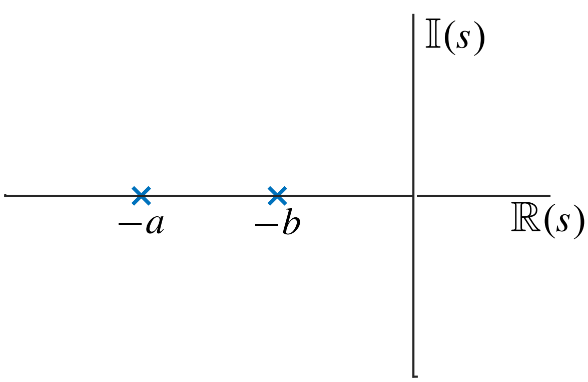
- If we make $D(s) = \dfrac{1}{T_I s}$, then as $s \to 0$, $D(s) \to \infty$

$$e_{ss} \to \frac{1}{1+\infty} = 0.$$

- Adding the integrator into the compensator has reduced error from $\dfrac{1}{1+K_p}$ to zero for systems that do not have any free integrators.
- Adding the integrator increases the system type, but as steady-state response improves, transient response often degrades.

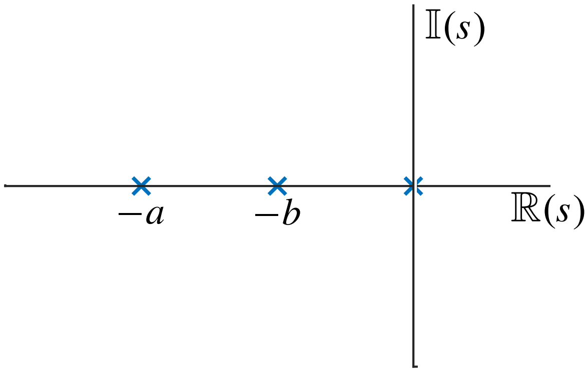
EXAMPLE: $G(s) = \dfrac{1}{(s+a)(s+b)}, \qquad a > b > 0.$

- Proportional feedback, $D(s) = 1$, $G(0) = \dfrac{1}{ab}$, $e_{ss} = \dfrac{1}{1 + \frac{K}{ab}}$.



- We can make e_{ss} small by making K very large, but this often leads to poorly-damped behavior and often requires excessively large actuators.

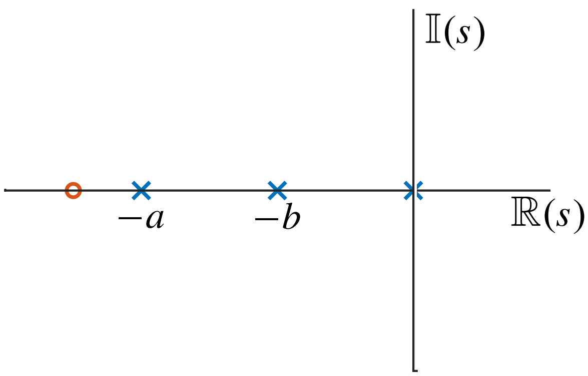
- Integral feedback, $D(s) = \dfrac{1}{T_I s}$, $e_{ss} = 0$.



- Increasing K to increase the speed of response pushes the pole toward the imaginary axis ➠ oscillatory.

3) Proportional-integral (PI) control

- Now, $D(s) = K\left[1 + \frac{1}{T_I s}\right] = K\left[\frac{s + (1/T_I)}{s}\right]$. Both a pole and a zero.



- Combination of proportional and integral (PI) solves many of the problems with just (I) integral.

4) Phase-lag control

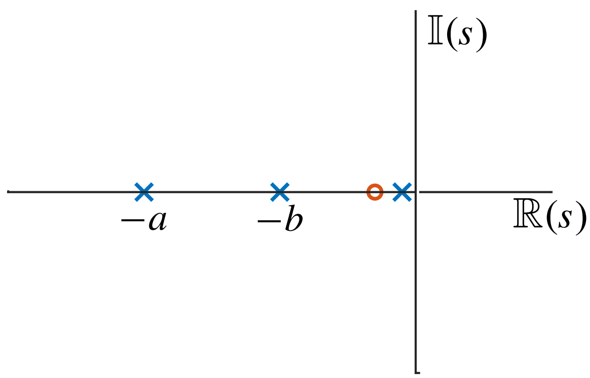
- The integrator in PI control can cause some practical problems; *e.g.*, "integrator windup" due to actuator saturation.
- PI control is often approximated by "lag control."

$$D(s) = \frac{(s - z_0)}{(s - p_0)}, \qquad |p_0| < |z_0|.$$

That is, the pole is closer to the origin than the zero.

- Because $|z_0| > |p_0|$, the phase ϕ added to the open-loop transfer function is negative... "phase lag"
- Pole often placed *very* close to the origin ($s = 0$). *e.g.*, $p_0 \approx 0.01$.

- Zero is placed near pole. *e.g.,* $z_0 \approx 0.1$. We want $|D(s)| \approx 1$ for all s to preserve transient response (and hence, have nearly the same root locus as for a proportional controller).
- Idea is to improve steady-state error but to modify the transient response as little as possible.
  - That is, using proportional control, we have pole locations we like already, but poor steady-state error.
  - So, we add a lag controller to minimally disturb the existing good pole locations, but improve steady-state error.



- Good steady-state error without overflow problems. Very similar to proportional control.

- The uncompensated system had loop gain $K_{\text{before}} = \lim_{s\to 0} G(s)$.
- The lag-compensated system has loop gain

$$K_{\text{after}} = \lim_{s\to 0} D(s)G(s) = (z_0/p_0)\lim_{s\to 0} G(s).$$

- Since $|z_0| > |p_0|$, there is an improvement in the position/velocity/etc. error constant of the system, and a reduction in steady-state error.
- Transient response is mostly unchanged, but slightly slower settling due to small-magnitude slow “tail” caused by lag compensator.

7.3: Improving transient response

- We have a number of options available to us if we wish to improve transient response

1) Proportional feedback

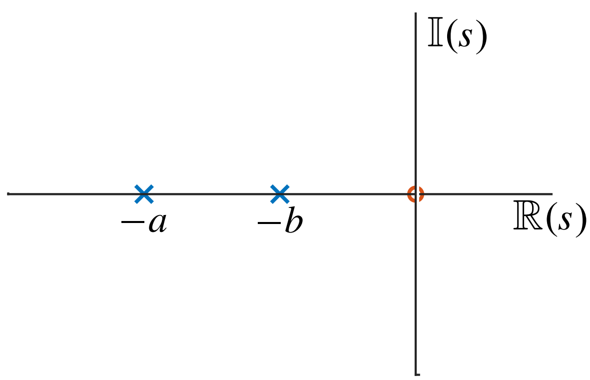
- Again, we could use a proportional feedback controller.
- It has the same benefits and limitations that we've already seen.

2) Derivative feedback

$$D(s) = T_D s, \qquad u(t) = K T_D \dot{e}(t).$$

- Does nothing to help the steady-state error. In fact, it can make it worse.
- But, derivative control provides feedback that is proportional to the rate-of-change of $e(t)$ ➠ control response *ANTICIPATES* future errors.
- Very beneficial—tends to smooth out response, reduce ringing.

EXAMPLE: $G(s) = \dfrac{1}{(s+a)(s+b)}, \qquad D(s) = T_D s.$

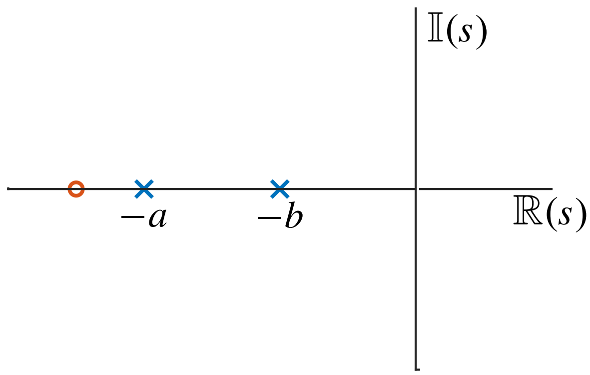


- No ringing. "Very" stable.

3) Proportional-derivative (PD) control

- Often, proportional control and derivative control go together.

$$D(s) = 1 + T_D s.$$



- No more zero at $s = 0$.
- Therefore better steady-state response.

4) Phase-lead control

- Derivative magnifies sensor noise.
- Instead of D-control or PD-control use "lead control."

$$D(s) = \frac{(s - z_0)}{(s - p_0)}, \qquad |z_0| < |p_0|.$$

That is, the zero is closer to the origin than the pole.

- Same form as lag control, but with different intent:
  - Lag control does not change locus much since $p_0 \approx z_0 \approx 0$. Instead, lag control improves steady-state error.
  - Lead control *DOES* change locus. Pole and zero locations chosen so that locus will pass through some desired point $s = s_1$.

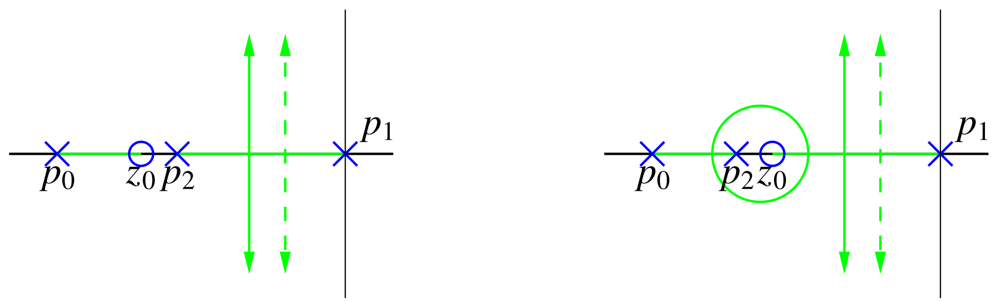
DESIGN METHOD I: Sometimes, we can be successful by choosing the value of z_0 to cancel a stable pole in the plant.

- Then, we solve for K and p_0 such that

$$[1 + KD(s)G(s)|_{s=s_1} = 0.$$

- That is, we force one closed-loop pole to be at $s = s_1$.
- This does not ensure that other poles do anything reasonable, so we must always test design.

- And, what about pole-zero cancelation? Can it occur?



- Either way, the locus is still okay. (What if we tried to cancel an unstable pole?)

DESIGN METHOD II: If there is no stable real pole to cancel, we can still use similar approach.

- Use somewhat modified version of lead compensator form

$$D(s) = \frac{a_1 s + a_0}{b_1 s + 1}.$$

- Choose a_0 to get specified dc gain (*e.g.*, open-loop gain=K_p, K_v, ...)

$$\left| \left[\frac{a_1 s + a_0}{b_1 s + 1} \right] G(s) \right|_{s=0} = \text{dc gain.}$$

$$|a_0||G(0)| = \text{dc gain.}$$

$$a_0 = \frac{\text{Desired dc gain}}{|G(0)|}.$$

- a_1 and b_1 are chosen to make locus go through $s = s_1$,

$$\left[\frac{a_1 s_1 + a_0}{b_1 s_1 + 1} \right] G(s_1) = -1$$

for that point to be on the root locus.

➠ Magnitude $\left|\dfrac{a_1 s_1 + a_0}{b_1 s_1 + 1}\right| |G(s_1)| = 1$

➠ Phase $\angle\left[\dfrac{a_1 s_1 + a_0}{b_1 s_1 + 1}\right] + \angle G(s_1) = 180°$.

(math happens)

$$\left.\begin{aligned} a_1 &= \frac{\sin(\beta) + a_0|G(s_1)|\sin(\beta-\psi)}{|s_1||G(s_1)|\sin(\psi)} \\ b_1 &= \frac{\sin(\beta+\psi) + a_0|G(s_1)|\sin(\beta)}{-|s_1|\sin(\psi)} \end{aligned}\right\} \quad \begin{aligned} s_1 &= |s_1|e^{j\beta} \\ G(s_1) &= |G(s_1)|e^{j\psi}. \end{aligned}$$

5) Proportional-integral-derivative (PID) control

- There is a similar design procedure for PID control:

$$D(s) = K\left[1 + \frac{1}{T_I s} + T_D s\right] = K_p + \frac{K_I}{s} + K_d s.$$

- Compute: $K_p = \dfrac{-\sin(\beta+\psi)}{|G(s_1)|\sin(\beta)} - \dfrac{2K_I\cos\beta}{|s_1|}$
- Compute: $K_d = \dfrac{\sin(\psi)}{|s_1||G(s_1)|\sin(\beta)} + \dfrac{K_I}{|s_1|^2}$, where $s_1 = |s_1|e^{j\beta}$ and $G(s_1) = |G(s_1)|e^{j\psi}$ for both cases.
- T_I chosen to match some design criteria. *e.g.*, steady-state error.
- Convert to first form via $K = K_p$; $T_I = K/K_I$; $T_D = K_d/K$.

6) Lead-lag control

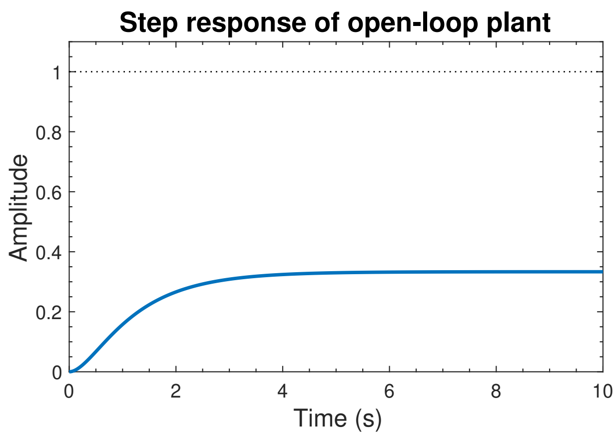
- If we must satisfy both a transient and steady-state spec:

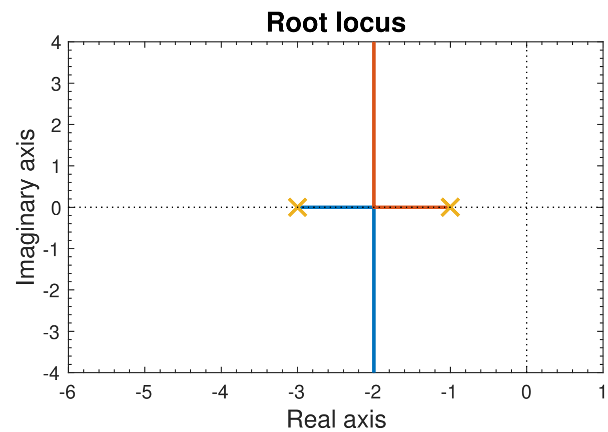
1. Design a lead controller to meet transient spec first;
2. Include lead controller with plant after its design is final;
3. Design a lag controller (where “plant” = actual plant and lead controller combined) to meet steady-state spec.

7.4: Examples (a)

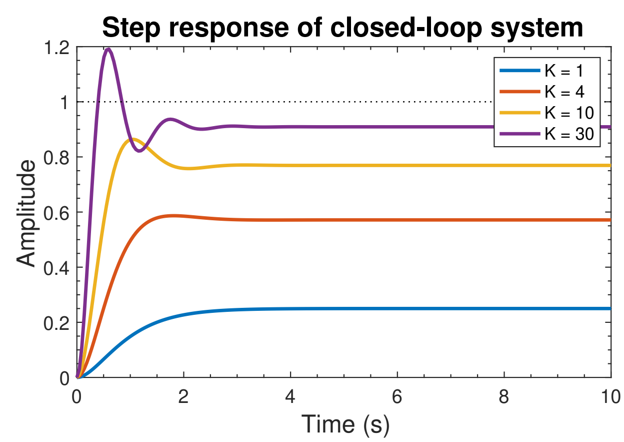
EXAMPLE I: We start with the plant $G(s) = \dfrac{1}{(s+1)(s+3)}$.

- The open-loop step response for $G(s)$ is plotted to the left.
- The root locus (assuming proportional control) is plotted to the right.





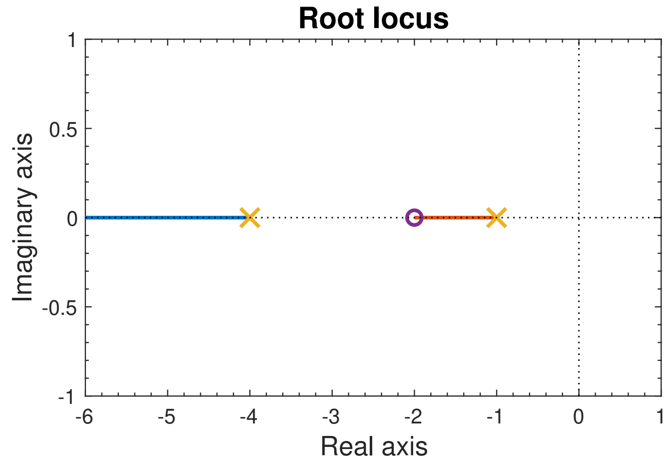
- We see that the open-loop response is smooth (good), slow (bad), and has very large steady-state error (bad).
- But, root locus shows that proportional control moves pole locations.
- The plot to the right shows step responses of closed-loop systems with proportional control.
- Changing K "shapes" the transient response.



- Higher values of K speed up the closed-loop response when compared to the open-loop response (good), decrease steady-state error (good), but also add ringing to the transient response (bad).

EXAMPLE II: We start with the plant $G(s) = \dfrac{s+2}{(s+1)(s+4)}$.

- Using proportional control, we wish to solve for the value of K that places a closed-loop pole at $s = -5$.



- First, we draw the locus to ensure that it does pass through $s = -5$.
- It does! Looking good so far.
- Next, we remember that the root-locus "magnitude condition" gives us

$$\begin{aligned} K &= \left. \frac{1}{|G(s)|} \right|_{s=-5} \\ &= \left| \frac{(s+1)(s+4)}{s+2} \right|_{s=-5} \\ &= \left| \frac{(-4)(-1)}{(-3)} \right| \\ &= \frac{4}{3}. \end{aligned}$$

- We're done, but we can further double-check that $s = -5$ is a point on the root locus using the "angle condition"

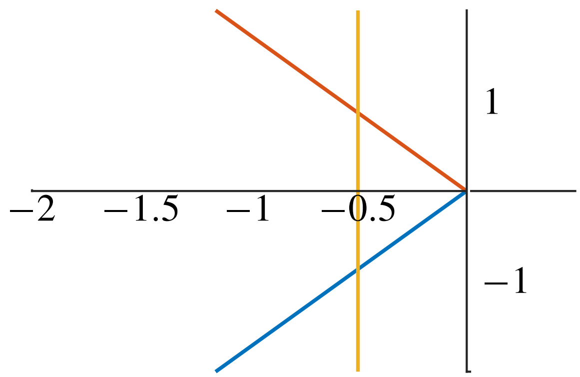
$$\begin{aligned} [\angle G(s)|_{s=-5} &= [\angle(s+2) - \angle(s+1) - \angle(s+4)|_{s=-5} \\ &= 180^\circ - 180^\circ - 180^\circ = -180^\circ. \end{aligned}$$

- So, the angle condition is satisfied as well (meaning we didn't have to draw the root locus to ensure that $s = -5$ was a valid locus point).

EXAMPLE III: We start with the plant

$$G(s) = \frac{1}{s(10s+1)}.$$

- Our goal is to have closed-loop



1. $M_p < 16\%$. This means that $\zeta \geq 0.5$.
2. $t_s < 10$ secs to 1%. This means that $\sigma \geq 0.46$.
3. e_{ss} for ramp input < 0.01 when slope of ramp $= 0.01$. This means that $K_v = 0.01/0.01 = 1.0$.

- Since we need to change transient response, we choose to use a lead controller.

- Since the plant has a stable real pole, we choose $D(s)$ to approximately cancel plant pole.

$$D(s) = \frac{10s+1}{s+p_0}.$$

- Initially, choose $s_1 = -0.5 + j$ to be a point on the locus. So, we want

$$\left[1 + K\left(\frac{10s+1}{s+p_0}\right)\left(\frac{1}{s(10s+1)}\right)\right|_{s=s_1} = 0$$

and

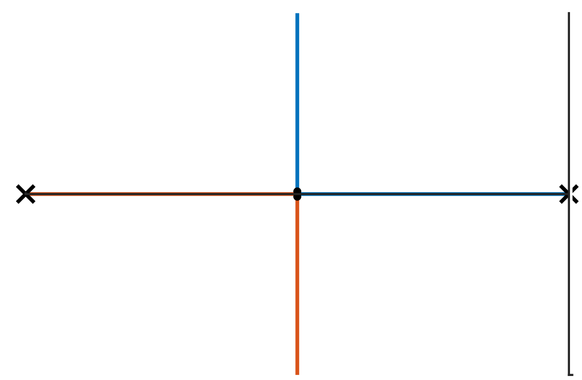
$$\lim_{s\to 0} s\left[K\left(\frac{10s+1}{s+p_0}\right)\left(\frac{1}{s(10s+1)}\right)\right] \geq 1.$$

- The steady-state error spec gives $K \geq p_0$. For simplicity, choose $K = p_0$.

- The transient spec gives

$$\left[1 + p_0\left(\frac{1}{s(s+p_0)}\right)\right|_{s=s_1} = 0$$
$$s_1(s_1+p_0)+p_0 = 0$$
$$s_1^2 + s_1p_0 + p_0 = 0$$
$$p_0(1+s_1) = -s_1^2$$
$$p_0 = -\frac{s_1^2}{1+s_1}.$$

- Solving gives $p_0 = 1.1 - 0.2j$. This is not a feasible design since p_0 must be real.
- Modify p_0 to $p_0 = 1.1$. This gives $K = 1.1$, $K_v = 1$, and poles at $-0.55 \pm 0.893j$.
- This gives $\omega_n \approx 1$ for pole locations, so $t_r \approx 1.8$ s.



- Could choose slightly larger K, still achieve transient-response specs, but have better steady-state response since $K \geq p_0$.

7.5: Examples (b)

EXAMPLE IV: Consider the plant $G(s) = \dfrac{1}{s^2}$.

- We want to design a compensator

$$D(s) = \frac{a_1 s + a_0}{b_1 s + 1}$$

  so the closed-loop system has a pole at $s_1 = 2\sqrt{2}e^{j135^\circ} = -2 + 2j$. (The point s_1 is chosen to achieve $\zeta = 0.707$ and $\tau = 0.5$ s.)

- Here, there is no stable real pole in $G(s)$, so we use the second design method for a lead compensator.

- Step 1, compute a_0: We cannot compute a_0 since $\left.\dfrac{1}{s^2}\right|_{s=0} \to \infty$. So, *arbitrarily* choose $a_0 = 2$.

- Step 2, compute a_1: Note, $\beta = 135^\circ$, $\psi = -270^\circ$ because

$$G(s_1) = \left.\frac{1}{s^2}\right|_{s=2\sqrt{2}e^{j135^\circ}} = \frac{1}{8}e^{-j270^\circ}.$$

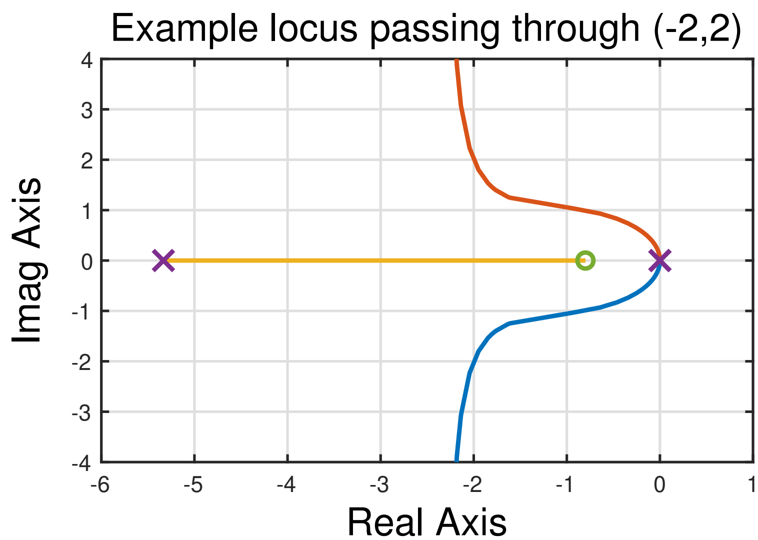
$$a_1 = \frac{\sin(135^\circ) + 2(1/8)\sin(45^\circ)}{(2\sqrt{2})(1/8)\sin(-270^\circ)} = \frac{(1/\sqrt{2})(1 + 1/4)}{\sqrt{2}/4} = \frac{5}{2}.$$

- Step 3, compute b_1:

$$b_1 = \frac{\sin(-135^\circ) + 2(1/8)\sin(135^\circ)}{-(2\sqrt{2})\sin(-270^\circ)} = \frac{-(1/\sqrt{2})(1 - 1/4)}{-2\sqrt{2}} = \frac{3}{16}.$$

- So, the compensator is:

$$D(s) = \frac{(5/2)s + 2}{(3/16)s + 1}.$$



EXAMPLE V: An alternative way to solve the prior problem uses coefficient matching.

- We have that $G(s) = \dfrac{1}{s^2}$, and have assumed that $D(s) = \dfrac{a_1 s + 2}{b_1 s + 1}$.
- We want two closed-loop poles at $s = -2 \pm 2j$, but recognize that there will be a total of three closed-loop poles (because of the added compensator pole).
- So, we can specify a ***desired*** characteristic equation

$$\begin{aligned}\chi_d(s) &= (s+\alpha)(s+2+2j)(s+2-2j)\\ &= (s+\alpha)(s^2+4s+8)\\ &= s^3 + (4+\alpha)s^2 + (8+4\alpha)s + 8\alpha = 0,\end{aligned}$$

where $s = -\alpha$ is the (unknown *a priori*) location of the third pole.

- The ***actual*** characteristic equation is

$$\begin{aligned}\chi_a(s) &= 1 + D(s)G(s) = 0\\ &= 1 + \left(\frac{a_1 s + 2}{b_1 s + 1}\right)\left(\frac{1}{s^2}\right)\\ &= b_1 s^3 + s^2 + a_1 s + 2 = 0.\end{aligned}$$

- The coefficient-matching method forces the polynomial coefficients of the desired and actual characteristic equations to be the same.
- Looking at the s^3 coefficients, we could set $b_1 = 1$, but then we would have problems because we cannot simultaneously have

$$4 + \alpha = 1 \quad \text{and} \quad 8\alpha = 2.$$

- So, we divide $\chi_a(s)$ by b_1, without changing its meaning:

$$\chi_a(s) = s^3 + \frac{1}{b_1}s^2 + \frac{a_1}{b_1}s + \frac{2}{b_1} = 0.$$

- This has given us another degree of freedom when solving. Now, we have

$$4 + \alpha = \frac{1}{b_1}, \quad 8 + 4\alpha = \frac{a_1}{b_1} \quad \text{and} \quad 8\alpha = \frac{2}{b_1}.$$

- Combining the first and third equations gives

$$\begin{aligned} 2(4 + \alpha) &= 8\alpha \\ 8 &= 6\alpha \\ \alpha &= \frac{4}{3}. \end{aligned}$$

- With this value of α, we have $b_1 = 3/16$ and $a_1 = 5/2$, as before.

EXAMPLE VI: Consider the compensated system of Example III.

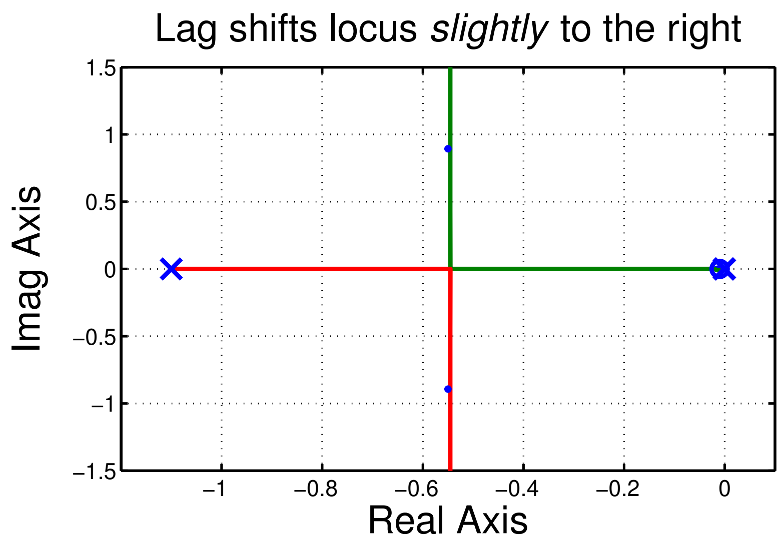
$$G(s) = \frac{1.1}{s(s + 1.1)}.$$

- We like the transient response (so want to leave it alone), but wish to improve the steady-state response by a factor of 10.
- This calls for a lag controller. Recall that

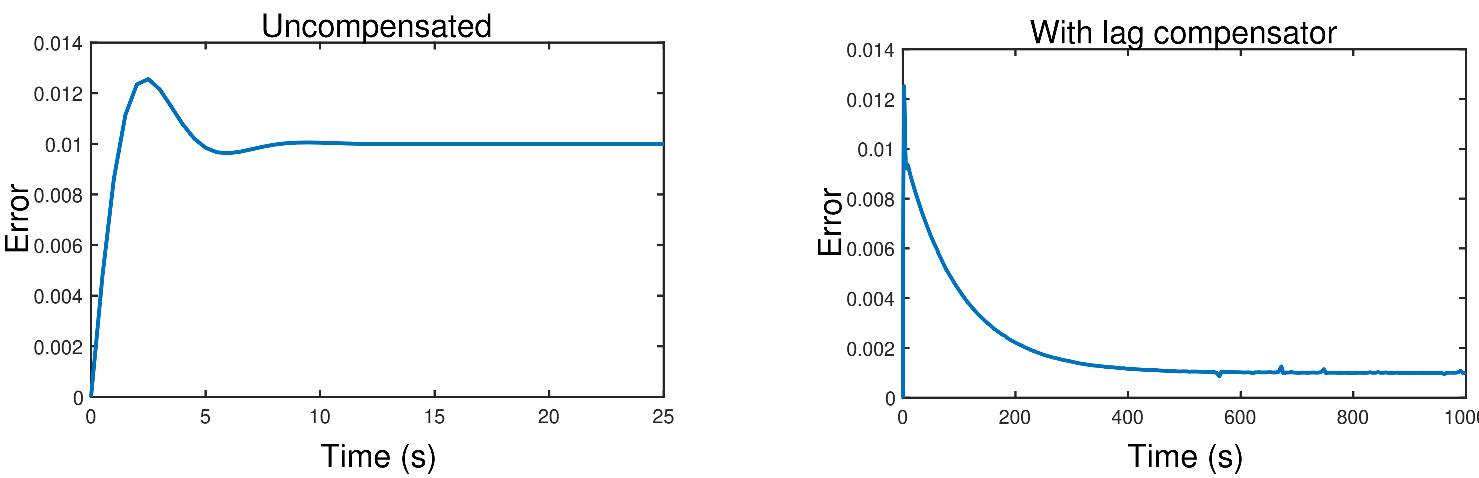
$$K_{\text{after}} = (z_0/p_0)\, K_{\text{before}},$$

so, we want $z_0/p_0 \ge 10$.

- Choose $p_0 = 0.001$. Then, $z_0 = 0.01$ and $D(z) = \dfrac{s + 0.01}{s + 0.001}$.



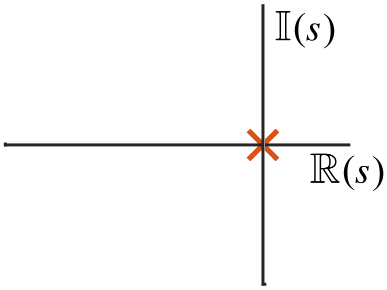
- Plots of error versus time without and with the new lag compensator (simulated using Simulink):



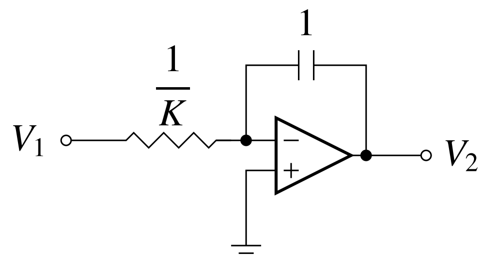
- Notice the different time scales: The lag adds a small-amplitude slow time constant to the output.

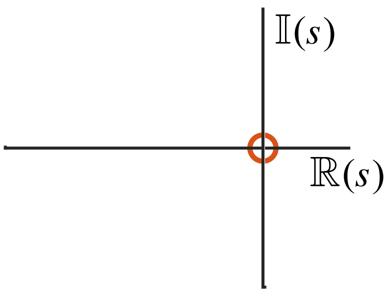
7.6: Compensator implementation

- Analog compensators commonly use op-amp circuits.
- See the following pages. . .

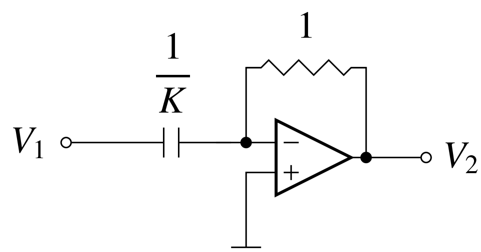


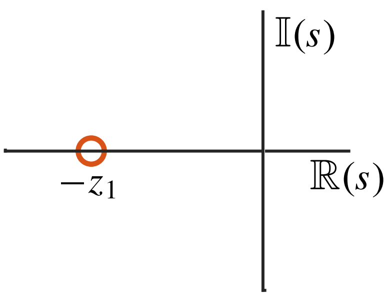
$$-\frac{K}{s}$$



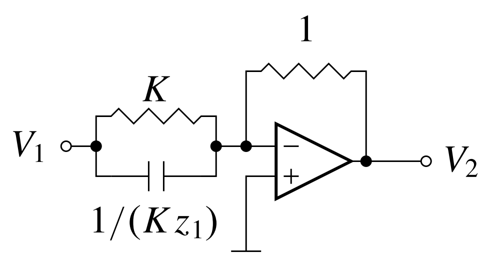


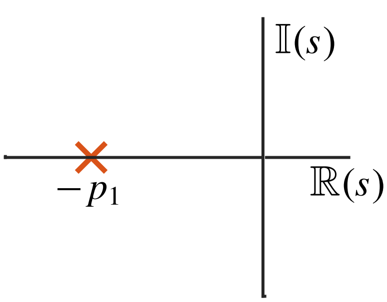
$$-Ks$$



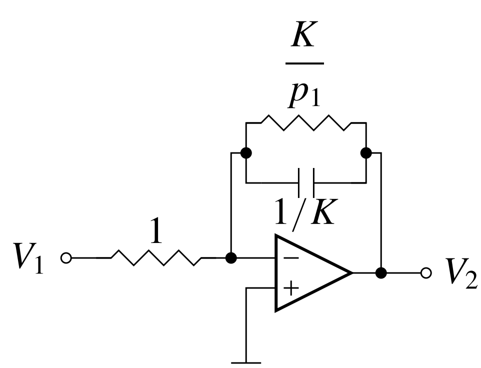


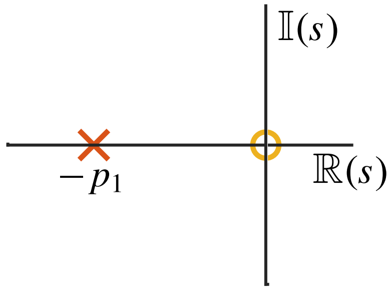
$$-K(s+z_1)$$



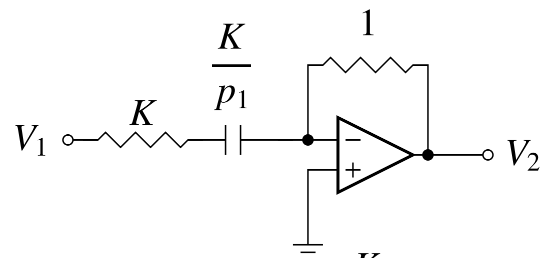


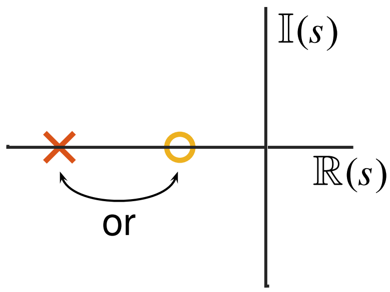
$$\frac{-K}{s+p_1}$$





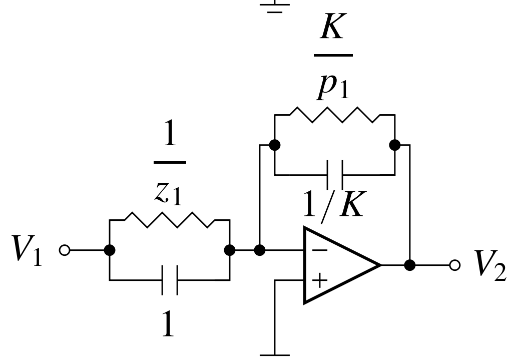
$$\frac{-Ks}{s+p_1}$$

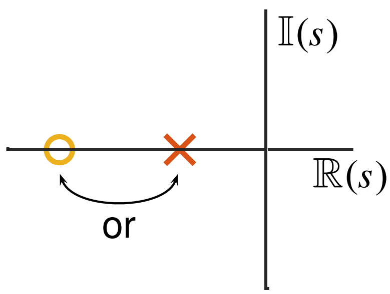




$$-K\frac{s+z_1}{s+p_1}$$

any p_1 and z_1

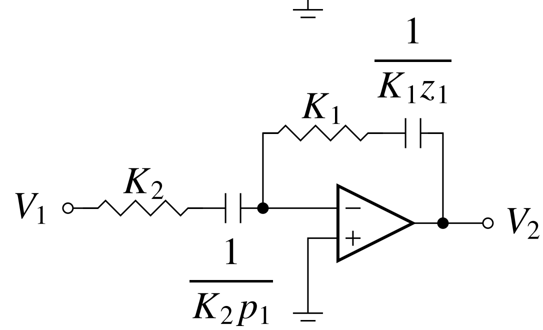


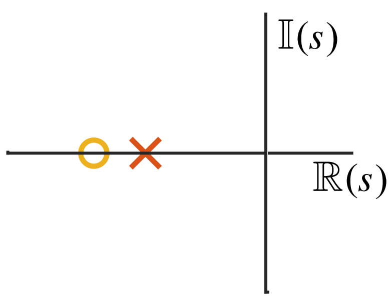


$$-K\frac{s+z_1}{s+p_1}$$

$$K=\frac{K_1}{K_2}$$

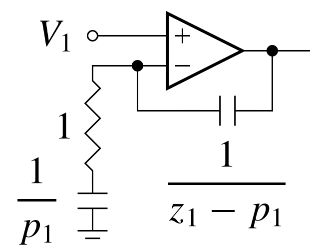
any p_1 and z_1



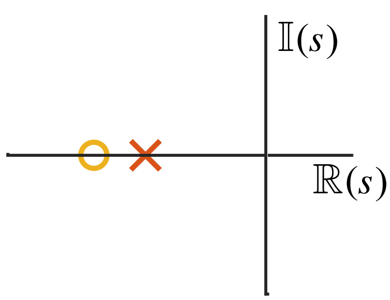


$$\frac{s+z_1}{s+p_1}$$

$z_1 > p_1$

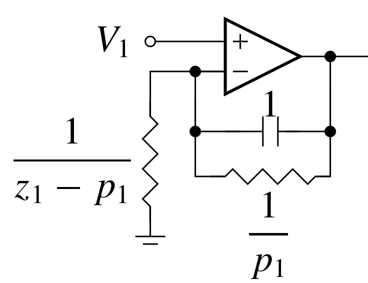


LAG

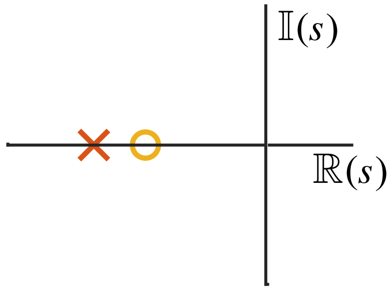


$$\frac{s+z_1}{s+p_1}$$

$z_1 > p_1$



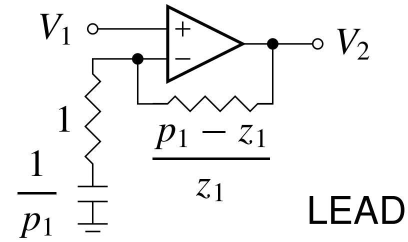
LAG

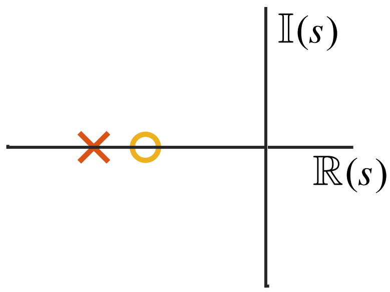


$$K\frac{s+z_1}{s+p_1}$$

$$p_1 > z_1$$

$$K = \frac{p_1}{z_1}$$

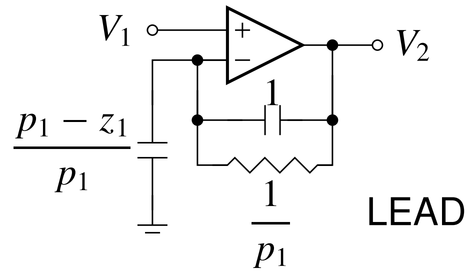


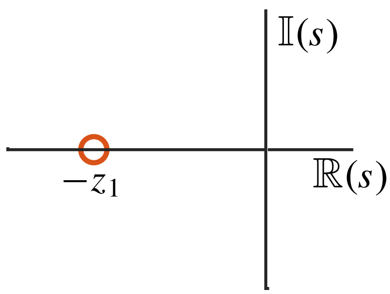


$$K\frac{s+z_1}{s+p_1}$$

$$p_1 > z_1$$

$$K = \frac{p_1}{z_1}$$





$$K(s+z_1)$$

$$K = \frac{1}{z_1}$$