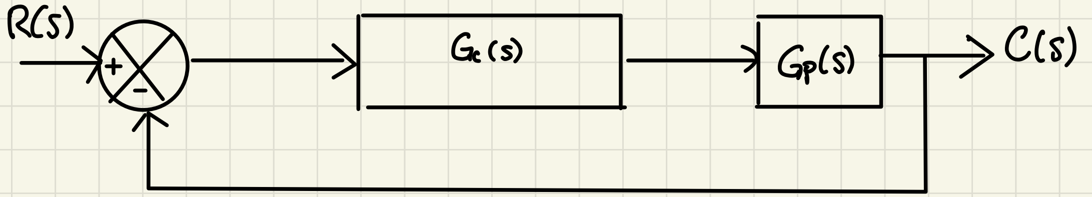
BACK TO USING ROOT LOCUS TO DESIGN CONTROL SYSTEMS: LEAD-LAG CONTROLLERS

Up until now, we have looked at controllers of this form:



Controllers in the feedforward path are called Cascade Compensators and PID controllers (and its variants) are one type.

Another common category of cascade compensators are called lead-lag compensators

Lag compensators: $G_c(s) = K\frac{(s+z_c)}{(s+p_c)}$

- ⇒ adds a pole and a zero
- ⇒ the pole at $-p_c$ is small
- ⇒ z_c is close to, but at the left of p_c ($|z_c|>|p_c|$)
- ⇒ Acts to improve (but not eliminate) SSE

Lead compensators: $G_c(s) = K\frac{(s+z_c)}{(s+p_c)}$

- ⇒ adds a pole and a zero
- ⇒ pole is more negative than zero ($|p_c|>|z_c|$)
- ⇒ works to improve transient response
  - → shifts centroid of RL further into LHP

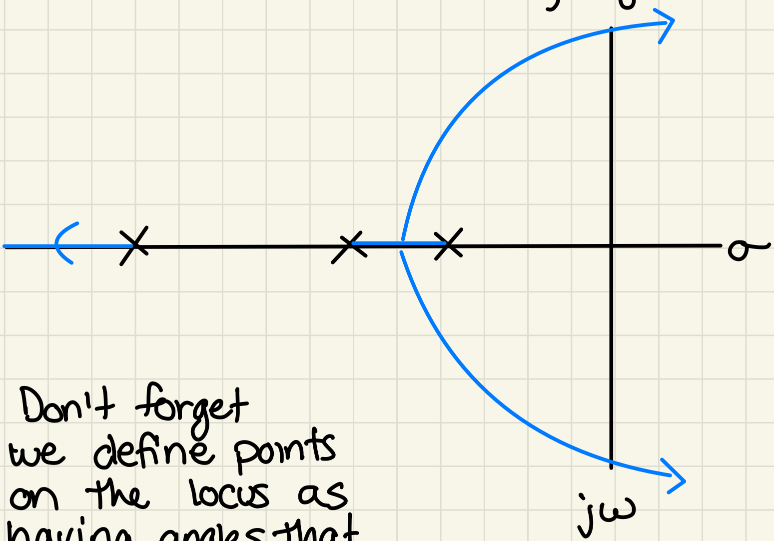
Lead-lag compensator: $G_c(s) = K\dfrac{(s+z_{lag})(s+z_{lead})}{(s+p_{lag})(s+z_{lead})}$

⇒ add 2 poles & 2 zeros
⇒ lag to help w/ SSE
⇒ lead to help w/ transient response
⇒ $-p_{lag}$ is small, with z_{lag} close by and to to the left of p_{lag}
⇒ p_{lead} is more negative than z_{lead}

We already saw how powerful adding poles and zeros can be in affecting our response, so let's explore the impact of doing this according to the structure of lead-lag compensators

Lag compensator and the root locus

Consider the following system

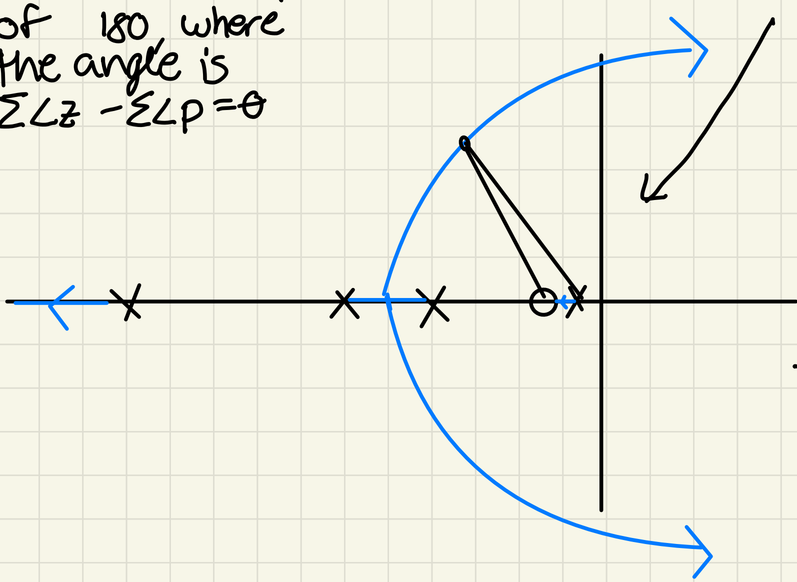


what happens if we add a lag compensator?

consider the ∠ criteria

if a zero and a pole are added close together, their angles to some point p are almost the same so $\angle z_c - \angle p_c \approx 0$

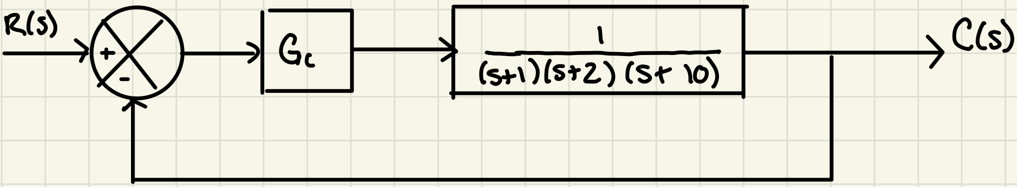
Don't forget we define points on the locus as having angles that are odd multiples of 180 where the angle is $\sum\angle z - \sum\angle p = \theta$



the new pole and zero do very little to change the shape and location of the locus, but the added pole near 0 improves the SSE

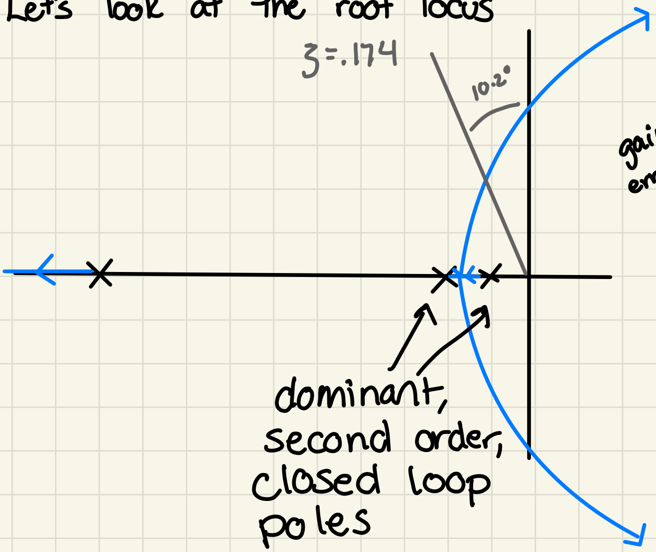
Let's consider how we might design one using an example

We have the system below, which we will subject to a unit step input.



First, let's look at the behavior of the system when compensated with only a gain K_p, which is adjusted such that the system is operating at $\zeta = .174$ to meet a design specification

Let's look at the root locus



We can use Matlab to help us find K at the intersection of the $\zeta = .174$ line

gain, not error const → $K = 164.6$

with closed loop poles at this point $s_{1,2} = -.694 \pm 3.9j$ and on the third branch $s_3 = -11.61$

$$e_{\infty step} = \frac{1}{1 + K_p}$$ ← error const, not gain!

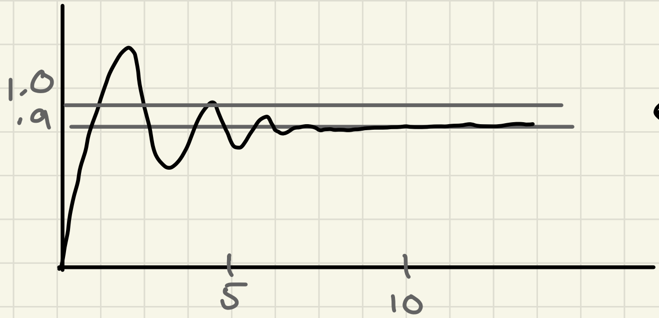
$$K_p = \lim_{s \to 0} G(s) = \lim_{s \to 0} \left(\frac{164.6}{(s+1)(s+2)(s+10)} \right)$$

$$= 8.23$$

So the

$$e_{\infty step} = 0.108$$

We can also use Matlab to plot our response in the time domain



It looks something like ← this

Now let's assume we want to use a lag compensator to improve the SSE by a factor of 10 when the system operates at $\zeta = .174$

Improving $e(\infty)$ tenfold means

$$e(\infty) = \frac{.108}{10} = .0108 = \frac{1}{1+K_p} \quad \text{so } K_p = \frac{1-.0108}{.0108} = 91.59$$

this is our SSE design goal

so how does the new pole and zero from the lead lag compensator affect K_p?

consider a general form for a transfer function:

$$G(s) = K\frac{(s+z_1)(s+z_2)\dots}{(s+p_1)(s+p_2)\dots} \qquad K_p = K\frac{(z_1)(z_2)(\dots)}{(p_1)(p_2)(\dots)}$$

a lag compensator adds a z_c and p_c, so the new $K_p = \frac{K z_1 z_2 \dots (z_c)}{p_1 p_2 \dots (p_c)}$

so $\frac{z_c}{p_c} = \frac{\text{desired } K_p}{\text{original } K_p}$

for our problem, $\frac{z_c}{p_c} = \frac{K_{p\,desired}}{K_{p\,original}} = \frac{91.59}{8.23} = 11.13$

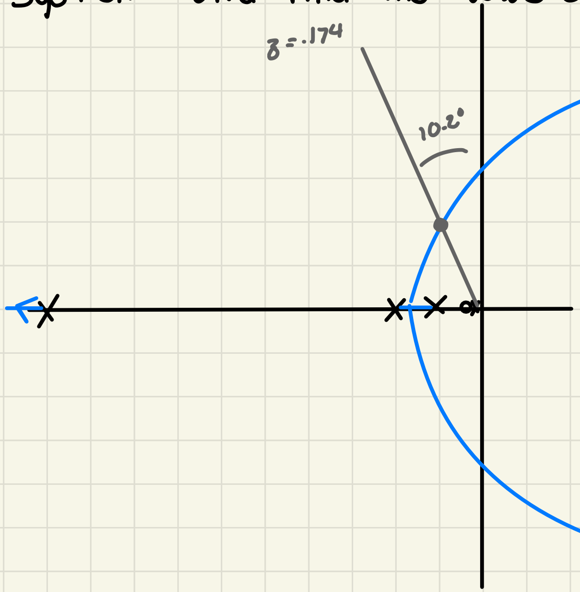
for a lag compensator, we choose p_c to be a value close to 0, so let's choose $p_c = 0.01$

Then we can find the $z_c = 11.13 p_c = .111$

Now we have a compensator design

$$G_c = \frac{K(s+0.111)}{(s+.01)}$$

now let's make our root locus for our compensated system and find the value of K for $\zeta = .174$



Intersection occurs at $K = 158.1$

$s_{1,2} = -.678 \pm 3.84j$

We have barely moved the dominant R2 branches

BUT look at the transient response

and one more thing - we designed for our K_p goal based on our inital value of $K = 164.6$. However, after adding the lag compensator, we shifted to $K = 158.1$. So let's calculate our actual SSE at $\zeta = .174$ for our new design.

$$K_p = \lim_{s \to 0} K G_c G_p = \frac{158.1\,(.111)}{(.01)(1)(2)(10)} = 87.75$$

$$e_\infty = \frac{1}{1 + K_p} = .011$$ ← very close to our target

Let's look at this in Matlab

lagcompensator.m