Root Locus Design Example

\[ G(s) := \frac{1643}{s(s + 16.64)(s + 53.78)} \]

\[ \sigma = \frac{-0 - 16.64 - 53.78}{3} = -23.473 \]

PI Zero steady-state error (for constant inputs) & perfect rejection of constant disturbances

\[ G(s) := \frac{1643}{s(s + 16.64)(s + 53.78)} \frac{s + 0.1}{s} \]

Add pole at 0 and zero at -0.1

\[ C(s) = k_p + \frac{k_i}{s} = k_p \frac{s + k_i}{s} \]

LAG An alternative is a Lag Compensator, here with a pole at -0.1 and a zero at -0.5

\[ G(s) = \frac{1643}{s(s + 16.64)(s + 53.78)} \frac{s + 0.5}{s + 0.1} \]

This works very much like the PI controller, but without the need for active components.
Let's keep the pole at 0 and zero at -0.1 for elimination of steady-state errors and rejection of disturbances.

CL poles at
\[ p = -7.06 + 7.06j \quad \text{and} \quad -7.06 - 7.06j \]

At gain of 3.44:
\[ \frac{\text{Im}(p)}{\text{Re}(p) + 53.78} = 8.593 \text{ deg} \]
\[ \frac{\text{Im}(p)}{\text{Re}(p) + 16.64} = 36.388 \text{ deg} \]

This is a point in the root locus because:
\[-8.6 \text{ deg} - 36.4 \text{ deg} - 135 \text{ deg} - 135 \text{ deg} + 135 \text{ deg} = -180 \text{ deg} \]

PD or PID Improve the dynamic response

Want to double the speed. Want poles to move to:
\[ p = -14 + 14j \quad -14 - 14j \]

Unfortunately, this point is NOT on the root locus:
\[ -\frac{\text{Im}(p)}{\text{Re}(p) + 53.78} - \frac{\text{Im}(p)}{\text{Re}(p) + 16.64} = 135 \text{ deg} = -233.71 \text{ deg} \]

Maybe we could add a zero so that its angle is:
\[ \theta_z = 233.71 \text{ deg} - 180 \text{ deg} \quad \theta_z = 53.71 \text{ deg} \]

\[ x = \text{Im}(p) \cdot \frac{1}{\tan(\theta_z)} = 10.28 \]
\[ z := \text{Re}(p) - \text{Im}(p) \cdot \frac{1}{\tan(\theta_z)} \]
\[ z = -24.28 \]

\[ G(s) := \frac{1643}{s(s + 16.64)(s + 53.78)} (s + 0.1)(s + 24.28) \]
\[ k = \frac{1}{|G(-14 + 14j)|} = 0.418 \]
Root Locus Design Example  p.3

We have designed our compensation with the following:

A pole at the origin
A zero at -0.1
A zero at -24.28
Gain of 0.418

Find the $k_p$, $k_i$, & $k_d$ of a PID controller.

\[
C(s) = \frac{k_p + \frac{k_i}{s} + s\cdot k_d}{s} = \frac{s^2\cdot k_p + k_i}{s^2 + \frac{s\cdot k_p}{k_d} + \frac{k_i}{k_d}}
\]

\[
= \frac{s\cdot k_p + k_i + s^2\cdot k_d}{s} = k_d\frac{s^2\cdot k_p + k_i}{s^2 + \frac{s\cdot k_p}{k_d} + \frac{k_i}{k_d}}
\]

\[
\text{gain} = k_d = 0.418
\]

\[
(s + 0.1)\cdot(s + 24.28) = s^2 + 24.38\cdot s + 2.48
\]

\[
= \frac{s^2 + \frac{s\cdot k_p}{k_d} + \frac{k_i}{k_d}}{s\cdot k_p + k_i + s^2\cdot k_d}
\]

\[
= \frac{k_i}{k_d} = 2.48 \quad k_i = k_d\cdot 2.48 \quad k_i = 1.037
\]

\[
\frac{k_p}{k_d} = 24.38 \quad k_p = k_d\cdot 24.38 \quad k_p = 10.191
\]

Notice: $\frac{k_i}{k_p} = 0.102 \approx 0.1$

**LEAD** An alternative to the differentiator is a Lead Compensator.

Instead of a single zero with: $\theta_z = 53.71^{\circ}$

How about a zero with $\theta_z = 70^{\circ}$ And a pole with $\theta_p = 70^{\circ} - 53.71^{\circ}$

\[
x = \text{Im}(p)\cdot \frac{1}{\tan(\theta_p)} = 5.096
\]

\[
z := \text{Re}(p) - \text{Im}(p)\cdot \frac{1}{\tan(\theta_z)} = -19.096
\]

\[
x_p = \text{Im}(p)\cdot \frac{1}{\tan(\theta_p)} = 47.907
\]

\[
p := \text{Re}(p) - \text{Im}(p)\cdot \frac{1}{\tan(\theta_p)} = -61.907
\]

This example is actually a PI-Lead controller.
Problems with the differentiator

1. Tries to differentiate a step input into an impulse -- not likely.
   You’ll have to consider how your differentiator will actually handle a step input and how your amplifier will saturate.

   If the differentiator and amplifiers saturate in such a way the the "area under the curve" approximates the impulse "area under the curve", then this may not be such a problem. It may not be as fast as predicted from the linear model, but it may be as fast as the system limits allow. (Pedal-to-the-metal.)

2. It’s a high-pass filter and can accentuate noise.
   This is actually common to all compensators that speed up the response.

3. Requires active components and a power supply to build.
   Usually no big deal since you amplifier (Source of gain) does too.

4. Is never perfect (always has higher-order poles), but then neither is anything else. Especially in mechanical systems, these poles usually are well beyond where they could cause problems.

Alternatives:

1. Lag-Lead or PI-Lead compensation. This eliminates the differentiator, but it is still a high-pass filter that can be a noise problem and it could still saturate the amplifier if the input changes too rapidly.
   Be sure to check for saturation problems.

2. Place the differentiator in the feedback loop. The output is much less likely to be a step or to change so rapidly that it causes problems.

Differentiation in the feedback

\[
\begin{align*}
F(s) \cdot C(s) &= \left( k_p \frac{k_i}{s} + 1 + k_d s \right) = k_p \cdot k_d \left( \frac{s + k_i}{s} \right) \left( \frac{1}{s + k_p} \right) \\
&= k_p \cdot k_d \left( \frac{s + k_i}{s} \right) \left( \frac{1}{s + k_p} \right) \\
&= k_p \cdot k_d \left( \frac{s + k_i}{s} \right) \left( \frac{1}{s + 24.28} \right) \\
&= k_d \frac{(s + 0.1) \cdot (s + 24.28)}{s} \\
&= k_d \frac{1}{24.38} \\
k_d &= 0.041 \\
k_p &= 10.191 \\
k_i &= k_p \cdot 0.1 \\
k_i &= 1.019
\end{align*}
\]

In this case the open-loop zero in the feedback loop IS NOT in the closed-loop. This turns out to make the step response slower than predicted by the second-order approximation, but try a simulation, you may be able to use significantly more gain with no more overshoot. The differentiator in this position inhibits overshoot.