In this Lecture, you will learn:

**Lead-Lag Compensation**
- Designing Leads
- Designing Lags
- Combining Leads and Lags

**Notch Filters**
- Providing extra zeros
- Eliminates annoying frequency components.
Recall: Pole-Zero Compensation

**Definition 1.**

A Pole-Zero Compensator is of the form

\[ K(s) = \frac{s + z}{s + p} \]

**Lead Compensation**

- \( p < z \)
- Replaces Pure Zero

**Lag Compensation**

- \( z < p \)
- Replaces Integrator
Lead Compensation

Example

\[ G(s) = \frac{1}{s(s+1)} \]

**Asymptotes:** ±90°

**Intercept:** \( \alpha = -0.5 \)

\[ \alpha = \frac{\sum p_i - \sum z_i}{n - m} = \frac{-1 - 0}{2 - 0} = -0.5 \]

**Break Point:** \( s = -0.5 \)

\[ n'd - d'n = 2s + 1 = 0 \]

**Conclusion:** At high gain, we get
- High Frequency Oscillation
- Lots of overshoot
- Fixed Settling Time
Lead Compensation

Example

Asymptotes: $\pm 90^\circ$

Intercept: $\alpha = -1$

\[
\alpha = \frac{\sum p_i - \sum z_i}{n - m}
\]

\[
= \frac{-5 + 4}{2 - 0} + \frac{-1 - 0}{2 - 0} = -0.5 - 0.5 = -1
\]

Break Point: $s = -0.508$

\[
n'd' - n'd
\]

\[
= (s + 5)(s^2 + s) - (3s^2 + 12s + 5)(s + 4)
\]

Conclusion: At high gain, we get

- Improved Settling time
- Slightly less overshoot
The effect of a **Lead Compensator**

- Add Phase at every point

\[ \angle(K(s)G(s)) = \angle K(s) + \angle G(s) \]

- The change in phase is **positive**.

\[ \Delta \angle = \theta_z - \theta_p \]

Points compensate by moving left.
Lead Compensation

Lead-Lag can be used to do pole-placement

\[ G(s) = \frac{1}{s(s + 1)} \]

Suppose we want:

- 20% Overshoot
- \( \omega_n = 2 \)
- \( T_s < 4 \)

We choose a desired point on the root locus:

- The intersection of
  - \( \omega_n = 2 \)
  - \( \sigma < -1 \)

\[ s_{1,2} = -1 \pm \sqrt{3}i \]

**Question:** Can we achieve this point exactly using Pole-Zero compensation?
Lead-Lag Compensation
Pole Placement

Let's start with a basic question:

- Is $s$ already on the root locus?

Let's check:

$$\angle G(s) = \sum \angle(s - z_i) - \sum \angle(s - p_i)$$

Working out the geometry:

$$\angle G(s) = -90^\circ - 120^\circ = -210^\circ$$

Not on the Root Locus!

The point $s$ lacks $30^\circ$ of phase.
Lead Compensation
Pole Placement

To place the point \( s \) on the root locus:

- we need to add 30° of phase at this point.

**Phase is sum of zeros minus poles**

- Zeros add phase
- Poles subtract phase.

We can add 30° if we use a pole-zero combo:

- Add a zero at 60°
- Add a pole at 30°
Lead Compensation

Pole Placement

\[ K(s) = \frac{s - z}{s - p} \]

Use reverse geometry to find \( p \) and \( z \).

**Zero:**

\[ \tan 60^\circ = \frac{\sqrt{3}}{x} \]

\[ x = \frac{\sqrt{3}}{\tan 60^\circ} = 1 \]

**Pole:**

\[ \tan 30^\circ = \frac{\sqrt{3}}{x} \]

\[ x = \frac{\sqrt{3}}{\tan 30^\circ} = 3 \]

\[ p = -1 - x = -4 \]

\[ z = -1 - x = -2 \]
Lead Compensation
Pole Placement

Now, the root locus passes through $s$.
To find the gain at this point
- Use `rlocfind`
- Use $k = \left| \frac{d(s)}{n(s)} \right|$.
For this example,

$$k = 6.00$$

Potential Problem: May adversely affect other poles.
Wiley+ Root Locus Demo 2

Make the phase $180^\circ$. 

The other big use of lead compensation is to change **Departure Angles**. Recall the Suspension system problem with integral feedback:

\[
G(s) = \frac{s^2 + s + 1}{s^4 + 2s^3 + 3s^2 + 1s + 1} \cdot \frac{1}{s}
\]

The poles are

- \( p_{1,2} = -0.957 \pm 1.23i \)
- \( p_{3,4} = -0.0433 \pm 0.641i \)

At pole \( p_{3,4} = -0.0433 + 0.641i \), the phase is \(-156^\circ\).

**Departure Angle:**

\[
\angle_{dep} = \angle G(s) + 180^\circ = 24^\circ
\]

**Goal:** Increase the departure angle to \(90^\circ\) or more.
Lead Compensation
Departure Angles

Suppose we want a departure angle of $\angle_{dep} = 100^\circ$.

- Recall
  \[ \angle_{dep} = \angle G(s) + 180^\circ \]

- Required Phase $\angle G(s)$
  \[ \angle_{req} G(s) = \angle_{dep} - 180^\circ = -80^\circ \]

- Required Phase Change:
  \[ \Delta \angle G(s) = \angle_{req} G(s) - \angle G'(s) = -80 + 156^\circ = 76^\circ \]
Lead Compensation
Departure Angles

We need to add $76^\circ$.

- Zero at $90^\circ$.
- Pole at $14^\circ$.

Recall departure point is $p_3 = -0.0433 + 0.641i$.

Zero:

- $\theta = 90^\circ$,
- $\Delta x = 0$
- $z = -0.0433$

Pole:

$$\tan 14^\circ = \frac{0.641}{\Delta x}$$

$$\Delta x = \frac{0.641}{\tan 14^\circ} = 2.57$$

So $p = -0.0433 - \Delta x = -2.61$.

Controller:

$$K(s) = \frac{s + 0.0433}{s + 2.61}$$
Lag Compensation
Steady-State Error

Predict the Steady-State Error.

\[ G(s) = \frac{s^2 + s + 1}{s^4 + 2s^3 + 3s^2 + 1s + 1} \]

\[ K(s) = \frac{s - z}{s - p} = \frac{n_k(s)}{d_k(s)} \]

\[ e_{ss} = \lim_{s \to 0} \frac{1}{1 + G(s)K(s)} \]

\[ = \frac{d_k(0)}{d_k(0) + k n_k(0) G(0)} \]

\[ = \frac{-p}{-p + -zkG(0)} \]

If
- \( p \) is small
- \( z \) is large

Then

\[ e_{ss} \approx \frac{p}{kz G(0)} \]
Lag Compensation

Steady-State Error

\[ G(s) = \frac{s^2 + s + 1}{s^4 + 2s^3 + 3s^2 + 1s + 1} \]

Say we want steady-state error less than .01.

\[ e_{ss} \approx \frac{p}{kz} \frac{1}{G(0)} = \frac{p}{kz} \leq .01 \]

or \( p \leq .01kz \)

- Assume \( k > 10 \)
- Choose \( p = .1 \)
- Result: \( z = 100 \)

Alternatively, \( p = 1, z = 1000. \)

- But there are dangers!
Lag Compensation

Notice some negative effects of Lag

• Asymptotes still at ±90°

**Center of Asymptotes:**

\[
\alpha = \frac{\sum p_i - \sum z_i}{n - m} = \frac{\sum p_{i,old} - \sum z_{i,old}}{n - m} + \frac{\sum p_{i,new} - \sum z_{i,new}}{n - m} = \alpha_{old} + \frac{p - z}{2}
\]

Creates a **Shift in Asymptotes** by

\[
\Delta \alpha \cong \frac{z}{2} = 50
\]

for the suspension problem \((z = -100)\).
Lead-Lag Compensation

To mitigate the effect of lag compensation:

- Add some Lead Compensation
  - Zero at $z = .01$
  - Pole at $p = 20$

**Phase at $s_1$**

$$\angle G(s) = -25.8^\circ$$

**Departure Angle:**

$$\angle_{dep} = 180 + \angle G(s) = 154.25^\circ$$
Use `rlocfind` to pick off

- Maximum stable gain

\[ k = 0.7768 \]
Notch-Filters

Conclusion:
• Lead-Lag Improves Performance
• Can’t do everything.

Problem Can’t Stabilize those poles at

\[ s = -0.0433 \pm 0.641i \]

One solution is to use a **Notch Filter**.

**Definition 2.**

A **Notch Filter** consists of
• Two Complex Zeros
  ▶ Used to Capture Troublesome Poles
• Two Real Poles far out in the LHP
Notch-Filter Example

To attack the poles at

\[ s = -0.0433 \pm 0.641i \]

Let's use a notch filter at

\[ z_{1,2} = -0.5 \pm 0.641i \]

Poles at

\[ p_{1,2} = -20 \]

\[
K(s) = \frac{(s + 0.5 + 0.641i)(s + 0.5 - 0.641i)}{(s + 20)^2} = \frac{s^2 + s + 0.66}{s^2 + 40s + 400}
\]
Combining the Notch-Filter with the Lag filter.

- Using `rlocfind`, we pick off the point

\[ k = 10 \]
Notch-Filter Example

Step Response

Time (sec)  Amplitude
0  1  2  3  4  5  6
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
Step Response

Reduced Steady-State Error

M. Peet
Lecture 17: Control Systems
Notch-Filter Example

Don’t Forget about the other poles!

\[ k_{max} = 19.4 \]
Notch-Filter Example

What about 30% overshoot?
Notch-Filter Example

What about 30% overshoot?

Don’t forget those other poles!
What have we learned today?

Lead-Lag Compensation
- Designing Leads
- Designing Lags
- Combining Leads and Lags

Notch Filters
- Providing extra zeros
- Eliminates annoying frequency components.

Next Lecture: The Frequency Domain