Outline

- PI control system design.
- PD control system design.
- PID control system design.

PI Control

- Add integral control to improve steady-state error, \((\text{increases type number one}) \Rightarrow \text{worse transient response or instability.}\)
- Add a proportional control \(\Rightarrow\) controller has pole at origin and zero.
- **Cascade compensation**: integral term in feedback path is equivalent to a differentiator in the forward path.

\[
G_c(s) = K_p + \frac{K_i}{s} = K_p \frac{s + a}{s}, \quad a = \frac{K_i}{K_p}
\]
PI Design Procedure

1. Design a proportional controller to meet the transient response specifications, i.e. place the dominant closed-loop system poles at a desired location $s_{cl} = -\zeta \omega_n \pm j \omega_d$
2. Add a PI controller with a zero at $-\zeta \omega_n / 10$
3. Tune the gain of the system to move the closed-loop pole closer to $s_{cl}$.
4. Check the time response and modify the design until it is acceptable.

Example: PI Design

Design a controller for the system for percentage overshoot less than 5% and zero steady-state error due to step.

$$G(s) = \frac{1}{(s + 3)(s + 4)}$$

Step 1: Proportional Control

Design proportional control to meet transient response specifications.

Percentage overshoot less than 5%

$$\zeta = \frac{|\ln(0.05)|}{\sqrt{|\ln(0.05)|^2 + \pi^2}} \approx 0.7$$

$$G(s) = \frac{1}{(s + 3)(s + 4)}$$

c.l. characteristic polynomial

$$s^2 + 7s + 12 + K = s^2 + 2(0.7)\omega_n s + \omega_n^2$$

Equate coefficients

$$\omega_n = 5 \text{rad/s} \quad K = 5^2 - 12 = 13$$
Step 2: Add PI Compensator

\[ G_c(s) = K_p \frac{s + a}{s} \]

\[ a = \frac{\xi \omega_n}{10} \approx 0.3 \]

\[ K \approx 13 \]

- For this example, an easier design canceling the pole at \(-3\) may be superior.

RL for PI Compensated System

Time Response for PI Example

Increase gain for faster response
**PD Control**

- Can be used in cascade or feedback control.
- Zero pulls RL to left: improves the transient response.
- Transfer Function of PD Controller
  \[ G_c(s) = K_p + K_ds = K_d(s + a), \quad a = \frac{K_p}{K_d} \]

**PD Design**

- Obtain the desired dominant pair from the design specifications.
- Angle of PD controller
  \[ \theta_c = \pm 180^\circ - \angle L(s_{cl}) \]

**Compensator (s + a)**

\[ a = \frac{\omega_d}{\tan(\theta_c)} + \zeta \omega_n \]

**PD Design Procedure**

1. Calculate \( \pi - \text{angle} [L(s_{cl})] \) using a calculator or MATLAB
   
   ```
   » scl=-zeta*wn+j*wd;
   » theta= pi–angle(evalfr(g, scl))
   ```

2. Calculate the zero location using
   
   \[ a = \frac{\omega_d}{\tan(\theta_c)} + \zeta \omega_n \]
   
   ```
   » a=wd/tan(thetac)+zeta*wn
   ```

3. Calculate the new loop gain function including the compensator then calculate the gain (magnitude condition) with a calculator or MATLAB
   
   ```
   » L = g*tf([1, a], 1)
   » K = 1/abs(evalfr(L, scl))
   ```

4. Check the time response of the PD compensated system and modify the design to meet the desired specifications if necessary.
Example

• Design a P-controller for \( \% \text{OS} < 10\% \)
• Design a PD controller to reduce \( T_s \) to 1 s.

![](image)

P-Control

• Percentage overshoot less than 10%

\[
\zeta = \frac{|\ln(0.1)|}{\sqrt{|\ln(0.1)|^2 + \pi^2}} \approx 0.59 \approx 0.6
\]

\[
G(s) = \frac{1}{s(s + 4)(s + 6)}
\]

• Find the intersection of constant \( \zeta \) line with the root locus using the angle condition by trial and error.

• The intersection is easy to find with MATLAB.

Root Locus for P-Control

PD Design

• Reduce \( T_s \) to 1 s

\[
T_s = \frac{4}{\zeta \omega_n} = 1 \text{ s} \Rightarrow \zeta \omega_n = 4 \text{ rad/s}
\]

\[
\zeta \approx 0.6 \Rightarrow \omega_n \approx 6.67 \text{ rad/s}
\]

• Find the c.l. pole location

\[
\omega_d = \omega_n \sqrt{1 - \zeta^2} \approx 5.33 \text{ rad/s}
\]

\[
s_{cl} = -4 + j5.33
\]

• Angle of PD controller

\[
\theta_c = 180^\circ - \angle G(s_{cl}) = 180^\circ - 73.7^\circ = 106.3^\circ
\]
PD Controller

- Compensator zero
  \[ a = \frac{\omega_d}{\tan(\theta_c)} + \zeta \omega_n = \frac{5.33}{\tan(106.3)} + 4 \approx 2.4 \]

Loop gain \( L(s) = G_c(s)G(s) = \frac{s+2.4}{s(s+4)(s+6)} \)

- Calculate the gain
  \[ K = \frac{1}{|L(s_{cl})|} \approx 3.63, \quad s_{cl} = -4 + j5.33 \]

Simulation results show that the system is too slow.

Modified PD Design

We need to speed up the time response.

1. Move zero closer to the \( j\omega \)-axis.
2. Increase the gain.

\[ G_c(s)G(s) = \frac{s + 2}{s(s + 4)(s + 6)} \]

\[ s_{cl} = -4.23 + j5.64 \]

\[ K = \frac{1}{|G_c(s_{cl})G(s_{cl})|} \approx 38.8 \]
Step Response $K = 70$

![Step Response Graph]

**PID Control**

$G_c(s) = K_p + \frac{K_i}{s} + K_d s = K_p \frac{s^2 + 2\zeta \omega_n s + \omega_n^2}{s}$,

$2\zeta \omega_n = \frac{K_d}{K_d}$,  \hspace{1em} \omega_n^2 = \frac{K_i}{K_d}$

**Procedure**

1. Design a PD controller to meet the transient response specifications.
2. Add a PI controller to meet the steady-state specifications without appreciably affecting the transient response.

---

**Example: PID Design**

- Design a PID controller for the system to meet the PD specifications (%OS < 10%, $T_s = 1s$) with zero steady-state error due to ramp.

$$G(s) = \frac{1}{s(s + 4)(s + 6)}$$

---

**PID Design**

- PD specifications met with earlier PD design
  $$G_{PD}(s) = 38.8(s + 2)$$

- PI design
  $$s_{cl} = -4.23 + j5.64 \Rightarrow \frac{\zeta \omega_n}{10} \approx 0.4$$

- Combine for PID
  $$G_{PID}(s) = 38 \frac{(s+2)(s+0.4)}{s}$$
Root Locus with PID

PID Design with $K = 38$

PID with $K = 70$, zero at $-0.2$

PD Design: Pole-Zero Cancellation

- Cancel pole with PD zero.
- Do not cancel RHP poles.
- Do not cancel pole at origin.
- Cancel closest real pole to the $j\omega$-axis.
- Imperfect implementation: almost cancel.
RHP Pole

- Zeros do not change the response due to the ICs.
- Imperfect cancellation: RHP closed-loop pole.

Example

\[ T(s) = \frac{K(s - 1)}{(s - 1)(s + 6)} \]

- Form of complete step response
\[ c(t) = a_0 + a_1 e^t + a_2 e^{-6t} \]

Example

For the closed-loop system with unity feedback, design a controller to reduce the percentage overshoot to less than 5% and the settling time to less than 1.5 second.

\[ G(s) = \frac{10}{s(s + 1)(s + 6)} \]

Root Locus: Uncompensated

With 10 included in MATLAB transfer function

Step Response

- Uncompensated closed-loop system

```matlab
>> step(feedback(g,1))
```
Design

\[ G(s) = \frac{10}{s(s + 1)(s + 6)} \]

- Cancel pole at \(-1\) with a zero

\[ G_{PD}(s) = K(s + 1) \]

\[ \zeta \omega_n = 3 \Rightarrow T_s = \frac{4}{\zeta \omega_n} = 1.333 \, s \]

- Choose \(K\) for \(\zeta = 0.7\) (gives <5% overshoot)
- 10 not included in compensator gain.

Equating Coefficients

- Loop gain

\[ L(s) = G_{PD}(s)G(s) = \frac{10K(s + 1)}{s(s + 1)(s + 6)} \]

- C.l. characteristic polynomial

\[ s^2 + 6s + 10K = s^2 + 2(0.7)\omega_n s + \omega_n^2 \]

\[ \omega_n = 4.29 \frac{rad}{s}, \quad 10K = 4.29^2 = 18.37 \]

Root Locus

\[ \omega_n = 4.28 \, rad/s, \quad K = 1.83 \]

Design Algorithm

1. Transient response OK?
   - Yes
     - Stop
   - No
     - Add PD
2. SS Error OK?
   - Yes
     - Stop
   - No
     - Add PI
3. SS Error OK?
   - Yes
     - Stop
   - No
     - Add PI