This lecture describes a method to tune PID controllers using pole placement.

For first-order systems, the approach is to:

- Use PI control and
- Select the gains to place the two closed-loop poles at desired locations.

The choice of natural frequency (time constant) is critical.
Design Approach: Pole Placement

1. Approximate the plant dynamics by a first or second-order ODE using the dominant pole approximation.
2. If the dynamics are first-order: Use a PI controller to place the two poles at a desired location.
2. If dynamics are second-order:
   • Use a PID controller to place the three poles.
   • Avoid use of derivative control if plant is well-damped. This will restrict the choice of the three poles.

A reasonable starting point is to place all poles at \( s = -\omega_n \).

The choice of natural frequency (time constant) is critical.
Design Approach: Pole Placement

1. Approximate the plant dynamics by a first or second-order ODE using the dominant pole approximation.

2. If the dynamics are first-order: Use a PI controller to place the two poles at a desired location.

2. If dynamics are second-order:
   • Use a PID controller to place the three poles.
   • Avoid use of derivative control if plant is well-damped. This will restrict the choice of the three poles.

3. Further tuning is often required. Use root locus to tune one gain at a time.

4. Implementation:
   • D-control: Use smoothed derivative or rate feedback
   • I-control: Use anti-windup (to be discussed later)
Dominant-Pole Approximation

The dominant poles of a higher-order system are the slowest poles (largest time constant).

We can often approximate a higher-order system by a:

1. First-order approximation if the dominant pole is real
2. Second-order approximation if the dominant pole(s) are a complex pair.

The approximation is accurate if the dominant pole(s) are significantly slower than the remaining poles.

(Dominant pole time constant is 5x larger than other poles)
Example

Consider the fifth-order system:

\[ G_2(s) = \frac{2.7 \times 10^5}{s^5 + 98s^4 + 2194s^3 + 36555s^2 + 107100s + 2.7 \times 10^5} \]

**Poles:** \( s = -1.5 \pm 2.6j, -10 \pm 17.3j, -75 \)

\( \omega_n = 3 \frac{rad}{sec} \) and \( \zeta = 0.5 \)

**Approximation:** \( G_{low,2}(s) = \frac{b_0}{s^2 + 3s + 9} \)

Select \( b_0 \) to match the DC gain: \( G_2(0) = G_{low,2}(0) \)

\[ b_0 = 9 \Rightarrow G_{low,2}(s) = \frac{9}{s^2 + 3s + 9} \]
Example

Fifth-order system and dominant pole approximation

\[ G_2(s) = \frac{2.7 \times 10^5}{s^5 + 98s^4 + 2194s^3 + 36555s^2 + 107100s + 2.7 \times 10^5} \]

\[ G_{low,2}(s) = \frac{9}{s^2 + 3s + 9} \]
Example plant model:

\[
\dot{y}(t) + a_0 y(t) = b_0 u(t) + b_0 d(t) \quad \text{where } a_0 = 2 \text{ and } b_0 = 3
\]

Formal design requirements can be stated. Roughly a faster closed-loop response will:

- lead to better reference tracking and disturbance rejection,
- but it will also increase the actuator effort and degrade the noise rejection.

Important: First-order ODE is typically an approximate model. Formal tools to assess the impact of model uncertainty later.

If the closed-loop is too fast then the unmodeled dynamics will degrade performance and may even cause instability.
Dynamics of the plant:

\[ \ddot{y}(t) + a_0 y(t) = b_0 u(t) + b_0 d(t) \quad \text{where } a_0 = 2 \text{ and } b_0 = 3 \]

PI Controller:

\[ u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau \]

Sub for \( u \) into plant dynamics and collect terms.

Closed-loop dynamics are:

\[ \ddot{y}(t) + (a_0 + b_0 K_p) \dot{y}(t) + b_0 K_i y(t) = b_0 K_p \dot{r}(t) + b_0 K_i r(t) + b_0 \dot{d}(t) \]

\[ := 2 \zeta \omega_n \]

\[ := \omega_n^2 \]
PI Tuning

Dynamics of the closed-loop:

\[
\ddot{y}(t) + (a_0 + b_0 K_p) \dot{y}(t) + b_0 K_i \dot{y}(t) = b_0 K_p \dot{r}(t) + b_0 K_i r(t) + b_0 \dot{d}(t)
\]

Pole Placement:

- Select the closed-loop \((\omega_n, \zeta)\) based on a desired settling time and peak overshoot. (Starting point is \(\zeta = 1\).)
- Closed-loop from \(r\) to \(y\) has a zero at \(s = -\frac{K_i}{K_p}\)

This zero increases overshoot and reduces rise time.
- Solve for controller gains:
  \[
  K_i = \frac{\omega_n^2}{b_0} \quad \text{and} \quad K_p = \frac{2\zeta \omega_n a_0}{b_0}.
  \]
- Integral control yields zero steady-state error.
Comparison of Two PI Controllers

\( K_1 \) is designed for faster response than \( K_2 \).

<table>
<thead>
<tr>
<th>Design</th>
<th>( \zeta )</th>
<th>( \omega_n, \text{ rad/sec} )</th>
<th>Poles, ( s_{1,2} )</th>
<th>( M_p )</th>
<th>( \tau_{\text{settle, sec}} )</th>
<th>( K_p )</th>
<th>( K_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_1(s) )</td>
<td>1.0</td>
<td>6.67</td>
<td>-6.67, -6.67</td>
<td>0</td>
<td>0.45</td>
<td>3.78</td>
<td>14.81</td>
</tr>
<tr>
<td>( K_2(s) )</td>
<td>1.0</td>
<td>2.86</td>
<td>-2.86, -2.86</td>
<td>0</td>
<td>1.05</td>
<td>1.24</td>
<td>2.72</td>
</tr>
</tbody>
</table>
Comparison of Two PI Controllers

$K_1$ is designed for faster response than $K_2$.

<table>
<thead>
<tr>
<th>Design</th>
<th>$\zeta$</th>
<th>$\omega_n$, rad/sec</th>
<th>Poles, $s_{1,2}$</th>
<th>$M_p$</th>
<th>$\tau_{settle}$, sec</th>
<th>$K_p$</th>
<th>$K_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1(s)$</td>
<td>1.0</td>
<td>6.67</td>
<td>-6.67, -6.67</td>
<td>0</td>
<td>0.45</td>
<td>3.78</td>
<td>14.81</td>
</tr>
<tr>
<td>$K_2(s)$</td>
<td>1.0</td>
<td>2.86</td>
<td>-2.86, -2.86</td>
<td>0</td>
<td>1.05</td>
<td>1.24</td>
<td>2.72</td>
</tr>
</tbody>
</table>

Step responses with $r(t) = 4$, $d(t) = 2$ for $t \geq 1.5$, and sensor noise for $t \geq 4$. 

![Graph showing step responses for $K_1(s)$ and $K_2(s)$]
PID Tuning for Second-Order Systems

Next, we describe a method to tune PID controllers using pole placement.

For second-order systems, the approach is to:

- Use PID control and
- Select the gains to place the three closed-loop poles at desired locations.
- A PI controller (without the D-term) should be used if the plant has sufficient damping.

The choice of natural frequency (time constant) is critical.
Design Approach: Pole Placement

1. Approximate the plant dynamics by a first or second-order ODE using the dominant pole approximation.

2. If the dynamics are first-order: Use a PI controller to place the two poles at a desired location.

2. If dynamics are second-order:
   - Use a PID controller to place the three poles.
   - Avoid use of derivative control if plant is well-damped. This will restrict the choice of the three poles.

A reasonable starting point is to place all poles at $s = -\omega_n$. The choice of natural frequency (time constant) is critical.
Design Approach: Pole Placement

1. Approximate the plant dynamics by a first or second-order ODE using the dominant pole approximation.

2. If the dynamics are first-order: Use a PI controller to place the two poles at a desired location.

2. If dynamics are second-order:
   • Use a PID controller to place the three poles.
   • Avoid use of derivative control if plant is well-damped. This will restrict the choice of the three poles.

3. Further tuning is often required. Use root locus to tune one gain at a time.

4. Implementation:
   • D-control: Use smoothed derivative or rate feedback
   • I-control: Use anti-windup (to be discussed later)
Example plant model:
\[ \ddot{y}(t) + a_1 \dot{y}(t) + a_0 y(t) = b_0 u(t) + b_0 d(t) \]
where \( a_1 = -2 \), 
\[ a_0 = 17 \] and \( b_0 = 17 \)

Formal design requirements can be stated. Roughly a faster closed-loop response will:

• lead to better reference tracking and disturbance rejection,
• but it will also increase the actuator effort and degrade the noise rejection.

Important: Second-order ODE is typically an approximate model. Formal tools to assess the impact of model uncertainty later.

If the closed-loop is too fast then the unmodeled dynamics will degrade performance and may even cause instability.
Dynamics of the plant:

\[ \ddot{y}(t) + a_1 \dot{y}(t) + a_0 y(t) = b_0 u(t) + b_0 d(t) \]

where \( a_1 = -2 \), \( a_0 = 17 \), and \( b_0 = 17 \)

PID controller in rate feedback form:

\[ u(t) = K_p e(t) + K_i \int_0^t e(\tau) \, d\tau - K_d \dot{y}(t) \]

Sub for \( u \) into plant dynamics and collect terms.

Closed-loop dynamics are:

\[ y^{[3]}(t) + (a_1 + b_0 K_d) \dot{y}(t) + (a_0 + b_0 K_p) \dot{y}(t) + (b_0 K_i) y(t) \]

\[ = b_0 K_p \dot{r}(t) + b_0 K_i r(t) + b_0 \dot{d}(t) \]

The closed-loop characteristic equation is:

\[ 0 = s^3 + (a_1 + b_0 K_d) s^2 + (a_0 + b_0 K_p) s + (b_0 K_i) \]
Closed-loop characteristic equation:

\[ 0 = s^3 + (a_1 + b_0 K_d) s^2 + (a_0 + b_0 K_p) s + (b_0 K_i) \]

Pole Placement:

- Select the desired poles to satisfy for some \((\zeta, \omega_n, p)\):
  \[ 0 = (s^2 + 2\zeta \omega_n s + \omega_n^2) \cdot (s + p) \]

Choose \(\zeta = 1\) and \(p = \omega_n\) as a starting point.

- The desired characteristic equation is:
  \[ 0 = s^3 + (p + 2\zeta \omega_n) s^2 + (2\zeta \omega_n p + \omega_n^2) s + \omega_n^2 p \]

- Match coefficients to the closed-loop characteristic equation:

\[
\begin{align*}
  a_1 + b_0 K_d &= p + 2\zeta \omega_n \\
  a_0 + b_0 K_p &= 2\zeta \omega_n p + \omega_n^2 \\
  b_0 K_i &= \omega_n^2 p 
\end{align*}
\]

Solve these equations for the three gains.
Comparison of Two PID Controllers

\( K_1 \) is designed for faster response than \( K_2 \).

<table>
<thead>
<tr>
<th>Design</th>
<th>( \zeta )</th>
<th>( \omega_n, \ \text{rad/sec} )</th>
<th>( p )</th>
<th>Poles, ( s_{1,2} ) and ( s_3 )</th>
<th>( M_p )</th>
<th>( \tau_{\text{settle}, \ \text{sec}} )</th>
<th>( K_p )</th>
<th>( K_i )</th>
<th>( K_d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_1(s) )</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>-6, -6, -6</td>
<td>0</td>
<td>0.5</td>
<td>5.35</td>
<td>12.7</td>
<td>1.18</td>
</tr>
<tr>
<td>( K_2(s) )</td>
<td>1</td>
<td>3.5</td>
<td>3.5</td>
<td>-3.5, -3.5, -3.5</td>
<td>0</td>
<td>0.86</td>
<td>1.16</td>
<td>2.52</td>
<td>0.735</td>
</tr>
</tbody>
</table>
Comparison of Two PID Controllers

$K_1$ is designed for faster response than $K_2$.

<table>
<thead>
<tr>
<th>Design</th>
<th>$\zeta$</th>
<th>$\omega_n$, $\text{rad/sec}$</th>
<th>$p$</th>
<th>Poles, $s_{1,2}$ and $s_3$</th>
<th>$M_p$</th>
<th>$\tau_{\text{settle, sec}}$</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1(s)$</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>-6, -6, -6</td>
<td>0</td>
<td>0.5</td>
<td>5.35</td>
<td>12.7</td>
<td>1.18</td>
</tr>
<tr>
<td>$K_2(s)$</td>
<td>1</td>
<td>3.5</td>
<td>3.5</td>
<td>-3.5, -3.5, -3.5</td>
<td>0</td>
<td>0.86</td>
<td>1.16</td>
<td>2.52</td>
<td>0.735</td>
</tr>
</tbody>
</table>

Step responses with $r(t) = 4$, $d(t) = 2$ for $t \geq 1.5$, and sensor noise for $t \geq 4$. 

[Graph showing step responses with $K_1(s)$ and $K_2(s)$]