ECE 486: Control Systems

Lecture 9A: PI Tuning for First-Order Systems

Key Takeaways

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)

PI Tuning For First-Order Systems

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.

Closed-Loop Model

Dynamics of the plant:

$$\dot{y}(t) + a_0 y(t) = b_0 u(t) + b_0 d(t) \qquad \text{where } a_0 = 2 \text{ and } b_0 = 3$$

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:

$$\ddot{y}(t) + \underbrace{(a_0 + b_0 K_p)}_{:=2\zeta\omega_n} \dot{y}(t) + \underbrace{b_0 K_i}_{:=\omega_n^2} y(t) = b_0 K_p \dot{r}(t) + b_0 K_i r(t) + b_0 \dot{d}(t)$$

PI Tuning

Dynamics of the closed-loop:

$$\ddot{y}(t) + \underbrace{(a_0 + b_0 K_p)}_{:=2\zeta\omega_n} \dot{y}(t) + \underbrace{b_0 K_i}_{:=\omega_n^2} y(t) = b_0 K_p \dot{r}(t) + b_0 K_i r(t) + b_0 \dot{d}(t)$$

Pole Placement:

- Select the closed-loop (ω_n, ζ) 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: $K_i = \frac{\omega_n^2}{b_0}$ and $K_p = \frac{2\zeta\omega_n - a_0}{b_0}$.
- Integral control yields zero steady-state error.

Comparison of Two PI Controllers

K_1 is designed for faster response than K_2.

Design	ζ	$\omega_n, \frac{rad}{sec}$	Poles, $s_{1,2}$	M_p	τ_{settle}, sec	K_p	K_i
$K_1(s)$	1.0	6.67	-6.67, -6.67	0	0.45	3.78	14.81
$K_2(s)$	1.0	2.86	-2.86, -2.86	0	1.05	1.24	2.72

Comparison of Two PI Controllers

K_1 is designed for faster response than K_2.

Design	ζ	$\omega_n, \frac{rad}{sec}$	Poles, $s_{1,2}$	M_p	τ_{settle}, sec	K_p	K_i
$K_1(s)$	1.0	6.67	-6.67, -6.67	0	0.45	3.78	14.81
$K_2(s)$	1.0	2.86	-2.86, -2.86	0	1.05	1.24	2.72

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