# Module 08
# Controller Designs: Compensators and PIDs

**Ahmad F. Taha**

**EE 3413: Analysis and Desgin of Control Systems**

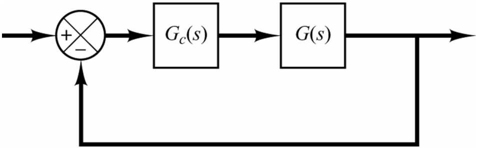
*Email:* ahmad.taha@utsa.edu

*Webpage:* http://engineering.utsa.edu/~taha

March 31, 2016

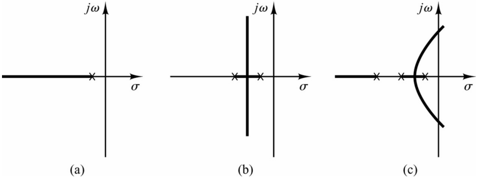
# Introduction

- Readings: 6.5–6.6, 8.1–8.2 Ogata; 7.6,10.3,10.5 Dorf & Bishop
- In Module 7, we learned to sketch the RL for any TF
- We saw how poles change as a function of the gain $K$
- '$K$' was a controller — a constant controller
- Many times, $K$ as a controller is not enough
- Example: system cannot be stabilized with a choice of $K$-gain
- Or, settling time is still high, overshoot still bad
- Today, we'll learn how to design more complicated controllers
- **Objective:** find $G_c(s)$ such that CLTF has desired properties such as settling time, maximum overshoot,...

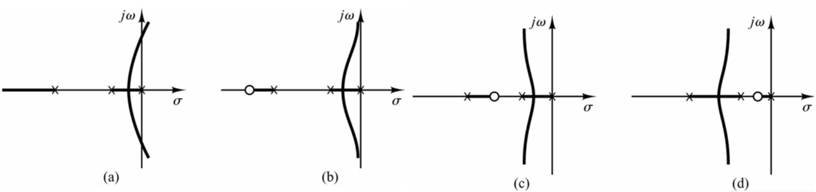


# Typical RL Plots

- (a) Root-locus plot of a single-pole system; (b) root-locus plot of a two-pole system; (c) root-locus plot of a three-pole system




- (a) Root-locus plot of a three-pole system; (b), (c), and (d) root-locus plots showing effects of addition of a zero to the three-pole system.

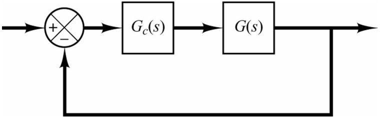


# Effects of Adding Poles and Zeros on RL

- Adding poles *pulls* the RL to the **right**
  – Systems become "less stable", settling is slower
- Adding zeros *pulls* the RL to the **left**
  – Systems become "more stable" (this is tricky), settling is faster
- **Question:** Can we conclude that a compensator (controller) $G_c(s)$ should always be a combination of zeros? Since, you know, it makes system *more stable* and settling is faster?
- Not really. **Why??** Because adding a zero amplifies the *high frequency noise*
- So, we can't add a zero alone (i.e., $G_c(s) = s + z$), and we can't add a pole alone either ($G_c(s) = \frac{1}{s+p}$). Solution?
- **Solution — Add a compensator of this form:**

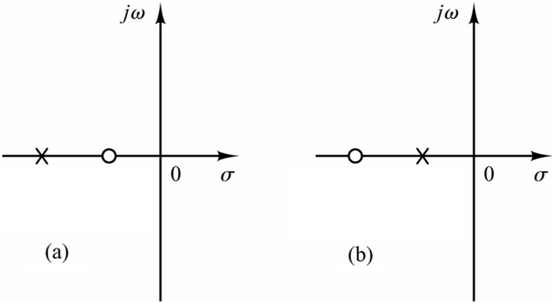
$G_c(s) = K\dfrac{s+z}{s+p}$ — **Objective: find K, z, p given certain desired properties**

# Two Controller Choices: Lead and Lag Compensators

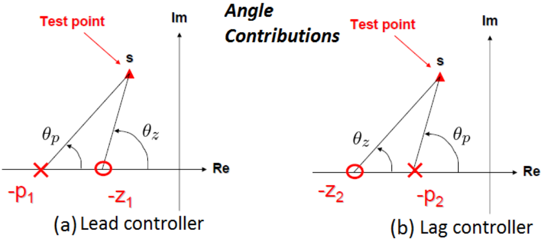



$G_c(s) = K\dfrac{s+z}{s+p}$ — **Objective: find K, z, p given certain desired properties**

- For $G_c(s)$ above, $K, z, p$ are all **real +ve** values to be found
- $\Rightarrow$ 2 combinations: (a) **lead controller**; (b) **lag controller**:



# Lead and Lag Compensators




- **Lead compensator** provides a +ve angle contribution:

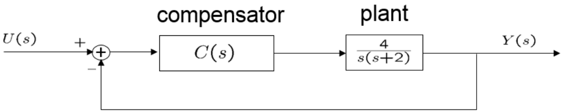
$$G_c^{ld}(s) = K\frac{s+z}{s+p} \;\Rightarrow\; \angle G_c^{ld}(s) = \angle(s+z) - \angle(s+p) = \theta_z - \theta_p = \theta_{lead} > 0$$

  - Speeds up transients by lowering rise time & decreasing overshoots
- **Lag compensator** provides a -ve angle contribution:

$$G_c^{ld}(s) = K\frac{s+z}{s+p} \;\Rightarrow\; \angle G_c^{ld}(s) = \angle(s+z) - \angle(s+p) = \theta_z - \theta_p = \theta_{lag} < 0$$

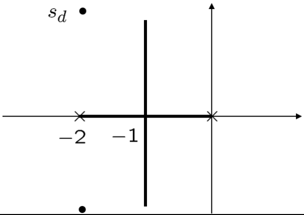
  - Improves the steady-state accuracy of the system for tracking inputs
- What if $p = z$? That's a constant gain (pole & zero cancel out)

# Lead Compensator Example




- Initially, the above system has $\zeta = 0.5$ and $\omega_n = 2$
- **Obj:** design $G_c^{ld}(s) = C(s) = K\frac{s+z}{s+p}$, such that $\zeta_d = 0.5, \omega_{nd} = 4$
- Can we do that via gain $K$? **No, see the RL below for** $C(s) = K$
- Hence, we can **never** reach $s_d$ via a constant gain, need compensator

$$s_d = -\zeta_d\omega_{nd} \pm \sqrt{1-\zeta_d^2}\,\omega_{nd} = -2 \pm j2\sqrt{3}$$



# Lead Compensator Example (Cont'd)

- **Objective:** design $G_c^{ld}(s) = K\frac{s+z}{s+p}$, such that $\zeta_d = 0.5, \omega_{nd} = 4$
- To find $K, z, p$, follow this algorithm:

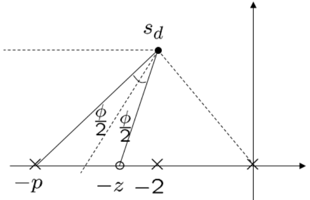
0. Find $s_d$ for $s_d^2 + 2\zeta_d\omega_{nd}s_d + \omega_{nd}^2 = 0$, $s_d = -2 + j2\sqrt{3}$
1. Find *angle of deficiency* $\phi$, as follows:

$$\theta = \angle G(s_d) = \angle G(-2 + j2\sqrt{3}) = -210\,\text{deg} \Rightarrow \phi = -180 - (\theta) = 30\,\text{deg}$$

2. Connect $s_d$ to the origin — OK
3. Draw a horizontal line to the left from $s_d$ — OK
4. Find the bisector of the above two lines — OK
5. Draw 2 lines that make angles $\phi/2$ & $-\phi/2$ with the bisector — OK
6. Their intersections with the real lines are $-p$ and $-z$ — OK

$z = 2.9$

$p = 5.4$



# Lead Compensator Example — Finding $K$

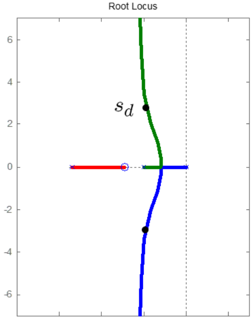
- We now know that $z = 2.9, p = 5.4 \Rightarrow G_c^{ld}(s) = C(s) = K\frac{s+2.9}{s+5.4}$
- We know that all points on the RL satisfy

$$1 + KG(s)G_c^{ld}(s) = 0 \Rightarrow 1 = |KG(s)G_c^{ld}(s)|$$

- We know **for sure** that $s_d$ belongs to the RL, so solve for $K$:

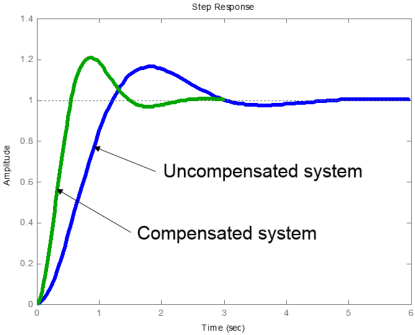
$$|4K\frac{s_d+2.9}{s_d(s_d+2)(s_d+5.4)}| = 1 \Rightarrow K = 4.68 \Rightarrow G_c^{ld}(s) = C(s) = 4.68\frac{s+2.9}{s+5.4}$$

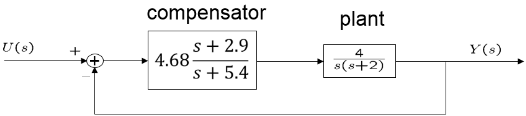
- *Compensated* RL plot:



# Step Response: Old vs. New Response

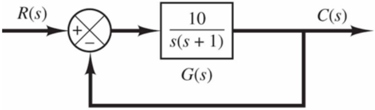
*Compensated system reaches SS faster (shorter rise, settling times), although it has a higher $M_p$. That said, we designed the compensator according to the design specs. Design specs weren't so smart, perhaps.*



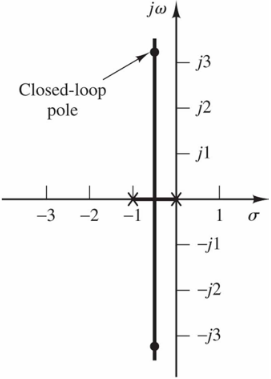



# Lead Compensator Example 2

- Given $G(s) = \dfrac{10}{s(s+1)}$, find $G_c^{ld}(s)$ such that the CLTF has $\zeta_d = 0.5$ and $\omega_{nd} = 3$
- Figures: (a) uncompensated control system (b) uncompensated root-locus plot



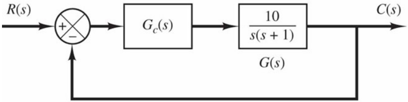

(a)



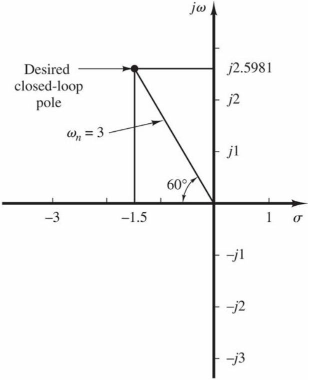

(b)

# Lead Compensator Example 2 (Cont'd)

- Given $G(s) = \dfrac{10}{s(s+1)}$, find $G_c^{ld}(s)$ such that the CLTF has $\zeta_d = 0.5$ and $\omega_{nd} = 3$
- Figures: (a) compensated system, (b) desired closed-loop pole location




(a)




(b)

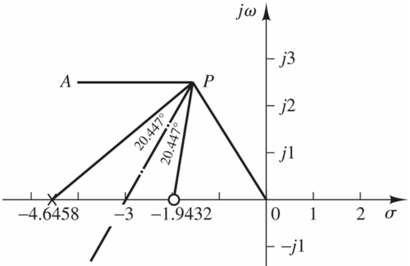
# Lead Compensator Example 2 (Cont'd)

- **Objective:** design $G_c^{ld}(s) = K\frac{s+z}{s+p}$, such that $\zeta_d = 0.5, \omega_{nd} = 3$

0. Find $s_d$ for $s_d^2 + 2\zeta_d\omega_{nd}s_d + \omega_{nd}^2 = 0$, $s_d = -1.5 \pm j2.58$
1. Find *angle of deficiency* $\phi$, as follows:

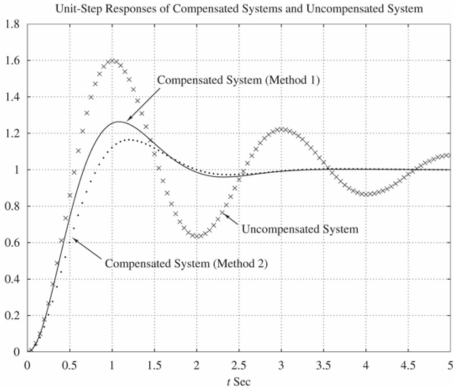
$$\theta = \angle G(s_d) = \angle G(-1.5+j2.58) = 138\,\text{deg} \Rightarrow \phi = -180-(138) = -318 \equiv 42\,\text{deg}$$

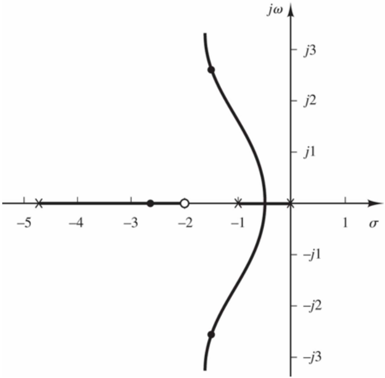
2. Connect $s_d$ to the origin — OK
3. Draw a horizontal line to the left from $s_d$ — OK
4. Find the bisector of the above two lines — OK
5. Draw 2 lines that make angles $\phi/2$ & $-\phi/2$ with the bisector — OK
6. Their intersections with the real lines are $-p$ and $-z$ — OK



# Compensated System, Example 2

**Unit-step response and RL plot for the compensated system:**



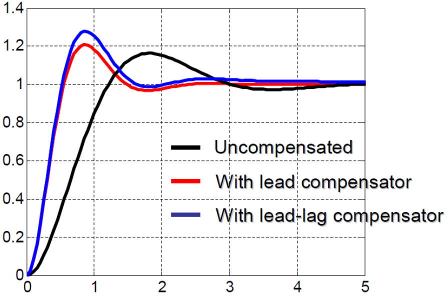



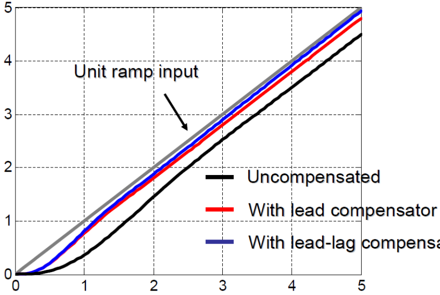
# Lag, Lead-Lag Compensators

- Now that we understand lead compensators, we can discuss lag and lead-lag compensators
- Recall that lead compensators: improve transient response and stability
- But they do not typically reduce SSE
- Lag compensators $G_c^{lg}(s)$: reduce SSE, so sometimes we want smaller SSE rather than shorter rise and settling time as in a lead compensator
- **Optimal solution:** lead-lag (LL) compensator— $G_c^{ll}(s) = G_c^{ld}(s)G_c^{lg}(s)$
- LL compensators provides the benefits of both lead and lag compensators

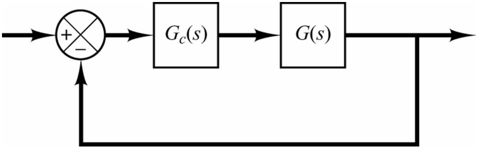
# Lead-Lag Compensators

- Unfortunately, we don't have time to cover LL compensator design
- Design procedure is simple, please read more about it from your textbooks
- But we'll show a figure that illustrates the difference in performance:
- Left figure (step response), right figure (ramp response)



# PID Control — Definitions and Basics




- **P**roportional **I**ntegral **D**erivative controller—PID control
- Without a doubt the most widely used controllers in industry today
- Bread and butter of control, 90% of control loops use PID control
- Proportional: $G_c(s) = K$, Integration: $G_c(s) = \frac{1}{Ts}$, Derivative: $G_c(s) = Ks$
- Can have combinations of the above controllers: P,I,D,PI,PD,ID,PID
- Major objectives for designing $G_c(s)$:

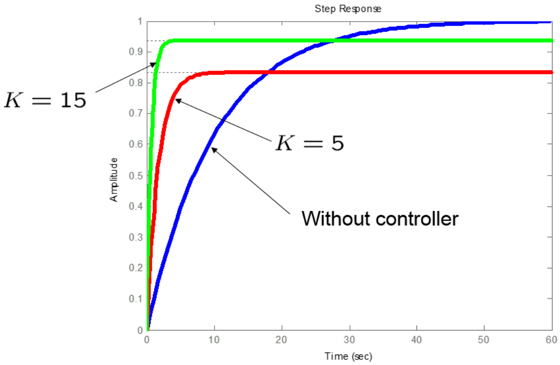
1. Stability—the most important objective: CLTF is stable
2. Steady-state error (SSE)—minimize this as much as we could
3. Time-specs—$M_p, t_r, t_s, \ldots$

# PID Controller Device


You can either use some tuning rules (which we will learn about during this module), or use an auto-tune function that figures out the parameters to a PID controller. Check http://www.omega.com/prodinfo/temperaturecontrollers.html for examples. Prices for common PID controllers range from \$20 to \$200, depending on size, quality, and performance.

# Example 1: P controller for FOS

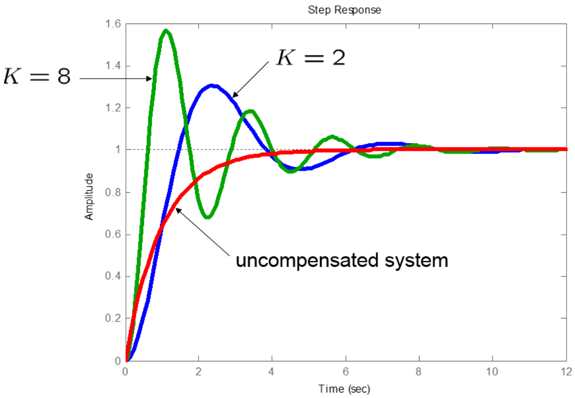
- Assume $G(s) = \frac{1}{Ts+1}$—first order system (FOS)
- We can design a P controller (i.e., $G_c(s) = K$)




- Result:
- Larger $K$ will increase the response speed
- SSE is present no matter how large $K$ is—recall the SSE Table ;)

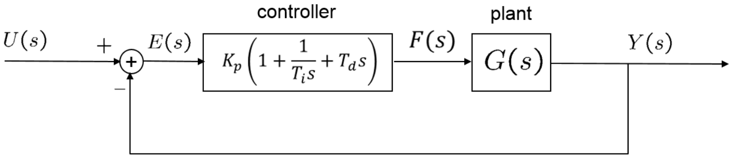
# Example 2: Integral (I) controller for FOS

- Assume $G(s) = \frac{1}{Ts+1}$—first order system (FOS)
- We can design an I controller (i.e., $G_c(s) = K/s$)




- Result:
- SSE for step input is completely eliminated
- But transients are bad—can cause instability for some $K$

# PID Controller

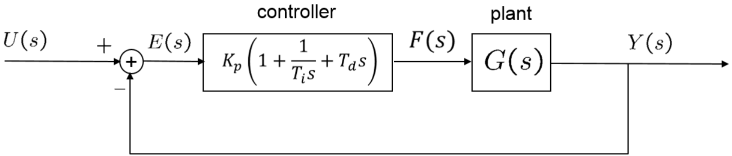



- PID (Proportional-Integral-Derivative) controller takes this form:

$$G_c(s) = K_p\left(1+\frac{1}{T_i s}+T_d s\right)$$

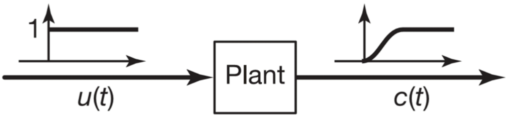
- Design objective: find parameters $K_p$, $T_i$, $T_d$ given required specs
- This process is called ***PID tuning***—process of adjusting $K_p$, $T_i$, $T_d$
- Many different **tuning rules** exist
- **Ziegler-Nichols Rule:** first PID tuning rules (first and second method)
- After finding these parameters, input them on the PID device

# More on PID Controllers




- Proportional term, i.e., $G_c^P(s) = K_p$:
  - Proportional term responds immediately to the current tracking error. Typically, however, it cannot achieve the desired tracking accuracy without excessively large gain.
- Integral term, i.e., $G_c^I(s) = \dfrac{K_p}{T_i s}$:
  - Integral term yields a zero steady-state error in tracking a step function (a constant set-point). This term is slow in response to the current tracking error.
- Derivative term, i.e., $G_c^D(s) = K_p T_d s$:
  - Derivative term is effective for plants with large dead-time
  - Reduces transient overshoots, but amplifies higher frequencies sensor noise

# Ziegler-Nichols Rule: First Method




Step 1 Obtain plant's unit step response experimentally[1]

- Unit step response is S-shaped for many plants
- Only valid if the step-response is S-shaped

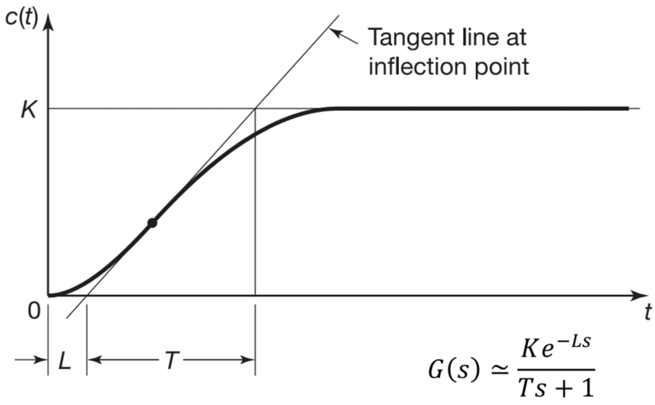
Step 2 Obtain delay time $L$ from the experimental plot

Step 3 Obtain time constant $T$ from the experimental plot

Step 4 Use tuning rule table to determine $K_p$, $T_i$, $T_d$ given $L$, $T$ (next slide)

[1] In industrial applications, control engineers usually specify the performance of the controlled system based on the system step response.

# Obtaining $L$, $T$ from Experimental Plot




- Of course, this is an approximation, but you have to be accurate with your computation of $L$ and $T$

# Obtaining $K_p$, $T_i$, $T_d$ via Tuning Method 1

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

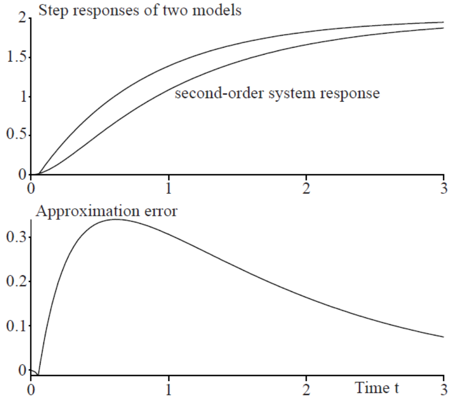
| Type of Controller | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | $\frac{T}{KL}$ | $\infty$ | 0 |
| PI | $\frac{0.9T}{KL}$ | $3.3L$ | 0 |
| PID | $1.2\frac{T}{KL}$ | $2L$ | $0.5L$ |

- If you want a PID controller, choose the third row and compute the parameters:

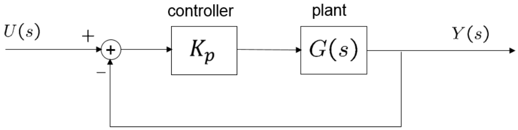
$$G_{PID}(s) = G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) = 0.6T\frac{\left(s + \frac{1}{L}\right)^2}{s}$$

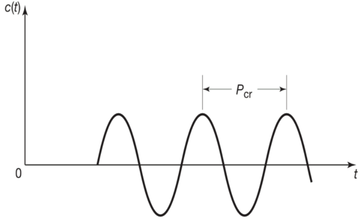
# PID Tuning, Method 1 Example

- Given a plant $G(s) = \dfrac{10}{s^2 + 6s + 5}$, find $K, L, T$ first
  - Given the procedure, we find that $\tilde{G}(s) \approx \dfrac{2e^{-0.05s}}{0.8s + 1}$
  - Plug these values in the table to obtain $G_c(s) = G_{PID}(s)$



# Ziegler-Nichols Rule: Second Method




**Step 1** Set $T_i = \infty$, $T_d = 0$ (above left figure) and increase $K_p$ until step response of the closed-loop system **has sustained oscillations**

- If no oscillation occurs for all values of $K_p$, this method is not applicable

**Step 2** Record $K_{cr}$ (critical value of gain $K_p$) and $P_{cr}$ (period of the oscillation); see above right figure

**Step 3** Use tuning rule table to determine $K_p, T_i, T_d$ given $K_{cr}, P_{cr}$ (next slide)

# Obtaining $K_p$, $T_i$, $T_d$ via Tuning Method 2

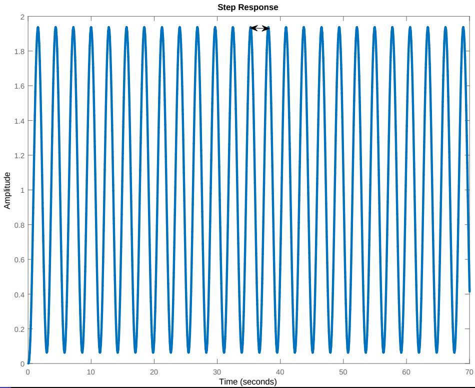
$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

| Type of Controller | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | $0.5K_{cr}$ | $\infty$ | 0 |
| PI | $0.45K_{cr}$ | $P_{cr}/1.2$ | 0 |
| PID | $0.6K_{cr}$ | $P_{cr}/2$ | $P_{cr}/8$ |

$$G_{PID}(s) = G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) = 0.075 K_{cr} P_{cr} \frac{\left(s + \frac{4}{P_{cr}}\right)^2}{s}$$

# Method 2 Example

- Given a plant $G(s) = \frac{1}{s(s+1)(s+5)}$, find the PID parameters using the second PID design method
- **Solution:** Experimentally, we plot the step response till we have sustained oscillations (Step 1)
- We can record $K_{cr} = 30, P_{cr} = 2.8$



# Method 2 Example

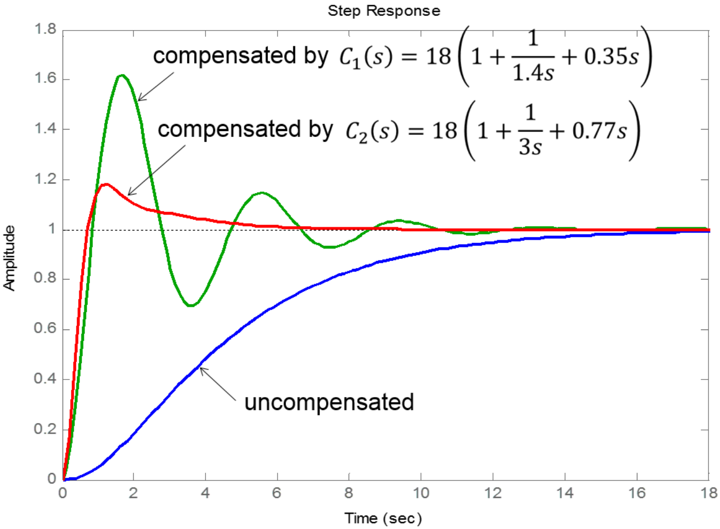
- Looking at the table, we can find $K_p$, $T_i$, $T_d$:

$$G_c(s) = 18\left(1 + \frac{1}{1.4s} + 0.35s\right)$$

- **Note:** we can find $K_{cr}$ by applying the RH table for the CP $(s^3 + 6s^2 + 5s + K_p)$
  - Then, you can find $K_p$ that would make the CP unstable $\Rightarrow K_{cr} = K_p^{max} = 30$
  - Then find the frequency $\omega_{cr}$ that solves this equation

$$(j\omega_{cr})^3+6(j\omega_{cr})^2+5j\omega_{cr}+30 = 0 \Rightarrow \omega_{cr} \Rightarrow \omega_{cr} = \sqrt{5} \Rightarrow P_{cr} = \frac{2\pi}{\omega_{cr}} = 2.8$$

# Step Response after PID Design



# PID Control Summary

- We only covered one type of PID control, called Type A PID control:

$$G_{PID-A}(s) = G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) = \alpha\frac{(s+\beta)^2}{s}$$

  where $\alpha$ and $\beta$ are the PID constants that depend on the plant's performance
- So, when do we use P,PI,PD, or PID control?
- Well, it depends on what you want

| $\nearrow$ Parameter | SSE | Response Speed | Stability | Oscillations | Overshoot |
|---|---|---|---|---|---|
| $\nearrow K_p$ | $\searrow$ | $\nearrow$ | $\searrow$ | $\nearrow$ | $\nearrow$ |
| $\nearrow K_i = \frac{1}{T_i}$ | $\searrow$ | $\searrow$ | $\searrow$ | $\nearrow$ | $\nearrow$ |
| $\nearrow K_d = T_d$ | $\nearrow$ | $\nearrow$ | $\nearrow$ | $\searrow$ | $\searrow$ |

# Course Progress

**Modeling (5-6 Weeks)**

- ~~Laplace Transforms~~
- ~~Transfer Functions~~
- ~~Solution of ODEs~~
- ~~Modeling of Systems~~
- ~~Block Diagrams~~
- ~~Linearization~~

**Analysis (7-8 Weeks)**

- ~~1st & 2nd Order Systems~~
  - ~~Time Response~~
  - ~~Transient & Steady State~~
- ~~Frequency Response~~
- ~~Bode Plots~~
- ~~RH Criterion~~
- ~~Stability Analysis~~

**Design (5-6 Weeks)**

- ~~Root-Locus~~
- ~~Design via RL, Compensators~~
- ~~PID Control~~
- Modern Control
- State-Space
- MIMO System Properties

# Questions And Suggestions?


**Thank You!**

Please visit

engineering.utsa.edu/~taha

**IFF** you want to know more ☺