

Control Systems

Subject Code: EC380

Lecture 19-24: The Design of Feedback Control Systems

Surajit Panja

Assistant Professor

Dept. of Electronics and Communication Engineering

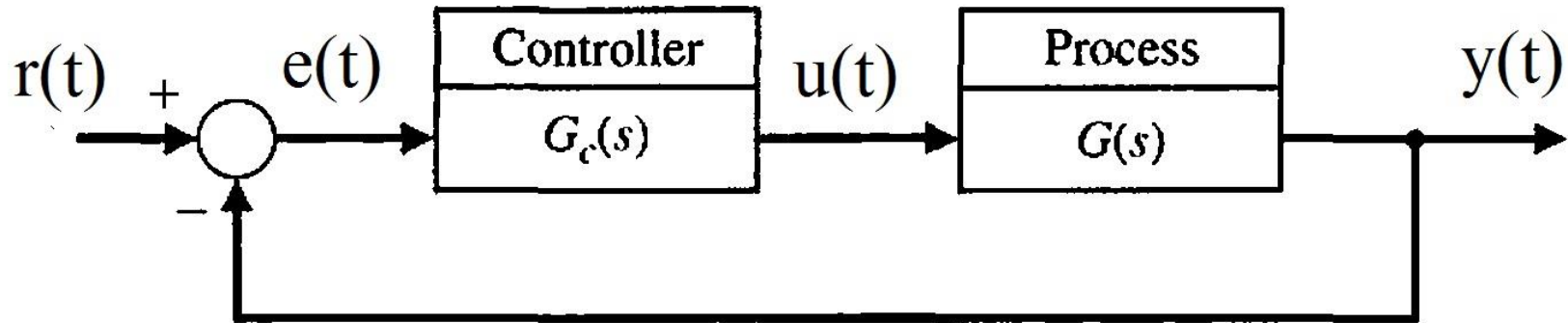


**Indian Institute of Information Technology Guwahati
Bongora, Guwahati-781015**

The Design of Feedback Control Systems

- **PID**
- **Compensation Networks**

Closed-loop system with PID controller

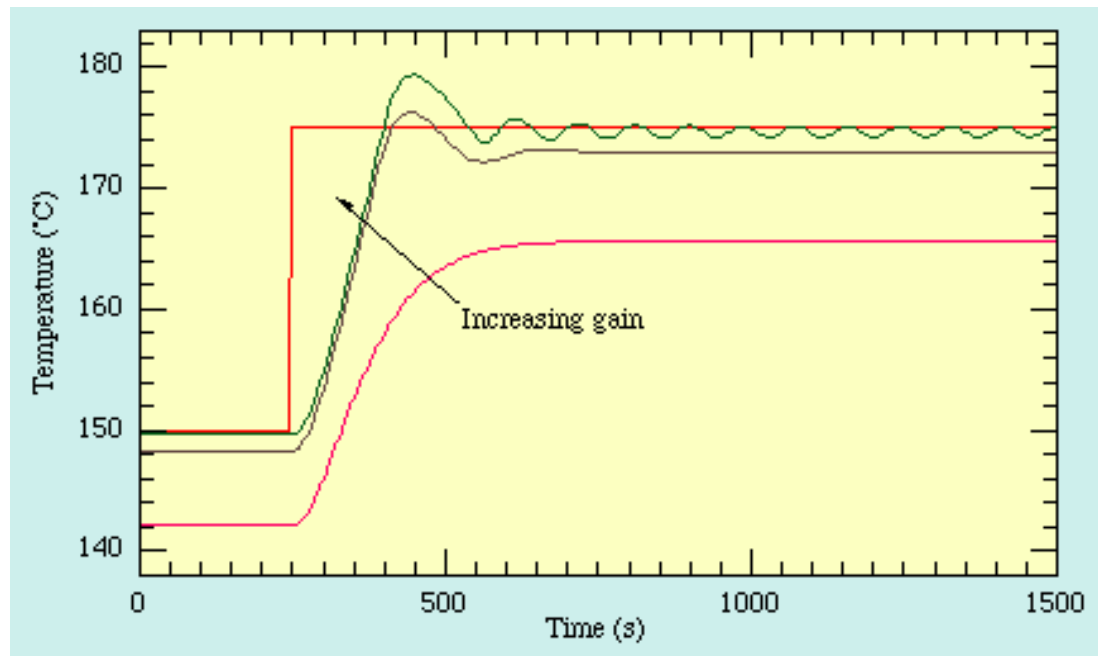


$$u(t) = K_p e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt}$$

$$G_c(s) = K_p + \frac{K_I}{s} + K_D s$$

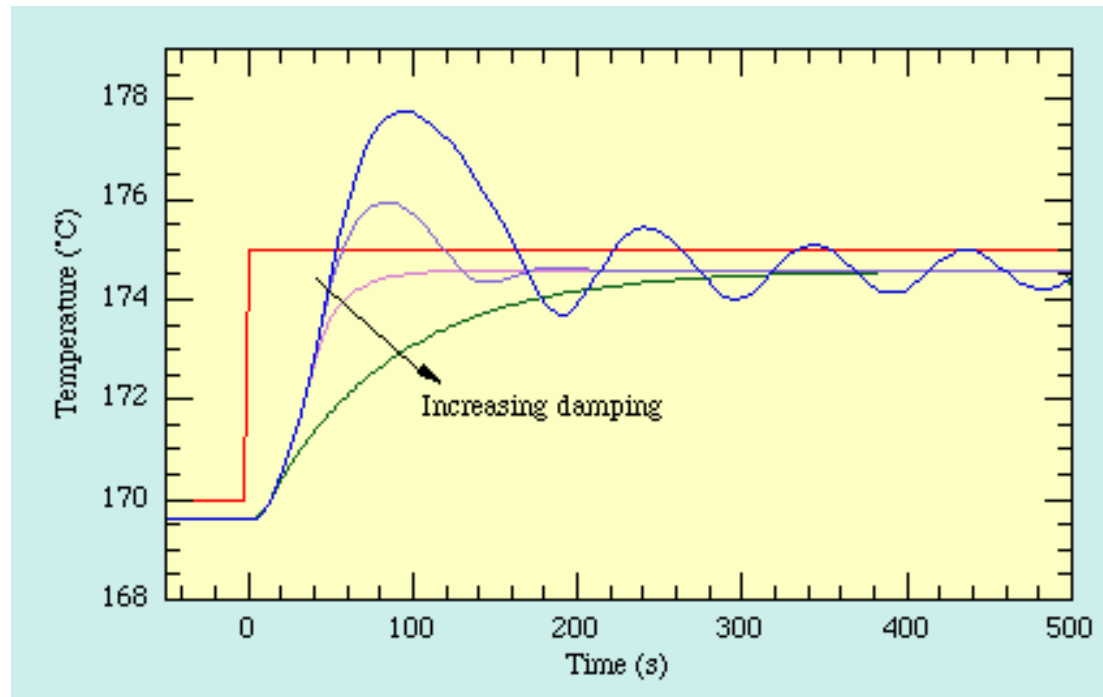
Proportional Control

A proportional controller attempts to perform better than the On-off type by applying power in proportion to the difference in temperature between the measured and the set-point. As the gain is increased the system responds faster to changes in set-point but becomes progressively underdamped and eventually unstable. The final temperature lies below the set-point for this system because some difference is required to keep the heater supplying power.



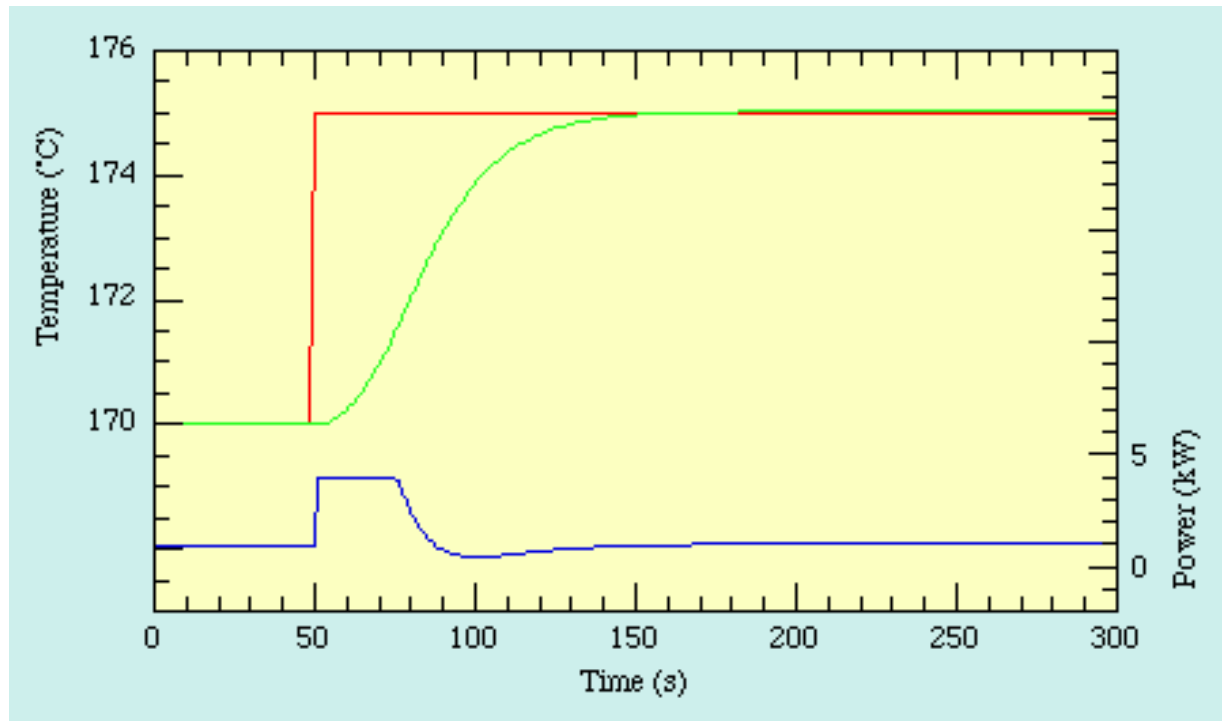
Proportional, Derivative Control

The stability and overshoot problems that arise when a proportional controller is used at high gain can be mitigated by adding a term proportional to the time-derivative of the error signal. The value of the damping can be adjusted to achieve a critically damped response.



Proportional+Integral+Derivative Control

Although PD control deals neatly with the overshoot and ringing problems associated with proportional control it does not cure the problem with the steady-state error. Fortunately it is possible to eliminate this while using relatively low gain by adding an integral term to the control function which becomes



The Characteristics of P, I, and D controllers

A proportional controller (K_p) will have the effect of reducing the rise time and will reduce, but never eliminate, the steady-state error.

An integral control (K_i) will have the effect of eliminating the steady-state error, but it may make the transient response worse.

A derivative control (K_d) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response.

Proportional Control

By only employing proportional control, a steady state error occurs.

Proportional and Integral Control

The response becomes more oscillatory and needs longer to settle, the error disappears.

Proportional, Integral and Derivative Control

All design specifications can be reached.

The Characteristics of P, I, and D controllers

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
K_p ↑	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small Change	Decrease	Decrease	Small Change

Tips for Designing a PID Controller

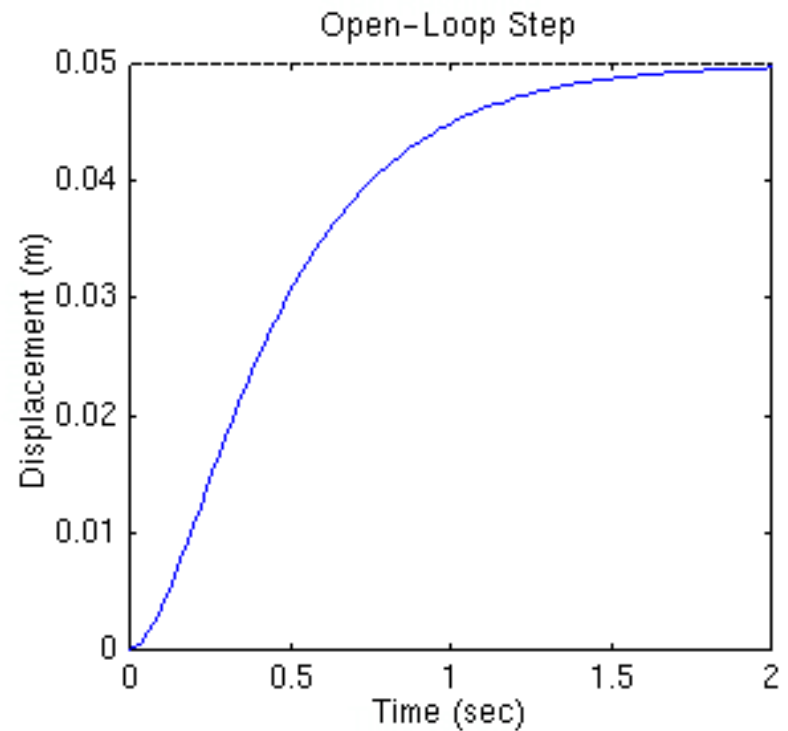
1. Obtain an open-loop response and determine what needs to be improved
2. Add a proportional control to improve the rise time
3. Add a derivative control to improve the overshoot
4. Add an integral control to eliminate the steady-state error
5. Adjust each of K_p , K_i , and K_d until you obtain a desired overall response.

Lastly, please keep in mind that you do not need to implement all three controllers (proportional, derivative, and integral) into a single system, if not necessary. For example, if a PI controller gives a good enough response (like the above example), then you don't need to implement derivative controller to the system. Keep the controller as simple as possible.

Open-Loop Control - Example

$$G(s) = \frac{1}{s^2 + 10s + 20}$$

```
num=1;  
den=[1 10 20];  
step(num,den)
```



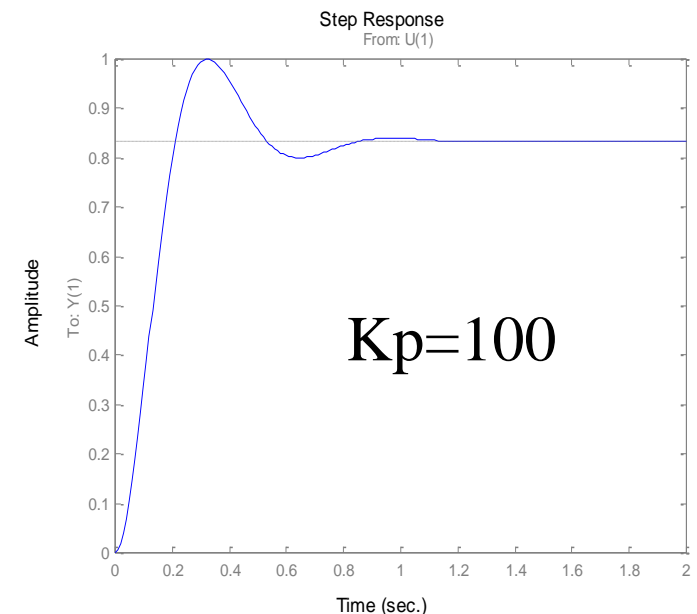
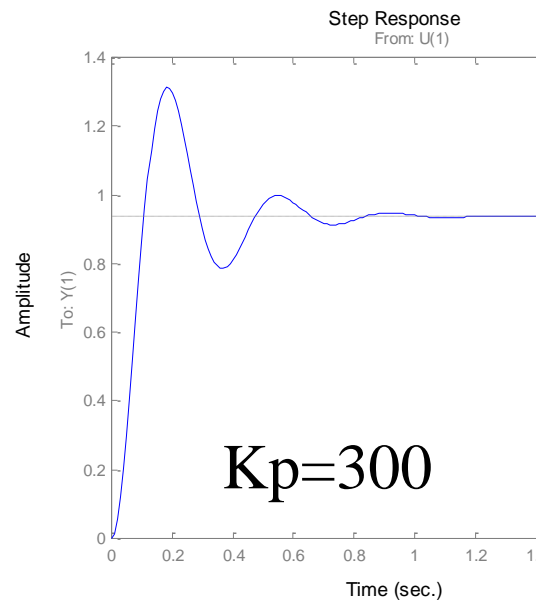
Proportional Control - Example

The proportional controller (K_p) reduces the rise time, increases the overshoot, and reduces the steady-state error.

$$T(s) = \frac{K_p}{s^2 + 10s + (20 + K_p)}$$

MATLAB Example

```
Kp=300;  
num=[Kp];  
den=[1 10 20+Kp];  
t=0:0.01:2;  
step(num,den,t)
```



Proportional - Derivative - Example

The derivative controller (K_d) reduces both the overshoot and the settling time.

MATLAB Example

$$T(s) = \frac{K_d \cdot s + K_p}{s^2 + (10 + K_d) \cdot s + (20 + K_p)}$$

$K_p=300;$

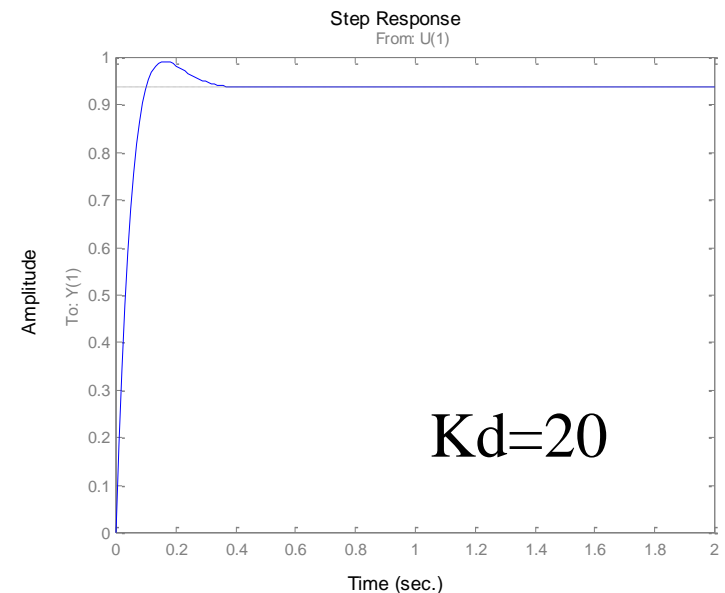
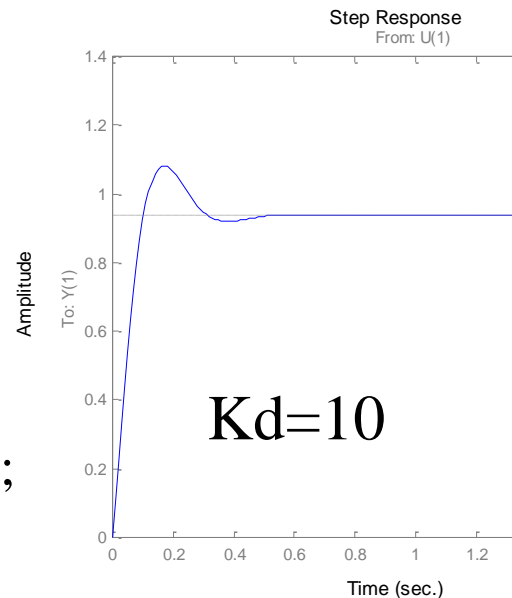
$K_d=10;$

$\text{num}=[K_d \ K_p];$

$\text{den}=[1 \ 10+K_d \ 20+K_p];$

$t=0:0.01:2;$

$\text{step}(\text{num},\text{den},t)$



Proportional - Integral - Example

The integral controller (K_i) decreases the rise time, increases both the overshoot and the settling time, and eliminates the steady-state error

$$T(s) = \frac{K_p \cdot s + K_i}{s^3 + 10s^2 + (20 + K_p) \cdot s + K_i}$$

MATLAB Example

$K_p=30$;

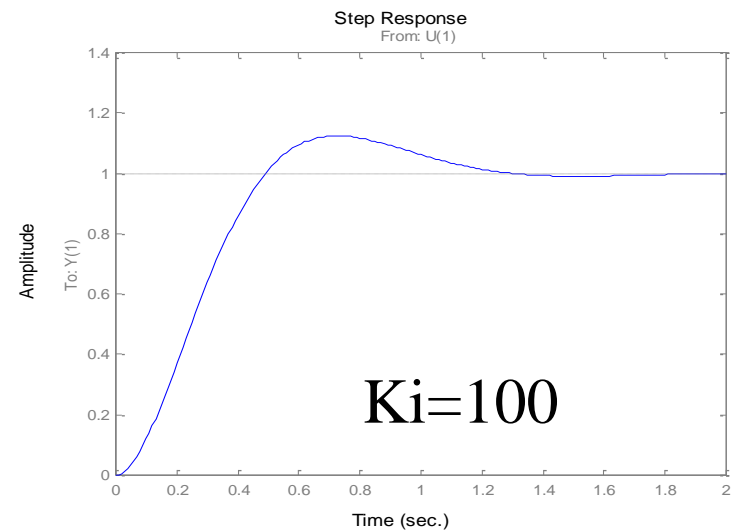
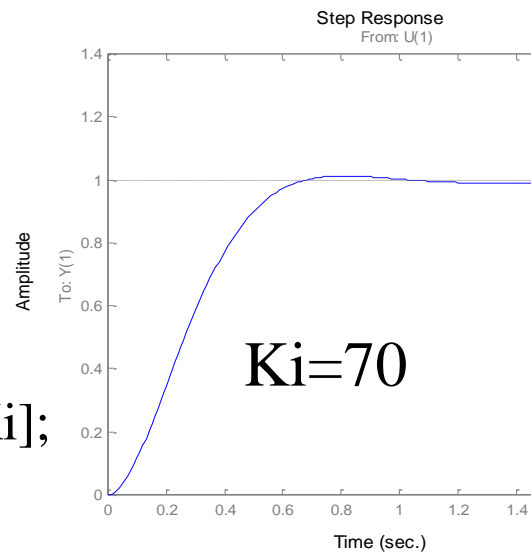
$K_i=70$;

$\text{num}=[K_p \ K_i]$;

$\text{den}=[1 \ 10 \ 20+K_p \ K_i]$;

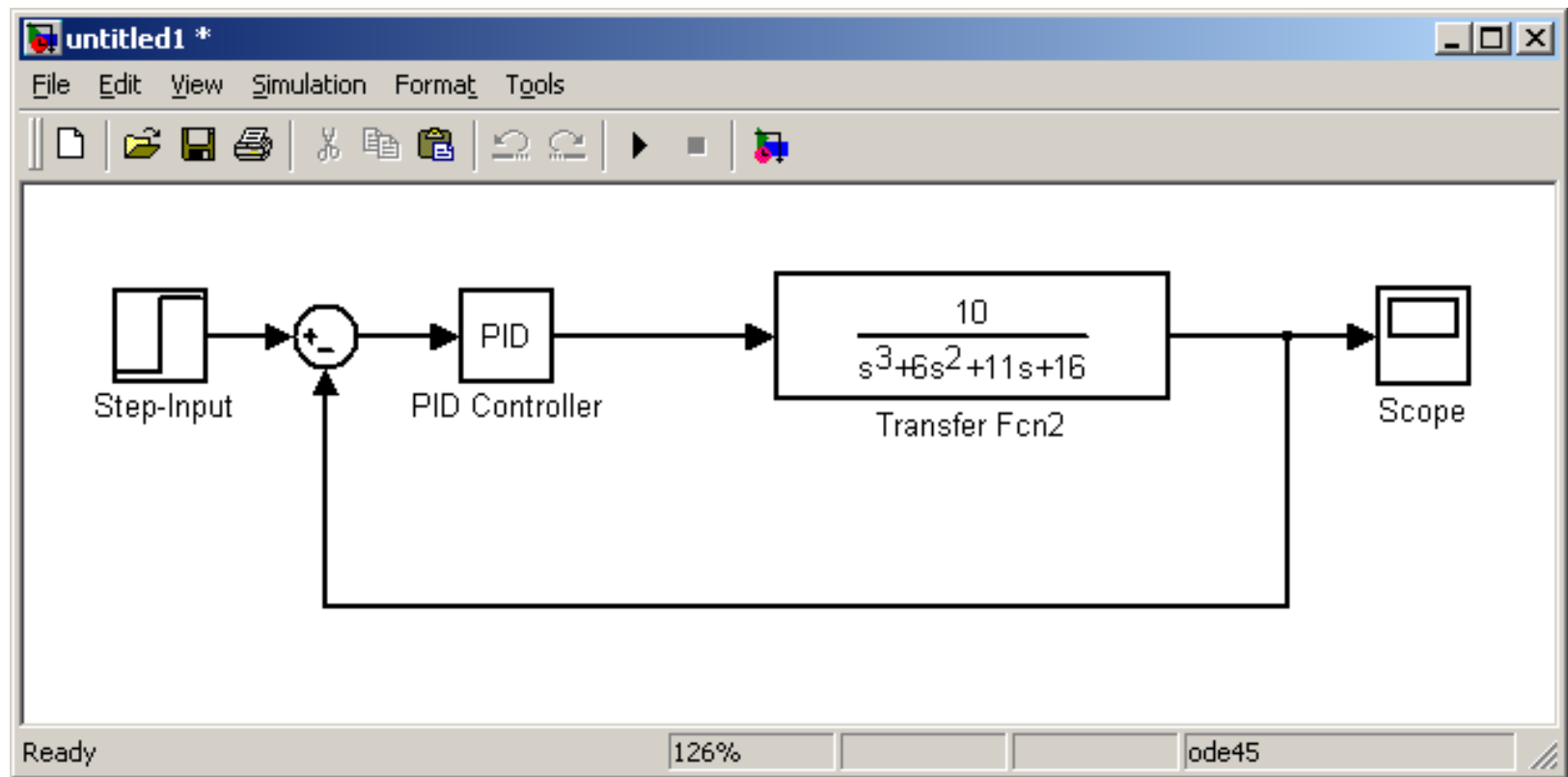
$t=0:0.01:2$;

$\text{step}(\text{num},\text{den},t)$



Example - Practice

Consider the following configuration:



Example - Practice

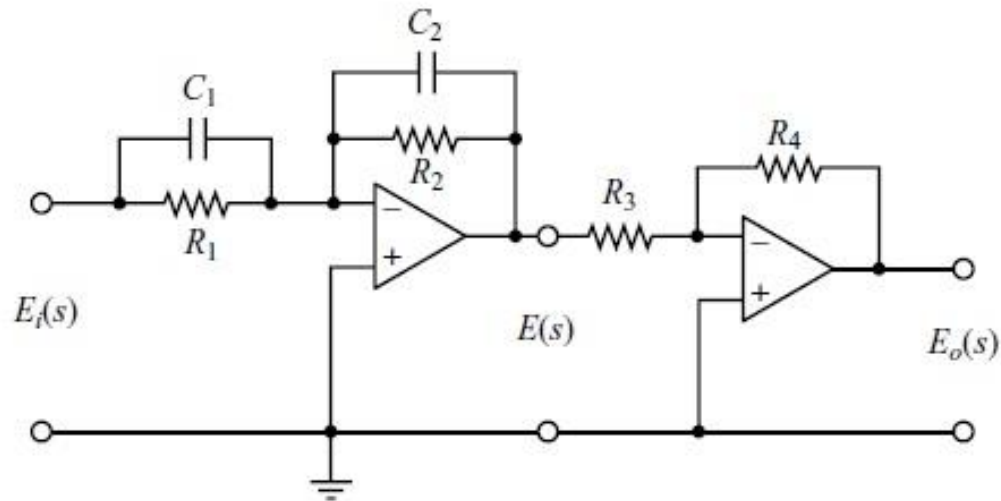
The design a system for the following specifications:

- Zero steady state error
- Settling time within 5 seconds
- Rise time within 2 seconds
- Only some overshoot permitted

Lead or Lag Compensator

A first-order lead or lag compensator can be realized as

$$G_c(s) = \frac{(s + z)}{(s + p)}$$



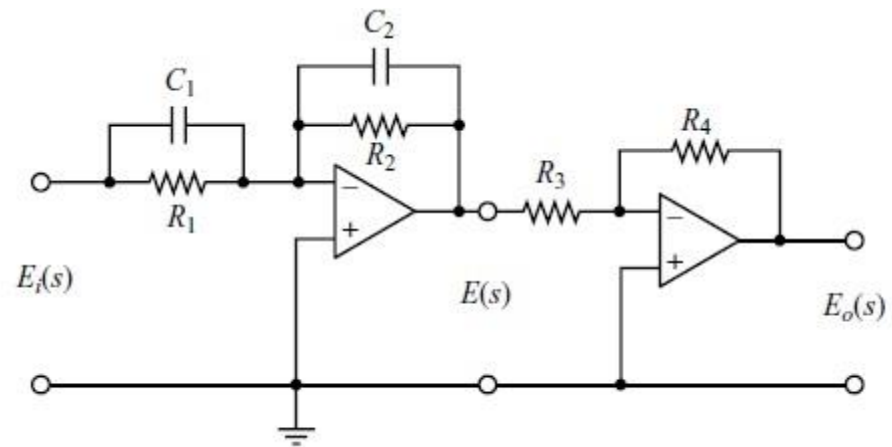
$$\frac{E_o(s)}{E_i(s)} = \frac{R_2 R_4}{R_1 R_3} \frac{R_1 C_1 s + 1}{R_2 C_2 s + 1} = \frac{R_4 C_1}{R_3 C_2} \frac{s + \frac{1}{R_1 C_1}}{s + \frac{1}{R_2 C_2}}$$

$$= K_c \alpha \frac{T s + 1}{\alpha T s + 1} = K_c \frac{s + \frac{1}{T}}{s + \frac{1}{\alpha T}}$$

where

$$T = R_1 C_1, \quad \alpha T = R_2 C_2, \quad K_c = \frac{R_4 C_1}{R_3 C_2}$$

Electronic circuit
that is a lead network
if $R_1 C_1 > R_2 C_2$ and a
lag network if
 $R_1 C_1 < R_2 C_2$.



Lead or Phase-Lead Compensator Using Root Locus

A first-order lead compensator can be designed using the root locus. A lead compensator in root locus form is given by

$$G_c(s) = \frac{(s + z)}{(s + p)}$$

where the magnitude of z is less than the magnitude of p . A phase-lead compensator tends to shift the root locus toward the left half plane. This results in an improvement in the system's stability and an increase in the response speed.

When a lead compensator is added to a system, the value of this intersection will be a larger negative number than it was before. The net number of zeros and poles will be the same (one zero and one pole are added), but the added pole is a larger negative number than the added zero. Thus, the result of a lead compensator is that the asymptotes' intersection is moved further into the left half plane, and the entire root locus will be shifted to the left. This can increase the region of stability as well as the response speed.

Lead or Phase-Lead Compensator Using Root Locus

In Matlab a phase lead compensator in root locus form is implemented by using the transfer function in the form

```
numlead=kc*[1 z];  
denlead=[1 p];
```

and using the `conv()` function to implement it with the numerator and denominator of the plant

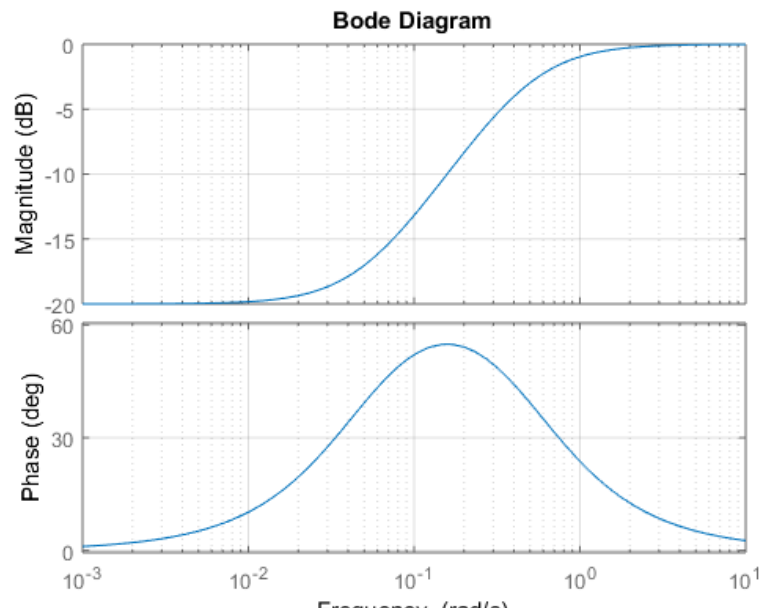
```
newnum=conv(num,numlead);  
newden=conv(den,denlead);
```

Lead or Phase-Lead Compensator Using Frequency Response

A first-order phase-lead compensator can be designed using the frequency response. A lead compensator in frequency response form is given by

$$G_C(s) = \frac{(1 + \alpha \cdot \tau \cdot s)}{\alpha \cdot (1 + \tau \cdot s)} \quad p = \frac{1}{\tau} \quad z = \frac{1}{\alpha \tau} \quad \omega_m = \sqrt{z \cdot p} \quad \sin(\phi_m) = \frac{\alpha - 1}{\alpha + 1}$$

In frequency response design, the phase-lead compensator adds positive phase to the system over the frequency range. A bode plot of a phase-lead compensator looks like the following



Lead or Phase-Lead Compensator Using Frequency Response

Additional positive phase increases the phase margin and thus increases the stability of the system. This type of compensator is designed by determining α from the amount of phase needed to satisfy the phase margin requirements, and determining τ to place the added phase at the new gain-crossover frequency.

Another effect of the lead compensator can be seen in the magnitude plot. The lead compensator increases the gain of the system at high frequencies (the amount of this gain is equal to α). This can increase the crossover frequency, which will help to decrease the rise time and settling time of the system.

Lead or Phase-Lead Compensator Using Frequency Response

In Matlab, a phase lead compensator in frequency response form is implemented by using the transfer function in the form

```
numlead=[aT 1];
```

```
denlead=[T 1];
```

and using the `conv()` function to multiply it by the numerator and denominator of the plant

```
newnum=conv(num,numlead);
```

```
newden=conv(den,denlead);
```

Lag or Phase-Lag Compensator Using Root Locus

A first-order lag compensator can be designed using the root locus. A lag compensator in root locus form is given by

$$G_c(s) = \frac{(s + z)}{(s + p)}$$

where the magnitude of z is greater than the magnitude of p . A phase-lag compensator tends to shift the root locus to the right, which is undesirable. For this reason, the pole and zero of a lag compensator must be placed close together (usually near the origin) so they do not appreciably change the transient response or stability characteristics of the system.

When a lag compensator is added to a system, the value of this intersection will be a smaller negative number than it was before. The net number of zeros and poles will be the same (one zero and one pole are added), but the added pole is a smaller negative number than the added zero. Thus, the result of a lag compensator is that the asymptotes' intersection is moved closer to the right half plane, and the entire root locus will be shifted to the right.

Lag or Phase-Lag Compensator Using Root Locus

It was previously stated that that lag controller should only minimally change the transient response because of its negative effect. If the phase-lag compensator is not supposed to change the transient response noticeably, what is it good for? The answer is that a phase-lag compensator can improve the system's steady-state response. It works in the following manner. At high frequencies, the lag controller will have unity gain. At low frequencies, the gain will be z_0/p_0 which is greater than 1. This factor z/p will multiply the position, velocity, or acceleration constant (K_p , K_v , or K_a), and the steady-state error will thus decrease by the factor z_0/p_0 . In Matlab, a phase lead compensator in root locus form is implemented by using the transfer function in the form

```
numlag=[1 z];
```

```
denlag=[1 p];
```

and using the `conv()` function to implement it with the numerator and denominator of the plant

```
newnum=conv(num,numlag);
```

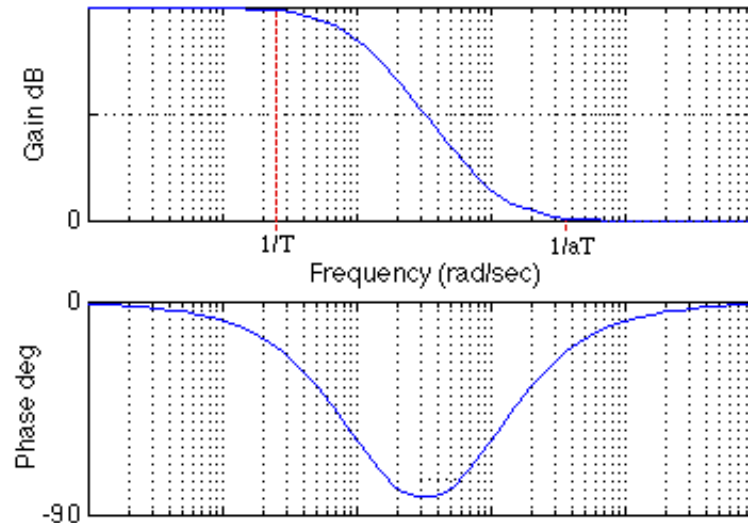
```
newden=conv(den,denlag);
```

Lag or Phase-Lag Compensator using Frequency Response

A first-order phase-lag compensator can be designed using the frequency response. A lag compensator in frequency response form is given by

$$G_c(s) = \frac{(1 + \alpha \cdot \tau \cdot s)}{\alpha \cdot (1 + \tau \cdot s)}$$

The phase-lag compensator looks similar to a phase-lead compensator, except that α is now less than 1. The main difference is that the lag compensator adds negative phase to the system over the specified frequency range, while a lead compensator adds positive phase over the specified frequency. A bode plot of a phase-lag compensator looks like the following



Lag or Phase-Lag Compensator using Frequency Response

In Matlab, a phase-lag compensator in frequency response form is implemented by using the transfer function in the form

$$\text{numlead} = [a \ T \ 1];$$
$$\text{denlead} = a \ [T \ 1];$$

and using the `conv()` function to implement it with the numerator and denominator of the plant

$$\text{newnum} = \text{conv}(\text{num}, \text{numlead});$$
$$\text{newden} = \text{conv}(\text{den}, \text{denlead});$$

Lead-lag Compensator using either Root Locus or Frequency Response

A lead-lag compensator combines the effects of a lead compensator with those of a lag compensator. The result is a system with improved transient response, stability and steady-state error. To implement a lead-lag compensator, first design the lead compensator to achieve the desired transient response and stability, and then add on a lag compensator to improve the steady-state response

Exercise - Dominant Pole-Zero Approximations and Compensations

The influence of a particular pole (or pair of complex poles) on the response is mainly determined by two factors: the real part of the pole and the relative magnitude of the residue at the pole. The real part determines the rate at which the transient term due to the pole decays; the larger the real part, the faster the decay. The relative magnitude of the residue determines the percentage of the total response due to a particular pole.

Investigate (using Simulink) the impact of a closed-loop negative real pole on the overshoot of a system having complex poles.

$$T(s) = \frac{pr \cdot \omega_n^2}{(s + pr) \cdot [s^2 + (2 \cdot \zeta \cdot \omega_n \cdot s) + \omega_n^2]}$$

Make pr to vary (2, 3, 5) times the real part of the complex pole for different values of (0.3, 0.5, 0.7).

Investigate (using Simulink) the impact of a closed-loop negative real zero on the overshoot of a system having complex poles.

$$T(s) = \frac{(s + zr)}{[s^2 + (2 \cdot \zeta \cdot \omega_n \cdot s) + \omega_n^2]}$$

Make zr to vary (2, 3, 5) times the real part of the complex pole for different values of (0.3, 0.5, 0.7).

Exercise - Lead and Lag Compensation

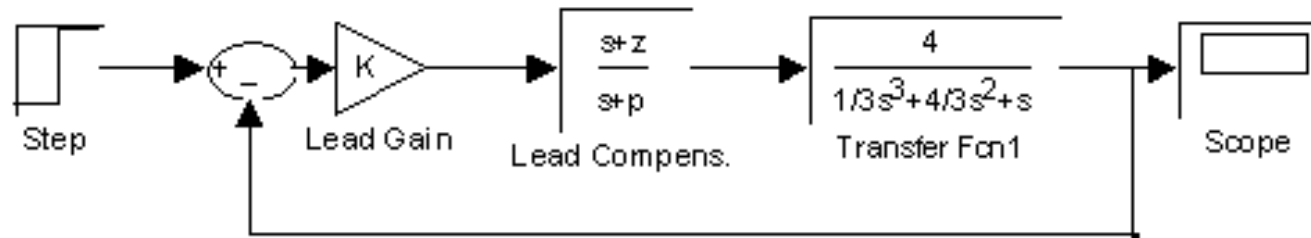
Investigate (using Matlab and Simulink) the effect of lead and lag compensations on the two systems indicated below. Summarize your observations. Plot the root-locus, bode diagram and output for a step input before and after the compensations.

Remember

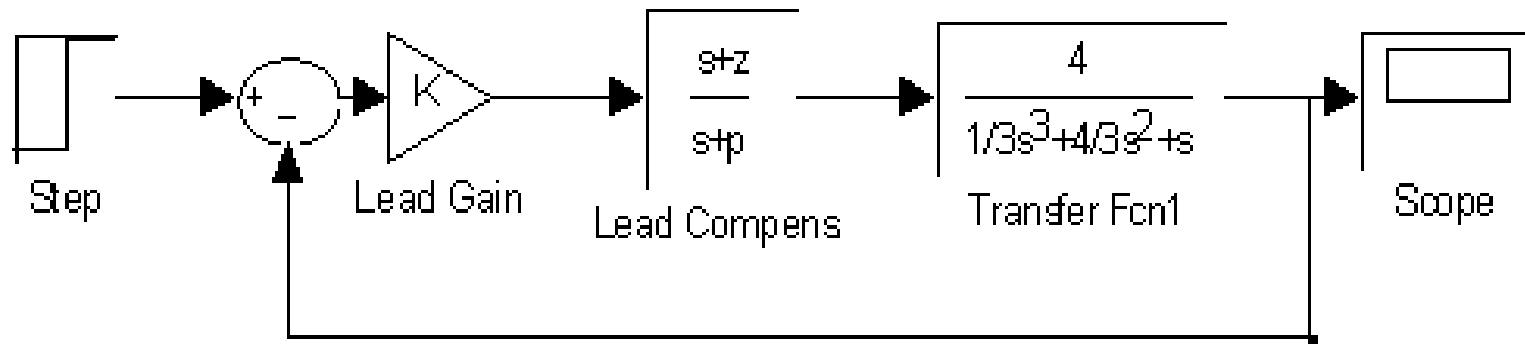
lead compensation: $z < p$ (place zero below the desired root location or to the left of the first two real poles)

lag compensation: $z > p$ (locate the pole and zero near the origin of the s-plane)

Lead Compensation (use $z=1.33$, $p=20$ and $K=15$).



Lag Compensation (use $z=0.09$, and $p=0.015$, $K=1/6$)



Summarize your findings

Problem 10.36

Determine a compensator so that the percent overshoot is less than 20% and K_v (velocity constant) is greater than 8.

