LABORATORY INSTRUCTION MANUAL

CONTROL SYSTEM II LAB

EE 693

ELECTRICAL ENGINEERING DEPARTMENT
JIS COLLEGE OF ENGINEERING
(AN AUTONOMOUS INSTITUTE)
KALYANI, NADIA

EXPERIMENT NO : CS – II/1

TITLE : FAMILIARIZATION WITH DIGITAL CONTROL SYSTEM TOOLBOX

OBJECTIVE : To study

I. Conversion of a transfer function from continuous domain to discrete domain.
II. Conversion of a transfer function from continuous domain to digital domain.
III. Pole Zero Map of a discrete transfer function.

Software Used: MATLAB/ SIMULINK

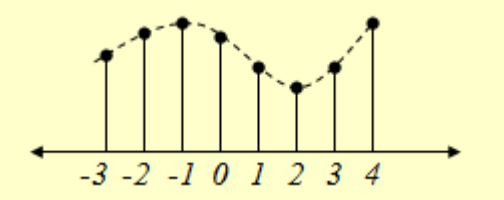
THEORY :

Sampling is a process by which a continuous time system can be converted to discrete domain. Discrete time signal x[n] often arises from periodic sampling of continuous time signal $x_c(t)$:

$$x[n] = x_c(nT) \qquad -\infty<n<\infty$$

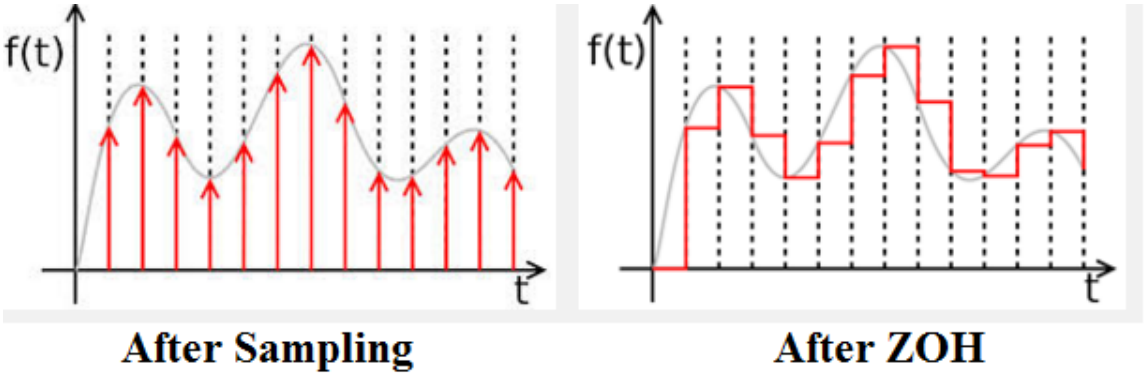
This system is called continuous to discrete time converter or sampler

T is the sampling period in second, $f_s = 1/T$ is the sampling frequency in Hz, Sampling frequency in radian-per-second $\Omega_s=2\pi f_s$ rad/sec.



After sampling a continuous signal an impulse train will be obtained. That impulses are difficult to generate and transmit. So it is more convenient to generate sample signal in form of Zero order hold. Such system samples a signal x(t) at a given instant and holds that value till the next instant at which sample is taken. The transfer function of ZOH is

$$G(s)_{zoh} = \frac{1-e^{-Ts}}{s}$$



MATLAB Commands used:

sysd = c2d (sys,Ts) is used to convert a transfer function (sys) from continuous to discrete domain at a sampling time of Ts.

sysd = c2d (sys,Ts, 'method') is used to convert a transfer function (sys) from continuous to discrete domain at a sampling time of Ts using a specific method such as ZOH or foh or tustin method.

pzmap(sysd) is used to obtain pole zero map of the discrete transfer function (sysd).

Example 1: To convert a transfer function (sys) from continuous to discrete domain at a sampling time of Ts.

```
num = [1]
den= [1 1 0]
Ts=0.1
sys = tf (num,den)
sysd = c2d(sys,Ts)

Output :

sys:
   1
-------
s^2 + s

sysd :

0.004837 z + 0.004679
----------------------
z^2 - 1.905 z + 0.9048

Sampling time: 0.1
```

Example 2: To convert a transfer function (sys) from continuous to discrete domain at a sampling time of Ts using ZOH method.

```
num = [1]
den= [1 1 0]
Ts=0.1
sys = tf (num,den)
sysd = c2d(sys,Ts, 'zoh')

Output :
sys:
   1
-------
s^2 + s

 sysd:
0.004837 z + 0.004679
----------------------
z^2 - 1.905 z + 0.9048

Sampling time: 0.1
```

Example 3: To convert a transfer function (sys) from continuous to discrete domain at a sampling time of Ts using FOH method.

```
num = [1]
den= [1 1 0]
Ts=0.1
sys = tf (num,den)
sysd = c2d(sys,Ts, 'foh')
```

Outputs:

sys:

```
  1
-------
s^2 + s
```

sysd:

```
0.001626 z^2 + 0.006344 z + 0.001547
-------------------------------------
     z^2 - 1.905 z + 0.9048
```

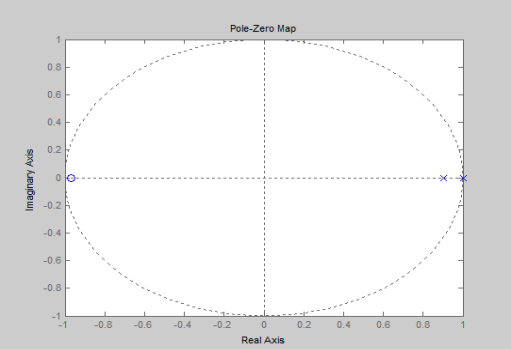
Sampling time: 0.1

Example 4: To convert a transfer function (sys) from continuous to discrete domain at a sampling time of Ts using FOH method.

```
num = [1]
den= [1 1 0]
Ts=0.1
sys = tf (num,den)
sysd = c2d(sys,Ts, 'tustin')

Outputs:

sys:
   1
-------
s^2 + s


sysd:
0.002381 z^2 + 0.004762 z + 0.002381
-------------------------------------
       z^2 - 1.905 z + 0.9048

Sampling time: 0.1
```

Example 5: Obtaining pole zero map of a discrete transfer function

```
num = [1]
den= [1 1 0]
Ts=0.1
sys = tf (num,den)
sysd = c2d(sys,Ts, 'zoh')
pzmap(sysd)
```



Assignments:

Obtain discrete domain transfer functions and pole zero maps of the following s – domain functions using
(a) ZOH method (b) FOH method (c) Tustin method

1. G(s) = $\frac{1}{s^2+s+4}$ 2. G(s) = $\frac{5}{s^2+9}$ 3. G(s) = $\frac{1}{(s^2+3s+5)(s+3)(s+5)}$

DISCUSSION :

1. Why continuous signal is to be converted in discrete domain?
2. What is the function of ZOH device?
3. What is sampling?

EXPERIMENT NO : CS– II/2

TILLE : DETERMINATION OF Z – TRANSFORM, INVERSE Z- TRANSFORM & POLE ZERO MAP OF DISCRETE SYSTEMS

OBJECTIVE : To determine

I. Z transform of a discrete time signal
II. Inverse Z transform of a discrete time signal
III. Factored form and partial fraction form of a rational z function
IV. Pole zero map of a digital system

Software Used: MATLAB

THEORY :

In mathematics and signal processing, the **Z-transform** converts a discrete time-domain signal, which is a sequence of real or complex numbers, into a complex frequency-domain representation.

The Z-transform, like many other integral transforms, can be defined as either a one-sided or two-sided transform.

Bilateral Z-transform

The bilateral or two-sided Z-transform of a discrete-time signal x[n] is the function X(z) defined as

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$$

.

Unilateral Z-transform

Alternatively, in cases where $x[n]$ is defined only for $n \geq 0$, the *single-sided* or *unilateral* Z-transform is defined as

$$X(z) = \mathcal{Z}\{x[n]\} = \sum_{n=0}^{\infty} x[n] z^{-n}$$

In signal processing, this definition is used when the signal is causal.

Inverse Z – Transform :

The inverse Z transform is defined as $x[n] = Z^{-1}[x(z)]$

Region of Convergence:

The set of values of z for which the z-transform G(z) converges is called its region of convergence (ROC)

Useful Commands:

ztrans is used to obtain z transform of a discrete time signal

iztrans is used to obtain inverse z transform of a z function

zp2sos is used to convert a transfer function from rational form to factored form.

resideuz is used to convert a transfer function from rational form to partial fraction form.

zplane (num,den) is used to obtain pole zero map of a Z function.

Example:

1. **Write a matlab code to obtain Z transform of the following discrete function**

X[n] = $\frac{1}{4^n}$ u[n]

Matlab Code:

```
syms z n
ztrans(1/4^n)

Output:
z/(z - 1/4)
```

2. **Write a matlab code to obtain Inverse Z transform of the following Z function**

X(z) = $\frac{2z}{2z-1}$

Matlab Code:

```
syms z n
iztrans(2*z/(2*z-1))
```

Output:
(1/2)^n

3. **Write a matlab code to convert rational form of the following z function in factored form**

$$G(z)= \frac{2z^4+16z^3+44z^2+56z+32}{3z^4+3z^3-15z^2+18z-12}$$

Matlab Code:

```
syms z n
num = [2 16 44 56 32]
den = [3 3 -15 18 -12]
[z,p,k] = tf2zp(num,den)
gzfct = zp2sos(z,p,k)
```

Output :

num =

2 16 44 56 32

den =

3 3 -15 18 -12

z =

-4.0000

-2.0000

-1.0000 + 1.0000i

-1.0000 - 1.0000i

p =

-3.2361

1.2361

0.5000 + 0.8660i

0.5000 - 0.8660i

k =

0.6667

gzfct =

0.6667 4.0000 5.3333 1.0000 2.0000 -4.0000

1.0000 2.0000 2.0000 1.0000 -1.0000 1.0000

4. Write a matlab code to convert rational form of the following z function in partial fraction form

$$G(z) = \frac{18z^3}{18z^3+3z^2-4z-1}$$

Matlab Code:

```
syms z n
num = [18 0 0 0]
den = [18 3 -4 -1]
[r,p,k] = residuez(num,den)

Outputs :
num =
    18     0     0     0
den =
    18     3    -4    -1
r =
    0.3600
    0.2400
    0.4000
p =
    0.5000
   -0.3333
   -0.3333
k =
     0
```

5. **Write a matlab code to obtain pole zero map of the following z function**

$$X(Z) = [Z^{-2} + Z^{-1}] / [1-2Z^{-1}+3Z^{-2}]$$

Matlab Code:

```
syms z n
num = [0 1 1]
den = [1 -2 3]
zeros = roots(num)
poles = roots(den)
zplane(num,den)
```

Output:

num =

0 1 1

den =

1 -2 3

zeros =

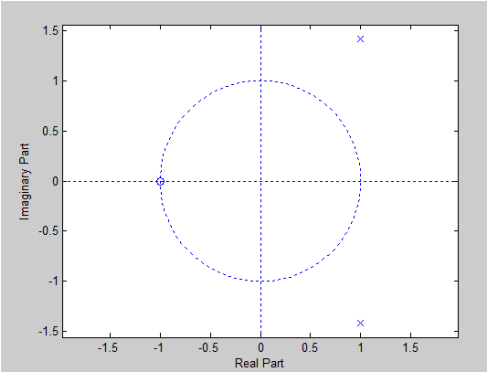
-1

poles =

1.0000 + 1.4142i

1.0000 - 1.4142i

Pole zero map



Assignments:

1. Determine Z transform of the following discrete functions:

   (a) X[n] = $(1/16^n)$ u(n) (b) x[n] = 0.5^n u(n)

2. Determine inverse Z transform of the following discrete functions:

   (a) X[z]= 3*Z/(Z+1) (b) x[z] = $\frac{18z^3}{18z^3+3z^2-4z-1}$

3. Obtain factored form and partial fraction form of the following z function, find pole zero values and also plot pole zero map using MATLAB.

$$X[z] = \frac{2z^4+16z^3+44z^2+56z+32}{3z^4+3z^3-15z^2+18z-12}$$

DISCUSSION :

1. What is Z transform?
2. Why Z transformation is needed in discrete systems?

EXPERIMENT NO : CS– II/3

TILLE : TO STUDY STEP RESPONSE OF A DISCRETE TIME SYSTEM AND EFFECT OF SAMPLING TIME ON SYSTEM RESPONSE

OBJECTIVE : To study

I. Closed loop response of a discrete time system
II. Comparison of time responses of continuous time and discrete time systems
III. Effect of sampling time on system response and system parameters

Software Used: MATLAB

THEORY :

Absolute stability is a basic requirement of all control systems. Apart from that, good relative stability and steady state accuracy are also required in any control system, whether continuous time or discrete time. Transient response corresponds to the system closed loop poles and steady state response corresponds to the excitation poles or poles of the input function.

In many practical control systems, the desired performance characteristics are specified in terms of time domain quantities. Unit step input is most commonly used in analysis of a system since it is easy to generate and represent a sufficiently drastic change thus providing useful information on both transient and steady state responses. The transient response of a system depends on the initial conditions. It is a common practice to consider the system initially at rest. Consider the digital control system shown in Figure1

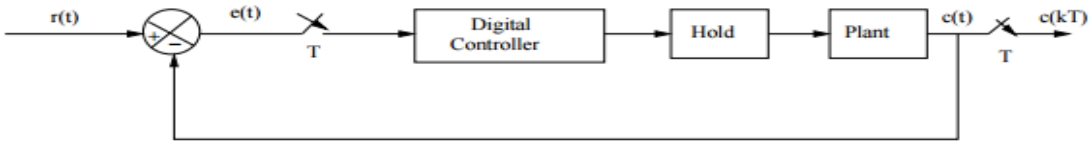


Figure 1: Block Diagram of a closed loop digital system

Similar to the continuous time case, transient response of a digital control system can also be characterized by the following.

1. Rise time (tr): Time required for the unit step response to rise from 0% to 100% of its final value in case of underdamped system or 10% to 90% of its final value in case of overdamped system.

2. Delay time (td): Time required for the the unit step response to reach 50% of its final value.

3. Peak time (tp): Time at which maximum peak occurs.

4. Peak overshoot (Mp): The difference between the maximum peak and the steady state value of the unit step response.

5. Settling time (ts): Time required for the unit step response to reach and stay within 2% or 5% of its steady state value.

However since the output response is discrete the calculated performance measures may be slightly different from the actual values. Figure 2 illustrates this. The output has a maximum value cmax whereas the maximum value of the discrete output is c ∗ max which is always less than or equal to cmax. If the sampling period is small enough compared to the oscillations of the response then this difference will be small otherwise c ∗ max may be completely erroneous.

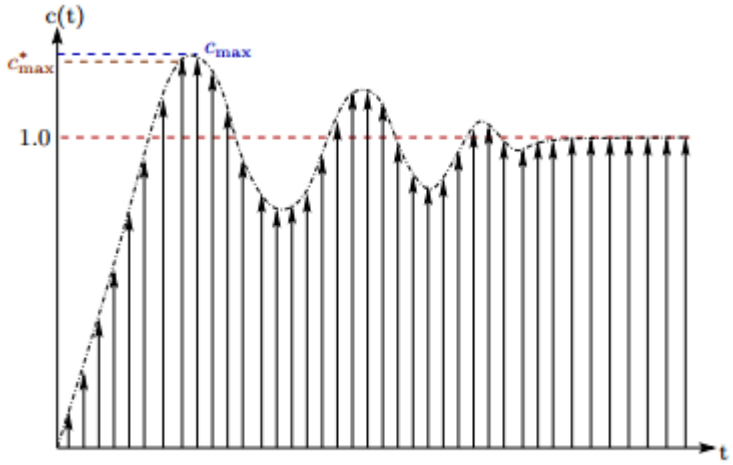


Figure 2: Unit step response of a discrete time system

Example 1: Consider a unity feedback control system having forward path transfer function G(s) = $\frac{1}{s(s+1)}$.

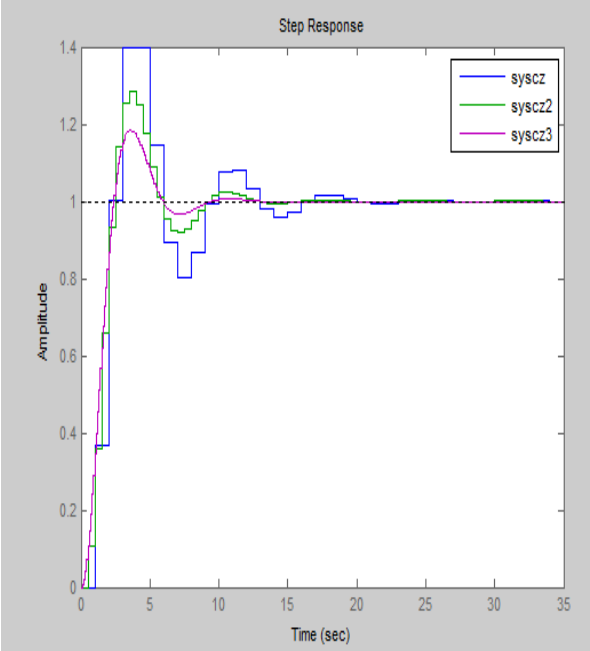
Determine (i) step response in continuous and discrete domain.

(ii) effect of sampling time on system response

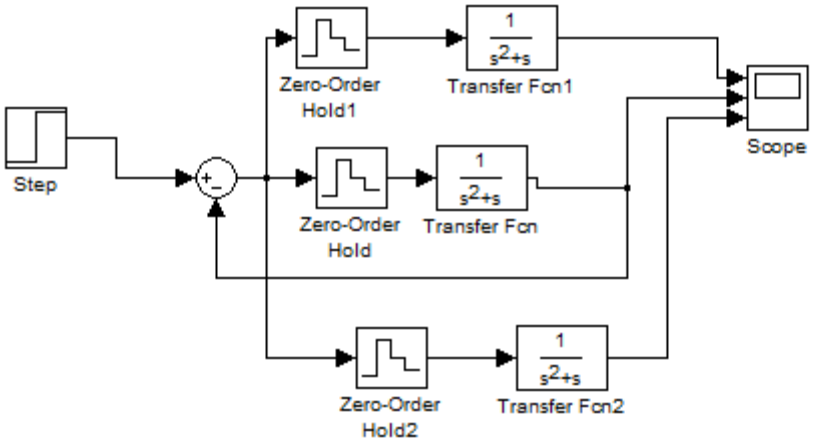
Matlab Code:

```
n=[1]
d=[1 1 0]
sys=tf(n,d)
sysz=c2d(sys,1,'zoh')
sysc=feedback(sys,1)
syscz=feedback(sysz,1)
step(syscz,'b')
hold on
sysz1=c2d(sys,.5,'zoh')
syscz2=feedback(sysz1,1)
step(syscz2,'g')
hold on
sysz2=c2d(sys,.10,'zoh')
syscz3=feedback(sysz2,1)
step(syscz3,'m')
hold on
```

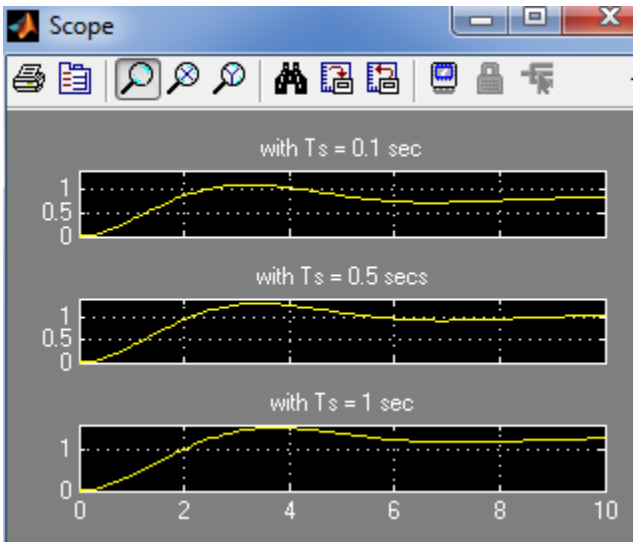
Matlab Result:



Simulink Model:



Simulink Output :



Assignments:

1. Build simulink model to obtain step response of a unity feedback system whose closed loop transfer function is given by:
   T(s) = $\frac{1}{s^2+s+1}$. Also show effect of sampling time on time response specification parameters.

2. Write down matlab code to obtain step response of a unity feedback system having forward path transfer function of G(s) = $\frac{1}{s^2+4s+3}$. Also show effect of sampling time on time response specification parameters.

Discussion:

1. How sampling time affects rise time, peak time, % overshoot, settling time of a system?
2. How practical sample and hold circuit works?
3. Derive expression for transfer function of a zero order hold.

EXPERIMENT NO : CS– II/4

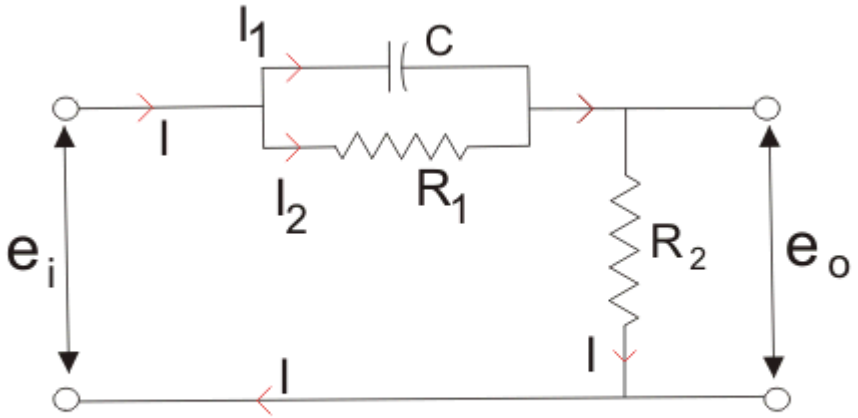
TILLE : DESIGN OF A LEAD COMPENSATOR USING BODE PLOT METHOD

OBJECTIVE : To design a lead compensator to obtain system response with desired accuracy, less overshoot.

Software Used: MATLAB

THEORY :

A system which has one pole and one dominating zero (the zero which is closer to the origin than all over zeros is known as dominating zero.) is known as lead network. If we want to add a dominating zero for **compensation in control system** then we have to select **lead compensation** network. The basic requirement of the phase lead network is that all poles and zeros of the transfer function of the network must lie on (-)ve real axis interlacing each other with a zero located at the origin of nearest origin. Given below is the circuit diagram for the phase **lead compensation** network.



Phase Lead Compensation Network From above circuit we get,

$$I_1 = C\frac{d}{dt}(e_i - e_o)$$

$$I_2 = \frac{e_i - e_o}{R_1}$$

$$I = I_1 + I_2 = C\frac{d}{dt}(e_i - e_o) + \frac{e_i - e_o}{R_1}$$

$$Again, I = \frac{e_o}{R_2}$$

Equating above expression of I we get,

$$\frac{e_o}{R_2} = C\frac{d}{dt}(e_i - e_o) + \frac{e_i - e_o}{R_1}$$

Now let us determine the transfer function for the given network and the transfer function can be determined by finding the ratio of the output voltage to the input voltage. So taking Laplace transform of both side of above equations,

$$\frac{1}{R_2}E_o(s) = \frac{1}{R_1}[E_i(s)-E_o(s)]+Cs[E_i(s)-E_o(s)] \quad (neglecting\ initial\ condition)$$

$$\Rightarrow \frac{1}{R_2}E_o(s) + \frac{1}{R_1}E_o(s) + CsE_o(s) = \frac{E_i(s)}{R_1} + CsE_i(s)$$

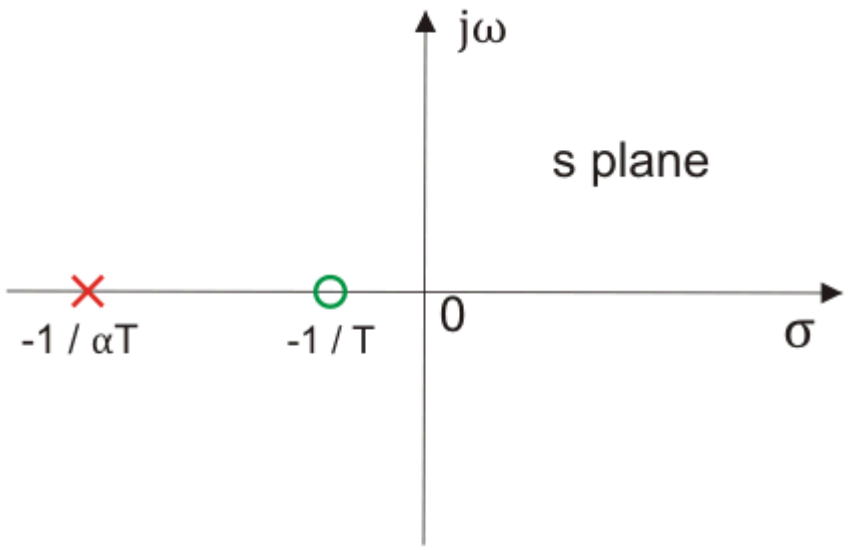
$$\Rightarrow \frac{E_o(s)}{E_i(s)} = \frac{\frac{1+sCR_1}{R_1}}{\frac{R_1+R_2+sR_1R_2C}{R_2R_1}}$$

$$\Rightarrow \frac{E_o(s)}{E_i(s)} = \frac{R_2}{R_1+R_2}\left[\frac{1+sCR_1}{1+\frac{sR_1R_2C}{R_1+R_2}}\right]$$

On substituting the $\alpha = (R_1 + R_2)/ R_2$ & $T = \{(R_1R_2) /(R_1 + R_2)\}$ in the above equation. Where T and α are respectively the time constant and attenuation constant, we have

$$Transfer\ function\ ,\ G_{lead}(s) = \frac{E_o(s)}{E_i s} = \frac{1}{\alpha}\left[\frac{1+\alpha sT}{1+sT}\right]$$

The above network can be visualized as an amplifier with a gain of 1/α. Let us draw the pole zero plot for the above transfer function.



Pole Zero Plot of Lead Compensating Network

Clearly we have -1/T (which is a zero of the transfer function) is closer to origin than the -1/(αT) (which is the pole of the transfer function).Thus we can say in the lead compensator zero is more dominating than the pole and because of this lead network introduces positive phase angle to the system when connected in series.

Let us substitute s = jω in the above transfer function and also we have α < 1. On finding the phase angle function for the transfer function we have

$$\theta(\omega) = tan^{-1}(\omega T) - \tan^{-1}(\alpha\omega T)$$

Now in order to find put the maximum phase lead occurs at a frequency let us differentiate this phase function and equate it to zero. On solving the above equation we get

$$\alpha = \frac{1 - \sin\theta_m}{1 + \sin\theta_m}$$

Where, θ_m is the maximum phase lead angle. And the corresponding magnitude of the transfer function at maximum θ_m is 1/a.

Effect of Phase Lead Compensation

The velocity constant K_v increases.
The slope of the magnitude plot reduces at the gain crossover frequency so that relative stability improves & error decrease due to error is directly proportional to the slope.
Phase margin increases.
Response become faster.

Advantages of Phase Lead Compensation

Let us discuss some of the advantages of the phase lead compensation-
Due to the presence of phase lead network the speed of the system increases because it shifts gain crossover frequency to a higher value.
Due to the presence of phase lead compensation maximum overshoot of the system decreases.
Disadvantages of Phase Lead Compensation
Some of the disadvantages of the phase lead compensation -
Steady state error is not improved.

Example:

1. **Write a matlab code to design a phase-lead compensator for the system $G(s)=\frac{1}{s(s+1)}$, such that the steady-state error is less than 0.1 for a unit ramp input and a % overshoot less than 28%.**

Steady-state error specification

$$K_V = \lim_{s\to 0} sG(s) = \lim_{s\to 0} s\frac{K\cdot 1}{s(s+1)} = K$$

$$e_{ss} = \frac{1}{K_V} = \frac{1}{K} < 0.1 \Rightarrow K \geq 10$$

% overshoot specification

$$\%\,\text{Overshoot} = 100e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$

To maintain an overshoot of 28% , ζ = 0.4

Then, the relationship between phase margin (PM) and damping ratio (ζ) for the special case of open-loop transfer function $G(s) = \dfrac{\omega_n^2}{s(s+2\zeta\omega_n)}$ which is given by

$$\text{PM} = \tan^{-1}\left(\frac{2\zeta}{\sqrt{\sqrt{1-4\zeta^2}-2\zeta^2}}\right)$$

Phase-lead design procedure:

i.) Choose the DC gain constant K such that the steady-state error specification is met. From above, we know K must be greater than or equal to 10, so let $K = 10$.

ii.) Obtain the gain margin and phase margin plots of the uncompensated system along with the DC gain constant K found in (i.) to determine the amount of phase lead θ_m needed to realize the required phase margin so that the percent overshoot specification is met.

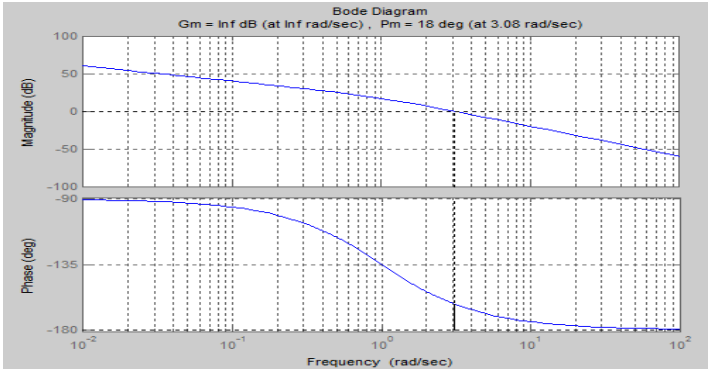


Figure. Bode plot of uncompensated system $K \cdot G(s)$.

From Figure above, the PM of the uncompensated system $\text{PM}_{uncomp} \approx 20^\circ$. Thus, choosing the PM of the compensated system as $\text{PM}_{comp} = 45^\circ$, then the additional amount of phase lead $\theta_m = \text{PM}_{comp} - \text{PM}_{uncomp} = 25^\circ$. Now that θ_m has been determined, the parameter α of the phase-lead compensator can be chosen using the equation given below:

$$\alpha = \frac{1 - \sin\theta_m}{1 + \sin\theta_m}$$

which has been chosen to be $\alpha = 0.3$ which corresponds to a maximum phase lead of 33°.

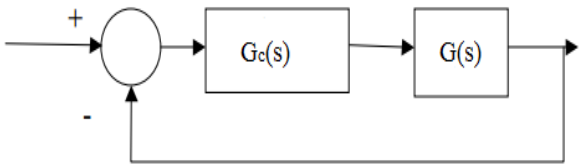
iii.) The maximum phase lead θ_m must be added around the new gain-crossover frequency ω_m. The phase-lead compensator contributes a gain around $-10\log(0.3) = 5.2\text{dB}$ at the new ω_m; therefore, one must determine the frequency at which the uncompensated system has a magnitude $10\log(0.3) = -5.2\text{dB}$. Thus, ω_m should equal this frequency so that it becomes the new 0-dB crossover frequency in the compensated system. From inspection of Figure , the magnitude of the uncompensated system equals –5.2dB at the frequency $\omega = 4.5\,\text{rad/sec}$. Let $\omega_m = 4.5\ \text{rad/sec}$.

iv.) Calculate the parameters of the phase-lead compensator based on the values obtained in steps (i.) thru (iii.). The transfer function of a phase-lead compensator is given as

$$Gc(s) = \frac{1}{\alpha}\cdot\frac{s + 1/T}{s + 1/\alpha T} \quad \text{or} \quad C(j\omega) = \frac{j\omega T + 1}{j\omega\alpha T + 1} \quad \text{with } \alpha < 1$$

where $T = \dfrac{1}{\omega_m\sqrt{\alpha}}$. Thus, for $\alpha = 0.3$, $T = 0.41\,\text{sec}$. This leads to a phase-lead compensator design of the following:

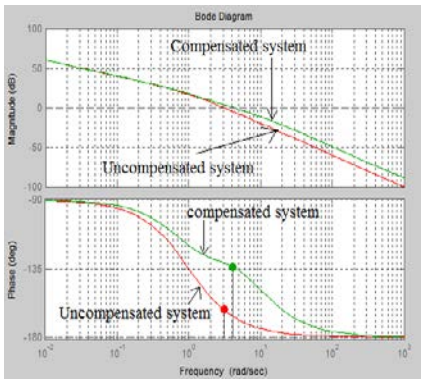
$$Gc(s) = \frac{0.41s + 1}{0.123s + 1}$$



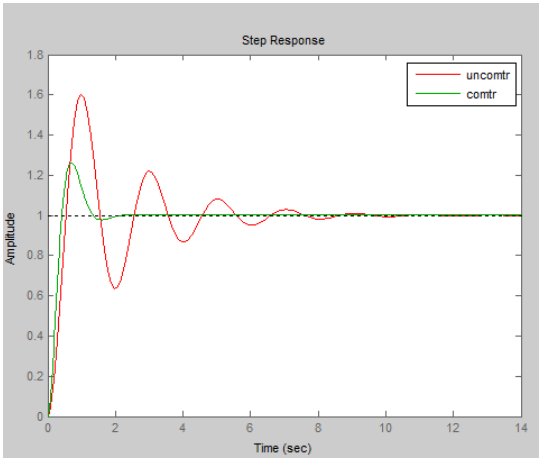
MATLAB CODE FOR PHASE LEAD COMPENSATION:

```
wm = 4.5;
alpha = 0.3;
T = 1/(wm*sqrt(alpha));
k= 10;
gnum = [k];
gden = [1 1 0];
uncompensated = tf(gnum,gden)
cnum = [T 1];
cden = [T*alpha 1];
compensator = tf(cnum,cden)
numo = conv(cnum,gnum);
deno = conv(cden,gden);
```

```
compensated = tf(numo,deno)
bode(uncompensated,'r',compensated ,'g')
```




```
uncomtr=feedback(uncompensated,1)
comtr=feedback(compensated,1)
step(uncomtr,'y')
hold on
step(comtr,'b')
```



Assignments:

1. **Consider a unity feedback system with G(s) = $\frac{4}{s(s+2)}$.**

   Design a lead compensator to achieve following requirements:

   i. **Static velocity error constant = 20**

   ii. **Phase margin> 50º**

DISCUSSION :

1. Draw magnitude & phase plot of a phase lead compensator.

EXPERIMENT NO : CS– II/5

TILLE : DESIGN OF A LAG COMPENSATOR USING BODE PLOT METHOD

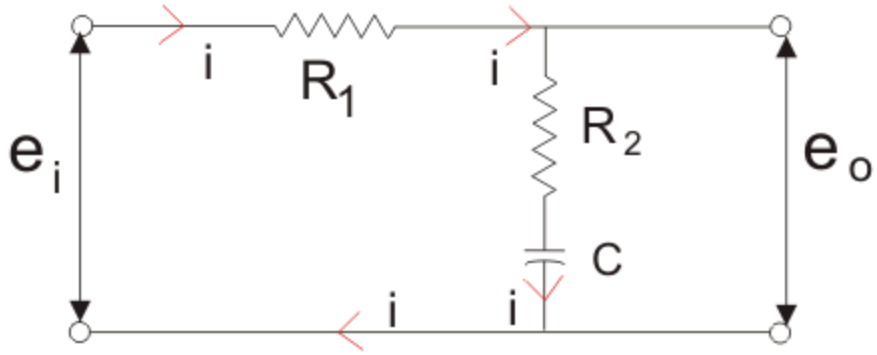
OBJECTIVE : To design a lag compensator to meet performance specification parameters

Software Used: MATLAB

THEORY :

A system which has one zero and one dominating pole (the pole which is closer to origin that all other poles is known as dominating pole) is known as lag network. If we want to add a dominating pole for compensation in control system then, we have to select a **lag compensation** network.

The basic requirement of the phase lag network is that all poles & zeros of the transfer function of the network must lie in (-)ve real axis interlacing each other with a pole located or on the nearest to the origin. Given below is the circuit diagram for the phase **lag compensation** network.



Phase Lag Compensating Network

We will have the output at the series combination of the resistor R_2 and the capacitor C.

From the above circuit diagram, we get

$$e_i = iR_1 + iR_2 + \frac{1}{C}\int idt$$

$$e_o = iR_2 + \frac{1}{C}\int idt$$

Now let us determine the transfer function for the given network and the transfer function can be determined by finding the ratio of the output voltage to the input voltage.

Taking Laplace transform of above two equation we get,

$$E_i(s) = R_1 I(s) + R_2 I(s) + \frac{1}{Cs} I(s)$$

$$E_o(s) = R_2 I(s) + \frac{1}{Cs} I(s)$$

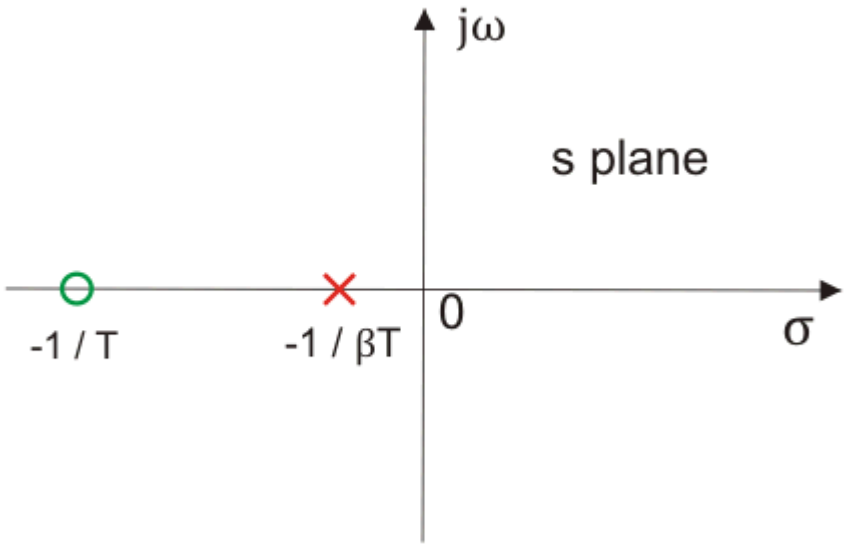
$$Transfer\ function,\ G_{lag}(s) = \frac{E_o(s)}{E_i(s)} = \frac{R_2 + \frac{1}{Cs}}{R_1 + R_2 + \frac{1}{Cs}}$$

$$\Rightarrow G_{lag}(s) = \frac{R_2 C s + 1}{(R_1 + R_2) C s + 1}$$

On substituting the $T = R_2C$ and $\beta = \{(R_2 + R_1) / R_1\}$ in the above equation (where T and β are respectively the time constant and dc gain), we have

$$Transfer\ function,\ G_{lag}(s) = \frac{1 + Ts}{1 + \beta Ts}$$

The above network provides a high frequency gain of 1 / β. Let us draw the pole zero plot for the above transfer function.



Pole Zero Plot of Lag Network Clearly we have -1/T (which is a zero of the transfer function) is far to origin than the -1 / (βT)(which is the pole of the transfer function). Thus we can say in the lag compensator pole is more dominating than the zero and because of this lag network introduces negative phase angle to the system when connected in series.

Let us substitute $s = j\omega$ in the above transfer function and also we have $a < 1$. On finding the phase angle function for the transfer function we have

$$\theta(\omega) = tan^{-1}(\omega T) - \tan^{-1}(\beta \omega T)$$

Now in order to find put the maximum phase lag occurs at a frequency let us differentiate this phase function and equate it to zero. On solving the above equation we get

$$\beta = \frac{1 - \sin\theta_m}{1 + \sin\theta_m}$$

Where, θ_m is the maximum phase lead angle. Remember β is generally chosen to be greater than 10.

Example:

Design a phase-lag compensator for the system $G(s) = \frac{1}{s(s+1)}$, such that the steady-state error is less than 0.1 for a unit ramp input and a percent overshoot less than 25%.

Steady-state error specification

$K \geq 10$.

Percent overshoot specification

$\text{PM}_{comp} \geq 45^\circ$.

Phase-lag design procedure:

i.) Choose the DC gain constant K such that the steady-state error specification is met. From above, we know K must be greater than or equal to 10, so let $K = 10$.

ii.) Obtain the gain margin and phase margin plots of the uncompensated system along with the DC gain constant K found in (i.) to estimate the frequency at which the PM of 50° occurs. Denote this frequency as the new gain-crossover frequency ω_m. From Figure 8., let $\omega_m = 0.84\ \text{rad/sec}$.

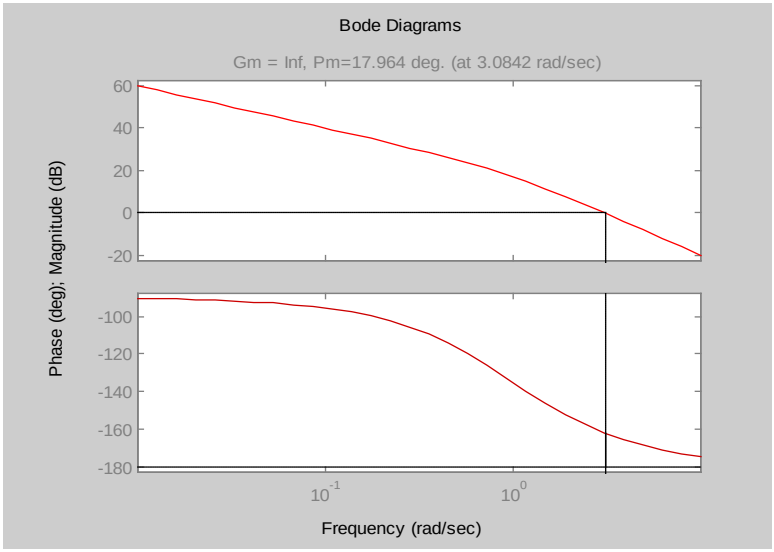


Figure 8. Bode plot of uncompensated system $K \cdot G(s)$.

iii.) Determine the magnitude of uncompensated system at $\omega_m = 0.84\,\text{rad/sec}$. From Figure 8., the magnitude of the uncompensated system at $\omega_m = 0.84\,\text{rad/sec}$ is 20 dB. To bring the magnitude curve down to 0 dB at ω_m, the phase-lag compensator must provide $20\log(\alpha) = 20\,\text{dB}$ or $\alpha = 10^{\frac{20}{20}} = 10$.

iv.) Calculate the parameters of the phase-lag compensator based on the values obtained in steps (i.) thru (iii.). The transfer function of a phase-lag compensator is given as

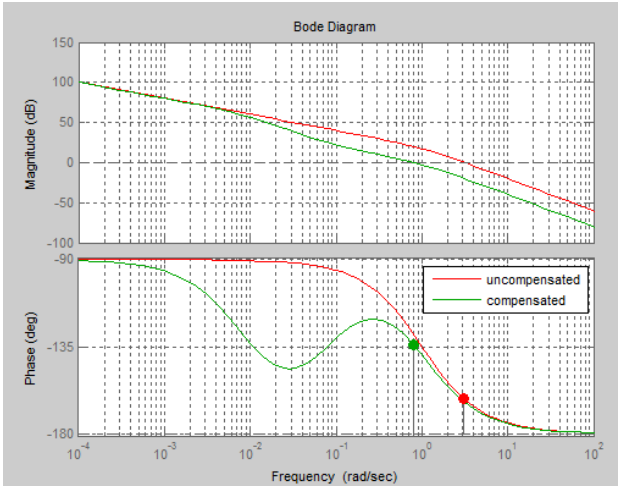
$$C(s) = \frac{1}{\alpha}\cdot\frac{s+1/T}{s+1/\alpha T} \quad \text{or} \quad C(j\omega) = \frac{j\omega T+1}{j\omega\alpha T+1} \quad \text{with } \alpha > 1$$

where $T = \frac{10}{\omega_m} = 11.9\,\text{sec}$. This is to ensure that the frequency at $\omega = \frac{1}{T}$ is one decade below the new gain-crossover frequency ω_m. This leads to a phase-lag compensator design of the following:

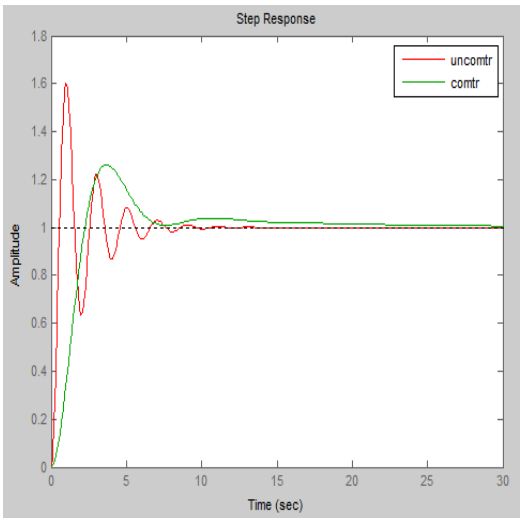
$$C(s) = \frac{11.9s+1}{119s+1}.$$

MATLAB CODE FOR PHASE LAG COMPENSATION:

```
wm = 0.84;
beta = 10;
T = 10/(wm);
k= 10;
gnum = [k];
gden = [1 1 0];
uncompensated = tf(gnum,gden)
cnum = [T 1];
cden = [T*beta 1];
compensator = tf(cnum,cden)
numo = conv(cnum,gnum);
deno = conv(cden,gden);
compensated = tf(numo,deno);
bode(uncompensated,'r', compensated,'g')
```



```
uncomtr=feedback(uncompensated,1)
comtr=feedback(compensated,1)
step(uncomtr,'y')
hold on
step(comtr,'b')
```



DISCUSSION :

1. How lag compensator works?

EXPERIMENT NO : CS– II/6

TILLE : DETERMINATION OF STEP RESPONSE OF A DIGITAL SYSTEM DUE TO VARIATION IN CONTROLLER PARAMETERS

OBJECTIVE : To study

I. The effect of variation in controller parameter on system response

Software Used: MATLAB

THEORY :

PID controllers use a 3 basic behavior types or modes: P - proportional, I - integrative and D - derivative. While proportional and integrative modes are also used as single control modes, a derivative mode is rarely used on it' s own in control systems. Combinations such as PI and PD control are very often in practical systems.

P Controller: In general it can be said that P controller cannot stabilize higher order processes. For the 1st order processes, meaning the processes with one energy storage, a large increase in gain can be tolerated. Proportional controller can stabilize only 1st order unstable process. Changing controller gain K can change closed loop dynamics. A large controller gain will result in control system with: a) smaller steady state error, i.e. better reference following b) faster dynamics, i.e. broader signal frequency band of the closed loop system and larger sensitivity with respect to measuring noise c) smaller amplitude and phase margin When P controller is used, large gain is needed to improve steady state error. Stable systems do not have problems when large gain is used. Such systems are systems with one energy storage (1st order capacitive systems). If constant steady state error can be accepted with such processes, than P controller can be used. Small steady state errors can be accepted if sensor will give measured value with error or if importance of measured value is not too great anyway.

PD Controller: D mode is used when prediction of the error can improve control or when it necessary to stabilize the system. From the frequency characteristic of D element it can be seen that it has phase lead of 90°.

Often derivative is not taken from the error signal but from the system output variable. This is done to avoid effects of the sudden change of the reference input that will cause sudden change in the value of error signal. Sudden change in error signal will cause sudden change in control output. To avoid that it is suitable to design D mode to be proportional to the change of the output variable. PD controller is often used in control of moving objects such are flying and underwater vehicles, ships, rockets etc. One of the reason is in stabilizing effect of PD controller on sudden changes in heading variable y(t). Often a "rate gyro" for velocity measurement is used as sensor of heading change of moving object.

PI Controller: PI controller will eliminate forced oscillations and steady state error resulting in operation of on-off controller and P controller respectively. However, introducing integral mode has a negative effect on speed of the response and overall stability of the system. Thus, PI controller will not increase the speed of response. It can be expected since PI controller does not have means to predict what will happen with the error in near

future. This problem can be solved by introducing derivative mode which has ability to predict what will happen with the error in near future and thus to decrease a reaction time of the controller. PI controllers are very often used in industry, especially when speed of the response is not an issue. A control without D mode is used when: a) fast response of the system is not required b) large disturbances and noise are present during operation of the process c) there is only one energy storage in process (capacitive or inductive) d) there are large transport delays in the system.

PID Controller: PID controller has all the necessary dynamics: fast reaction on change of the controller input (D mode), increase in control signal to lead error towards zero (I mode) and suitable action inside control error area to eliminate oscillations (P mode). Derivative mode improves stability of the system and enables increase in gain K and decrease in integral time constant Ti, which increases speed of the controller response. PID controller is used when dealing with higher order capacitive processes (processes with more than one energy storage) when their dynamic is not similar to the dynamics of an integrator (like in many thermal processes). PID controller is often used in industry, but also in the control of mobile objects (course and trajectory following included) when stability and precise reference following are required. Conventional autopilot is for the most part PID type controllers.

Effects of Coefficients:

Parameter	Rise time	Overshoot	Settling time	Steady-state error
K_p	Decrease	Increase	Small change	Decrease
K_i	Decrease	Increase	Increase	Decrease significantly
K_d	Minor decrease	Minor decrease	Minor decrease	No effect in theory

A discrete implementation of proportional control is identical to continuous. The continuous is

$u(t) = k_p e(t)$; $D(s) = k_p$

$u(k) = k_p e(k)$; $D(z) = k_p$

The discrete is

where $e(t)$ or $e(k)$ is the error signal and k_p is proportional gain.

The continuous time derivative control can be expressed as

$u(t) = k_d \dot{e}(t)$; $D(s) = k_d s$

The discrete derivative control equation can be written as

$u(k) = k_d \frac{e(k) - e(k-1)}{T}$; $D(z) = k_d \frac{1-z^{-1}}{T} = k_d \frac{z-1}{Tz}$

The continuous time integral control can be expressed as

$u(t) = k_i \int e(t)dt$; $D(s) = ki/s$

$u(k) = u(k-1)+k_i Te(k)$; $D(z) = k_i \frac{T}{1-z^{-1}} = k_i \frac{Tz}{z-1}$

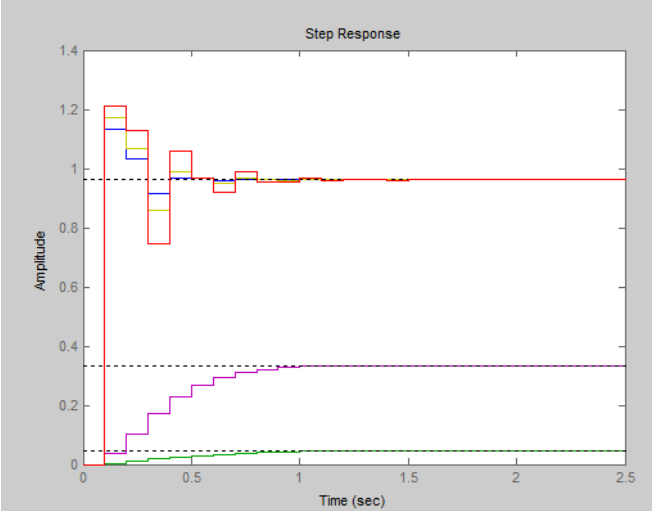
The discrete PID controller can be represented as

$G_{pid}(z) = k_p + k_i \frac{Tz}{z-1} + k_d \frac{z-1}{Tz}$

Example:

1. Consider a unity feedback system with forward path transfer function G(s) = $\frac{1}{s^2+10s+20}$. Convert this system in z-domain with T = 0.1 sec. Show the effect of addition of a PD controller on the system performance.

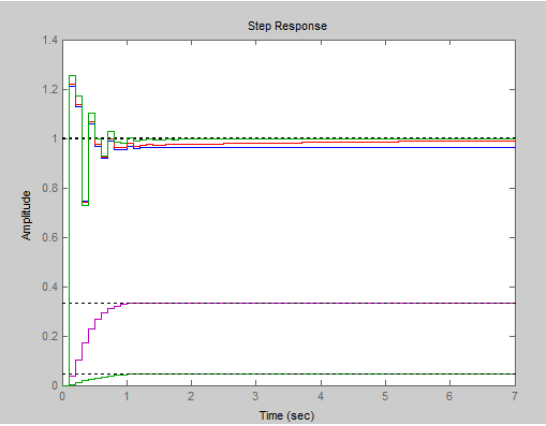
```
num=1;
den=[1 10 20];
g1=tf (num,den)
t1=feedback(g1,1)
d1=c2d(t1,.1,'zoh')
step(d1,'g')
hold on

num1=10;
den1=[1 10 20];
g2=tf (num1,den1)
t2=feedback(g2,1)
d2=c2d(t2,.1, 'zoh')
step(d2,'m')
hold on
Kp=500;
Kd=10;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
t3=feedback(g3,1)
d3= c2d(t3,.1,'zoh')
step(d3,'b')
hold on
Kp=500;
Kd=5;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
t3=feedback(g3,1)
d3= c2d(t3,.1,'zoh')
step(d3,'y')
hold on
Kp=500;
Kd=.01;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
t3=feedback(g3,1)
d3= c2d(t3,.1,'zoh')
step(d3,'r')
hold on
```



2. Consider a unity feedback system with forward path transfer function G(s) = $\frac{1}{s^2+10s+20}$. Convert this system in z-domain with T = 0.1 sec. Show the effect of addition of a PI controller on the system performance.

```
num=1;
den=[1 10 20];
g1=tf (num,den)
t1=feedback(g1,1)
d1=c2d(t1,.1,'zoh')
step(d1,'g')
hold on

num1=10;
den1=[1 10 20];
g2=tf (num1,den1)
t2=feedback(g2,1)
d2=c2d(t2,.1, 'zoh')
step(d2,'m')
hold on
Kp=500;
Ki = 1
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
deno=conv(den,denc)
g3=tf(numo,deno)
t3=feedback(g3,1)
d3= c2d(t3,.1,'zoh')
step(d3,'b')
hold on
Kp=500;
Ki = 100
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
```

```
deno=conv(den,denc)
g3=tf(numo,deno)
t3=feedback(g3,1)
d3= c2d(t3,.1,'zoh')
step(d3,'r')
hold on
Kp=500;
Ki = 500
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
deno=conv(den,denc)
g3=tf(numo,deno)
t3=feedback(g3,1)
d3= c2d(t3,.1,'zoh')
step(d3,'g')
hold on
```



Assignments:

1. Consider a unity feedback system with forward path transfer function G(s) = $\frac{1}{s(s+1.6)}$. Convert this system in z-domain with T = 0.1 sec. Show the effect of addition of a PD controller on the system performance.
2. Consider a unity feedback system with forward path transfer function G(s) = $\frac{1}{s(s+1.6)}$. Convert this system in z-domain with T = 0.1 sec. Show the effect of addition of a PI controller on the system performance.
3. Consider a unity feedback system with forward path transfer function G(s) = $\frac{1}{s(s+1.6)}$. Convert this system in z-domain with T = 0.1 sec. Show the effect of addition of a PID controller on the system performance.

Discussion:

1. What is the effect of derivative gain in system performance?
2. What is the effect of integral gain in system performance?

EXPERIMENT NO : CS /7

TITLE : DETERMINATION OF STATE SPACE MODEL FROM TRANSFER FUNCTION MODEL & VICE VERSA.

OBJECTIVE : To obtain

I. Transfer function model from a state model
II. State model from transfer function model
III. Step response of a system represented by its state model

Software Used: MATLAB/ SIMULINK

Example 1: Obtain transfer function and its step response of a state model given by:

$$\begin{matrix}\dot{x}_1 \\ \dot{x}_2\end{matrix} = \begin{bmatrix}0 & 1 \\ -1 & -1\end{bmatrix}\begin{pmatrix}x_1 \\ x_2\end{pmatrix} + \begin{bmatrix}0 \\ 1\end{bmatrix} u$$

$$Y = [1 \quad 0]\begin{pmatrix}x_1 \\ x_2\end{pmatrix}$$

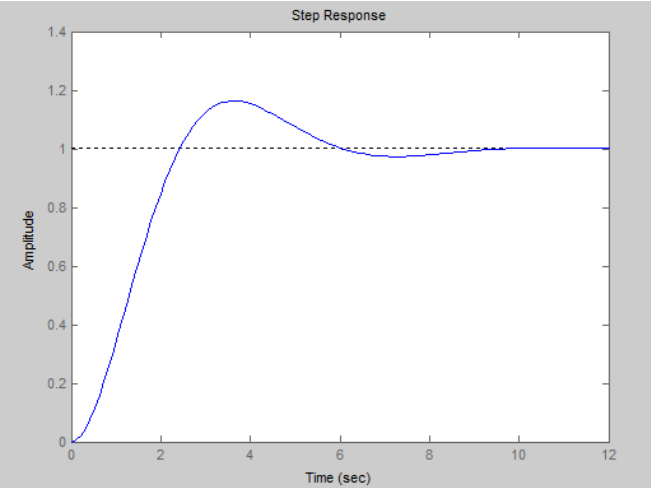
Matlab Code:

```
a = [0 1; -1 -1]
b = [0;1]
c = [1 0]
d=0
[num,den] = ss2tf(a,b,c,d)
g = tf(num,den)
step(g)
```

Output :

Transfer function:

```
    1
-----------
s^2 + s + 1
```



Example 2: Obtain state model and its step response of a transfer function given by:

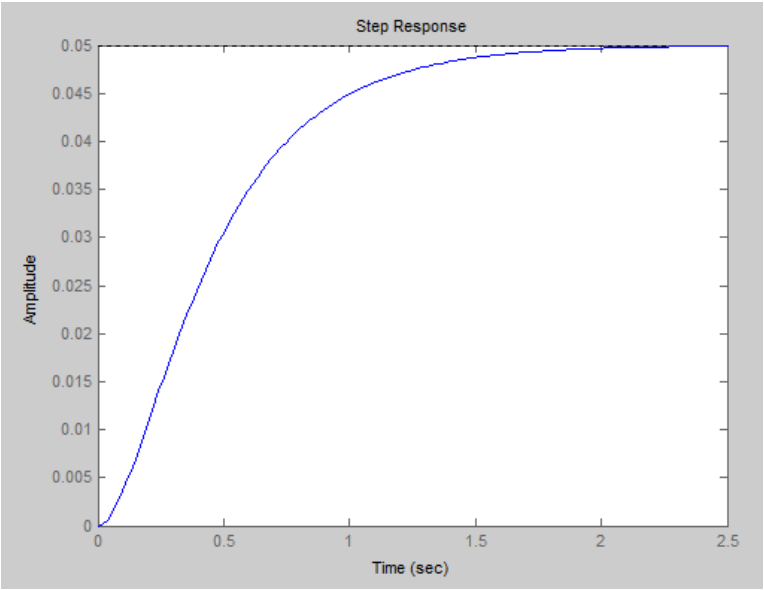
$$G(s) = \frac{1}{s^2+10s+20}$$

Matlab Code:

```
num = [1]
den = [1 10 20]
[A,B,C,D]= tf2ss(num,den)
step(A,B,C,D)
```

Output:

```
A =

   -10   -20
     1     0

B =

     1
     0

C =

     0     1

D =

     0
```



EXPERIMENT NO : CS /8

TITLE : DETERMINATION OF EIGEN VALUES FROM STATE MODEL & STABILITY ANALYSIS

OBJECTIVE : To determine

I. Eigen values from state model
II. Eigen values from transfer function model
III. Stability of a system

Software Used: MATLAB/ SIMULINK

Example 1: Obtain transfer function and its step response of a state model given by:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -1 & -1\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix} u$$

$$Y = [1 \quad 0]\begin{pmatrix}x_1\\ x_2\end{pmatrix}$$

Matlab Code:

```
a = [0 1; -1 -1]
b = [0;1]
c = [1 0]
d=0
[num,den] = ss2tf(a,b,c,d)
g = tf(num,den)
eig(a)
if (eig(a)< 0)
    system = 1
else
    system = 0
end
```

***[Note: system = 1 represents stable system ; system = 0 represents unstable system]**

Output:

```
Transfer function:
     1
-----------
s^2 + s + 1

ans =

  -0.5000 + 0.8660i
  -0.5000 - 0.8660i

system =

     1
```

Example 2: Obtain eigen values of a transfer function given by:

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

Matlab Code:

```
num = [1]
den = [1 10 20]
[A,B,C,D]= tf2ss(num,den)
eig (A)
if (eig(a) < 0)
    system = 1
else
    system = 0
end
```

***[Note: system = 1 represents stable system ; system = 0 represents unstable system]**

Output:

```
A =

   -10   -20
     1     0

B =

     1
     0

C =

     0     1

D =

     0
ans =

   -7.2361
   -2.7639


system =


     1
```

EXPERIMENT NO : CS /9

TITLE : STUDY THE EFFECT OF COMMON NON – LINEARITIES INTRODUCED TO THE FORWARD PATH TRANSFER FUNCTION OF A 2ND ORDER UNITY FEEDBACK CONTROL SYSTEM

OBJECTIVE : To study

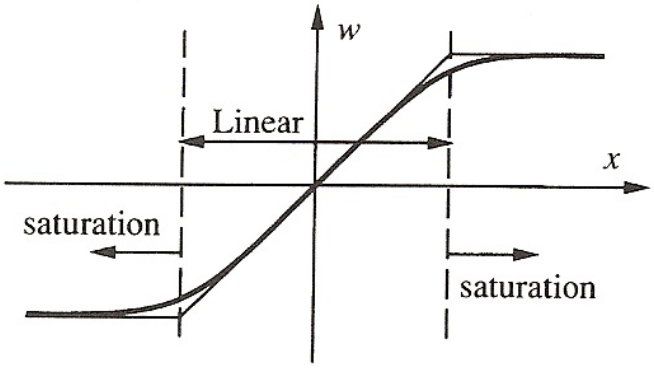
I. the effect of common non linearities such as relay, dead zone, saturation on response of a 2nd order control system.

THEORY:

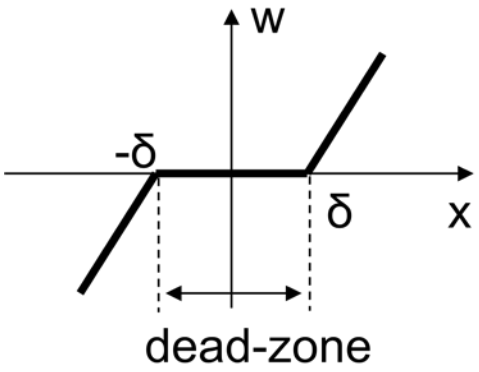
Consider the typical block shown in Figure 1. It is composed of four parts: a plant to be controlled, sensors for measurement, actuators for control action, and a control low, usually implemented on a computer. Nonlinearities may occur in any part of the system, thus make it a nonlinear control system.

Nonlinearities can be classified as continuous and discontinuous. Because discontinuous nonlinearities cannot be locally approximated by linear functions, they are also called "hard" nonlinearities. Hard nonlinearities are commonly found in control systems, both in small range operation and large range operation. Whether a system in small range operation should be regarded as nonlinear or linear depends on the magnitude of the hard nonlinearities and on the extent of their effects on the system performance.

Saturation: When one increases the input to a physical device, the following phenomenon is often observed: when the input is small, its increase leads to a corresponding (often proportional) increase of output: but when the input reaches a certain level, its further increase does produce little or no increase of the output. The output simply stays around its maximum value. The device is said to be *saturation* when this happen. A typical saturation nonlinearity is represented in Figure 2, where the thick line is the real nonlinearity and the thin line is an idealized saturation nonlinearity.



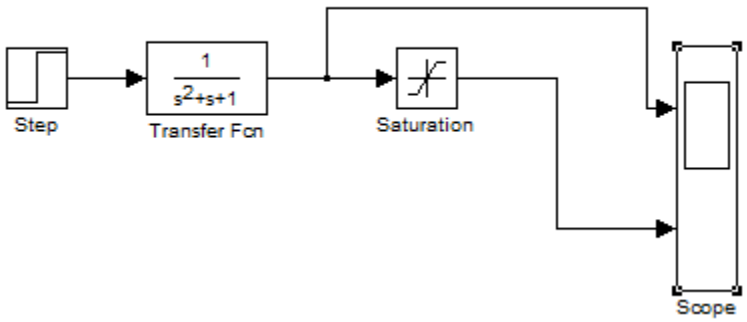
Dead-Zone: Consider the dead-zone characteristics shown in Figure 6, with the dead-zone with being 2δ and its slope k



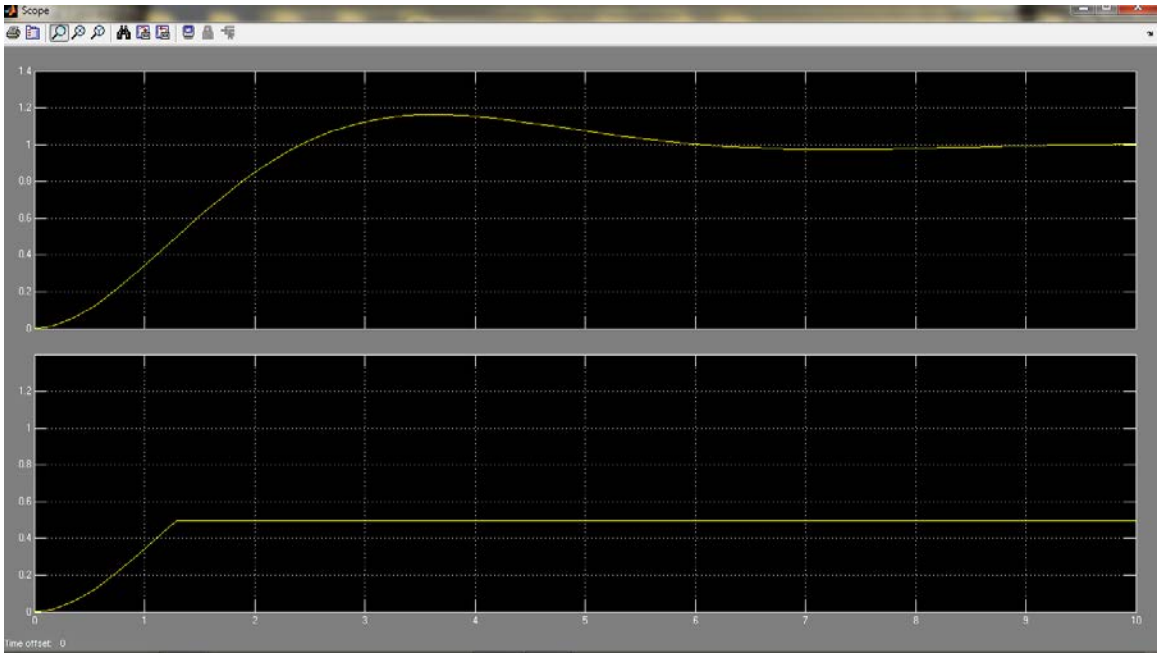
Dead-zones can have a number of possible effects on control systems. Their most common effect is to decrease static output accuracy. They may also lead to limit cycles or system instability because of the lack of response in the dead-zone. The response corresponding to a sinusoidal input x(t)=Asin(ωt) into a dead-zone of width 2δ and slope k, with A≥δ, is plotted in Figure 7. Since the characteristics is an odd function, a_1=0. The response is also seen to be symmetric over the four quarters of a period. In one quarter of a period, i.e., when 0≤ωt≤p/2, one has

Software Used: MATLAB/ SIMULINK

Example 1: Show the effect of saturation with limit 0.5 introduced in the forward path of a open loop system having G(s) = $\frac{1}{s^2+s+1}$.

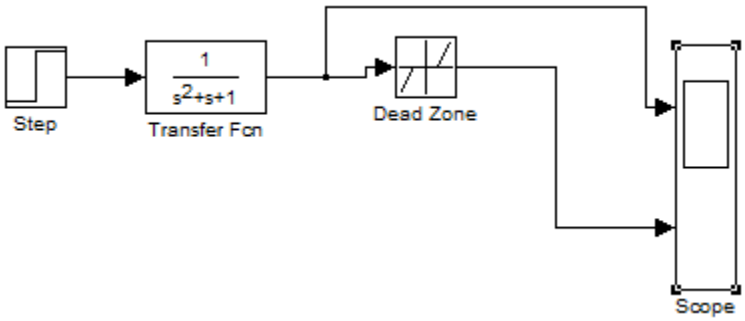


SIMULINK MODEL

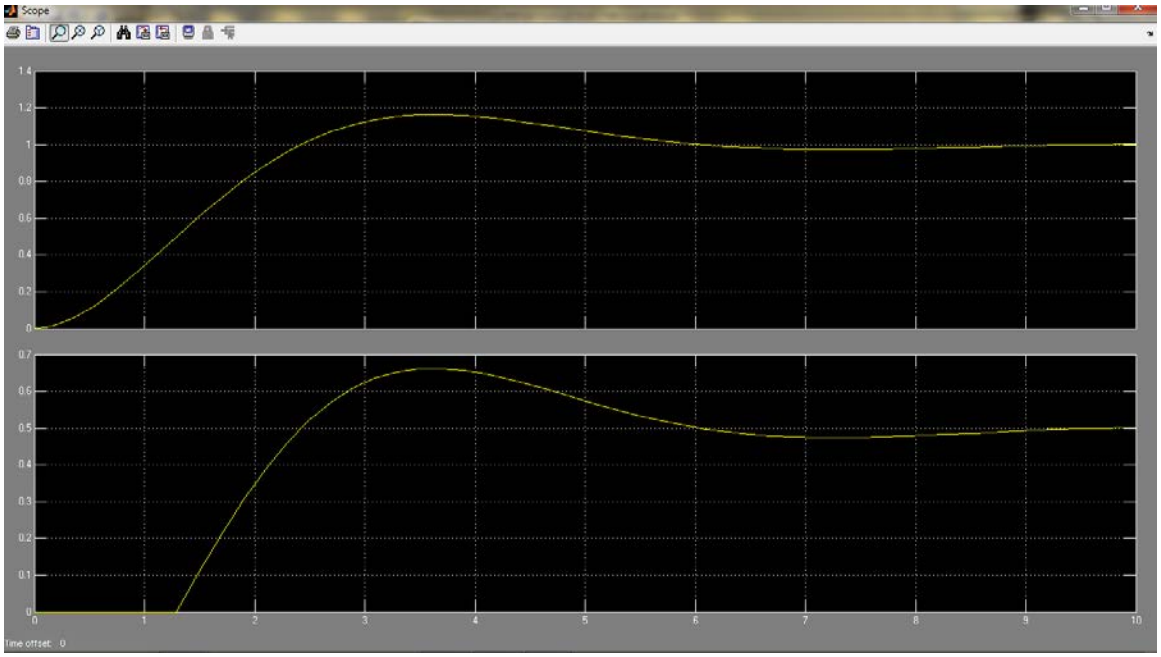


MATLAB RESULT

Example 2: Show the effect of dead zone with limit 0.5 introduced in the forward path of a open loop system having G(s) = $\frac{1}{s^2+s+1}$.

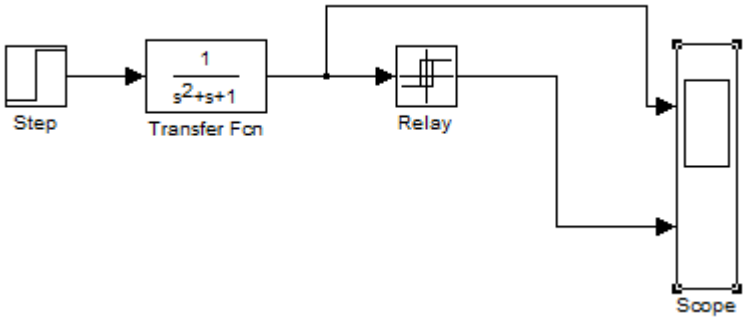


SIMULINK MODEL

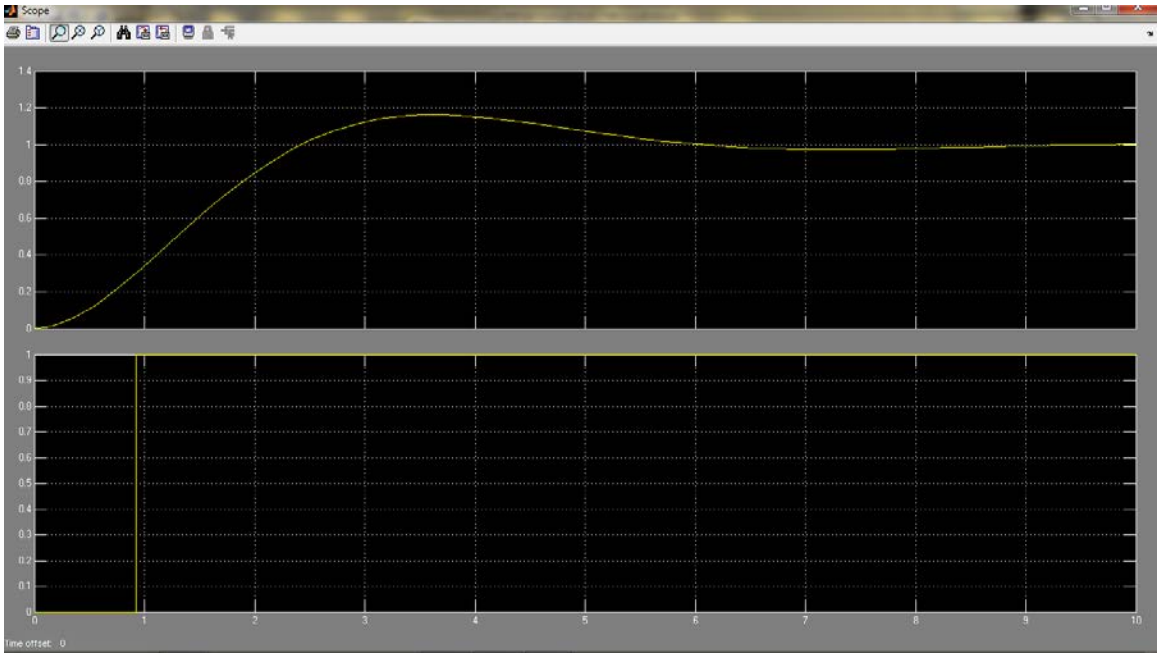


MATLAB RESULT

Example 3: Show the effect of relay with on & off point of 0.3 & 0.2 respectively introduced in the forward path of a open loop system having G(s) = $\frac{1}{s^2+s+1}$.

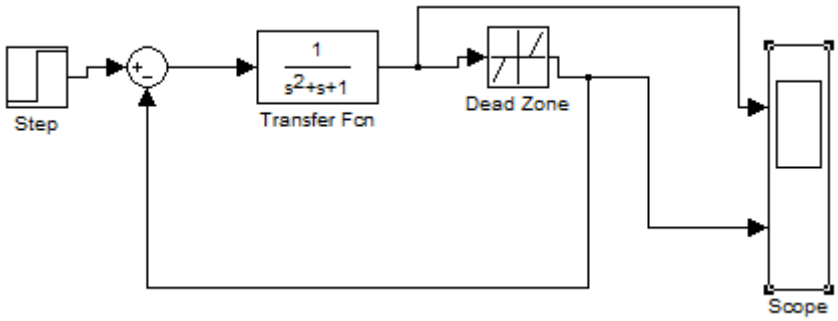


SIMULINK MODEL

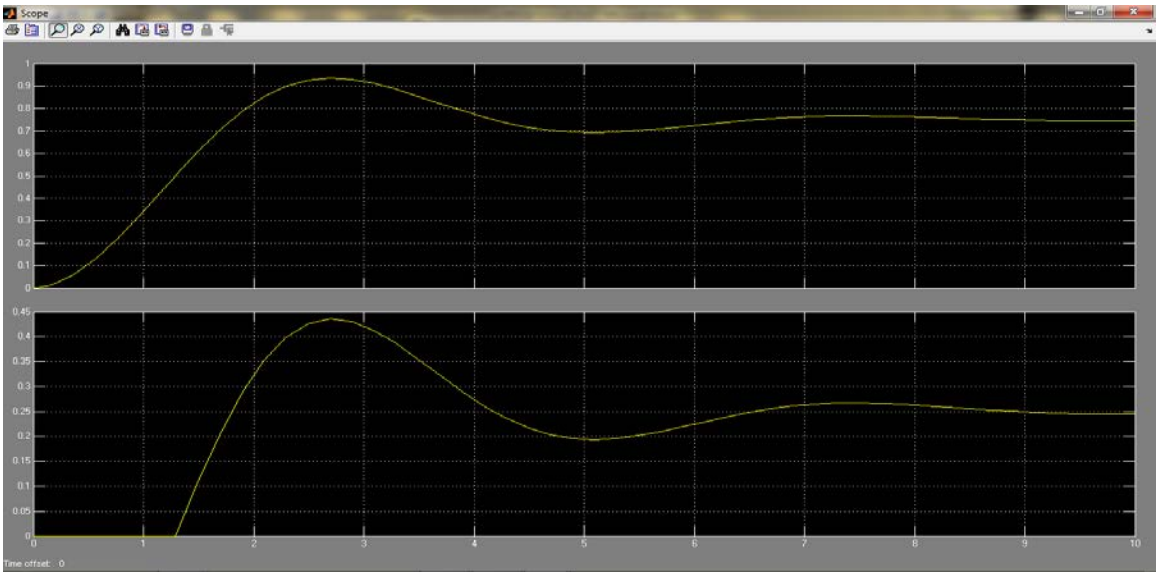


MATLAB RESULT

Example 4: Show the effect of dead zone with limit 0.5 introduced in the forward path of a unity feedback closed loop system having G(s) = $\frac{1}{s^2+s+1}$.

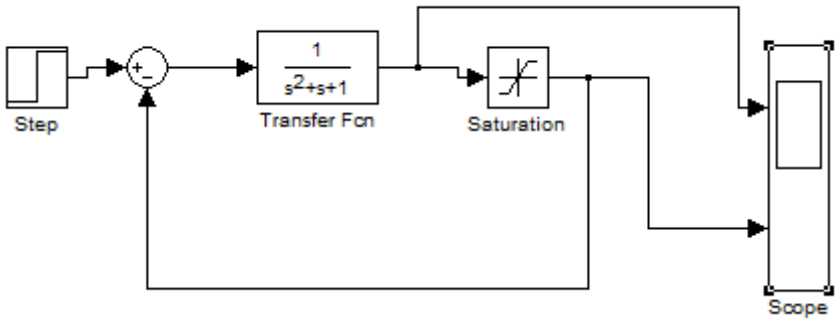


SIMULINK MODEL

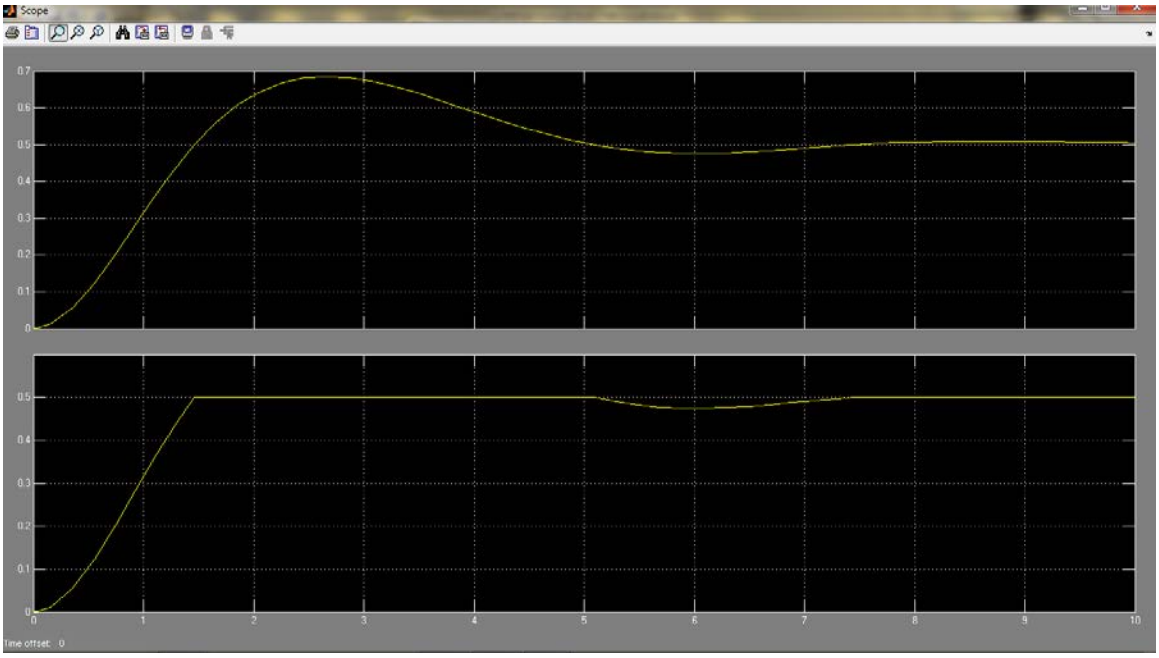


MATLAB RESULT

Example 5: Show the effect of saturation with limit 0.5 introduced in the forward path of a unity feedback closed loop system having G(s) = $\frac{1}{s^2+s+1}$.

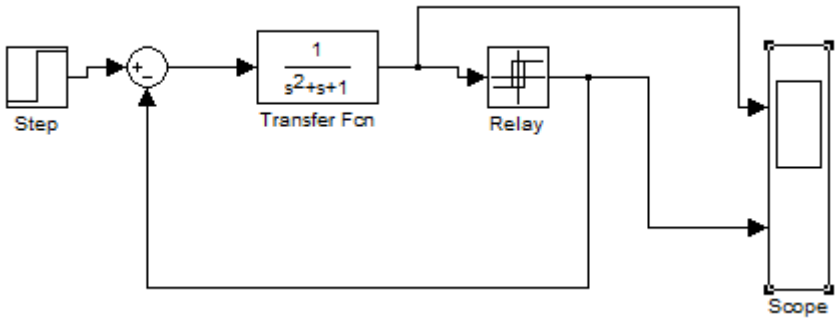


SIMULINK MODEL

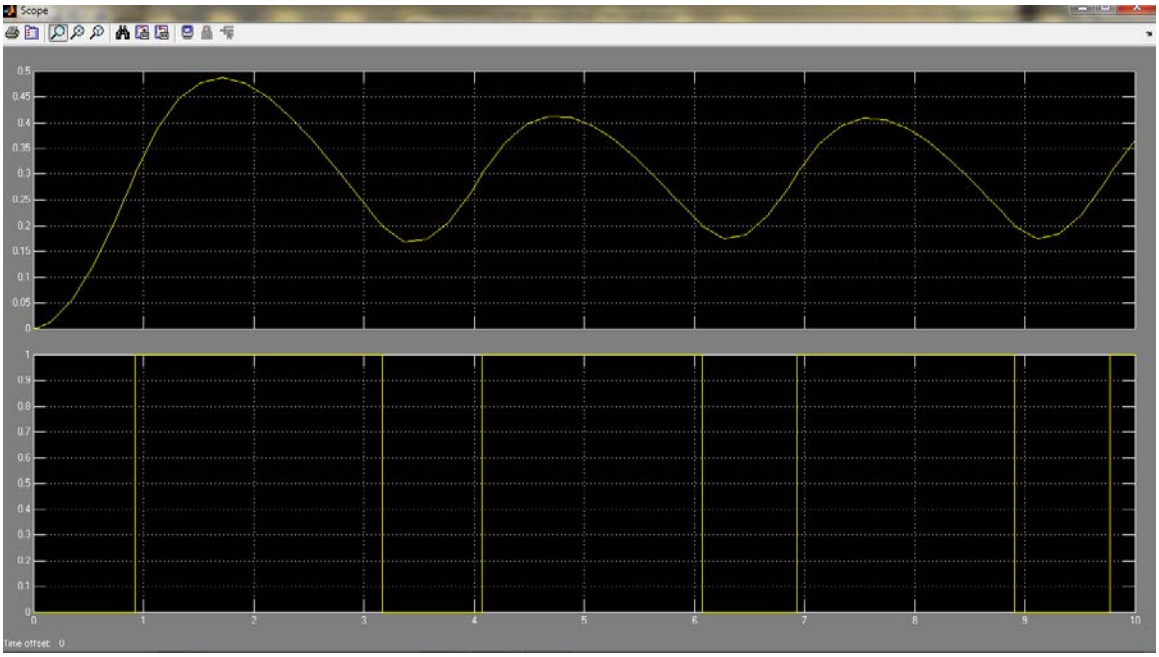


MATLAB RESULT

Example 6: Show the effect of relay with on & off point of 0.3 & 0.2 respectively introduced in the forward path of a unity feedback closed loop system having G(s) = $\frac{1}{s^2+s+1}$.



SIMULINK MODEL



MATLAB RESULT