

# سایت اختصاصی مهندسی کنترل



<https://controlengineers.ir>



<https://t.me/controlengineers>



<https://www.instagram.com/controlengineers.ir>

# LABORATORY INSTRUCTION MANUAL

## CONTROL SYSTEM I LAB

EE 593

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

## EXPERIMENT NO : CSI/1

### TITLE : FAMILIARIZATION WITH MATLAB CONTROL SYSTEM TOOL BOX, MATLAB/SIMULINK TOOL BOX.

**OBJECTIVE :** To obtain

- I. Pole, zero, gain values from a given transfer function
- II. Transfer function model from pole, zero, gain values
- III. Pole, zero plot of a transfer function

### THEORY:

A transfer function is also known as the network function is a mathematical representation, in terms of spatial or temporal frequency, of the relation between the input and output of a (linear time invariant) system. The transfer function is the ratio of the output Laplace Transform to the input Laplace Transform assuming zero initial conditions. Many important characteristics of dynamic or control systems can be determined from the transfer function. The transfer function is commonly used in the analysis of single-input single-output electronic system, for instance. It is mainly used in signal processing, communication theory, and control theory. The term is often used exclusively to refer to linear time-invariant systems (LTI). In its simplest form for continuous time input signal  $x(t)$  and output  $y(t)$ , the transfer function is the linear mapping of the Laplace transform of the input,  $X(s)$ , to the output  $Y(s)$ .

Zeros are the value(s) for  $z$  where the numerator of the transfer function equals zero. The complex frequencies that make the overall gain of the filter transfer function zero.

Poles are the value(s) for  $z$  where the denominator of the transfer function equals zero. The complex frequencies that make the overall gain of the filter transfer function infinite.

The general procedure to find the transfer function of a linear differential equation from input to output is to take the Laplace Transforms of both sides assuming zero conditions, and to solve for the ratio of the output Laplace over the input Laplace.

The transfer function provides a basis for determining important system response characteristics without solving the complete differential equation. As defined, the transfer function is a rational function in the complex variable 's' that is It is often convenient to factor the polynomials in the numerator and the denominator, and to write the transfer function in terms of those factors:

$$G(s) = \frac{N(s)}{D(s)} = K \frac{(s-z_1)(s-z_2)\dots\dots\dots(s-z_n)}{(s-p_1)(s-p_2)\dots\dots\dots(s-p_m)}$$

where, the numerator and denominator polynomials,  $N(s)$  and  $D(s)$ .

The values of  $s$  for which  $N(s) = 0$ , are known as zeros of the system. i.e; at  $s = z_1, z_2, \dots, z_n$ .

The values of  $s$  for which  $D(s) = 0$ , are known as poles of the system. i.e; at  $s = p_1, p_2, \dots, p_n$ .

Example 1: Obtain pole, zero & gain values of a transfer function  $G(s) = \frac{s^2+4s+3}{(s+5)(s^2+4s+7)}$ . Also obtain pole zero plot.

```
num = [1 4 3]
den= conv([1 5], [3 4 7])
g = tf (num,den)
[z,p,k] = tf2zp(num,den)
pzmap(g)
```

Output :

Transfer function:

$$s^2 + 4 s + 3$$


---


$$3 s^3 + 19 s^2 + 27 s + 35$$

z =

-3  
-1

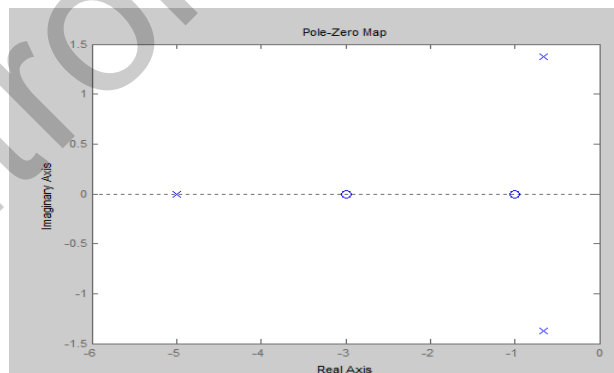
p =

-5.0000  
-0.6667 + 1.3744i  
-0.6667 - 1.3744i

k =

0.3333

Pole Zero Plot:



Example 2: Write matlab code to obtain transfer function of a system from its pole ,zero, gain values. Assume pole locations are -2, 1, zero at -1 and gain is 7.

Matlab Code :

```
p= [-2 1]
z= [-1]
```

```
k=7
[num,den]= zp2tf(z',p',k)
g=tf(num,den)
```

Output:

Transfer function:

$$\frac{7s + 7}{s^2 + s - 2}$$

### Assignments:

Obtain Pole, zero, gain values of the transfer functions given below. **Also verify your result theoretically**

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

Obtain Transfer function of the systems (Theoretically & Practically).

1. Poles = -1+i, -1-i, -4. Zeros = -2, -5, gain = 1
2. Poles = -1+4i, -1-4i, -5. Zeros = -8, -5, gain = .75

### DISCUSSION:

1. What do you mean by poles?
2. What is the significance of transfer function?



**EXPERIMENT NO : CS I/2**

**TITLE : DETERMINATION OF STEP & IMPULSE RESPONSE FOR A FIRST ORDER UNITY FEEDBACK SYSTEM**

**OBJECTIVE :** To determine

- I. Step response of 1<sup>st</sup> order system
- II. Impulse response of 1<sup>st</sup> order system

**THEORY:** A first order system is one in which highest power of s in denominator if transfer function defines order of the system.

For first order system,

$$\frac{C(s)}{R(s)} = \frac{1}{sT+1}$$

$$C(s) = \frac{1}{sT+1} R(s) \dots\dots\dots (1)$$

Since the laplace transform of the unit step function is 1/s , substituting R(s)=1/s in equation (1)

$$C(s) = \frac{1}{sT+1} * \frac{1}{s}$$

Expanding C(s) into partial fractions gives ,

$$C(s) = \frac{1}{s} - \frac{1}{sT+1}$$

$$C(s) = \frac{1}{s} - \frac{1}{s+(1/T)} \dots\dots\dots (2)$$

Taking the inverse laplace transform of equation (2), we get

$$C(t) = 1 - e^{-t/T} \text{ for } t \geq 0 \dots\dots\dots (3)$$

Equation (3) shows that initially (when t=0), the output c(t) is zero and finally (t → ∞) e<sup>-t/T</sup> is zero and the output c(t) becomes unity .

At t = T,

$$C(t) = 1 - e^{-1} = 1 - 0.368 = 0.632$$

That's , the output response has reached 63.2 % of it's final value . T is known as the time constant . Thus , the time constant T is defined as the time required for the output response to attain 63.2% of its final value or steady state value .

Equation (3) shows that the response curve is exponential in nature as shown on figure.

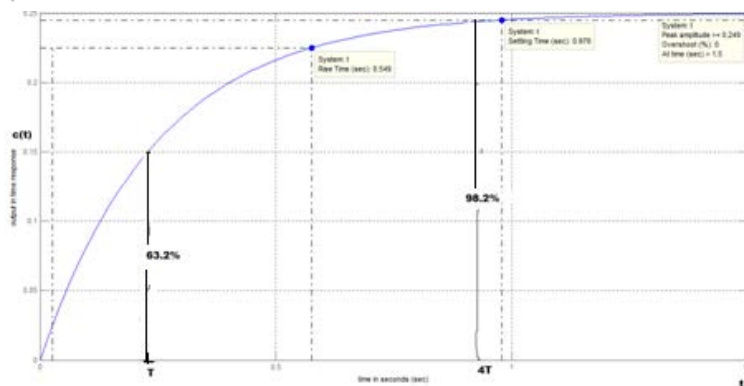


Fig. : Response curve of first order unit step input

Response of the first order system with unit impulse response

For the unit – impulse input

$$R(s) = 1$$

$$C(s) = \frac{1}{sT+1} * R(s) \dots\dots\dots(1)$$

Substituting the value of R(s) = 1 in equation (1), we get

$$C(s) = \frac{1}{sT+1} * 1$$

$$C(s) = \frac{1}{T} * \frac{1}{s+1/T} \dots\dots\dots(2)$$

Taking the inverse laplace transform of the equation of (2) , we get the output response as

$$C(t) = \frac{1}{T} e^{-t/T} \text{ for } t \geq 0 \dots\dots\dots(3)$$

The output response curve shown in the figure

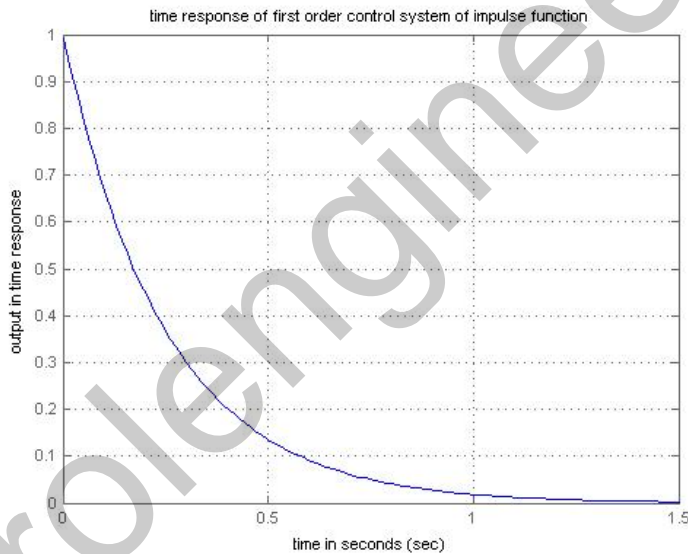


Fig. : unit impulse response of first order system

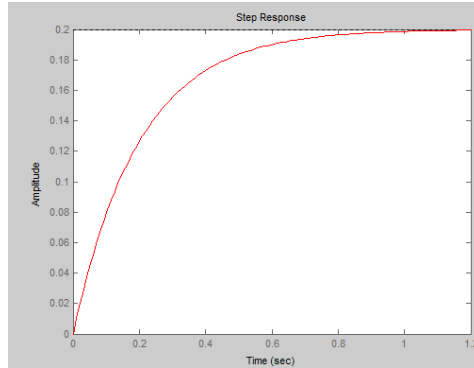
**Example 1: Obtain step response of a unity feedback system having forward path transfer function of**

$$G(s) = \frac{1}{s+4}$$

**Matlab Code:**

```
num = [1];
den = [1 4]
g = tf (num,den)
t = feedback(g,1)
step(t, 'r')
```

Output :

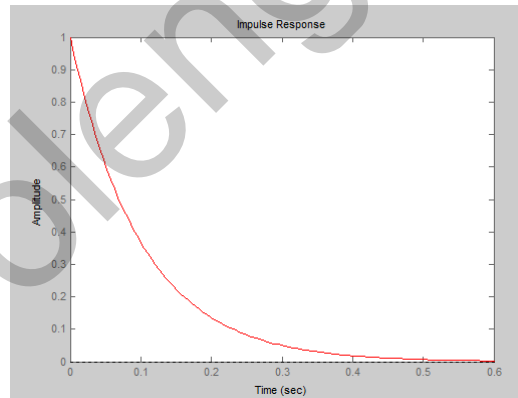


**Example 2: Obtain impulse response of a unity feedback system having forward path transfer function of  $G(s) = \frac{1}{s+9}$**

Matlab Code :

```
num = [1];
den = [1 9]
g = tf (num,den)
t = feedback(g,1)
impulse(t, 'r')
```

Output:



**Assignments:**

**Obtain step and impulse response of the following systems with unity feedback connection. Also verify your result theoretically**

$$1. G(s) = \frac{1}{s+11} \quad 2. G(s) = \frac{1}{s+2}$$

**DISCUSSION:**

**1. Explain why a series RL circuit with high inductance has a slow response?**

## EXPERIMENT NO : CS I/3

**TITLE :** DETERMINATION OF STEP & IMPULSE RESPONSE FOR A SECOND ORDER UNITY FEEDBACK SYSTEM

**OBJECTIVE :** To determine

- I. Step response of 2<sup>nd</sup> order system
- II. Impulse response of 2<sup>nd</sup> order system

**THEORY:** The time response has utmost importance for the design and analysis of control systems because these are inherently time domain systems where time is independent variable. During the analysis of response, the variation of output with respect to time can be studied and it is known as time response. To obtain satisfactory performance of the system with respect to time must be within the specified limits. From time response analysis and corresponding results, the stability of system, accuracy of system and complete evaluation can be studied easily. Due to the application of an excitation to a system, the response of the system is known as time response and it is a function of time. The two parts of response of any system:  
Transient response; Steady-state response.

**Transient response:** The part of the time response which goes to zero after large interval of time is known as transient response.

**Steady state response:** The part of response that means even after the transients have died out is said to be steady state response.

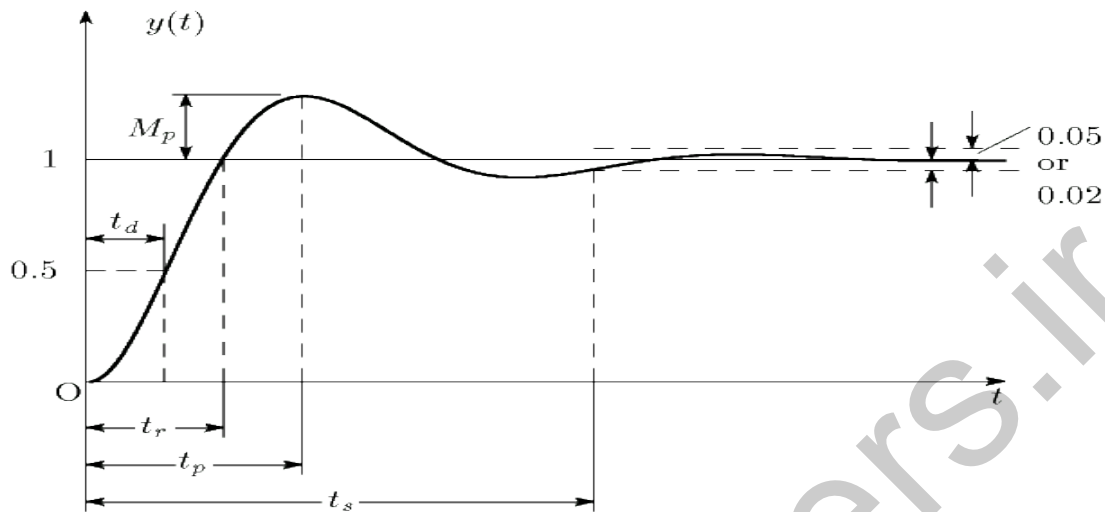
The total response of a system is sum of transient response and steady state response:  $C(t) = C_{tr}(t) + C_{ss}(t)$

**Time Response Specification Parameters:** The transfer function of a 2-nd order system is generally represented by the following transfer function:

$$\frac{Y(S)}{R(S)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The dynamic behavior of the second-order system can then be described in terms of two parameters: the damping ratio and the natural frequency.

If the damping ratio is between 0 and 1, the system poles are complex conjugates and lie in the left-half  $s$  plane. The system is then called **underdamped**, and the transient response is **oscillatory**. If the damping ratio is equal to 1 the system is called **critically damped**, and when the damping ratio is larger than 1 we have **overdamped** system. The transient response of critically damped and overdamped systems do not oscillate. If the damping ratio is 0, the transient response does not die out.

**Delay time ( $t_d$ )**

The delay time is the time required for the response to reach half the final value the very first time.

**Rise time ( $t_r$ )**

The rise time is the time required for the response to rise from 10% to 90%, 5% to 95%, or 0% to 100% of its final value. For underdamped second-order systems, the 0% to 100% rise time is normally used. For overdamped systems, the 10% to 90% rise time is commonly used.

**Peak time ( $t_p$ )**

The peak time is the time required for the response to reach the first peak of the overshoot.

**Maximum (percent) overshoot ( $M_p$ )**

The maximum overshoot is the maximum peak value of the response curve measured from unity. If the final steady-state value of the response differs from unity, then it is common to use the maximum percent overshoot. It is defined by

$$\frac{y(t_p) - c(\infty)}{c(\infty)} \times 100\%$$

**Settling time ( $t_s$ )**

The settling time is the time required for the response curve to reach and stay within a range about the final value of size specified by absolute percentage of the final value (usually 2% or 5%). The settling time is related to the largest time constant of the control system.

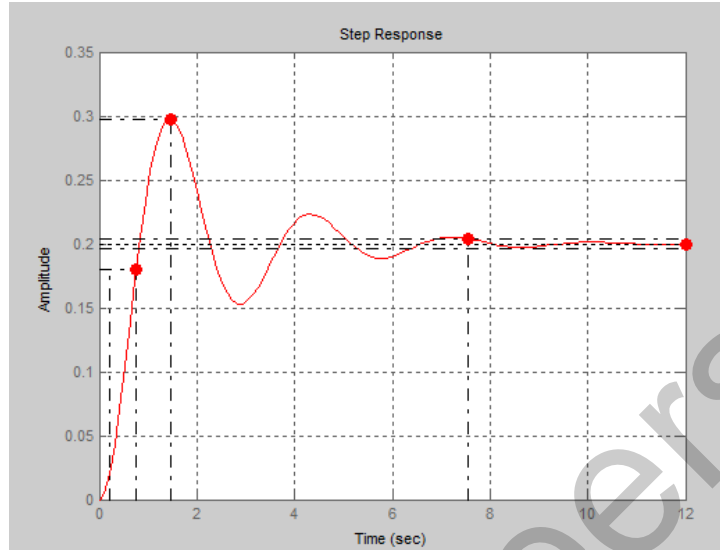
**Example 1: Obtain step response of a unity feedback system having forward path transfer function of**

$$G(s) = \frac{1}{s^2 + s + 4}$$

**Matlab Code:**

```
num = [1];
den = [1 1 4]
g = tf (num,den)
t = feedback(g,1)
step(t, 'r')
```

Output :



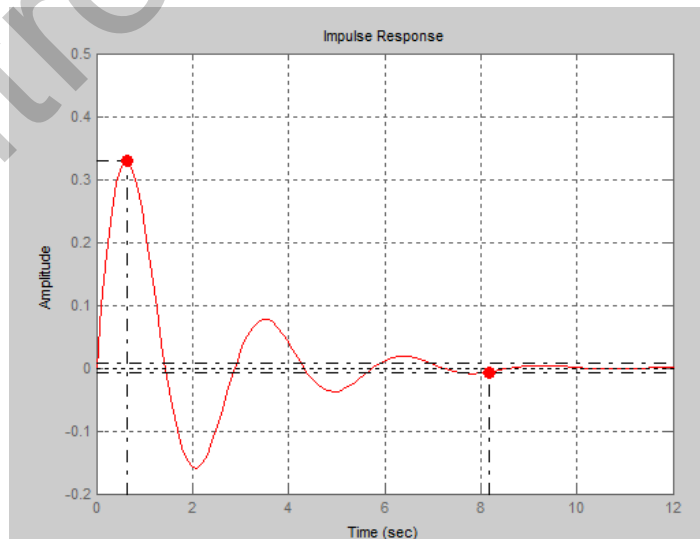
**Example 2: Obtain impulse response of a unity feedback system having forward path transfer function**

$$\text{of } G(s) = \frac{1}{s^2 + s + 4}$$

Matlab Code :

```
num = [1];
den = [1 9]
g = tf (num,den)
t = feedback(g,1)
impulse(t, 'r')
```

Output:



**Assignments:**

**Obtain step and impulse response of the following systems with unity feedback connection. Also verify your result theoretically**

1. 
$$G(s) = \frac{19}{(s+4)(s+8)}$$

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

**DISCUSSION:**

1. What do you mean by rise time? Derive its expression for a unity feedback 2<sup>nd</sup> order control system.
2. Why is less overshoot desired for practical systems?

controlengineers.ir

**EXPERIMENT NO : CS I/4**

**TITLE : DETERMINATION OF STEP & IMPULSE RESPONSE FOR A TYPE ‘0’, TYPE ‘1’, TYPE ‘2’ SYSTEMS**

**OBJECTIVE :** To determine

- I. Step response of Type ‘0’ system
- II. Impulse response of Type ‘0’ system
- III. Step response of Type ‘1’ system
- IV. Impulse response of Type ‘1’ system
- V. Step response of Type ‘2’ system
- VI. Impulse response of Type ‘2’ system

**THEORY:**

A real time system can be expressed by its transfer function. Based on presence of poles at origin of s plane , transfer functions can be classified as Type ‘0’, Type ‘1’, Type ‘2’ ..... Systems. Let open loop transfer function a system is expressed as:

$$G(s)H(s) = \frac{K(1+sT_a)(1+sT_b).....}{s^N(1+sT_1)(1+sT_2).....} \dots\dots\dots (i)$$

From equation (i) it is clear that ‘N’ determines the number of poles at origin.

For a Type ‘0’ system, N = 0

For a Type ‘1’ system, N = 1

For a Type ‘2’ system, N = 2

⋮  
⋮  
⋮  
⋮

For a Type ‘N’ system, N = N

The steady state error can be found out by the following equation

$$e_{ss} = \lim_{s \rightarrow 0} \frac{sR(s)}{1 + G(s)H(s)}$$

Type	Step Input		Ramp input		Parabolic input	
	K <sub>P</sub>	E <sub>SS</sub>	K <sub>V</sub>	E <sub>SS</sub>	K <sub>a</sub>	E <sub>SS</sub>
0	K	$\frac{1}{1+K}$	0	$\infty$	0	$\infty$
1	$\infty$	0	K	$\frac{1}{K}$	0	$\infty$
2	$\infty$	0	$\infty$	0	K	$\frac{1}{K}$

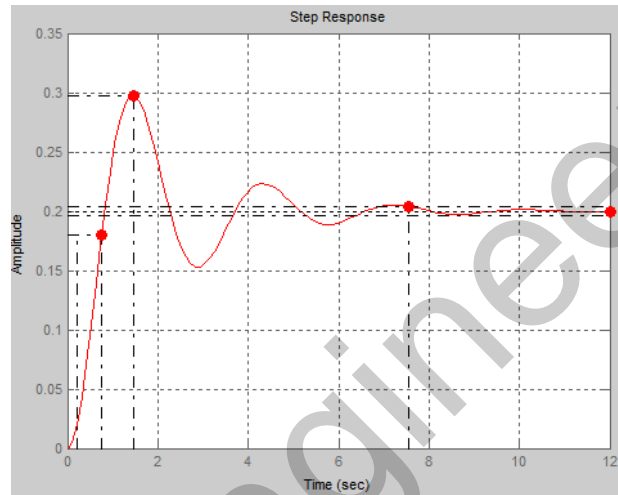
**Example 1: Obtain step response of a Type '0' system having forward path transfer function of  $G(s) =$**

$$\frac{1}{s^2+s+4}$$

**Matlab Code :**

```
num = [1];
den = [1 1 4]
g = tf (num,den)
t = feedback(g,1)
step(t, 'r')
```

Output:



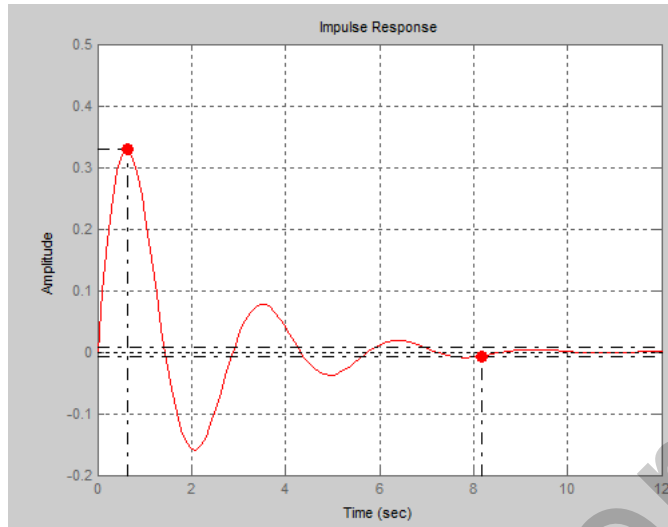
**Example 2: Obtain impulse response of a Type '0' system having forward path transfer function of  $G(s)$**

$$= \frac{1}{s^2+s+4}$$

**Matlab Code :**

```
num = [1];
den = [1 1 4]
g = tf (num,den)
t = feedback(g,1)
impulse(t, 'r')
```

Output:

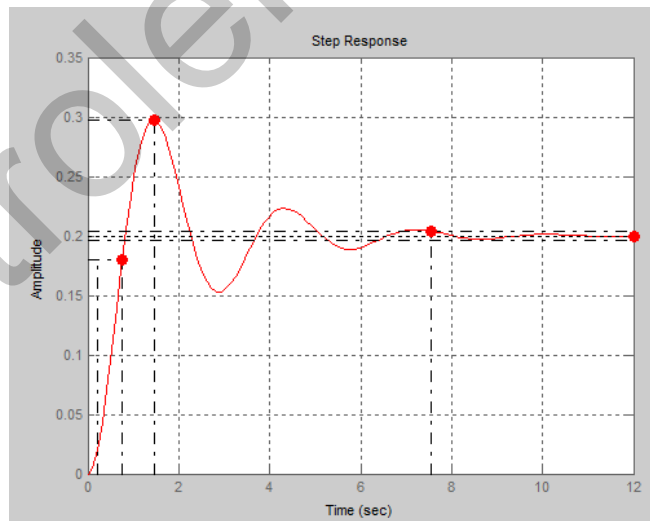


**Example 3: Obtain step response of a Type '1' system having forward path transfer function of  $G(s) = \frac{1}{s(s^2+s+4)}$**

**Matlab Code :**

```
num = [1];
den = [0 1 1 4]
g = tf (num,den)
t = feedback(g,1)
step(t, 'r')
```

Output :

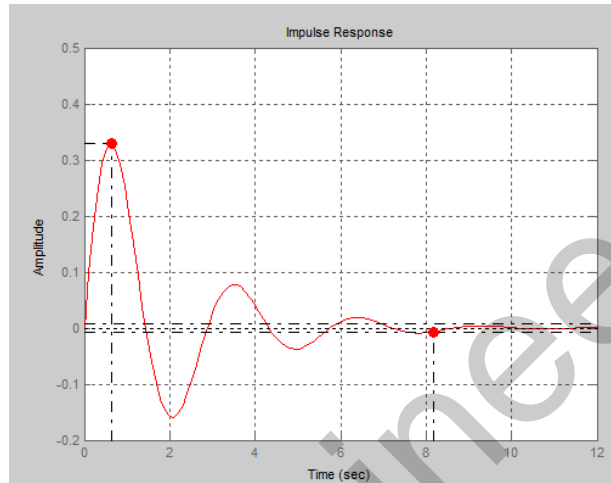


**Example 4: Obtain Impulse response of a Type '1' system having forward path transfer function of  $G(s) = \frac{1}{s(s^2+s+4)}$**

**Matlab Code :**

```
num = [1];
den = [0 1 1 4]
g = tf (num,den)
t = feedback(g,1)
impulse(t,'r')
```

Output:



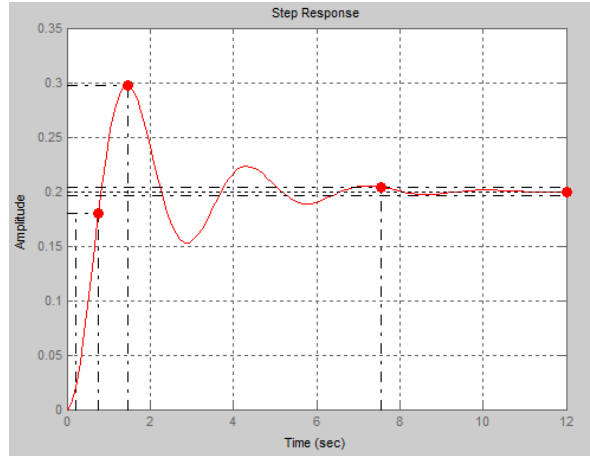
**Example 5: Obtain Step response of a Type '2' system having forward path transfer function of  $G(s) =$**

$$\frac{1}{s^2(s^2+s+4)}$$

**Matlab Code :**

```
num = [1];
den = [0 0 1 1 4]
g = tf (num,den)
t = feedback(g,1)
step(t,'r')
```

Output:



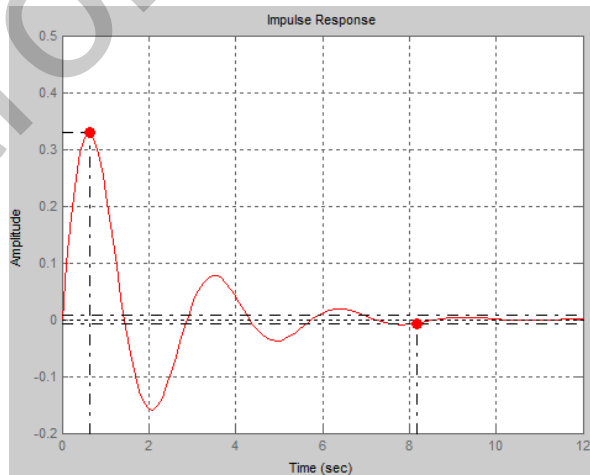
**Example 6: Obtain Impulse response of a Type ‘2’ system having forward path transfer function of G(s)**

$$= \frac{1}{s^2(s^2+s+4)}$$

**Matlab Code :**

```
num = [1];
den = [0 0 1 1 4]
g = tf (num,den)
t = feedback(g,1)
impulse(t, 'r')
```

**Output :**



**DISCUSSION:**

1. What would be steady state error for a type 1 system if unit ramp input is applied?

## EXPERIMENT NO : CS I/5

**TITLE :** DETERMINATION OF BODE PLOT USING MATLAB CONTROL SYSTEM TOOLBOX FOR 2<sup>ND</sup> ORDER SYSTEM & OBTAIN CONTROLLER SPECIFICATION PARAMETERS.

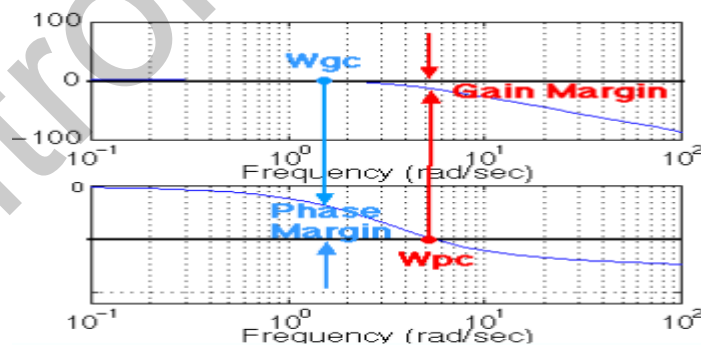
**OBJECTIVE :** To determine

- I. Bode plot of a 2<sup>nd</sup> order system
- II. Frequency domain specification parameters

**THEORY:** The frequency response method may be less intuitive than other methods you have studied previously. However, it has certain advantages, especially in real-life situations such as modeling transfer functions from physical data. The frequency response of a system can be viewed two different ways: via the Bode plot or via the Nyquist diagram. Both methods display the same information; the difference lies in the way the information is presented. We will explore both methods during this lab exercise. The frequency response is a representation of the system's response to sinusoidal inputs at varying frequencies. The output of a linear system to a sinusoidal input is a sinusoid of the same frequency but with a different magnitude and phase. The frequency response is defined as the magnitude and phase differences between the input and output sinusoids. In this lab, we will see how we can use the open-loop frequency response of a system to predict its behavior in closed-loop. To plot the frequency response, we create a vector of frequencies (varying between zero or "DC" and infinity i.e., a higher value) and compute the value of the plant transfer function at those frequencies. If  $G(s)$  is the open loop transfer function of a system and  $\omega$  is the frequency vector, we then plot  $G(j\omega)$  vs.  $\omega$ . Since  $G(j\omega)$  is a complex number, we can plot both its magnitude and phase (the Bode plot) or its position in the complex plane (the Nyquist plot).

The gain margin is defined as the change in open loop gain required to make the system unstable. Systems with greater gain margins can withstand greater changes in system parameters before becoming unstable in closed loop.

The phase margin is defined as the change in open loop phase shift required to make a closed loop system unstable.



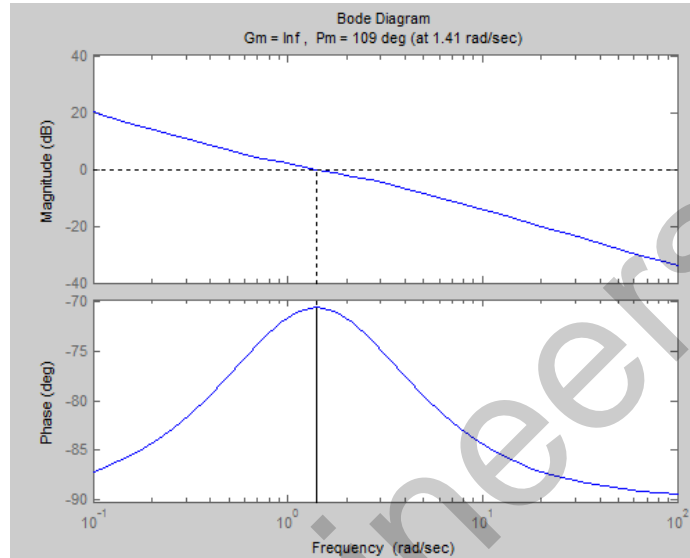
**Example 1:** Obtain Bode Plot of the system having forward path transfer function of  $G(s) = \frac{1+s}{s(1+0.5s)}$

**Matlab Code:**

```
num = [1 1]
den = conv([1 0],[.5 1])
```

```
g = tf(num,den);
bode(g)
margin(g)
```

Output :



### Assignments:

Obtain bode diagram of the following transfer function. Also verify your result theoretically.

$$G(s) = \frac{4(2+s)}{s(5+s)(s+10)}$$

### DISCUSSION:

1. What do you mean by GM & PM?
2. How GM & PM affects system?
3. What is gain cross over frequency?

## EXPERIMENT NO : CS I/6

**TITLE :** DETERMINATION OF ROOT LOCUS PLOT USING MATLAB CONTROL SYSTEM TOOLBOX FOR 2<sup>ND</sup> ORDER SYSTEM & OBTAIN CONTROLLER SPECIFICATION PARAMETERS.

**OBJECTIVE :** To determine Root Locus plot of a 2<sup>nd</sup> order system

**THEORY:** rlocus computes the Evans root locus of a SISO open-loop model. The root locus gives the closed-loop pole trajectories as a function of the feedback gain k (assuming negative feedback). Root loci are used to study the effects of varying feedback gains on closed-loop pole locations. In turn, these locations provide indirect information on the time and frequency responses. rlocus(sys) calculates and plots the rootlocus of the open-loop SISO model sys. This function can be applied to any of the following feedback loops by setting sys appropriately. If sys has transfer function  $G(s) = \frac{N(s)}{D(s)}$ , The closed-loop poles are the roots of  $d(s) + k*n(s)=0$

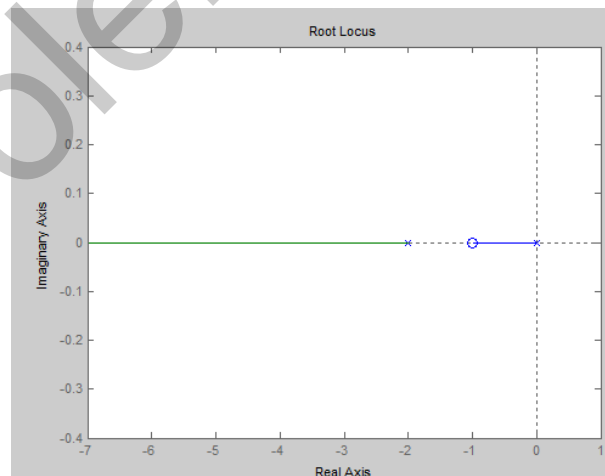
**Example 1: Obtain Root Locus Plot of a system having forward path transfer function of**  $G(s) =$

$$\frac{1+s}{s(1+0.5s)}$$

**Matlab Code:**

```
num = [1 1]
den = conv([1 0],[.5 1])
g = tf(num,den);
rlocus(g)
```

**Output :**



**Example 2: Obtain Root Locus Plot of a system having forward path transfer function of**  $G(s) =$

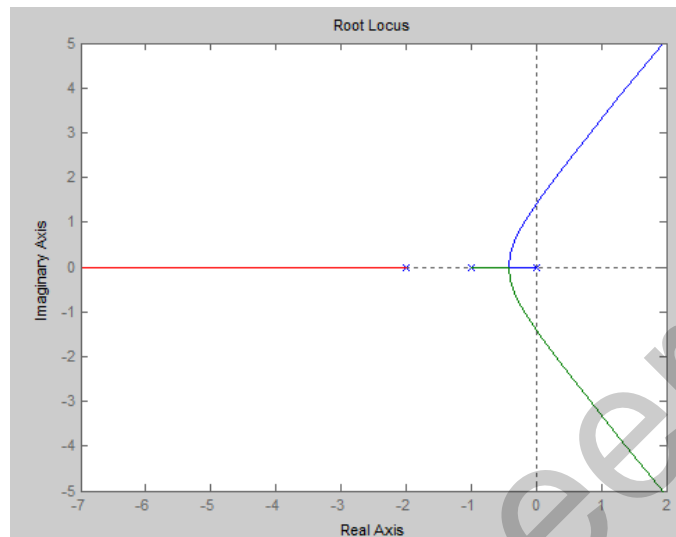
$$\frac{1}{s(1+s)(s+2)}$$

**Matlab Code :**

```
num = [1]
den = poly([0 -1 -2])
g = tf(num,den);
```

rlocus(g)

Output :



### Assignment:

Obtain Root Locus Plot of the following transfer function. Also verify your result theoretically.

$$G(s) = \frac{(s+6)}{(3+s)(s+6)}$$

### DISCUSSION:

1. What is relative stability? How can you measure relative stability using Root locus?
2. What do you mean by break away point?
3. What is asymptotic line?

## EXPERIMENT NO : CS I/7

### TITLE : DETERMINATION OF NYQUIST PLOT USING MATLAB CONTROL SYSTEM TOOLBOX.

**OBJECTIVE :** To determine nyquist plot of a 2<sup>nd</sup> order system

**THEORY:** A stability test for time invariant linear systems can also be derived in the frequency domain. It is known as Nyquist stability criterion. It is based on the complex analysis result known as Cauchy's principle of argument. Note that the system transfer function is a complex function. By applying Cauchy's principle of argument to the open-loop system transfer function, we will get information about stability of the closed-loop system transfer function and arrive at the Nyquist stability criterion (Nyquist, 1932). The importance of Nyquist stability lies in the fact that it can also be used to determine the relative degree of system stability by producing the so-called phase and gain stability margins. These stability margins are needed for frequency domain controller design techniques.

We present only the essence of the Nyquist stability criterion and define the phase and gain stability margins. The Nyquist method is used for studying the stability of linear systems with pure time delay. For a SISO feedback system the closed-loop transfer function is given by:

$$T(s) = \frac{G(s)}{1+G(s)H(s)}$$

Since the system poles are determined as those values at which its transfer function becomes infinity, it follows that the closed-loop system poles are obtained by solving the following equation.

In the following we consider the complex function

$$D(s) = 1 + G(s)H(s)$$

whose zeros are the closed-loop poles of the transfer function. In addition, it is easy to see that the poles of  $D(s)$  are the zeros of  $T(s)$ . At the same time the poles of  $D(s)$  are the open-loop control system poles since they are contributed by the poles of  $G(s)H(s)$ , which can be considered as the open-loop control system transfer function—obtained when the feedback loop is open at some point. The Nyquist stability test is obtained by applying the Cauchy principle of argument to the complex function  $D(s)$ . First, we state Cauchy's principle of argument.

Let  $F(s)$  be an analytic function in a closed region of the complex plane except at a finite number of points. It is also assumed that  $F(s)$  is analytic at every point on the contour. Then, as  $s$  travels around the contour in the  $s$ -plane in the clockwise direction, the function  $F(s)$  encircles the origin in the  $F(s)$ -plane in the same direction  $N$  times, given by  $N = Z - P$

Where  $Z$  and  $P$  stand for the number of zeros and poles (including their multiplicities) of the function inside the  $F(s)$  contour.

The above result can be also written as

$$\text{Arg}(F(s)) = 2\pi N$$

#### Nyquist Criterion:

It states that the number of unstable closed-loop poles is equal to the number of unstable open-loop poles plus the number of encirclements of the origin of the Nyquist plot of the complex function  $D(s)$ . This can be easily justified by applying Cauchy's principle of argument to the function  $D(s)$  with the  $s$ -plane contour. Note that  $Z$  and  $P$  represent the numbers of zeros and poles, respectively, of in the unstable part of the complex plane. At

the same time, the zeros of  $D(s)$  are the closed-loop system poles, and the poles of  $D(s)$  are the open-loop system poles (closed-loop zeros).

The above criterion can be slightly simplified if instead of plotting the function  $D(s) = 1 + G(s)H(s)$

, we plot only the function  $G(s)H(s)$  and count encirclement of the Nyquist plot of around the point  $(-1+j0)$ , so that the modified Nyquist criterion has the following form. The number of unstable closed-loop poles ( $Z$ ) is equal to the number of unstable open-loop poles ( $P$ ) plus the number of encirclements ( $N$ ) of the point  $(=1+j0)$ ,

$$Z=N+P$$

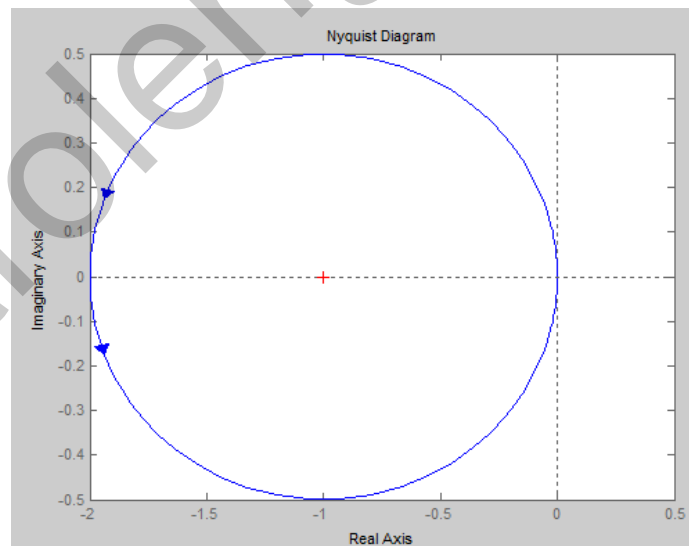
**Example 1: Obtain Nyquist Plot of a system having forward path transfer function of**

$$G(s) = \frac{2+s}{(s+1)(s-1)}$$

**Matlab Code:**

```
num = [1 2]
den = conv([1 1], [1 -1])
g = tf(num,den);
nyquist(g)
```

Output :



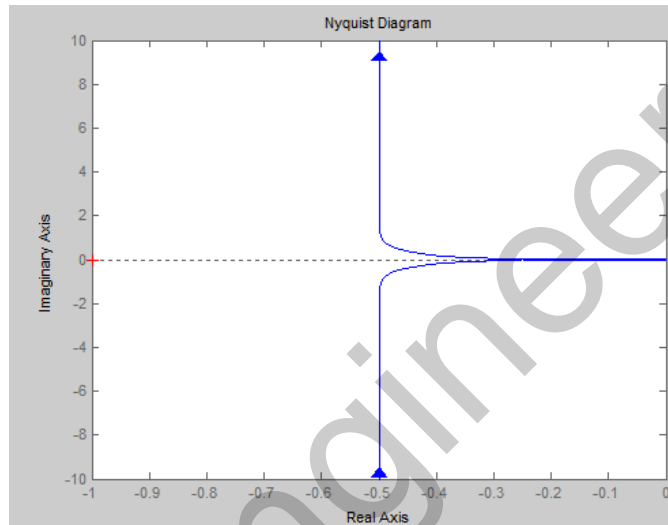
**Example 2: Obtain Nyquist Plot of a ufb system having forward path transfer function of**

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

**Matlab Code :**

```
num = [1]
a=[1 0]
b= [1 2 2]
den =conv(a,b)
g = tf(num,den)
nyquist(g)
```

Output :

**Assignment:**

Obtain Nyquist Plot of the following transfer function. Also verify your result theoretically.

$$G(s) = \frac{(s+8)}{s(3+s)(s+6)}$$

**DISCUSSION:**

1. What is nyquist contour?
2. What is polar plot? How it differs from Nyquist plot?

## EXPERIMENT NO : CS I/8

**TITLE :** STUDY THE EFFECT OF PI & PD CONTROLLER ON SYSTEM PERFORMANCE

**OBJECTIVE :** To determine

- I. Effect of PI controller on system performance
- II. Effect of PD controller on system performance

**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 its 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  $90^\circ$ .

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  $T_i$ , 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:

1. **Example 1:** Consider a unity feedback system with forward path transfer function  $G(s) = \frac{1}{s^2+10s+20}$ . Show the effect of addition of a PD controller on the system performance.

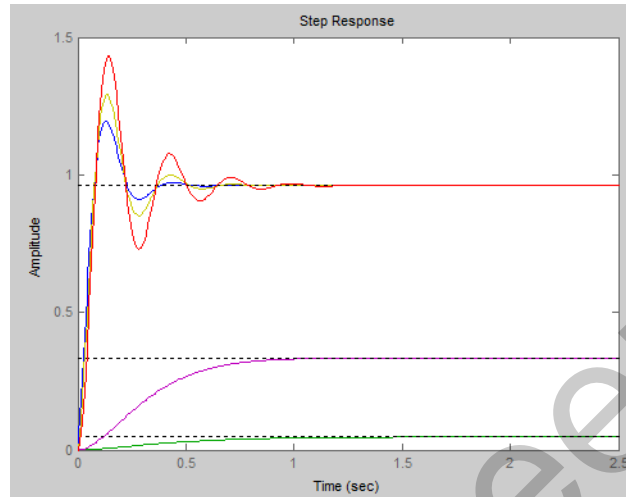
**Matlab Code:**

```

num=1;
den=[1 10 20];
g1=tf (num,den)
t1=feedback(g1,1)
step(t1, 'g')
hold on
num1=10;
den1=[1 10 20];
g2=tf (num1,den1)
t2=feedback(g2,1)
step(t2, 'm')
hold on
Kp=500;
Kd=10;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
t3=feedback(g3,1)
step(t3, 'b')
hold on
Kp=500;
Kd=5;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)
t4=feedback(g3,1)
step(t4, 'y')
hold on
Kp=500;
Kd=.01;
numc=[Kd Kp];
numo=conv(numc,num)
deno=den
g3=tf(numo,deno)

```

```
t5=feedback(g3,1)
step(t5,'r')
hold on
Output:
```



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

Matlab Code:

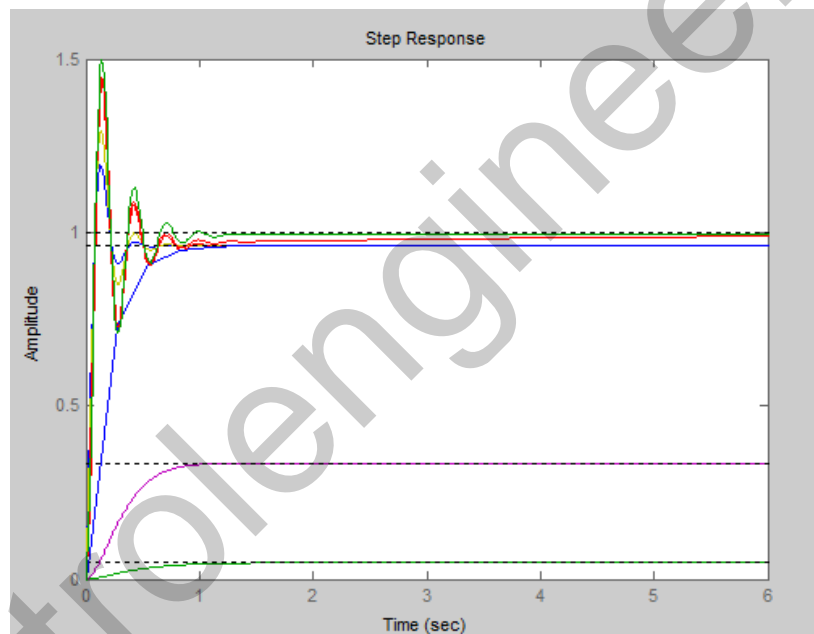
```
num=1;
den=[1 10 20];
g1=tf (num,den)
t1=feedback(g1,1)
step(t1,'g')
hold on
num1=10;
den1=[1 10 20];
g2=tf (num1,den1)
t2=feedback(g2,1)
step(t2,'m')
hold on
Kp=500;
Ki = 1
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
deno=conv(den,denc)
g3=tf(numo,deno)
t3=feedback(g3,1)
step(t3,'b')
hold on
Kp=500;
Ki = 100
numc=[Kp Ki];
denc= [1 0]
numo=conv(numc,num)
```

```

deno=conv(den,denc)
g3=tf(numo,deno)
t4=feedback(g3,1)
step(t4,'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)
t5=feedback(g3,1)
step(t5,'g')
hold on

```

Output :



### Assignments:

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

## EXPERIMENT NO : CS I/9

**TITLE :** STUDY THE EFFECT OF ADDITION OF ZEROS TO THE FORWARD PATH TRANSFER FUNCTION OF A CLOSED LOOP SYSTEM

**OBJECTIVE :** To study

- I. Effect of addition of zeros to forward path of an open loop system.
- II. Effect of addition of zeros to forward path of a closed loop system.

**THEORY:** The forward path transfer function of a general second order system is given by,

$$FPTF \quad G(s) = \frac{W_n^2}{s(s + 2\zeta W_n)}$$

Addition of zero to forward path transfer function:

When we add a zero the forward path transfer function becomes,

$$G'(s) = \frac{W_n^2(T_z s + 1)}{s(s + 2\zeta W_n)}$$

$$G'(s) = \frac{W_n^2(s + z)}{s(s + 2\zeta W_n)z}$$

$$\frac{C(s)}{R(s)} = \frac{G'(s)}{1 + G'(s)H(s)} = \frac{G'(s)}{1 + G'(s)}$$

$$\frac{C(s)}{R(s)} = \frac{W_n^2(s + z)}{zs^2 + (W_n^2 + 2\zeta W_n z)s + W_n^2 z}$$

Example 1. Consider an open loop system having forward path transfer function of  $G(S)=1/S*(S+1)$ . Write a MATLAB code to show the effect of addition of zeros at -2, -1, -0.5.

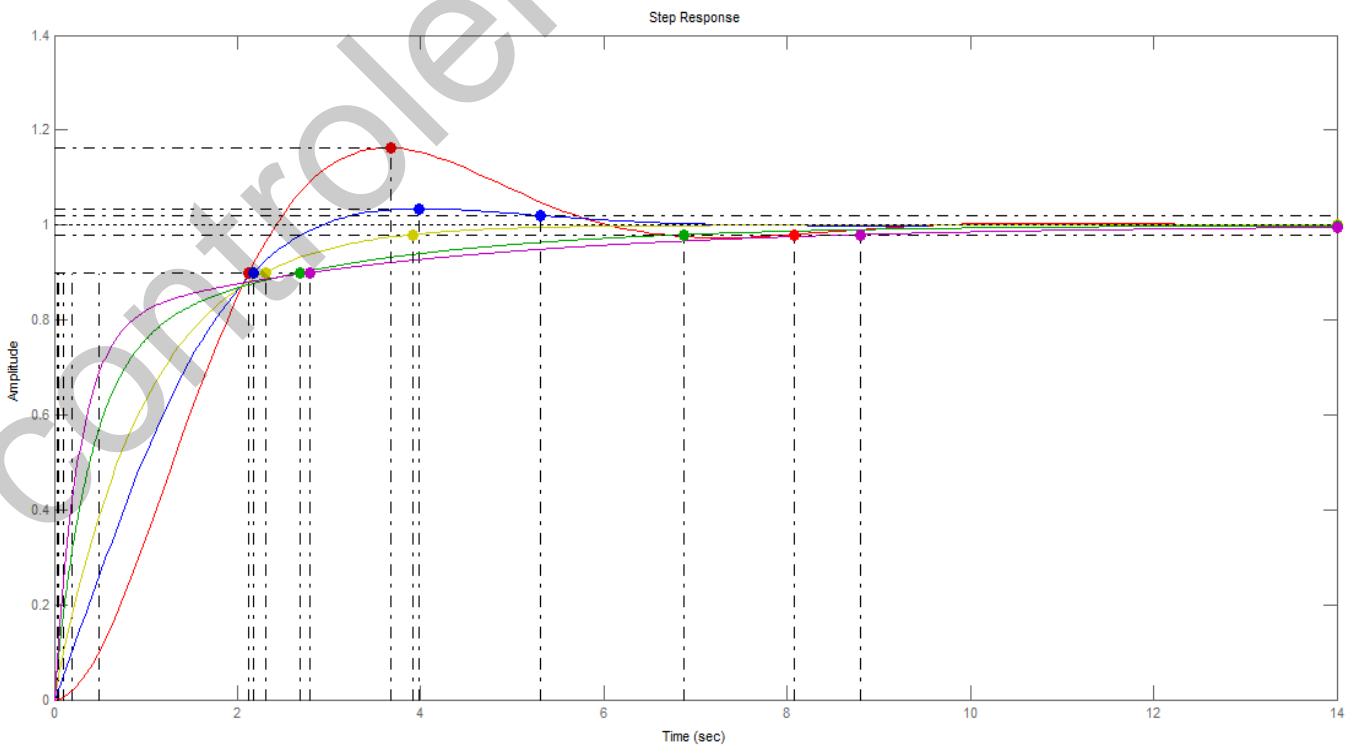
**MATLAB CODE:**

```
n1=1
d1=[1 1 0]
g1=tf(n1,d1)
t1=feedback(g1,1)
step(t1,'r')
hold on
Tz=0.5
Z1=[Tz 1]
```

```

n2=conv(n1,Z1)
g2=tf(n2,d1)
t2=feedback(g2,1)
step(t2,'b')
hold on
Tz=1
Z2=[Tz 1]
n3=conv(n1,Z2)
g3=tf(n3,d1)
t3=feedback(g3,1)
step(t3,'y')
hold on
Tz=2
Z3=[Tz 1]
n4=conv(n1,Z3)
g4=tf(n4,d1)
t4=feedback(g4,1)
step(t4,'g')
hold on
Tz=3
Z4=[Tz 1]
n5=conv(n1,Z4)
g5=tf(n5,d1)
t5=feedback(g5,1)
step(t5,'m')

```



**Observation Table:**

Tz	Tr	Tp	Ts	%Mp
0.5				
1				
2				
3				

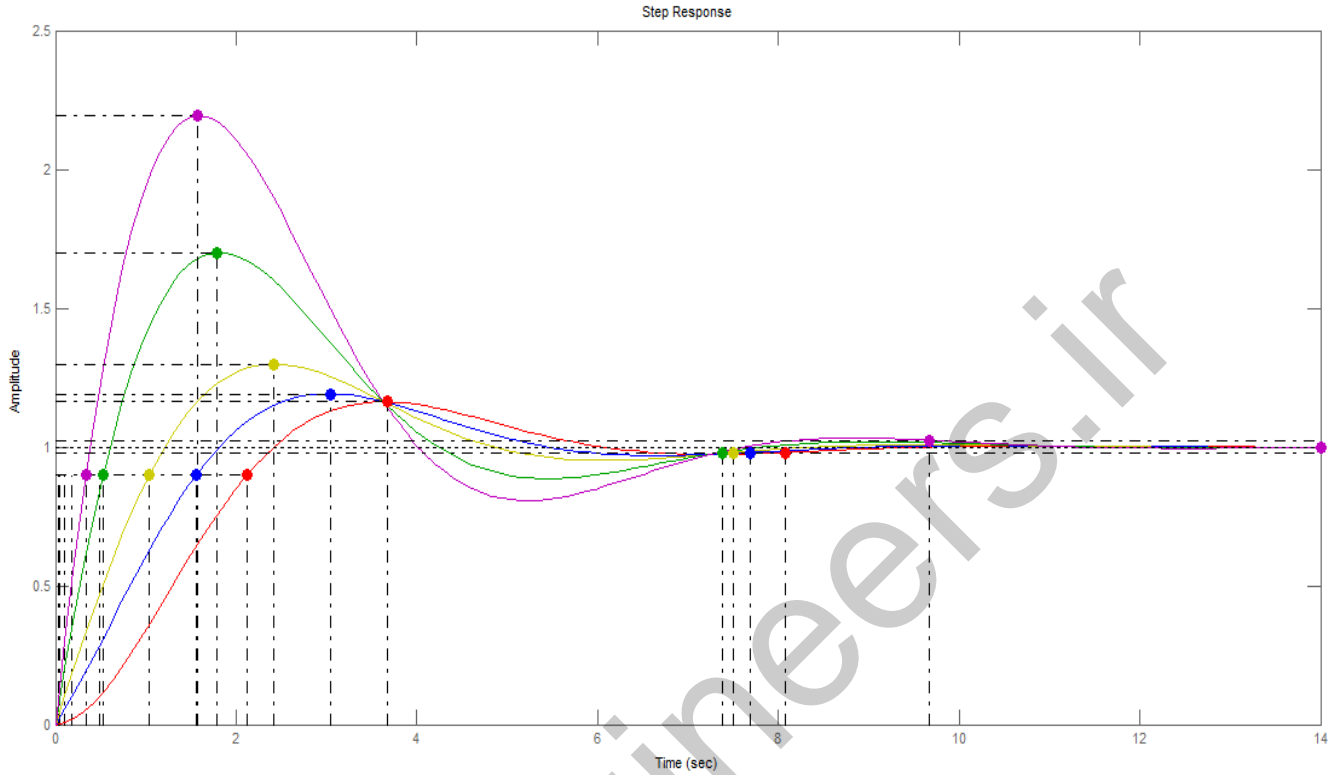
Example 2. Consider a closed loop system having overall transfer function of  $T(S)=1/(S^2+S+1)$ . Write a MATLAB code to show the effect of addition of zeros at -2, -1, -0.5.

**MATLAB CODE:**

```

n1=[1]
d1=[1 1 1]
t1=tf(n1,d1)
step(t1,'r')
hold on
Tz=0.5
Z1=[Tz 1]
n2=conv(n1,Z1)
t2=tf(n2,d1)
step(t2,'b')
hold on
Tz=1
Z2=[Tz 1]
n3=conv(n1,Z2)
t3=tf(n3,d1)
step(t3,'y')
hold on
Tz=2
Z3=[Tz 1]
n4=conv(n1,Z3)
t4=tf(n4,d1)
step(t4,'g')
hold on
Tz=3
Z4=[Tz 1]
n5=conv(n1,Z4)
t5=tf(n5,d1)
step(t5,'m')

```



**Observation Table:**

$T_z$	$t_r$	$T_p$	$T_s$	%Mp
0.5				
1				
2				
3				

**DISCUSSION:**

1. Discuss the effect of adding zeros to overall transfer function on system response curve.

## EXPERIMENT NO : CS I/10

**TITLE :** STUDY THE EFFECT OF ADDITION OF POLES TO THE FORWARD PATH TRANSFER FUNCTION OF A CLOSED LOOP SYSTEM

**OBJECTIVE :** To study

- I. Effect of addition of poles to forward path of an open loop system.
- II. Effect of addition of poles to forward path of a closed loop system.

The forward path transfer function of a general second order system is given by,

$$FPTF \quad G(s) = \frac{W_n^2}{s(s + 2\zeta W_n)}$$

Addition of pole to forward path transfer function:

When we add a pole, the transfer function becomes,

$$G'(s) = \frac{W_n^2}{s(s + 2\zeta W_n)(T_p s + 1)}$$

$$G'(s) = \frac{W_n^2 P}{s(s + 2\zeta W_n)(s + P)}$$

$$\frac{C(s)}{R(s)} = \frac{G'(s)}{1 + G'(s)H(s)} = \frac{G'(s)}{1 + G'(s)}$$

$$\frac{C(s)}{R(s)} = \frac{W_n^2 P}{s^3 + s^2(P + 2\zeta W_n) + 2\zeta W_n P s + W_n^2 P}$$



Example 1. Consider a open loop system having forward path transfer function of  $G(S)=1/S*(S+2)$ . Write a MATLAB code to show the effect of addition of poles at -1, -0.5, -0.25.

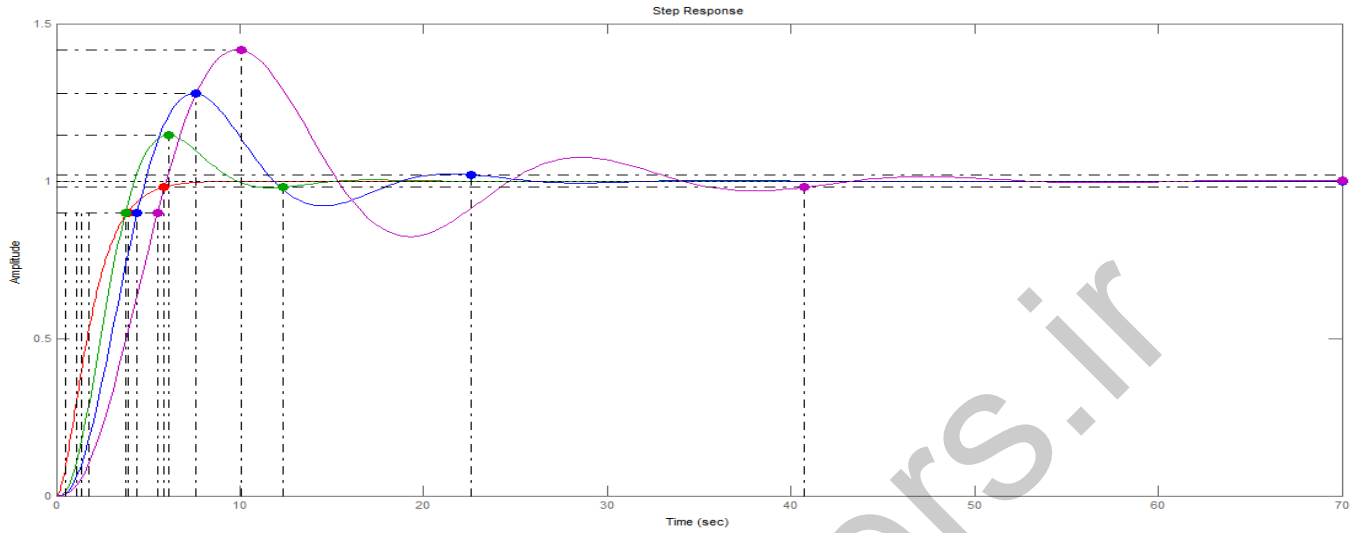
$$G_{\text{new}}(S)=[1/S*(S+2)]*[1/(TpS+1)]$$

### MATLAB CODE:

```

n1=1
d1=[1 2 0]
g1=tf(n1,d1)
t1=feedback(g1,1)
step(t1,'r')
hold on
Tp=1
P1=[Tp 1]
d2=conv(d1,P1)
g2=tf(n1,d2)
t2=feedback(g2,1)
step(t2,'g')
hold on
Tp=2
P2=[Tp 1]
d3=conv(d1,P2)
g3=tf(n1,d3)
t3=feedback(g3,1)
step(t3,'b')
hold on
Tp=4
P3=[Tp 1]
d4=conv(d1,P3)
g4=tf(n1,d4)
t4=feedback(g4,1)
step(t4,'m')

```



**Observation Table:**

Tp	tr	Tp	Ts	%Mp
1				
2				
4				

Example 2. Consider a closed loop system having overall transfer function of  $T(S)=1/(S^2+ 0.6S+1)$ . Write a MATLAB code to show the effect of addition of zeros at -2, -1, -0.5.

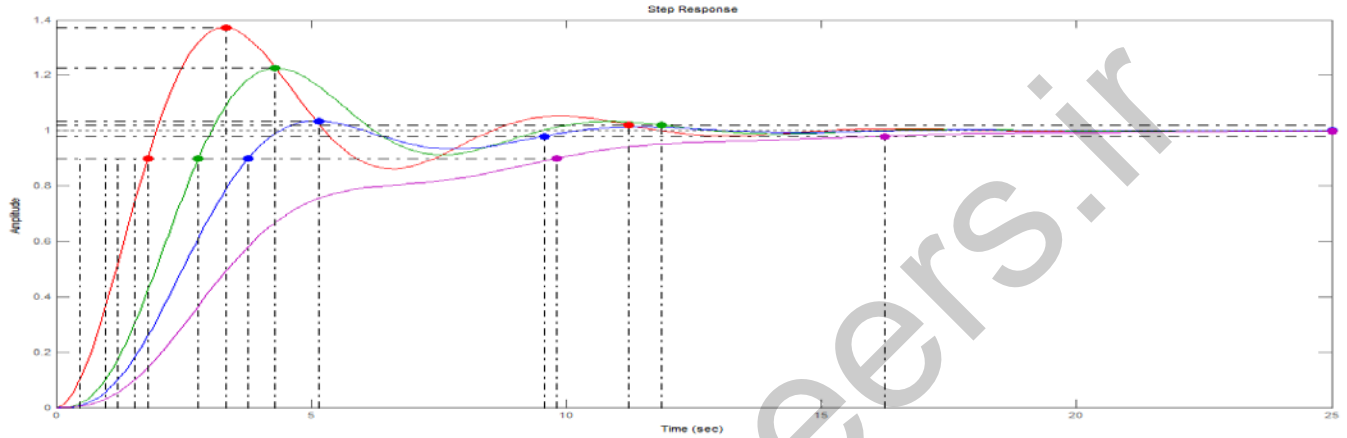
**MATLAB CODE:**

```

n1=1
d1=[1 0.6 1]
t1=tf(n1,d1)
step(t1,'r')
hold on
Tp=1
P1=[Tp 1]
d2=conv(d1,P1)
t2=tf(n1,d2)
step(t2,'g')
hold on
Tp=2
P2=[Tp 1]
d3=conv(d1,P2)
t3=tf(n1,d3)
step(t3,'b')
hold on
Tp=4
    
```



```
P3=[Tp 1]
d4=conv(d1,P3)
t4=tf(n1,d4)
step(t4,'m')
```



**Observation Table:**

Tp	tr	Tp	ts	%Mp
1				
2				
4				

**DISCUSSION:**

1. Discuss the effect of adding poles to overall transfer function on system response curve.