Date-15/02/2024

EXPERIMENT - 2

TIME RESPONSE ANALYSIS OF SECOND ORDER SYSTEM

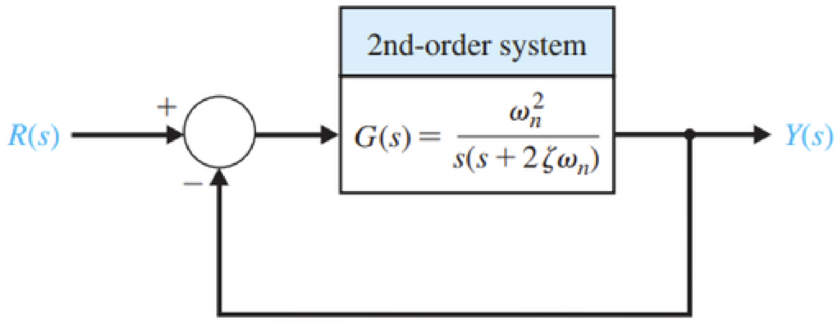
Aim-

The aim of this experiment is to evaluate a second-order system's performance and dynamic behavior in time response to input signals over time through software tools such as MATLAB

Software used-

MATLAB R2022b

Block Diagram-



Equations used-

Let us consider a single-loop second-order system and determine its response to a unit step input. A closed-loop feedback control system is shown in above figure. The closed-loop transfer function is

$$Y(s) = \frac{G(s)}{1 + G(s)} R(s). \quad \text{--(1)}$$

We may rewrite Equation (1) as-

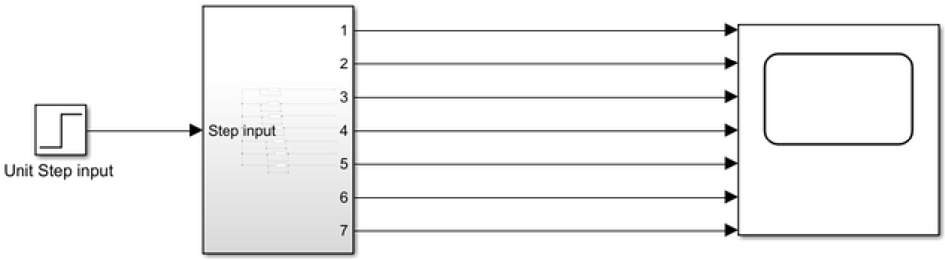
$$Y(s) = \frac{k\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} R(s).$$

where ζ is the dimensionless damping ratio

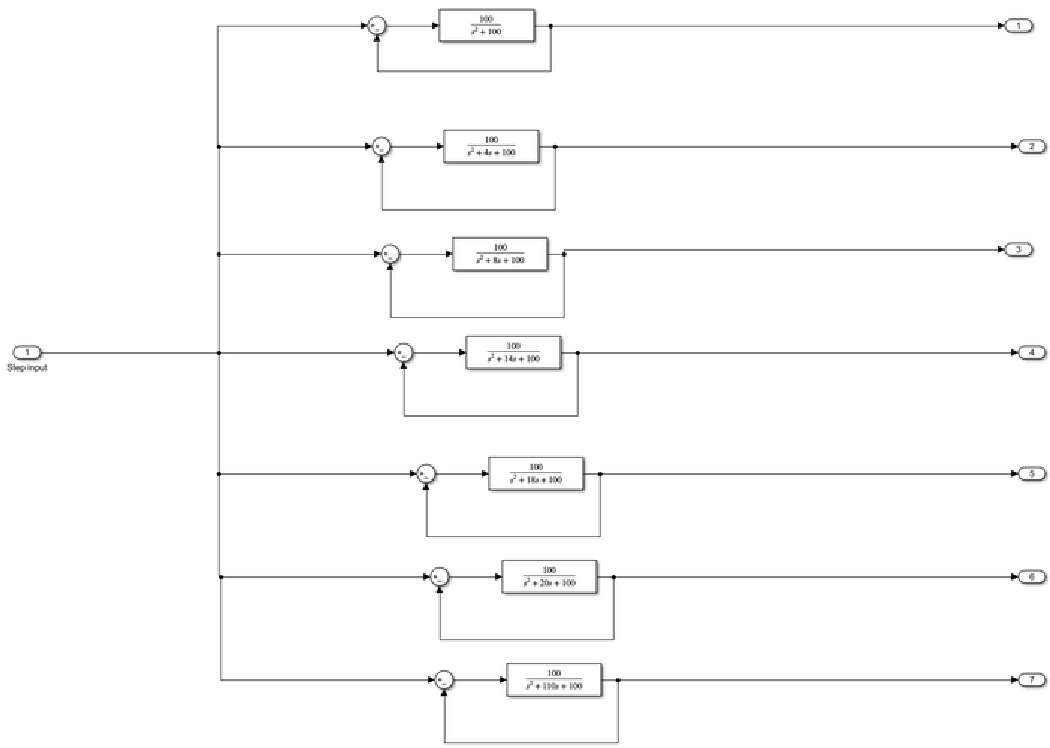
ω_n is the natural frequency of the system

- When ζ = 0, the system is Undamped.
- When ζ = 1, the system is Critically damped.
- When 0 < ζ < 1, the system is Under damped.
- When ζ > 1, the system is over damped.

Simulations-



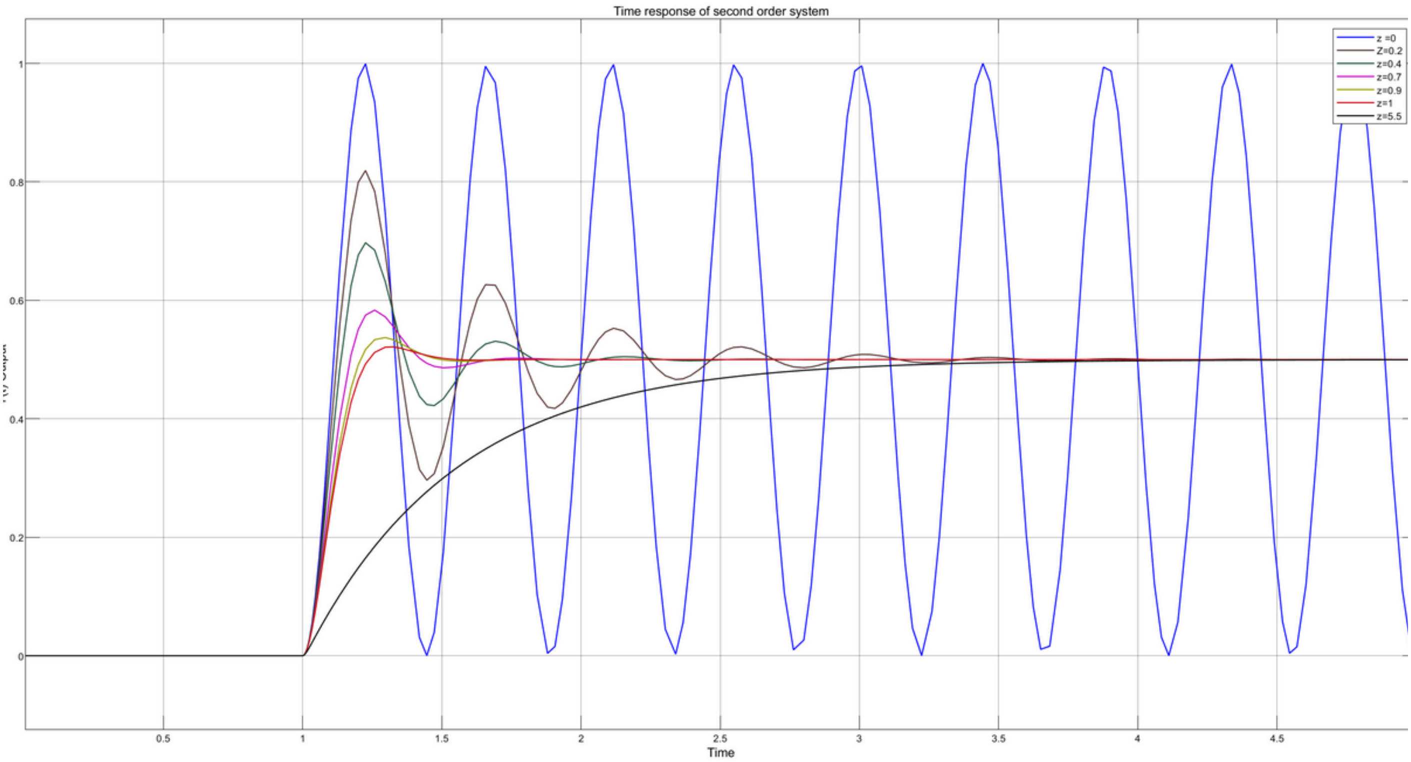
Inside subsystem-



Here for a second order system , UNIT STEP signal is given as input , from above equations we get

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

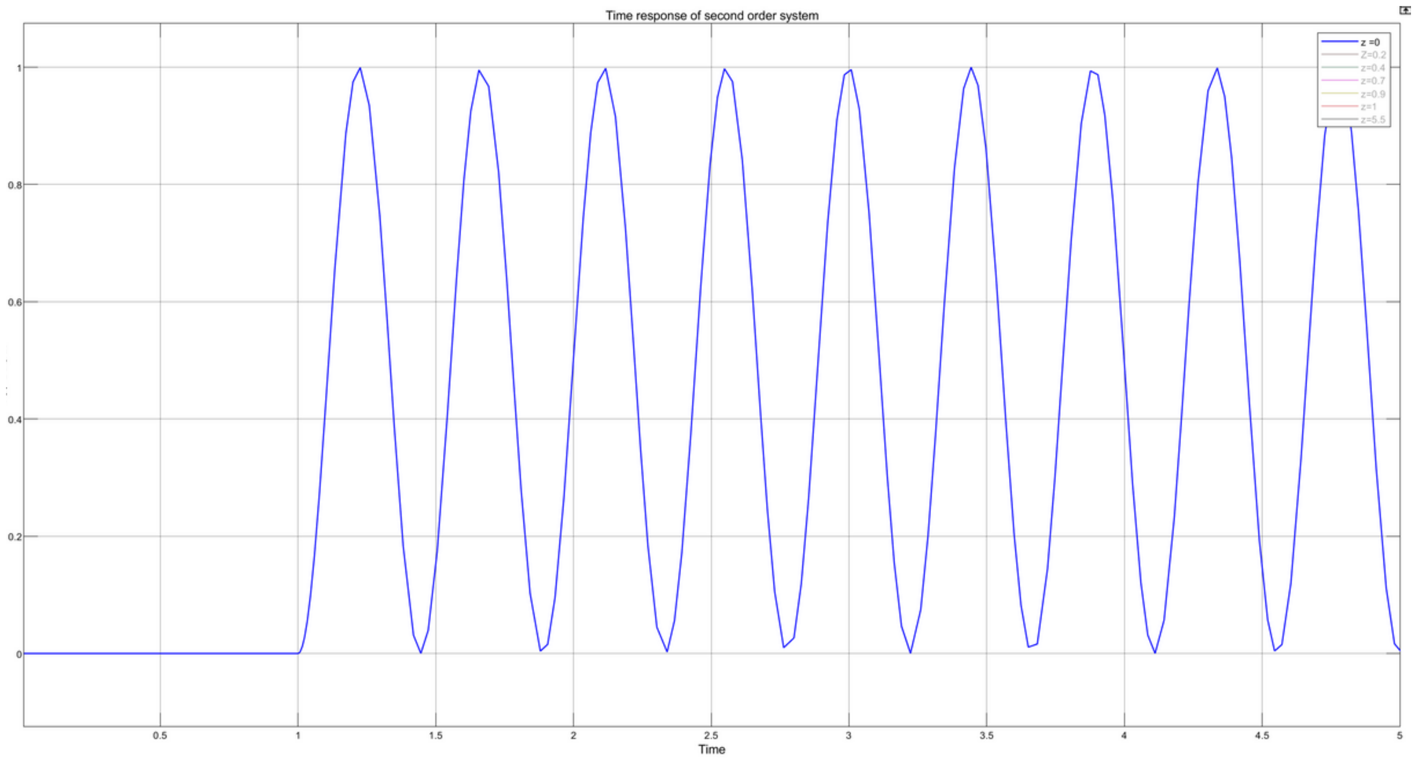
The response of this second order system for various values of the damping ratio ζ is shown in below scope .



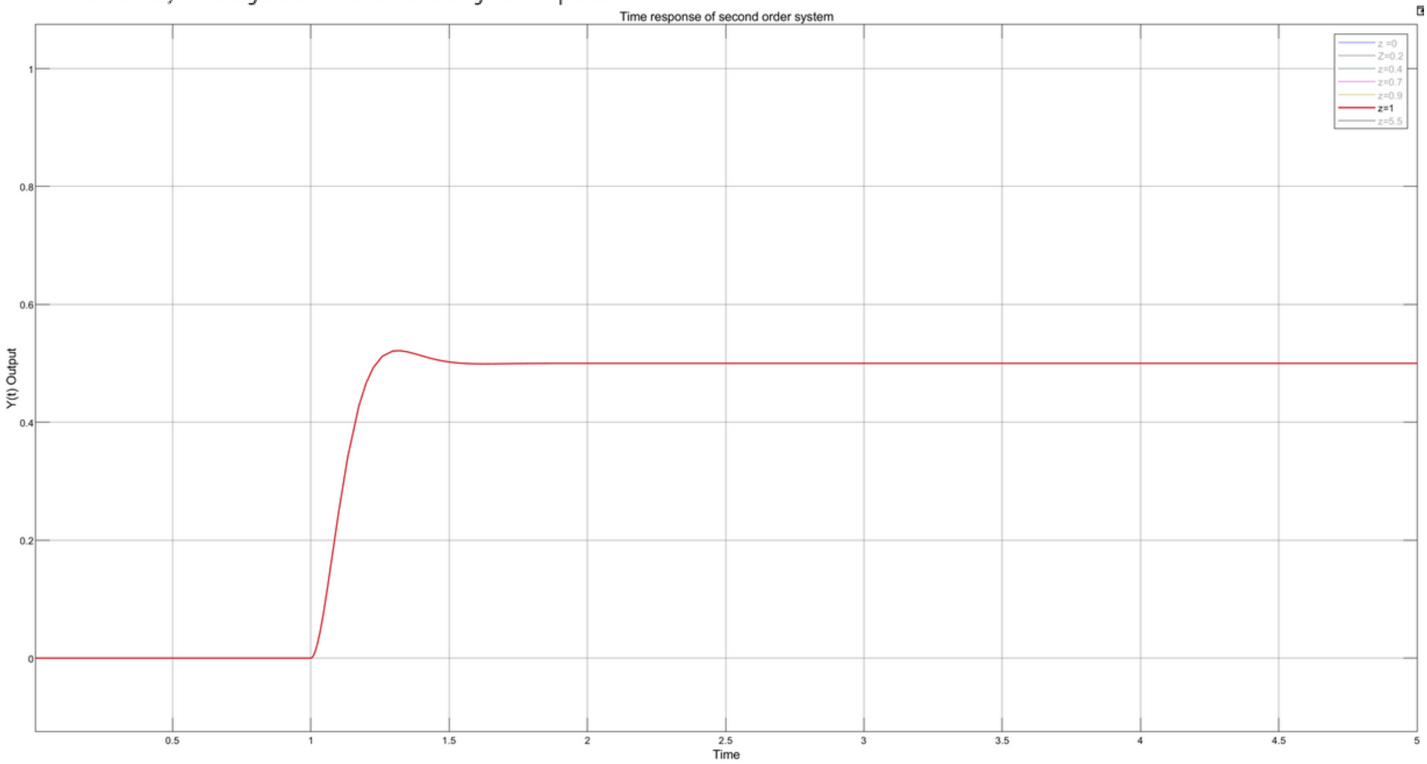
Obseravtion -

As the damping ratio decreases, the closed-loop poles approach the imaginary axis, and the response becomes increasingly oscillatory.

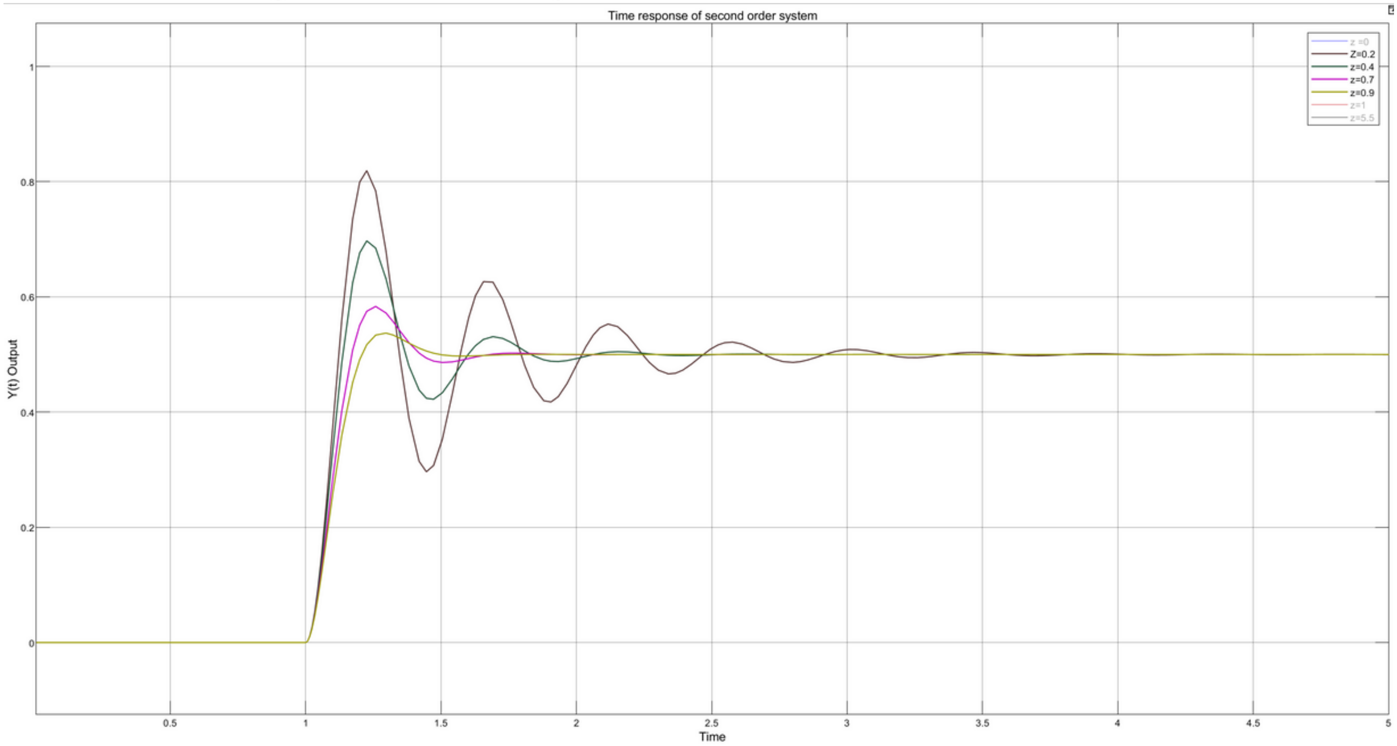
When ξ = 0, the system is Undamped



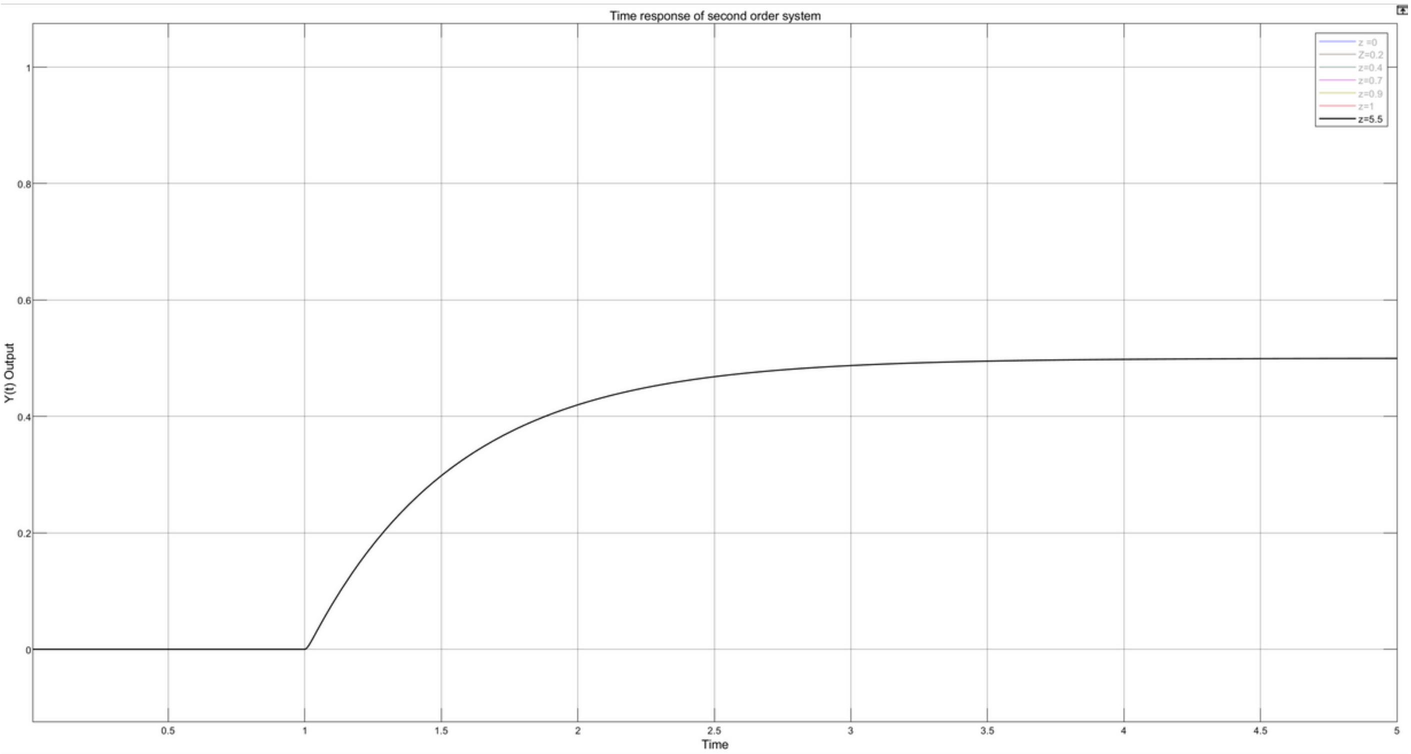
When ζ= 1, the system is Critically damped.



When 0 <ζ< 1, the system is Under damped.



When $\zeta > 1$, the system is over damped.



INFERENCE-

Four types of damping characteristics can be observed in a second-order system: undamped, underdamped, critically damped, and overdamped. When the damping ratio (ξ) is 0, the system is undamped. For ξ values between 0 and 1, the system is underdamped. At $\xi = 1$, the system is critically damped. When ξ is greater than 1, the system is overdamped. By varying the damping ratio, different system parameters change. Increasing ξ results in longer rise and peak times but reduced peak overshoot and steady-state errors. Higher ξ values lead to more accurate system output.

In addition to the damping characteristics, varying the damping ratio (ξ) also affects other parameters of the system's response. When ξ is increased, the rise time and peak time of the system response become longer, indicating a slower response to changes in input signals. This behavior is characteristic of increased damping in the system.

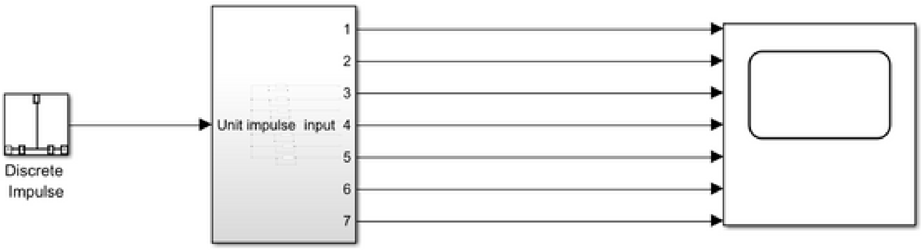
Moreover, as the damping ratio increases, the peak overshoot decreases. Peak overshoot refers to the maximum deviation of the response from its steady-state value. A higher damping ratio reduces oscillations in the system's response, resulting in a smoother transition to the desired output.

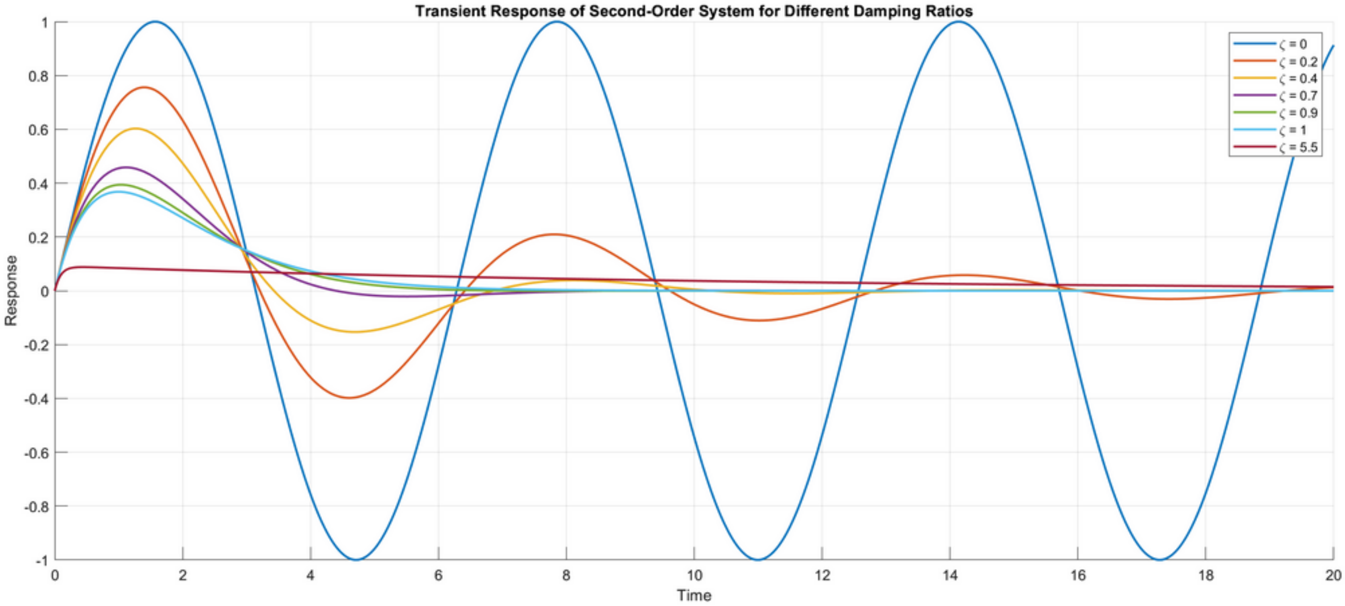
Furthermore, increasing the damping ratio leads to a decrease in steady-state errors. Steady-state error is the difference between the desired output and the actual output of the system once it has settled to a constant value. A higher damping ratio helps the system reach its steady-state more accurately and quickly.

Overall, adjusting the damping ratio allows for fine-tuning the performance of the system, balancing factors such as response speed, stability, and accuracy according to the specific requirements of the application.

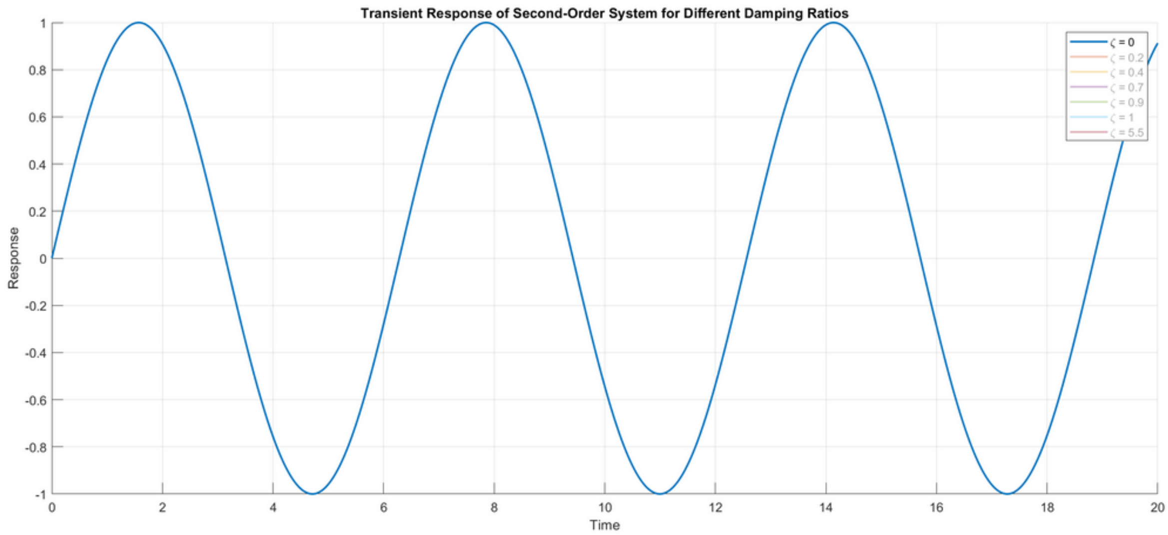
Similarly , when the input given is a unit impulse signal, then from above equation we get,

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

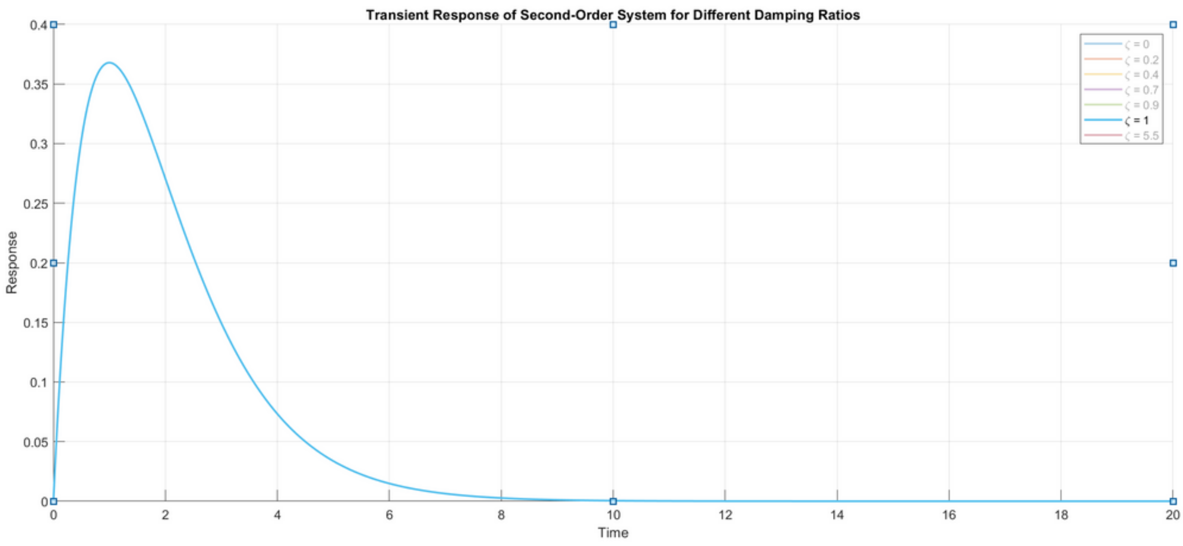




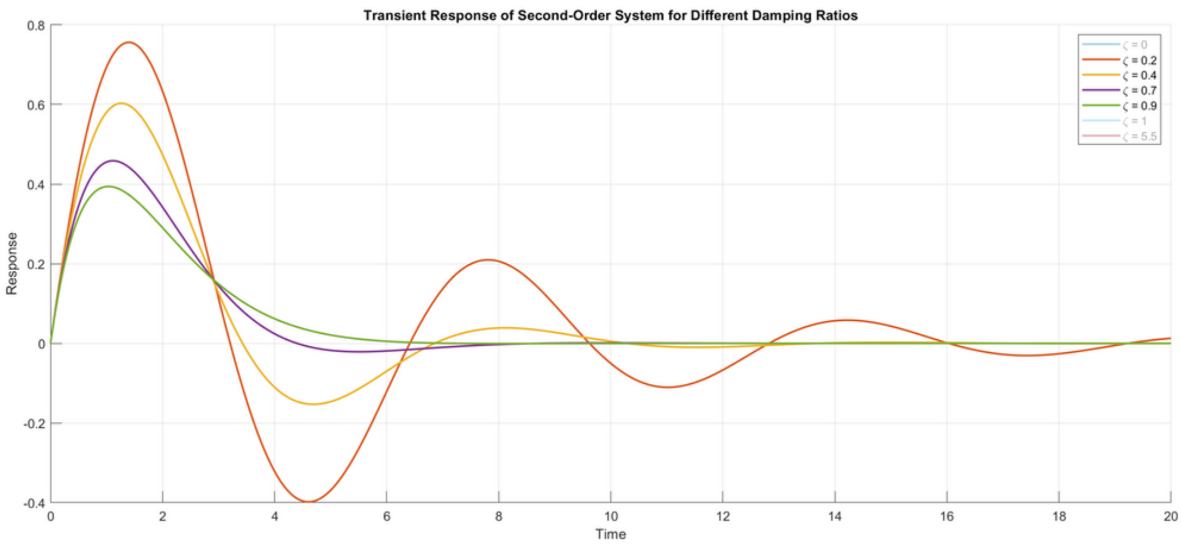
When ξ = 0, the system is Undamped



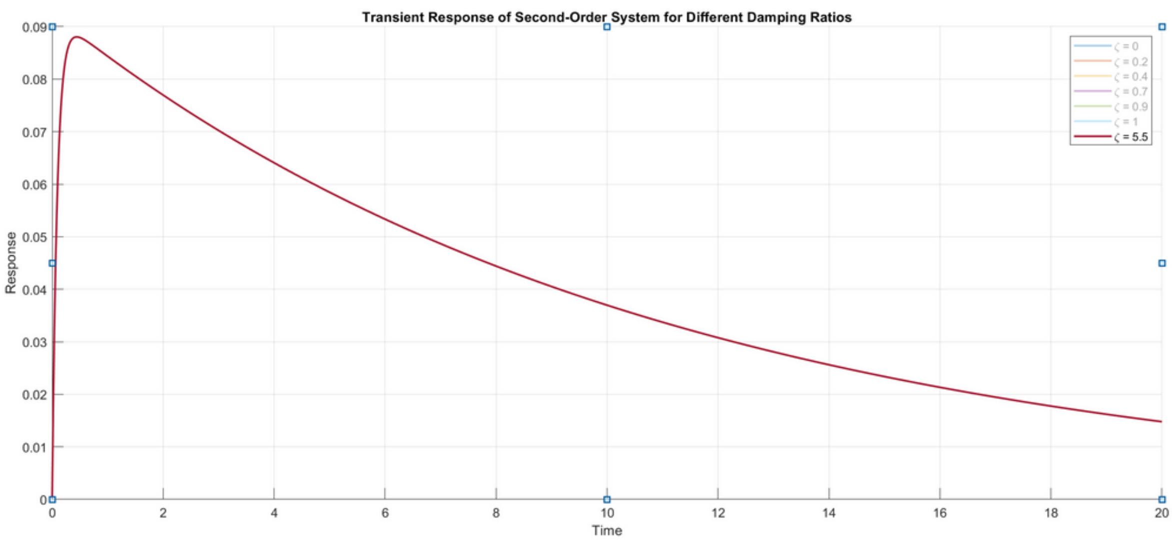
When $\zeta = 1$, the system is Critically damped.



When $0 < \zeta < 1$, the system is Under damped.



When $\zeta > 1$, the system is over damped.



MATLAB PROGRAMMING-

%% How to create a transfer function and cascade it;

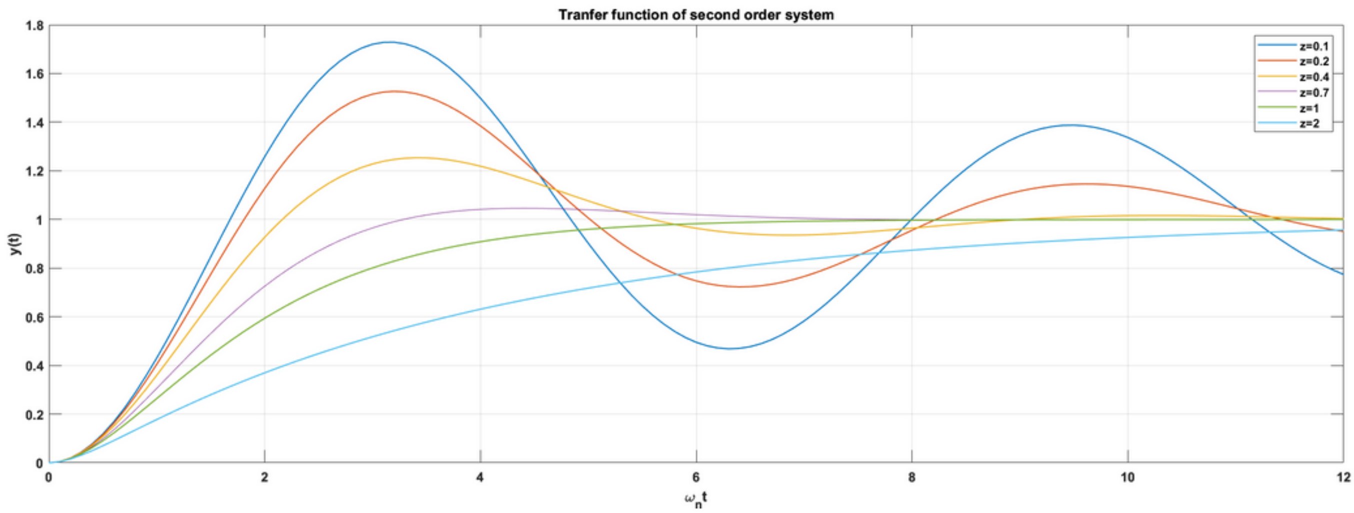
```
>> num1=10;
den1=[1 2 5];
sys1=tf(num1,den1)

sys1 =

          10
  -------------
  s^2 + 2 s + 5

Continuous-time transfer function.

>> num2=1;
den2=[1 1];
sys2=tf(num2,den2);
sys=series(sys1,sys2)

sys =

              10
  ---------------------
  s^3 + 3 s^2 + 7 s + 5

Continuous-time transfer function.
```

%% How to create a transfer function and cascade If feedback is given;

```
>> numg=1;
deng=[500 0 0];
sys2=tf(numg,deng)

sys2 =

     1
  -------
  500 s^2

Continuous-time transfer function.

>> numc=[1 1];
denc=[1 2];
sys1=tf(numc,denc)

sys1 =

  s + 1
  -----
  s + 2

Continuous-time transfer function.

>> sys3=series(sys1,sys2)

sys3 =

        s + 1
  ------------------
  500 s^3 + 1000 s^2

Continuous-time transfer function.

sys =

            s + 1
  --------------------------
  500 s^3 + 1000 s^2 + s + 1

Continuous-time transfer function.
```

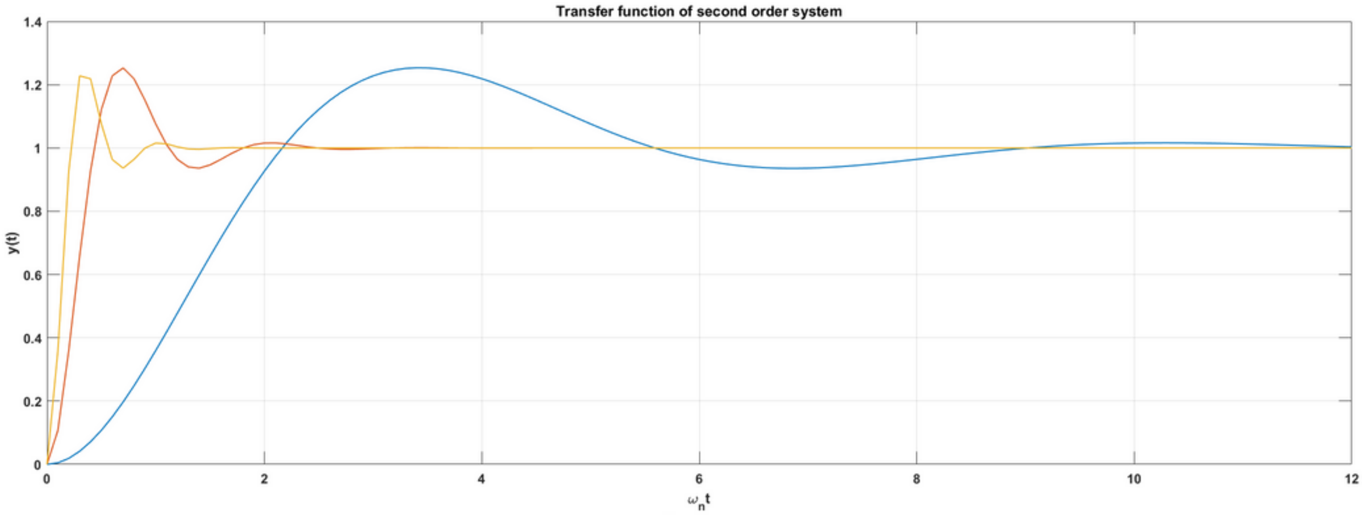
%% Verifying the types of damping conditions in the second order system varying zeta;

```
>> t=0:00.1:12;
num=1;
zeta1=0.1;
den=[1 2*zeta1 1];
sys1=tf(num,den);
zeta2=0.2;
den=[1 2*zeta2 1];
sys2=tf(num,den);
zeta3=0.4;
den=[1 2*zeta3 1];
sys3=tf(num,den);
zeta4=0.7;
den=[1 2*zeta4 1];
sys4=tf(num,den);
zeta5=1;
den=[1 2*zeta5 1];
sys5=tf(num,den);
zeta6=2;
den=[1 2*zeta6 1];
sys6=tf(num,den);
[y1,T1]=step(sys1,t);
[y2,T2]=step(sys2,t);
[y3,T3]=step(sys3,t);
[y4,T4]=step(sys4,t);
[y5,T5]=step(sys5,t);
[y6,T6]=step(sys6,t);
plot(T1,y1,T2,y2,T3,y3,T4,y4,T5,y5,T6,y6)
xlabel('\omega_nt')
ylabel('y(t)')
title('Tranfer function of second order system'),grid
```



%% verifying the types of damping condition in the second order system by varying Omega values;

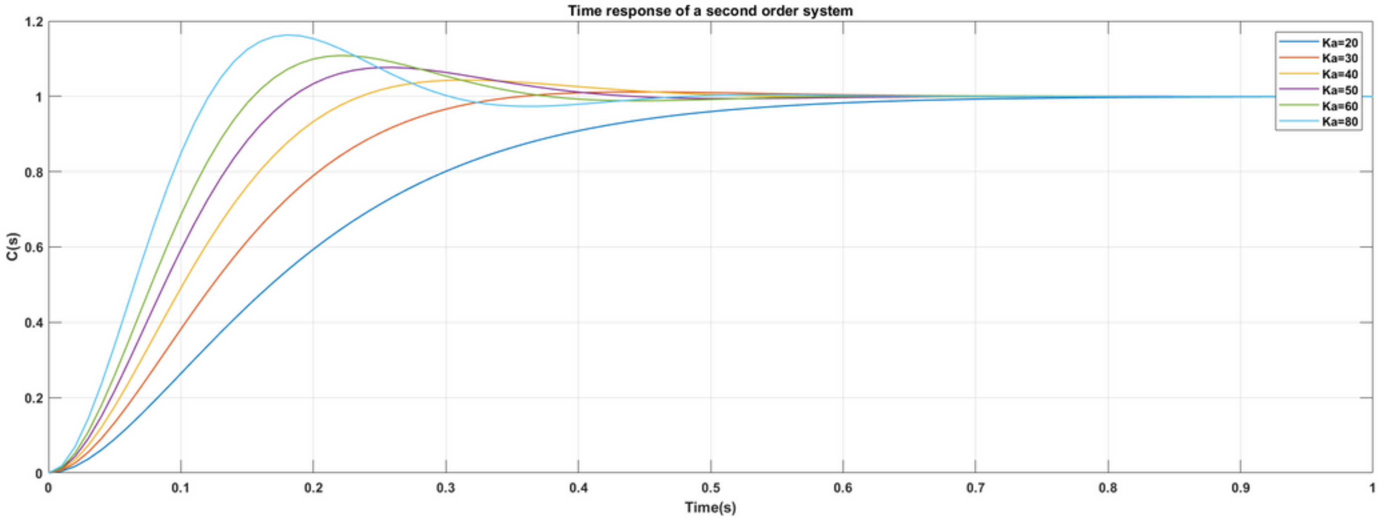
```
>> t=[0:0.1:12];
zeta=0.4;
w1=1;
num1=[w1^2];den1=[1 2*w1*zeta w1^2];sys1=tf(num1,den1);
w2=5;
num2=[w2^2];den2=[1 2*w2*zeta w2^2];sys2=tf(num2,den2);
w3=10;
num3=[w3^2];den3=[1 2*w3*zeta w3^2];sys3=tf(num3,den3);
[y1,T1]=step(sys1,t);
[y2,T2]=step(sys2,t);
[y3,T3]=step(sys3,t);
plot(T1,y1,T2,y2,T3,y3)
xlabel('\omega_nt')
ylabel('y(t)')
title('Transfer function of second order system'),grid
```



When keeping zeta constant and varying the omega values, it's observed that increasing omega leads to a decrease in settling time. Moreover, higher omega values result in improved working performance, enhancing the accuracy of the system's response.

```
>> Ka = 20,30,40,50,60,80
t=[0:0.01:1];
km=5;
ka1=20*km;
ka2=30*km;
ka3=40*km;
ka4=50*km;
ka5=60*km;
ka6=80*km
num=[1];
den=[1 20 0];
sysa=tf(num,den);
sys1=series(sysa,ka1);
sys2=series(sysa,ka2);
sys3=series(sysa,ka3);
sys4=series(sysa,ka4);
sys5=series(sysa,ka5);
sys6=series(sysa,ka6);
f1=feedback(sys1,[1]);
f2=feedback(sys2,[1]);
f3=feedback(sys3,[1]);
f4=feedback(sys4,[1]);
f5=feedback(sys5,[1]);
f6=feedback(sys6,[1]);
[y1,t]=step(f1,t);
[y2,t]=step(f2,t);
[y3,t]=step(f3,t);
[y4,t]=step(f4,t);
[y5,t]=step(f5,t);
[y6,t]=step(f6,t);
plot(t,y1,t,y2,t,y3,t,y4,t,y5,t,y6);
grid;
xlabel('Time(s)');
ylabel('C(s)');
title('Time response of a second order system')
```

```
Ka =
    20
ans =
    30
ans =
    40
ans =
    50
ans =
    60
ans =
    80
ka6 =
   400
```



VALUES OF Ka	tr(Rise time)	tp(Peak time)	%Mp(Peak overshoot)	ts(Settling time)
20	0.3359	1.1900	0	0.5835
30	0.2067	0.4421	1.1758	0.3176
40	0.1519	0.3132	4.3210	0.4216
50	0.1225	0.2579	7.6893	0.3790
60	0.1040	0.2210	10.8433	0.3404

INFERENCE

From the plot, it's evident that increasing the Ka value transitions the system from an undamped state to a critically damped state, accompanied by a reduction in settling time. Moreover, increasing the controller constant enhances the system's working condition.

RESULTS

Using Matlab text code, an analysis of the time response characteristics and damping conditions of a second-order system was conducted, resulting in the generation of a graph.

REFERENCES-

Richard C. Dorf and Robert H. Bishop, "Modern Control Systems", Pearson, 2011.

V V N S C MADHAVA

CB.EN.U4EEE22051