INDIAN INSTITUTE OF INFORMATION TECHNOLOGY NAGPUR

Control Systems

========= Project Report =========

“Implementation of Control System Experiments in MATLAB”

=================================

-: Implemented By :-
Ambarish Prashant Chandurkar
(BT17ECE021)

-: Under the Guidance of :-
Dr. Chandrashekhar Sakode

Table of Contents

Date : 5 Apr 2019

Lab 1 : Implementation of Time Response in MATLAB

Aim : To implement the time response of the given transfer function of the system in MATLAB

Apparatus : MATLAB Software.

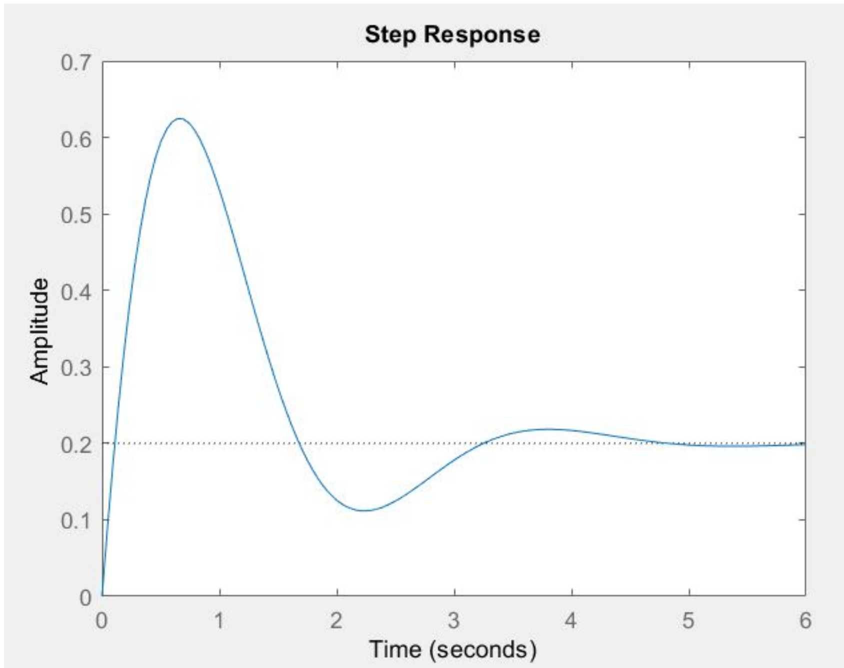
MATLAB Code :

```
%% Experiment : Plot the time response of a system

clc;clearvars;close all;

%   2s + 1
% ------------   =>    This is the first System
% s^2 + 2s + 5

num1 = [2 1]; %The numerator
den1 = [1 2 5]; %The denominator
sys1 = tf(num1,den1); %Create the system
figure(1);
step(num1,den1) %Plot the Step response.Save it,Print it and affix in the
                %journal along with code


%    s
% -------   =>    This is the second System
%  s + 3

num2 = [1 0]; %The numerator
den2 = [1 3]; %The denominator
sys2 = tf(num2,den2); %Create the system
figure(2);
step(num2,den2) %Plot the Step response.Save it,Print it and affix in the
                %journal along with code

%      s + 1
% ----------------   =>    This is the third System, in pole zero form
%   (s + 3)(s + 2)

G = zpk([-1],[-2 -3],1); %z = zeros, p = poles, k = gain
figure(3);
step(G) %Plot the Step response.Save it,Print it and affix in the
        %journal along with code

```

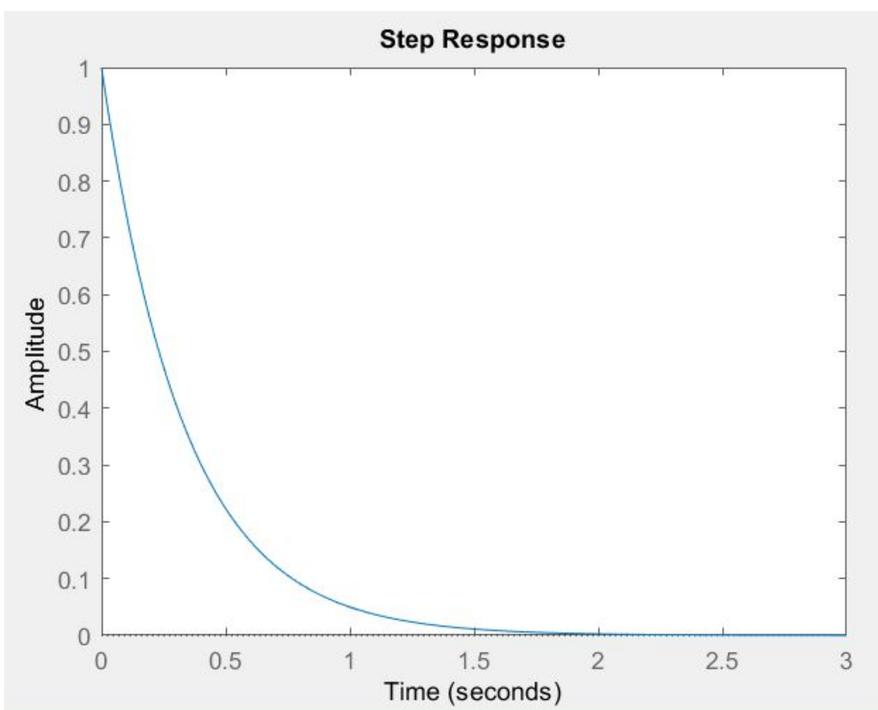
```
%       s + 1
% ----------------    =>    This is the third System, in coefficient form
%   (s + 3)(s + 2)

num3 = [1 1]; %The numerator
den3 = [1 5 6]; %The denominator
sys3 = tf(num3,den3); %Create the system
figure(4);
step(num3,den3) %Plot the Step response.Save it,Print it and affix in the
                %journal along with code


%        s + 1
%  ---------------  =>    This is the fourth System, as a closed loop
%  s^2 + 11 s + 13

%                     |     s + 1      |
%R(s) -------------->|----------------|----------------> C(s)
%            |        | s^2 + 11 s + 13|        |
%            |                                  |
%            |<-------------|8|<----------------|


num4 = [1 1]; %The numerator
den4 = [1 3 5]; %The denominator
Gofs = tf(num4,den4); %Create the Open loop Transfer func G(s)
Hofs = 8; % Create the Feedback Function H(s)
CLF1 = feedback(Gofs,Hofs); %This is the closed loop System
figure(5);
step(CLF1) %Plot the Step response.Save it,Print it and affix in the
           %journal along with code


%             (s+1)  (s+2)  (s+3)
%---------------------------------------------  => This is the sixth system
%(s^2 + 4.407s + 5.171) (s^2 + 3.593s + 5.995)

num5 = [1 1]; %The numerator
den5 = [1 3 5]; %The denominator
Gofs = tf(num5,den5); %Create the Open loop Transfer func G(s)
Hofs = zpk([-1],[-2 -3],1); %Create the feedback function
CLF2 = feedback(Gofs,Hofs); %This is the closed loop System
figure(6);
step(CLF2) %Plot the Step response.Save it,Print it and affix in the
           %journal along with code

```

Step Response Outputs :

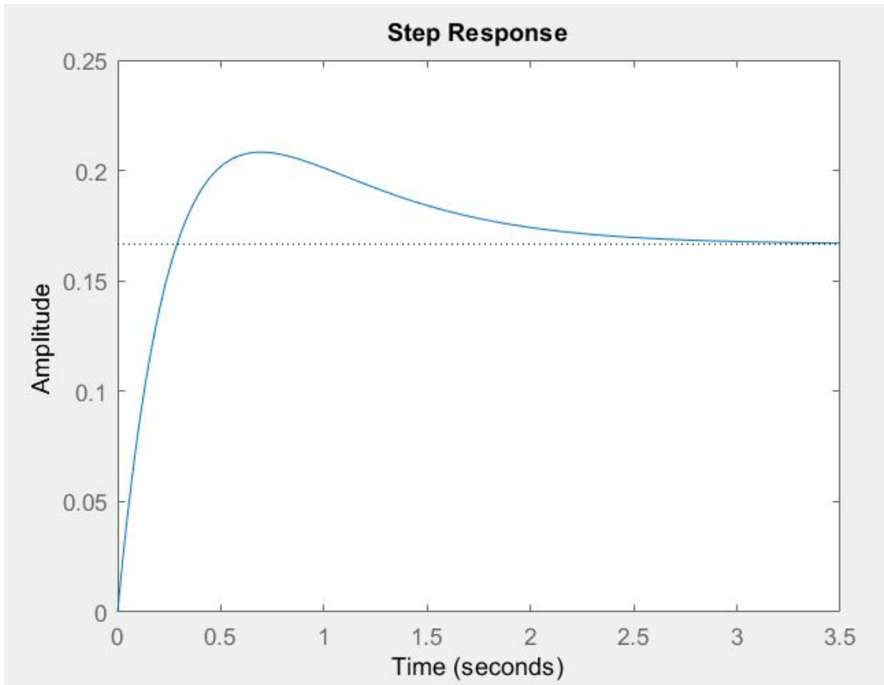
System 1



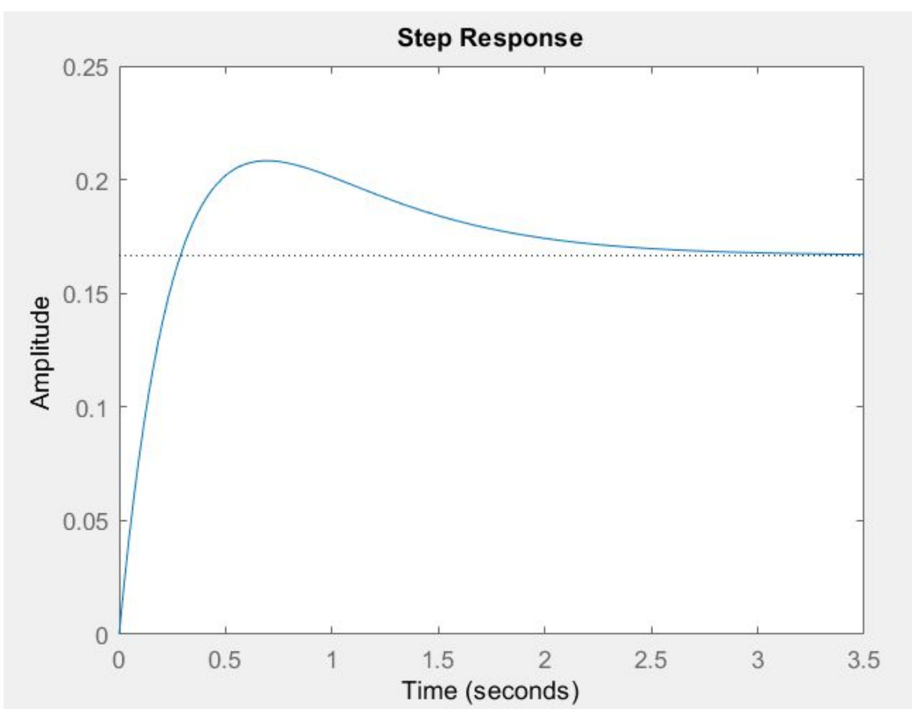
System 2



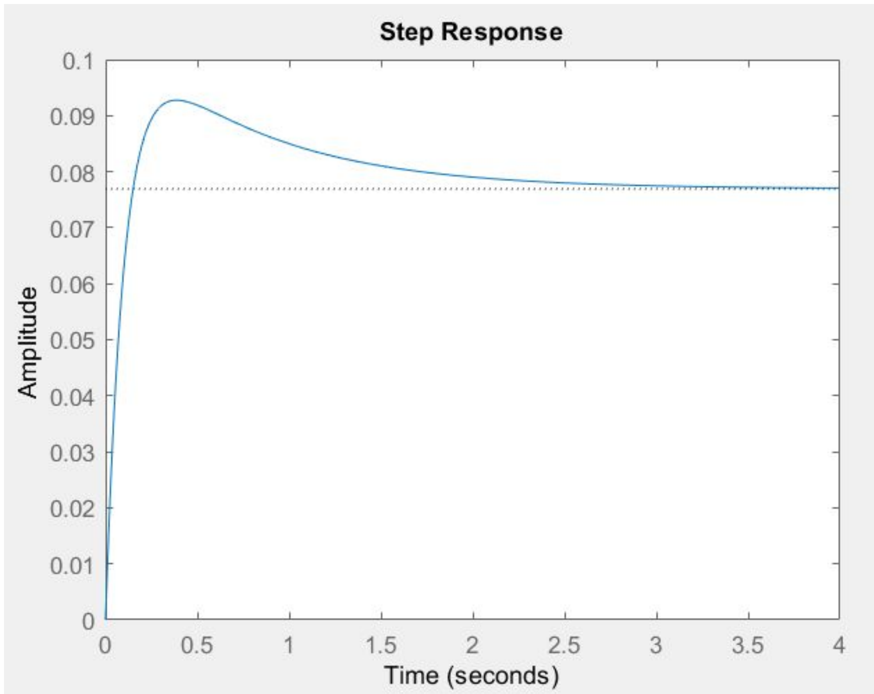
System 3



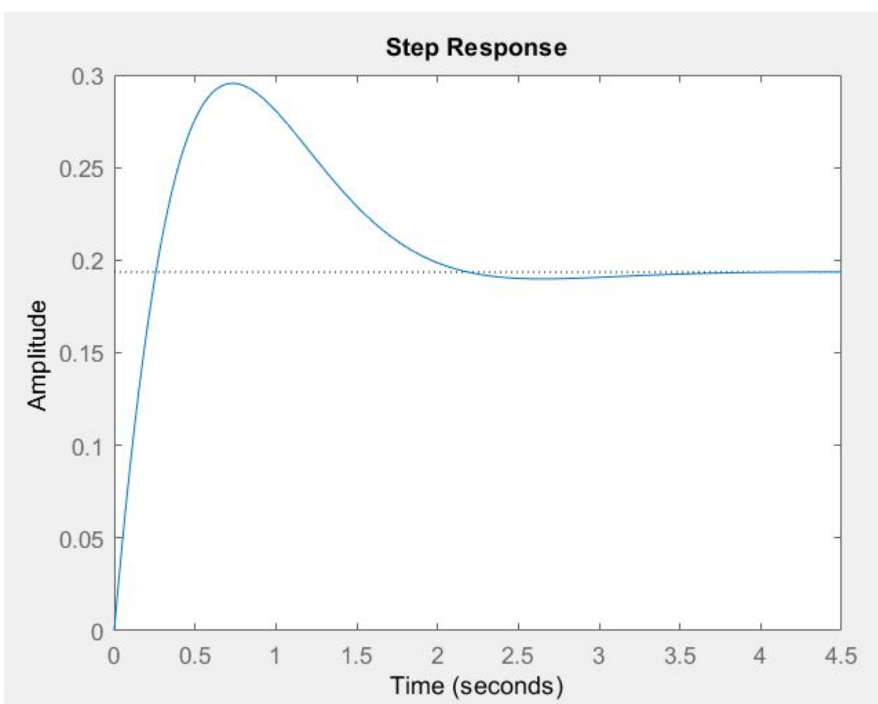
System 4



System 5



System 6



Conclusion :

In this experiment, Control System Design and Simulation was thoroughly understood , through the aid of MATLAB.

(GO TO NEXT PAGE)

Date : 5 Apr 2019

Lab 2 : Study of Root Locus Plot using MATLAB

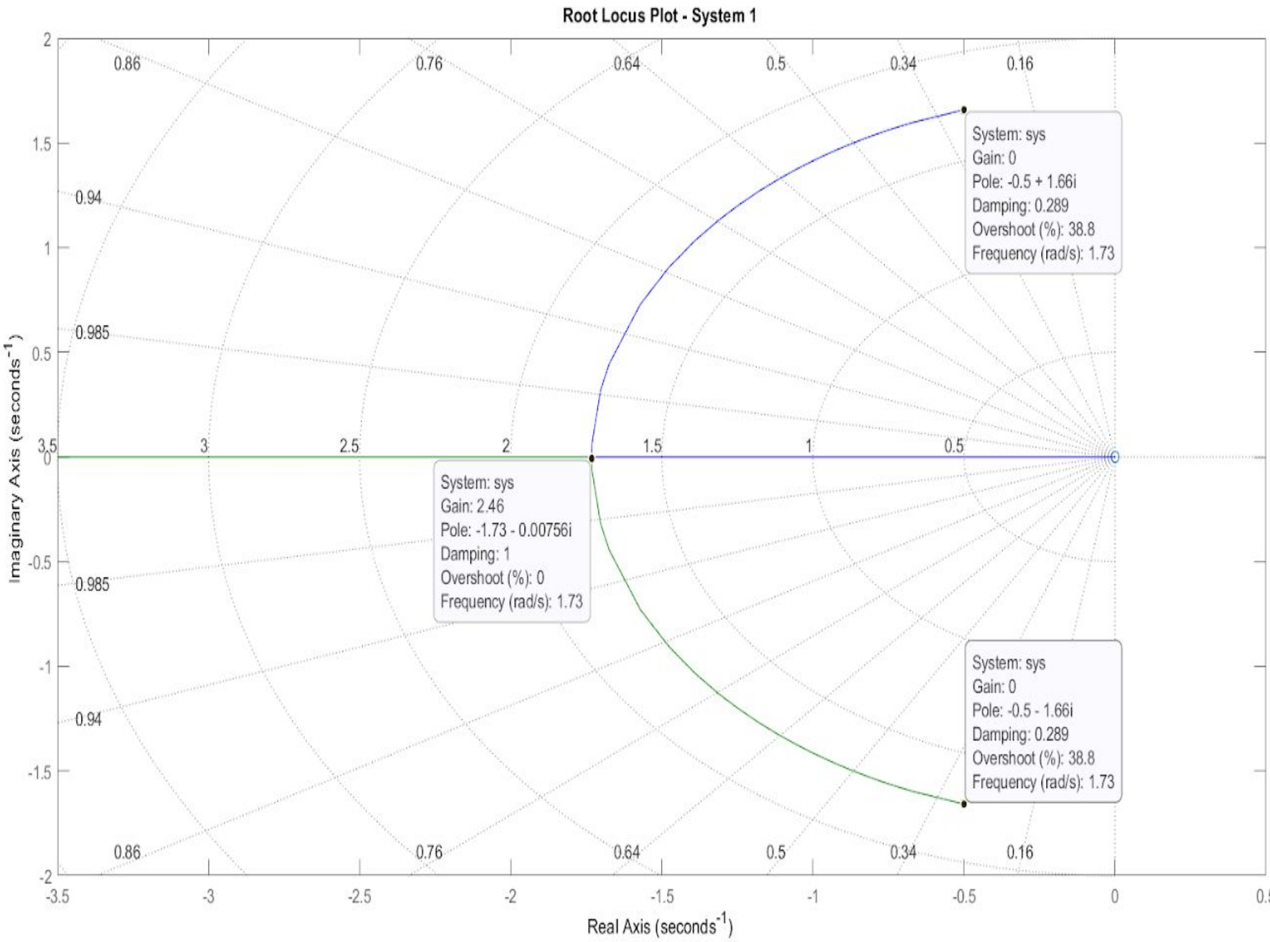
Aim : To plot the Root Locus of the given system function; using MATLAB

Apparatus : MATLAB Software.

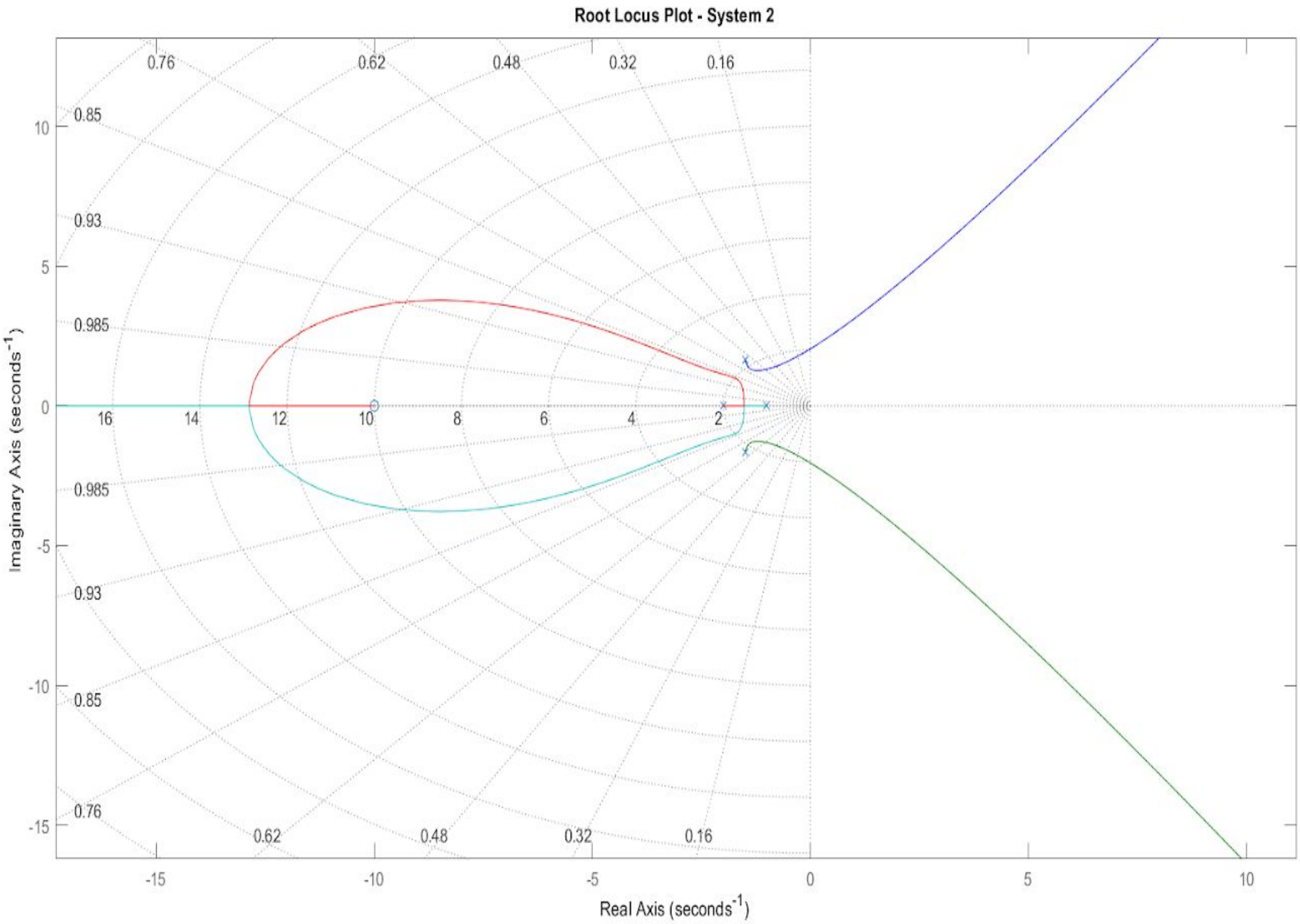
MATLAB Code :

```
%% Experiment : Plot the root locus
%Solve theoretically and then cross check with plot
clc;close all;clearvars;
%% Example 1:
%        s
%   -----------  <== System 1
%   s^2 + s + 3

num1 = [0 1 0]; %Numerator Coeffcients
den1 = [1 1 3]; %Denominator Coefficients
sys1 = tf(num1,den1) %Creation of System
figure(1);
rlocus(num1,den1); %Plotting of Root Locus Plot
title('Root Locus Plot - System 1');
%% Example 2:
%                s + 10
%   --------------------------------  <== System 2
%   s^4 + 6 s^3 + 16 s^2 + 21 s + 10

num2 = [0 1 10]; %Numerator Coeffcients
p1 = [0 1 1]; %Denominator Factors p1 p2 p3
p2 = [0 1 2];
p3 = [1 3 5];
p4 = conv(p1,p2);
p5 = conv(p4,p3); %Denominator Coefficients
sys2 = tf(num2,p5) %Creation of System
figure(2)
rlocus(num2,p5); %Plotting of Root Locus Plot
title('Root Locus Plot - System 2');
```

Root Locus Output Plots :



(GO TO NEXT PAGE)



Conclusion :

The root locus plot of the given systems have been plotted theoretically and verified against the MATLAB Outputs

Date : 18 Apr 2019

Lab 3 : Plotting Bode Plots through MATLAB

Aim : To plot the Bode Plots of the given System transfer functions using MATLAB

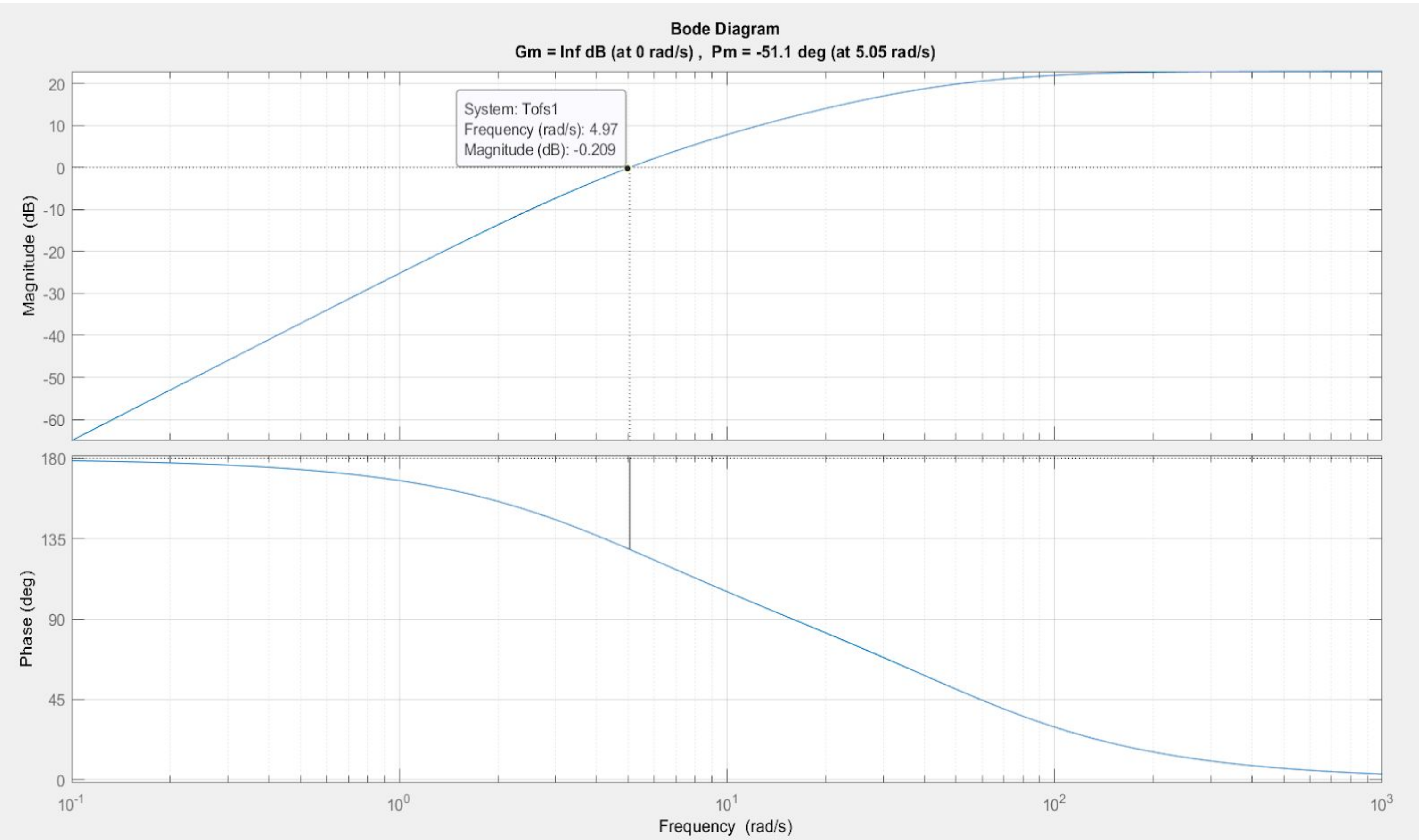
Apparatus : MATLAB Software.

MATLAB Code :

```
%% Experiment : To plot the Bode plots of given Systems
clc;close all;clearvars;
%% Q7 : Find the Gain 'K' for which Gain Cross over Frequncy is '5 rad/sec'
% Ans : Gain K = 0.056, as seen from Bode plot (Solution in Assignement)
%       0.056 s^2
%   ----------------------  <== System 1
%   0.004 s^2 + 0.22 s + 1

Num1 = [0.056 0 0]; %Numerator Coeffcients
Den1 = [0.2 1]; %Denominator factor 1 Coefficients
Den2 = [0.02 1]; %Denominator factor 2 Coefficients
Tofs1 = tf(Num1,conv(Den1,Den2)); %Creation of System
figure(1)
margin(Tofs1) %Plots the Bode Plot and also gives the "Gain Crossover Frequency"

%% Q8 : Find the Phase Crossover Frequency
% Ans = 9.976 (Theoretically), Verified from bode plot
%               1000
%   ------------------------------  <== System 2
%   s^3 + 24.95 s^2 + 99.53 s + 999

Num2 = 1000; %Numerator Coeffcients
Den3 = [1 22.5]; %Denominator factor 1 Coefficients
Den4 = [1 2.45 44.4]; %Denominator factor 2 Coefficients
Tofs2 = tf(Num2,conv(Den3,Den4)); %Creation of System
figure(2)
margin(Tofs2) %Plots the Bode Plot and also gives the "Phase Crossover Frequency"
```

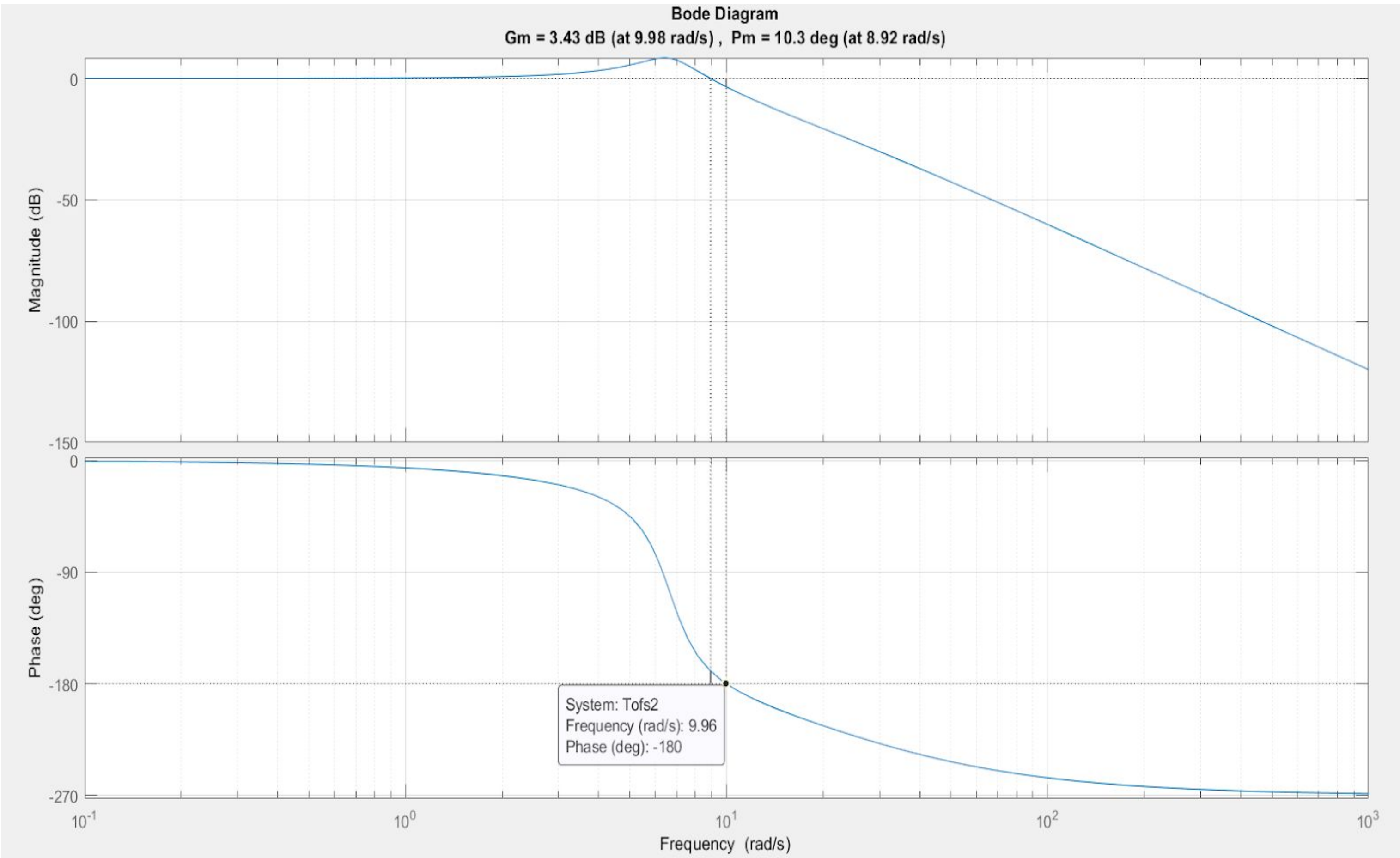
Bode Plots :

System Q.7



From the above Bode plot, we observe that for K = 0.056, Gain Crossover Frequency is wp = 5 rad/sec, as required.

System Q.8



From the above Bode Plot , it is clear that the Phase Crossover Frequency is 9.96 rad/sec.

Conclusion :

The Q7 and Q8 of Assignment 2 have been solved theoretically, and verified against the output of MATLAB

Date : 18 Apr 2019

Lab 4 : Effect of variation of Kp , Kd and Ki of PID controller on system Parameters

Aim : To obtain ζ , wn , tss and Mp of the system when the parameters Kp , Kd, Ki are varied for a PID Controller.

Apparatus : MATLAB Software.

MATLAB Code :

```
%% Q1 : Given parameters are Kp = Kd = Ki = 0
% T(s) = 0 ; System Function

clc;close all;clearvars;
Kp = 0; %Given parameters
Kd = 0;
Ki = 0;
PIDController = pid(Kp,Kd,Ki);
PIDController.u = 'E';  %Input of PID Block
PIDController.y = 'PIDOut'; %Output of PID Block
GofS = zpk([],[0,-10],8); %The Open loop Transfer function
GofS.u = 'PIDOut';  %Input of Gofs
GofS.y = 'Cofs'; %Output of Gofs
Sum = sumblk('E = Rofs - Cofs'); %Calculation of Error
Tofs = connect(GofS,PIDController,Sum,'Rofs','Cofs'); %Creation of Closed Loop system

step(Tofs);
SysCharac = stepinfo(Tofs);

%% Q2 Case I: Given parameters are Kp = 5 ; Kd = Ki = 0
%         40
%   ---------------- = T(s) ; System Function
%   (s^2 + 10s + 40)

clc;close all;clearvars;
Kp = 5; %Given parameters
Kd = 0;
Ki = 0;
PIDController = pid(Kp,Kd,Ki);
PIDController.u = 'E';  %Input of PID Block
PIDController.y = 'PIDOut'; %Output of PID Block
GofS = zpk([],[0,-10],8); %The Open loop Transfer function
GofS.u = 'PIDOut';  %Input of Gofs
GofS.y = 'Cofs'; %Output of Gofs
Sum = sumblk('E = Rofs - Cofs'); %Calculation of Error
Tofs = connect(GofS,PIDController,Sum,'Rofs','Cofs'); %Creation of Closed Loop system

step(Tofs);
SysCharac = stepinfo(Tofs);
```

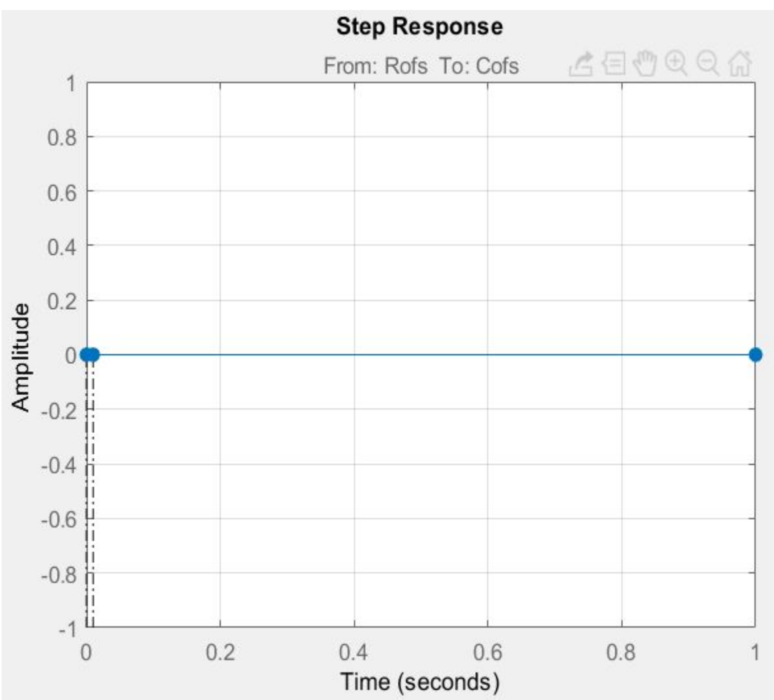
```
%% Q2 Case II : Given parameters are Kp = 10 ; Kd = Ki = 0
%        80
%  ---------------- = T(s) ; System Function
%   (s^2 + 10s + 80)

clc;close all;clearvars;
Kp = 10; %Given parameters

Kd = 0;
Ki = 0;
PIDController = pid(Kp,Kd,Ki);
PIDController.u = 'E';  %Input of PID Block
PIDController.y = 'PIDOut'; %Output of PID Block
GofS = zpk([],[0,-10],8); %The Open loop Transfer function
GofS.u = 'PIDOut';  %Input of Gofs
GofS.y = 'Cofs'; %Output of Gofs
Sum = sumblk('E = Rofs - Cofs'); %Calculation of Error
Tofs = connect(GofS,PIDController,Sum,'Rofs','Cofs'); %Creation of Closed Loop system

step(Tofs);
SysCharac = stepinfo(Tofs);

%% Q3 Case I : Given parameters are Kd = 5; Kp = Ki = 0
%                  40
%  --------------------------------- = T(s) ; System Function
%   (s+10.37) (s^2 - 0.3718s + 3.857)

clc;close all;clearvars;
Kp = 0; %Given parameters
Kd = 5;
Ki = 0;
PIDController = pid(Kp,Kd,Ki);
PIDController.u = 'E';  %Input of PID Block
PIDController.y = 'PIDOut'; %Output of PID Block
GofS = zpk([],[0,-10],8); %The Open loop Transfer function
GofS.u = 'PIDOut';  %Input of Gofs
GofS.y = 'Cofs'; %Output of Gofs
Sum = sumblk('E = Rofs - Cofs'); %Calculation of Error
Tofs = connect(GofS,PIDController,Sum,'Rofs','Cofs'); %Creation of Closed Loop system

step(Tofs);
SysCharac = stepinfo(Tofs);
```

```
%% Q3 Case II : Given parameters are Kd = 10; Kp = Ki = 0
%                  80
%  -------------------------------- = T(s) ; System Function
%   (s+10.7) (s^2 - 0.6989s + 7.477)
clc;close all;clearvars;
Kp = 0; %Given parameters
Kd = 10;
Ki = 0;
PIDController = pid(Kp,Kd,Ki);
PIDController.u = 'E';  %Input of PID Block
PIDController.y = 'PIDOut'; %Output of PID Block
GofS = zpk([],[0,-10],8); %The Open loop Transfer function
GofS.u = 'PIDOut';  %Input of Gofs
GofS.y = 'Cofs'; %Output of Gofs
Sum = sumblk('E = Rofs - Cofs'); %Calculation of Error
Tofs = connect(GofS,PIDController,Sum,'Rofs','Cofs'); %Creation of Closed Loop system

step(Tofs);
SysCharac = stepinfo(Tofs);

%% Q4 Case I : Given parameters are Ki = 5; Kp = Kd = 0
%    40 s
%  -------- = T(s) ; System Function
%   s (s+50)

clc;close all;clearvars;
Kp = 0; %Given parameters
Kd = 0;
Ki = 5;
PIDController = pid(Kp,Kd,Ki);
PIDController.u = 'E';  %Input of PID Block
PIDController.y = 'PIDOut'; %Output of PID Block
GofS = zpk([],[0,-10],8); %The Open loop Transfer function
GofS.u = 'PIDOut';  %Input of Gofs
GofS.y = 'Cofs'; %Output of Gofs
Sum = sumblk('E = Rofs - Cofs'); %Calculation of Error
Tofs = connect(GofS,PIDController,Sum,'Rofs','Cofs'); %Creation of Closed Loop system

step(Tofs);
SysCharac = stepinfo(Tofs);
```

```
%% Q4 Case II : Given parameters are Ki = 10; Kp = Kd = 0
%    80 s
%  -------- = T(s) ; System Function
%  s (s+90)

clc;close all;clearvars;
Kp = 0; %Given parameters
Kd = 0;
Ki = 10;
PIDController = pid(Kp,Kd,Ki);
PIDController.u = 'E';  %Input of PID Block
PIDController.y = 'PIDOut'; %Output of PID Block
GofS = zpk([],[0,-10],8); %The Open loop Transfer function
GofS.u = 'PIDOut';  %Input of Gofs
GofS.y = 'Cofs'; %Output of Gofs
Sum = sumblk('E = Rofs - Cofs'); %Calculation of Error
Tofs = connect(GofS,PIDController,Sum,'Rofs','Cofs'); %Creation of Closed Loop
system

step(Tofs);
SysCharac = stepinfo(Tofs);
```

Step Response and System Parameters :

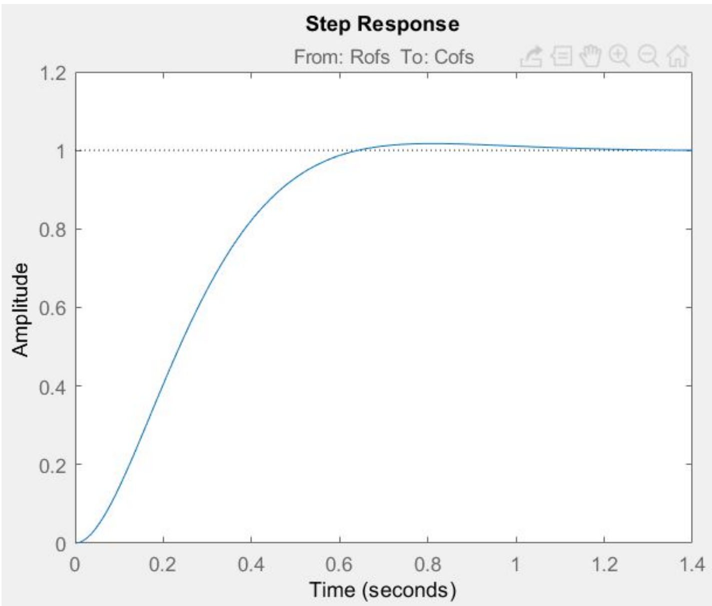
System Q.1



Command Window

```
>> SysCharac

SysCharac =

  struct with fields:

        RiseTime: 0
    SettlingTime: 0
     SettlingMin: 0
     SettlingMax: 0
       Overshoot: Inf
      Undershoot: 0
            Peak: 0
        PeakTime: 0
```

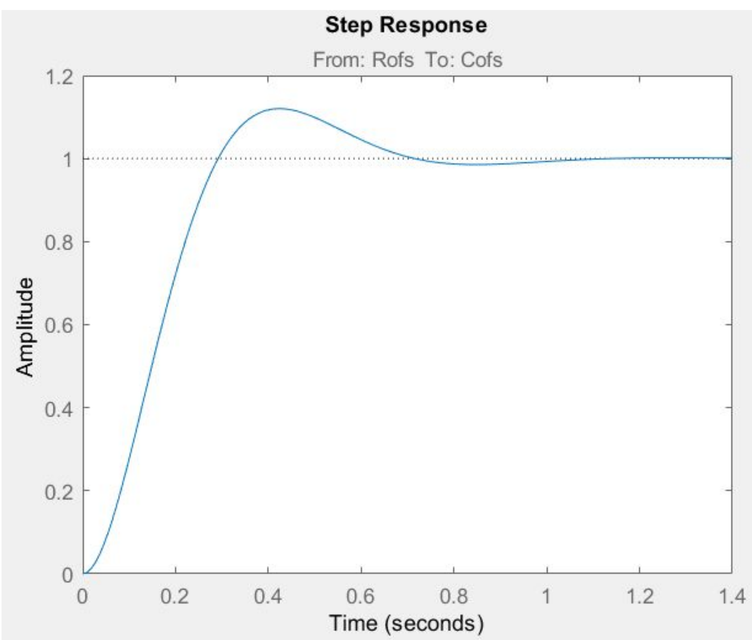
System Q.2 - Case I



Command Window

```
>> SysCharac

SysCharac =

  struct with fields:

        RiseTime: 0.3847
    SettlingTime: 0.5824
     SettlingMin: 0.9040
     SettlingMax: 1.0173
       Overshoot: 1.7322
      Undershoot: 0
            Peak: 1.0173
        PeakTime: 0.8105
```

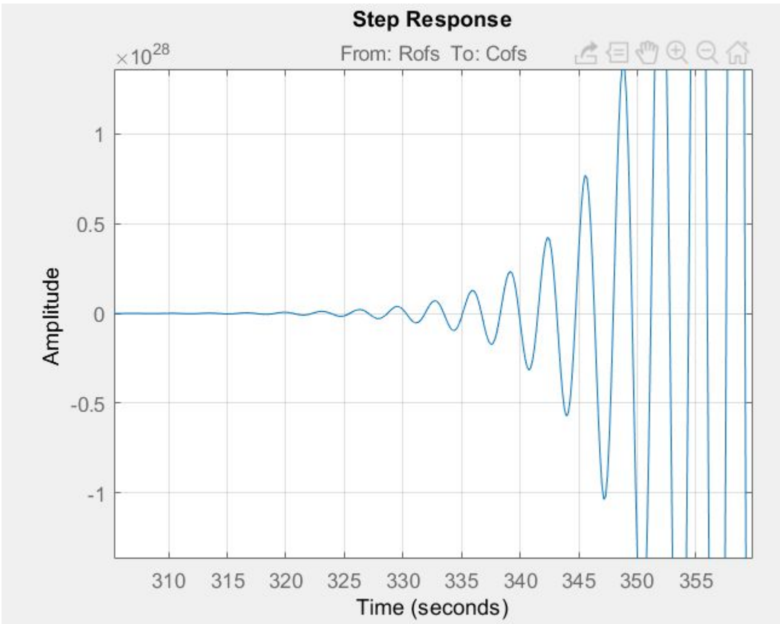
System Q.2 - Case II



Command Window

```
>> SysCharac

SysCharac =

  struct with fields:

        RiseTime: 0.1968
    SettlingTime: 0.6544
     SettlingMin: 0.9174
     SettlingMax: 1.1203
       Overshoot: 12.0265
      Undershoot: 0
            Peak: 1.1203
        PeakTime: 0.4237
```

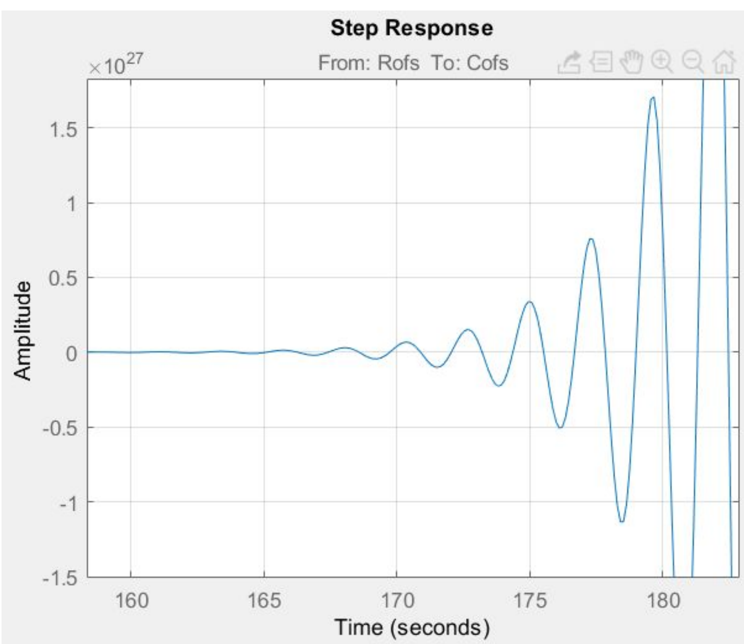
System Q.3 - Case I



Command Window

```
>> SysCharac

SysCharac =

  struct with fields:

        RiseTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```

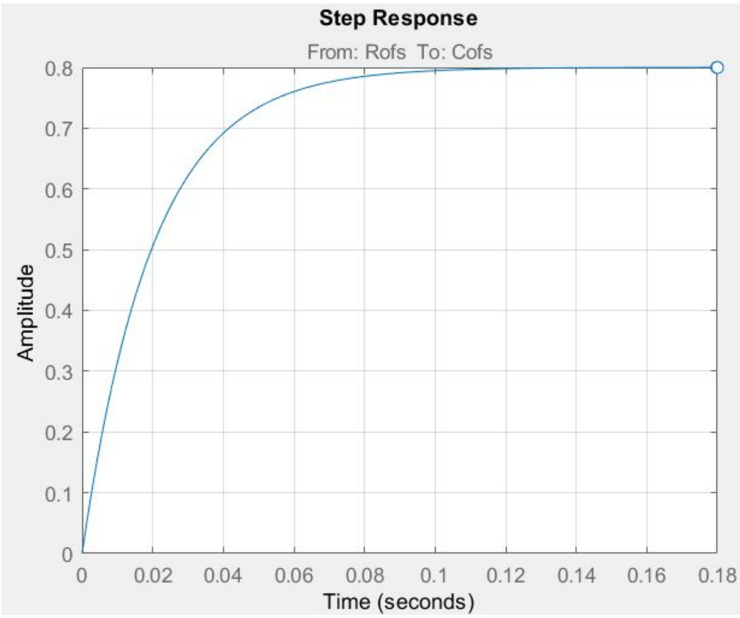
System Q.3 - Case II



Command Window

```
>> SysCharac

SysCharac =

  struct with fields:

        RiseTime: NaN
    SettlingTime: NaN
     SettlingMin: NaN
     SettlingMax: NaN
       Overshoot: NaN
      Undershoot: NaN
            Peak: Inf
        PeakTime: Inf
```

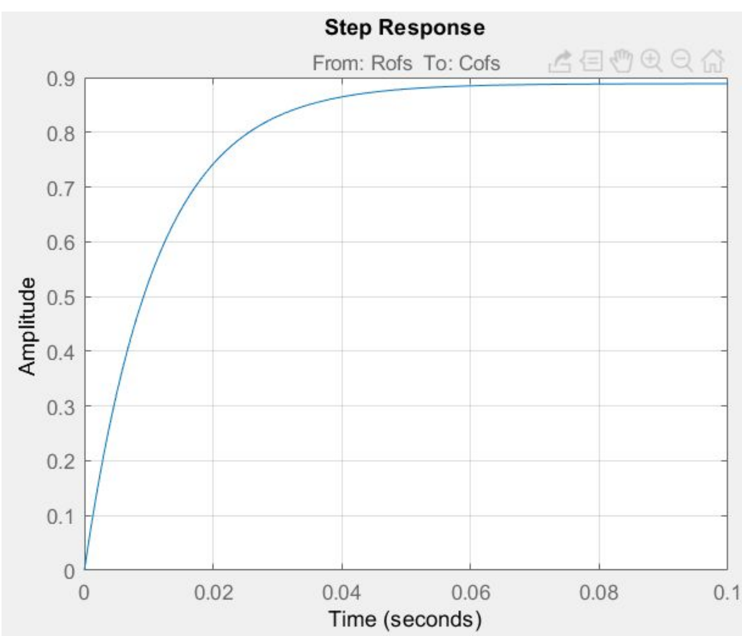
System Q.4 - Case I



Command Window

```
>> SysCharac

SysCharac =

  struct with fields:

        RiseTime: 0.0439
    SettlingTime: 0.0782
     SettlingMin: 0.7200
     SettlingMax: 0.8000
       Overshoot: 0
      Undershoot: 0
            Peak: 0.8000
        PeakTime: 0.6162
```

System Q.4 - Case II



Command Window

```
>> SysCharac

SysCharac =

  struct with fields:

        RiseTime: 0.0244
    SettlingTime: 0.0435
     SettlingMin: 0.8000
     SettlingMax: 0.8889
       Overshoot: 0
      Undershoot: 0
            Peak: 0.8889
        PeakTime: 0.3423
```

Conclusion :

From this experiment, it is now well understood the effect of variation of the parameters of the PID Controller on the transient response of the system.