Visvesvaraya National Institute of Technology (VNIT), Nagpur

Control Theory (ECL312 ECP312)

Lab Report

Submitted by :
Bipasha Parui (BT19ECE019)
Semester 4

Submitted to :
Dr.Punit K. Bhavsar and Dr.Deep Gupta
(Course Instructors)
Department of Electronics and Communication Engineering,
VNIT Nagpur

Contents

Experiment-1:

Problem statement: Solve the 2nd order differential equations for a simple pendulum with the following given conditions: theeta(0) = pi/3 ; theeta'(0) = 0 ; L = 2 meters.

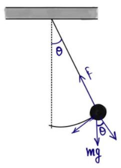


Figure 1: Problem diag

Mathematical formulation: The second order differential equation that forms is

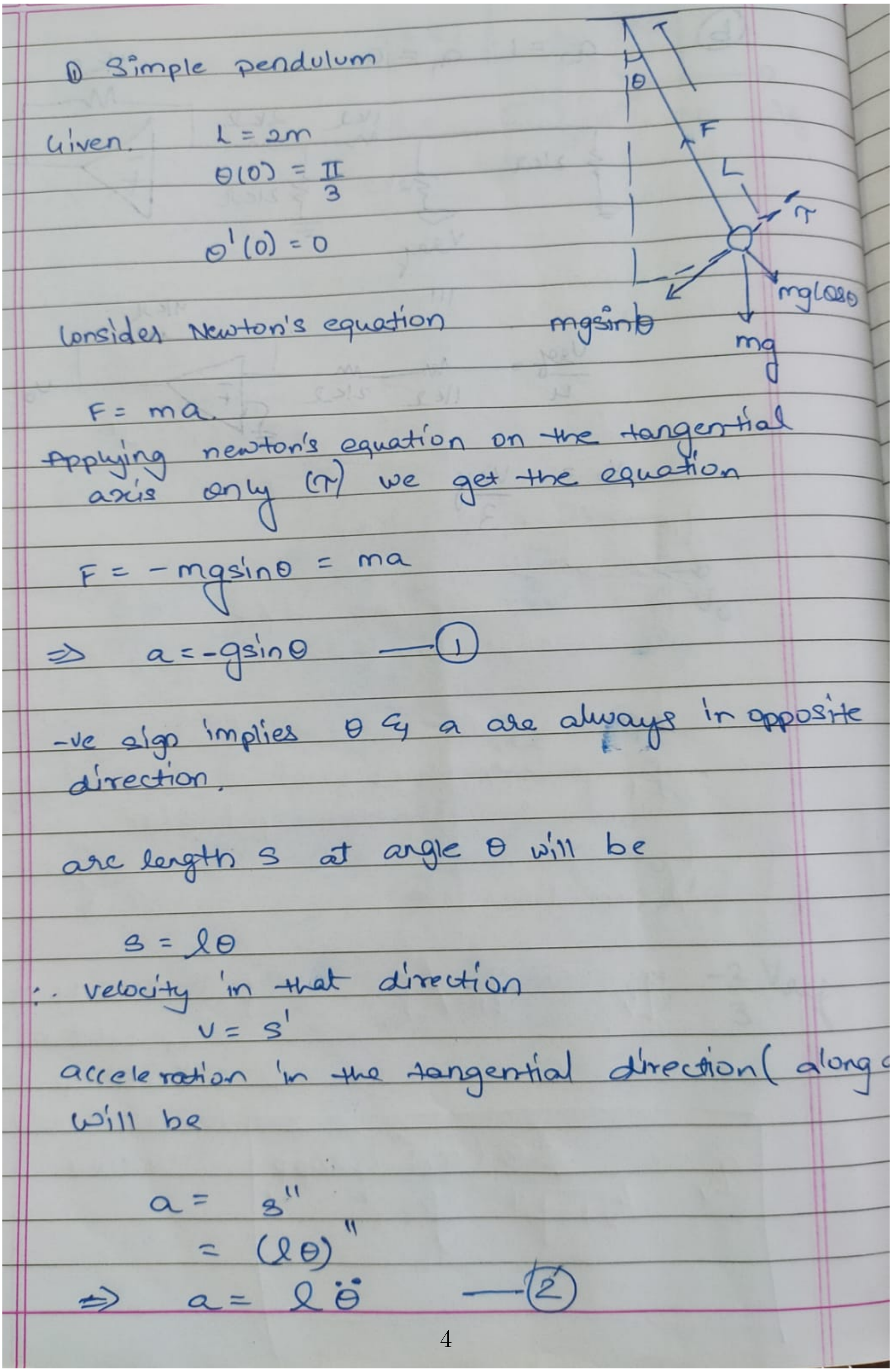
$$\ddot{\theta} - \frac{g}{L} sin(\theta) = 0$$

where we have 2 state variables forming 2nd order differential equation.

Code:

```
clc;
clear all;
clear;
t = 0.05;
n = 20;
tp = [0:t:n];
x_0 = [pi/3,0]

func = @(t,x)[x(2,1);-(9.8/2)*sin(x(1,1))];
[time,y] = ode45(func,tp,x_0);


plot(time,y(:,1),'r')

title('x')
grid
hold on
plot(time,y(:,2),'b')
xlabel('Time in seconds');
legend({'displacement in x','derivative in x'});
```

Manual Solution:



① Simple pendulum

Given. $L = 2m$

$\theta(0) = \frac{\pi}{3}$

$\theta'(0) = 0$

Consider Newton's equation

$F = ma.$

Applying newton's equation on the tangential axis only (T) we get the equation

$F = -mg\sin\theta = ma$

$\Rightarrow \quad a = -g\sin\theta \quad — ①$

-ve sign implies θ & a are always in opposite direction.

arc length s at angle θ will be

$s = l\theta$

∴ velocity in that direction

$v = s'$

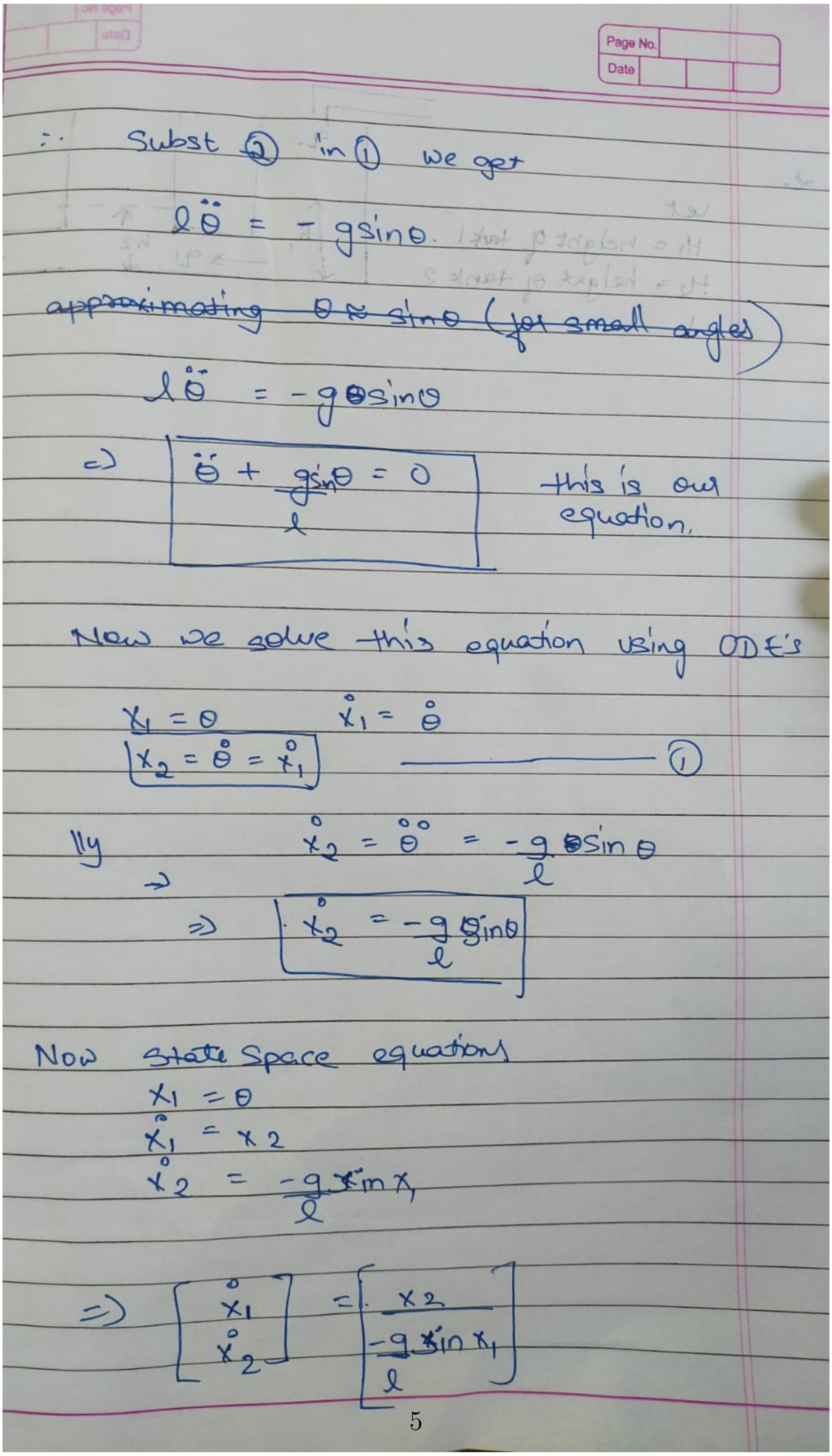
acceleration in the tangential direction(along
will be

$a = s''$

$= (l\theta)''$

$\Rightarrow \quad a = l\ddot{\theta} \quad — ②$

Figure 2: Motion of simple pendulum



∴ Subst (2) in (1) we get

$$l\ddot{\theta} = -g\sin\theta$$

~~approximating $\theta \approx \sin\theta$ (for small angles)~~

$$l\ddot{\theta} = -g\sin\theta$$

$$\Rightarrow \boxed{\ddot{\theta} + \frac{g\sin\theta}{l} = 0}$$ this is our equation.

Now we solve this equation using ODE's

$x_1 = \theta \qquad \dot{x}_1 = \dot{\theta}$

$\boxed{x_2 = \dot{\theta} = \dot{x}_1}$ ——— (1)

||ly $\dot{x}_2 = \ddot{\theta} = -\frac{g}{l}\sin\theta$

$$\Rightarrow \boxed{\dot{x}_2 = -\frac{g}{l}\sin\theta}$$

Now state space equations

$x_1 = \theta$

$\dot{x}_1 = x_2$

$\dot{x}_2 = -\frac{g}{l}\sin x_1$

$$\Rightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{l}\sin x_1 \end{bmatrix}$$

Figure 3: Motion of simple pendulum

Results:

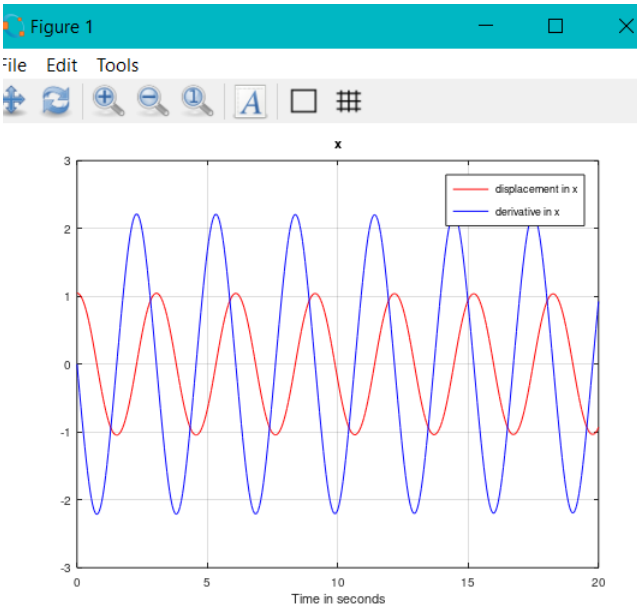


Figure 4: Motion of simple pendulum

Analysis Conclusions:

- We observe that the motion in terms of displacement as well as velocity of a simple pendulum turns out to be oscillatory without any damping.
- This is because factors such as air resistance , external force have not been taken into consideration. as a result the pendulum experiences free oscillations.
- Hence oscillations will take place till infinity .

Hence we analysed the second order differential equation of motion of simple pendulum using ODE solver.

Experiment-2:

Problem statement: Solve the 2nd order differential equations for a two tank system shown below

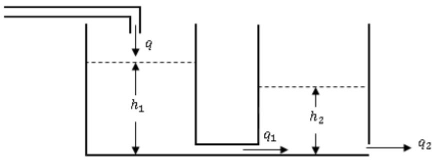


Figure 5: Problem diag2

Mathematical formulation: The second order differential equations that we get are is

$$T_1 \frac{dh_1}{dt} + h_1 - h_2 = qR_1$$

and

$$T_2 R_1 \frac{dh_2}{dt} = (h_1 - h_2)R_2 - h_2 R_1$$

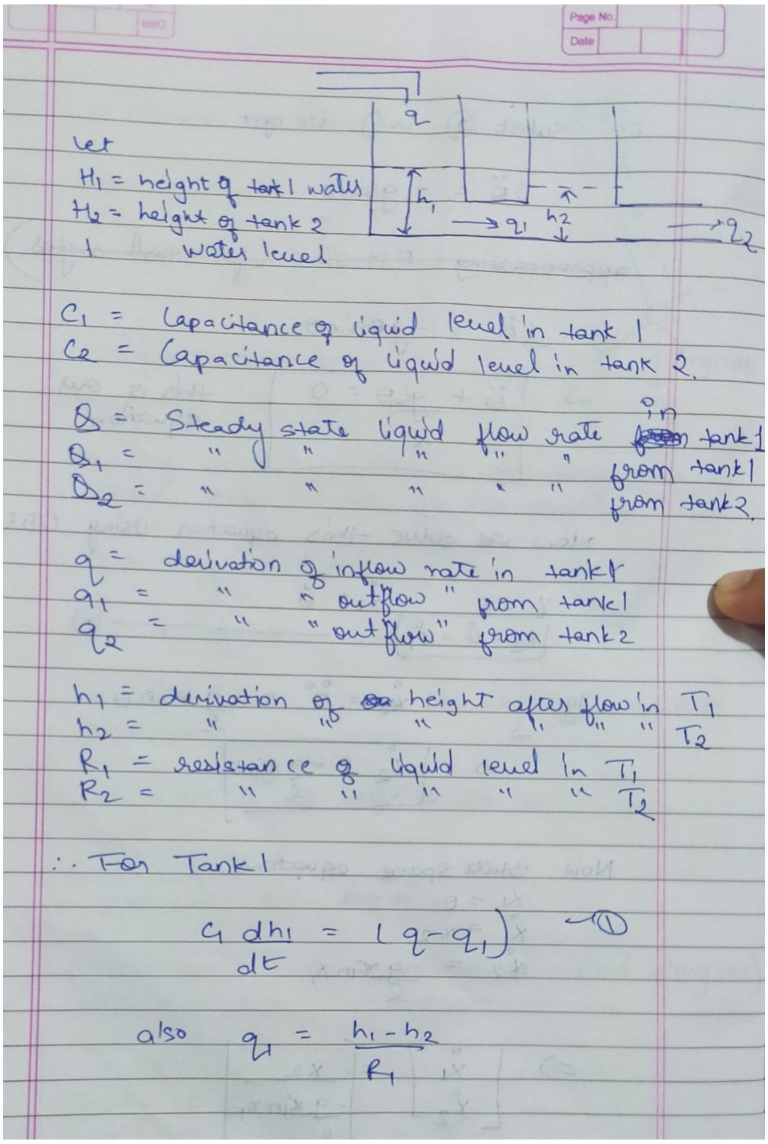
where we have 2 state variables forming 2nd order differential equation. Transfer function can be found in the manual solution

Code:

```
clc;
clear;
close all;

Res1=2;
Res2=5;
Cap1=1.1;
Cap2=1.4;
h10=10;
h20=20;
q=5;

t = 0.5;
n = 100;
tp = [0:t:n];
y_0 = [h10; h20];
```

```

func = @(t, y)[(1/Cap1)*(q-(y(1)-y(2))/Res1); ...
    (1/Cap2)*((y(1)-y(2))/Res1 - y(2)/Res2)];
[time,y] = ode45(func, tp, y_0);

figure(1)
plot(time,y(:,1),'r');
grid
hold on
plot(time,y(:,2),'b');
xlabel('t');
legend({'Movement in h1', 'Movement in h2'});
```

Manual Solution:



let

H_1 = height of tank 1 water

H_2 = height of tank 2 water level

C_1 = Capacitance of liquid level in tank 1

C_2 = Capacitance of liquid level in tank 2.

Q = Steady state liquid flow rate in tank 1

Q_1 = " " " " " from tank 1

Q_2 = " " " " " from tank 2.

q = deviation of inflow rate in tank 1

q_1 = " " outflow " from tank 1

q_2 = " " outflow " from tank 2

h_1 = derivation of height after flow in T_1

h_2 = " " " " " " T_2

R_1 = resistance of liquid level in T_1

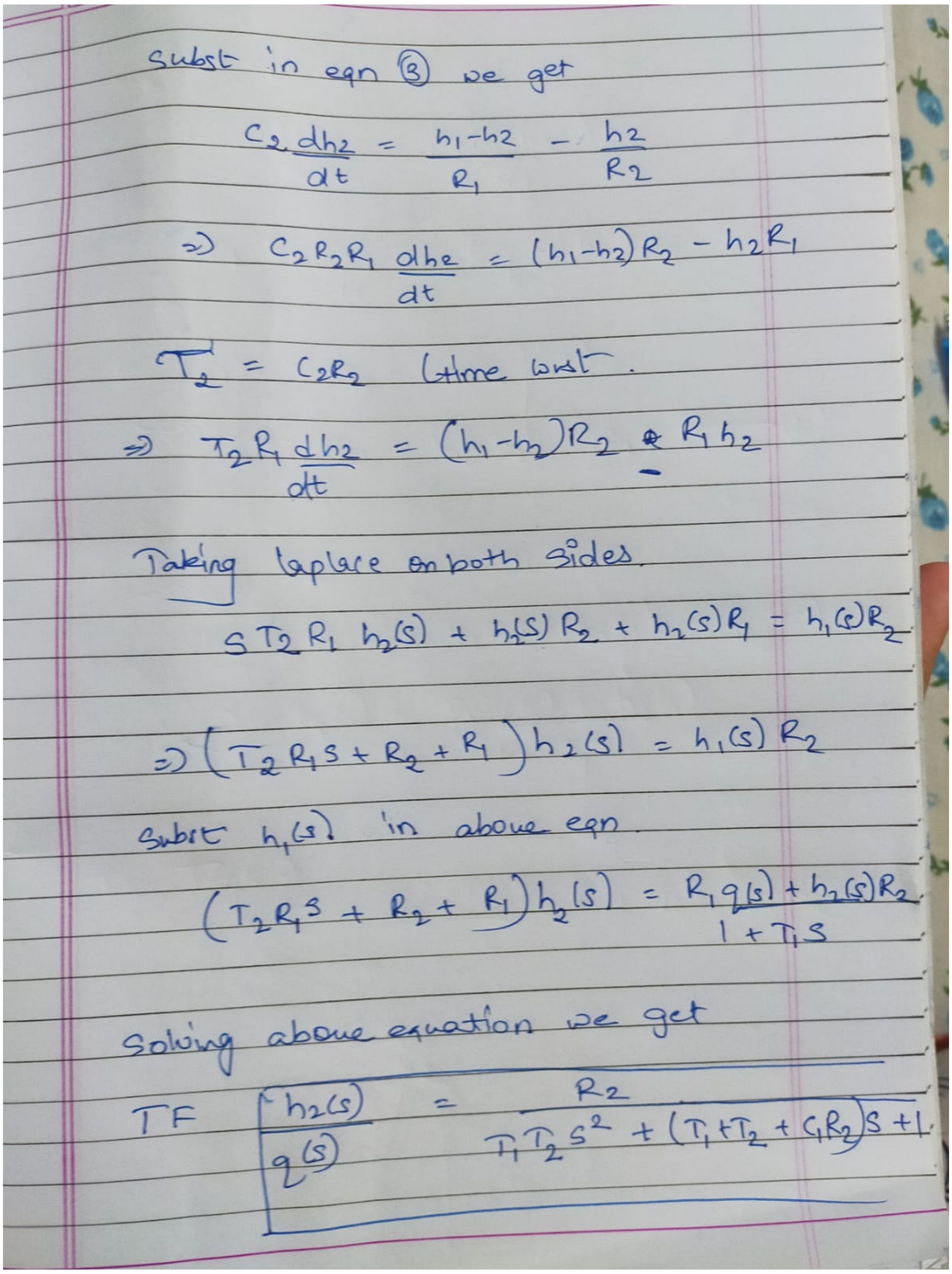
R_2 = " " " " " T_2

∴ For Tank 1

$$C_1 \frac{dh_1}{dt} = (q - q_1) \quad \text{--- (1)}$$

also $q_1 = \dfrac{h_1 - h_2}{R_1}$

Figure 6: sol1



subst in eqn (3) we get

$$C_2 \frac{dh_2}{dt} = \frac{h_1-h_2}{R_1} - \frac{h_2}{R_2}$$

$$\Rightarrow \quad C_2R_2R_1 \frac{dh_2}{dt} = (h_1-h_2)R_2 - h_2R_1$$

$T_2 = C_2R_2$ (time const.

$$\Rightarrow \quad T_2R_1\frac{dh_2}{dt} = (h_1-h_2)R_2 \cancel{+} - R_1h_2$$

Taking laplace on both sides

$$sT_2R_1h_2(s) + h_2(s)R_2 + h_2(s)R_1 = h_1(s)R_2$$

$$\Rightarrow (T_2R_1s + R_2 + R_1)h_2(s) = h_1(s)R_2$$

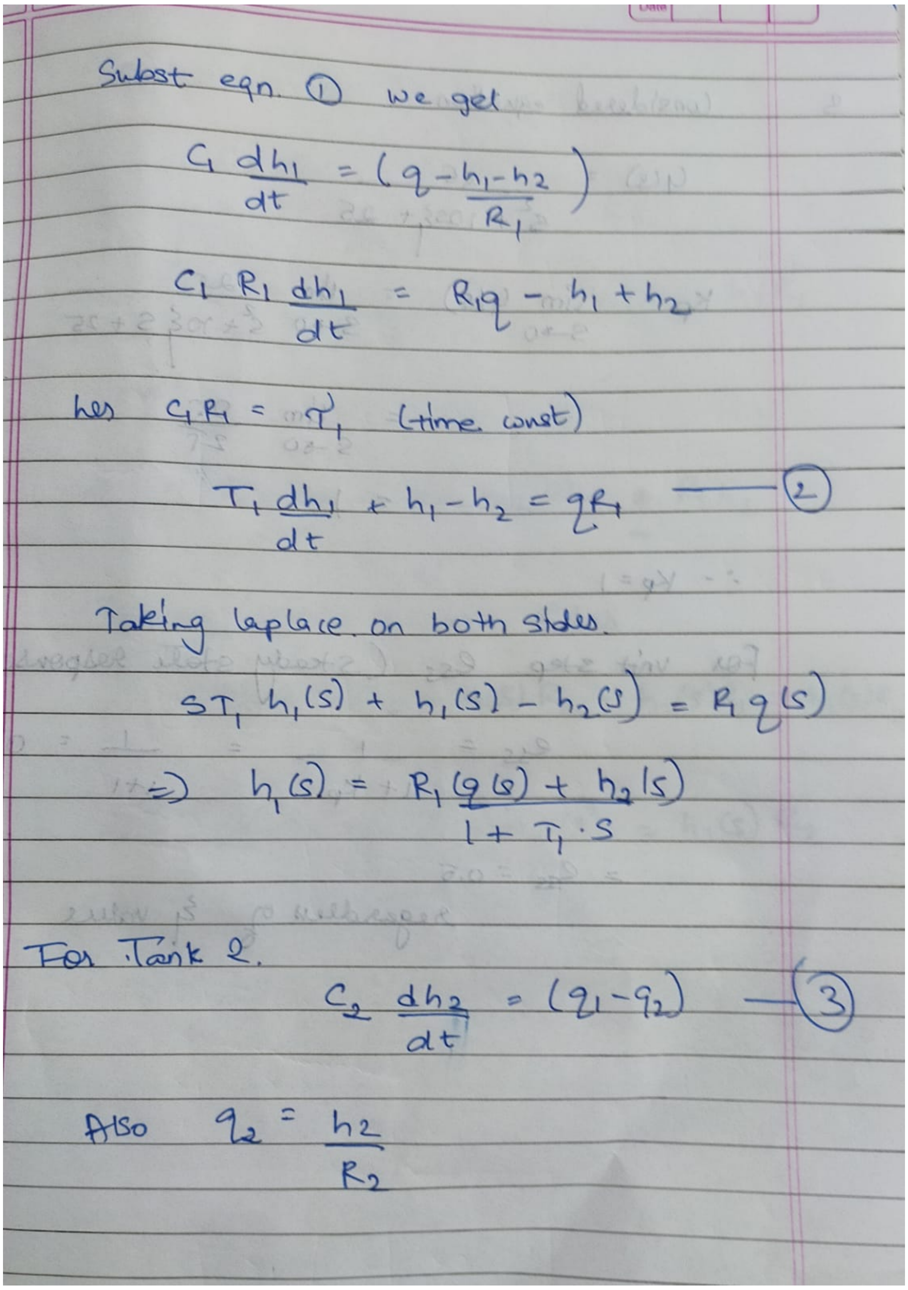
subst $h_1(s)$ in above eqn

$$(T_2R_1s + R_2 + R_1)h_2(s) = \frac{R_1q(s) + h_2(s)R_2}{1+T_1s}$$

Solving above equation we get

$$TF \quad \boxed{\frac{h_2(s)}{q(s)} = \frac{R_2}{T_1T_2s^2 + (T_1+T_2+C_1R_2)s + 1}}$$

Figure 7: sol2



Subst eqn. 1 we get

$$C_1 \frac{dh_1}{dt} = \left(q - \frac{h_1 - h_2}{R_1}\right)$$

$$C_1 R_1 \frac{dh_1}{dt} = R_1 q - h_1 + h_2$$

here $C_1 R_1 = \tau_1$ (time const)

$$T_1 \frac{dh_1}{dt} + h_1 - h_2 = qR_1 \quad \text{—— } (2)$$

Taking laplace on both sides.

$$sT_1 h_1(s) + h_1(s) - h_2(s) = R_1 q(s)$$

$$\Rightarrow h_1(s) = \frac{R_1 (q(s) + h_2(s))}{1 + T_1 \cdot s}$$

For Tank 2.

$$C_2 \frac{dh_2}{dt} = (q_1 - q_2) \quad \text{—— } (3)$$

Also $q_2 = \dfrac{h_2}{R_2}$

Figure 8: sol3

Results:

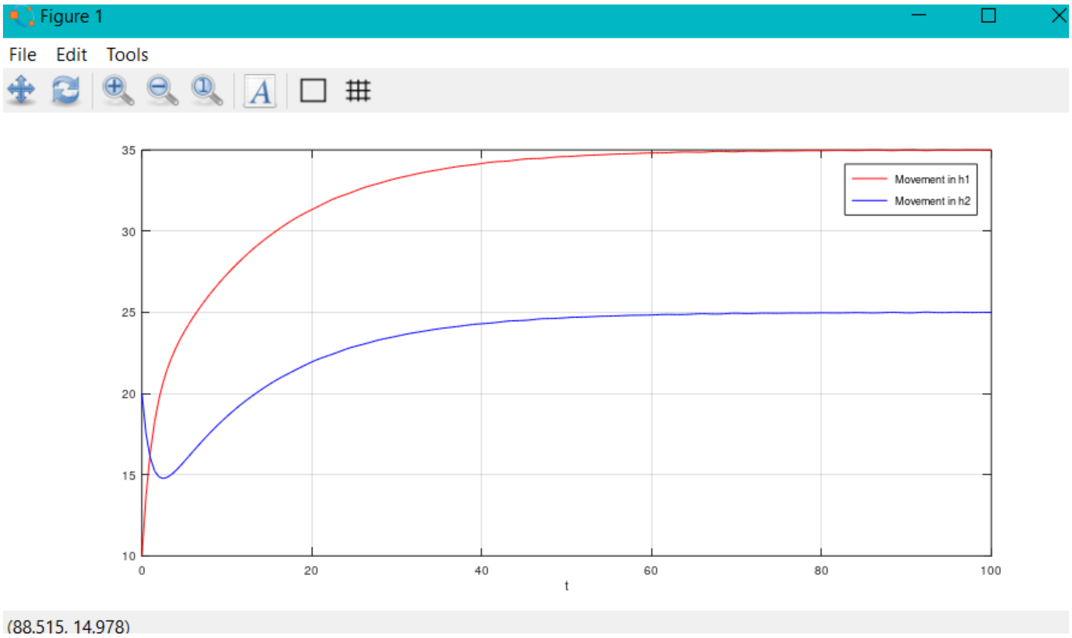


Figure 9: Flow in Two tank system

Analysis Conclusions:

- We have thus observed the flow in a 2 tank system which generates a 2nd order differential equation with h1 and h2 as the variables of flow.
- The graph is observed to be of 2 different types. For tank 1 we see that the graph increases for some time and then attains a stable const value indicating that the water initially flows and increases in into the tank and then reaches a constant level.
- in contrast for the tank 2 we observe that there is first a steep dip in the the value and then it increases and finally attains a constant level just like tank 1 .This is evident as in tank 2 initially the water flows out of the pipe till the height level equalises and then after height level equalises. the tank 2 starts filling till it reaches its max stable level.

Hence we analysed the second order differential equation of a Two tank system using ODE solver.

Experiment-3:

Problem statement: a. Observe the effects on the different parameters(transient) with vary ing values of damping ratio for second order system

Mathematical formulation: The second order equation taken into consideration is

$$= \frac{25}{s^2 + 10 * \zeta + 25}$$

where wn = 5 with varying zeta value

Code:

```
clc;
clear;

w_n = 5;
s = tf('s');
zeta1 = 0.1;
trans_func1 = tf(25,[1 2*5*zeta1 25]);
zeta2 = 0.3;
trans_func2 = tf(25,[1 2*5*zeta2 25]);
zeta3 = 0.5;
trans_func3 = tf(25,[1 2*5*zeta3 25]);
zeta4 = 0.7;
trans_func4 = tf(25,[1 2*5*zeta4 25]);
zeta5 = 1;
trans_func5 = tf(25,[1 2*5*zeta5 25]);
zeta6 = 2;
trans_func6 = tf(25,[1 2*5*zeta6 25]);

figure(1)
step(trans_func1);
hold on
step(trans_func2);
hold on
step(trans_func3);
hold on
step(trans_func4);
hold on
step(trans_func5);
hold on
step(trans_func6);
legend('Zeta = 0.1','Zeta = 0.3','Zeta = 0.5','Zeta = 0.7','Zeta ...
    = 1','Zeta = 2');
```

Results:

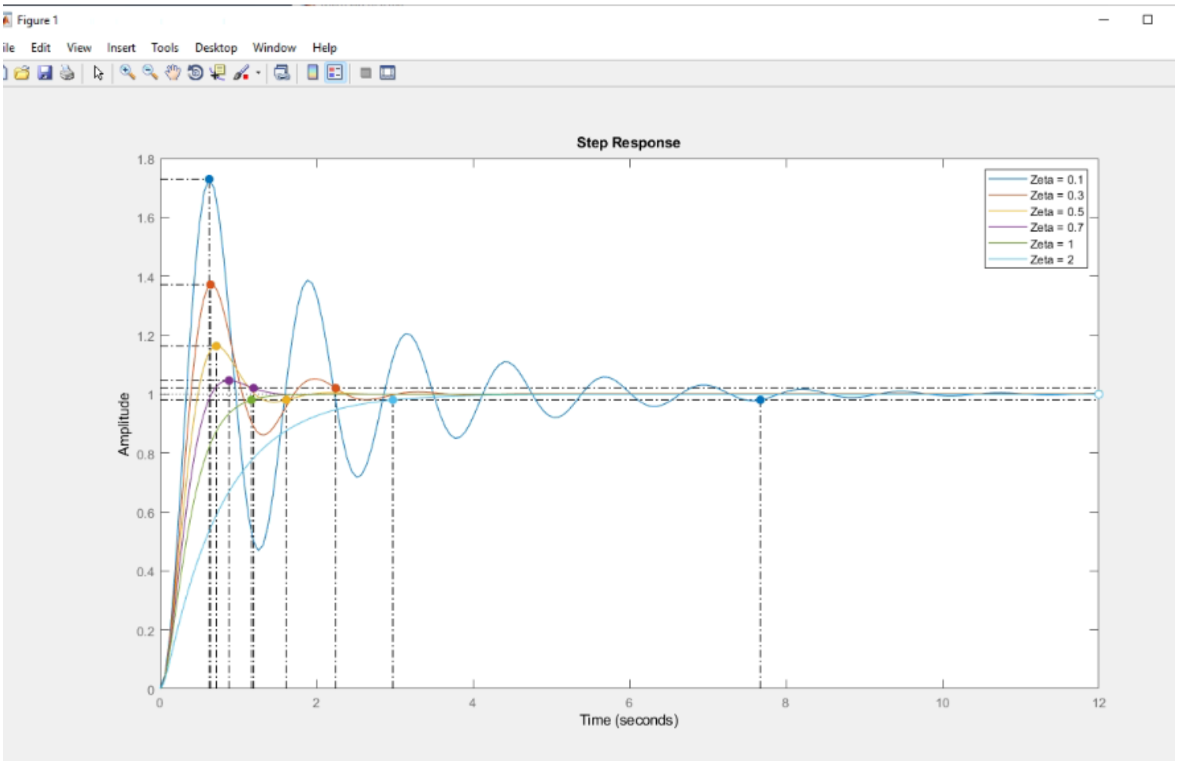


Figure 10: Varying zeta

ξ	Tr (s)	Tp (s)	Mp (%)	Ts (s)	Steady State Error
0.1	0.2254	0.6283	72	7.6746	0.5
0.3	0.2647	0.6447	37	2.246	0.5
0.5	0.3278	0.7184	16.3	1.6152	0.5
0.7	0.4254	0.8816	4.6	1.1958	0.5
1	0.6717	2.39	0	1.1668	0.5
2	1.6462	5.4654	0	2.9758	0.5

Figure 11: Varying zeta parameters

Analysis:

- For $0 < \zeta < 1$, the system is found to be overdamped and has significant overshoot.
- At $\zeta = 1$, the system is found to be critically damped and has 0 overshoot for step response.
- For $\zeta < 1$, the system is found to be over damped with no overshoot for step response.
- As the zeta value increases we observe the following trends
  1. Percentage of peak overshoot decreases.
  2. Peak time and rise time also increases as the oscillations decreases 3. It does not have any effect on the steady state error.

section b) Show in all plots by holding curserr and verify with numeric params.

Results:

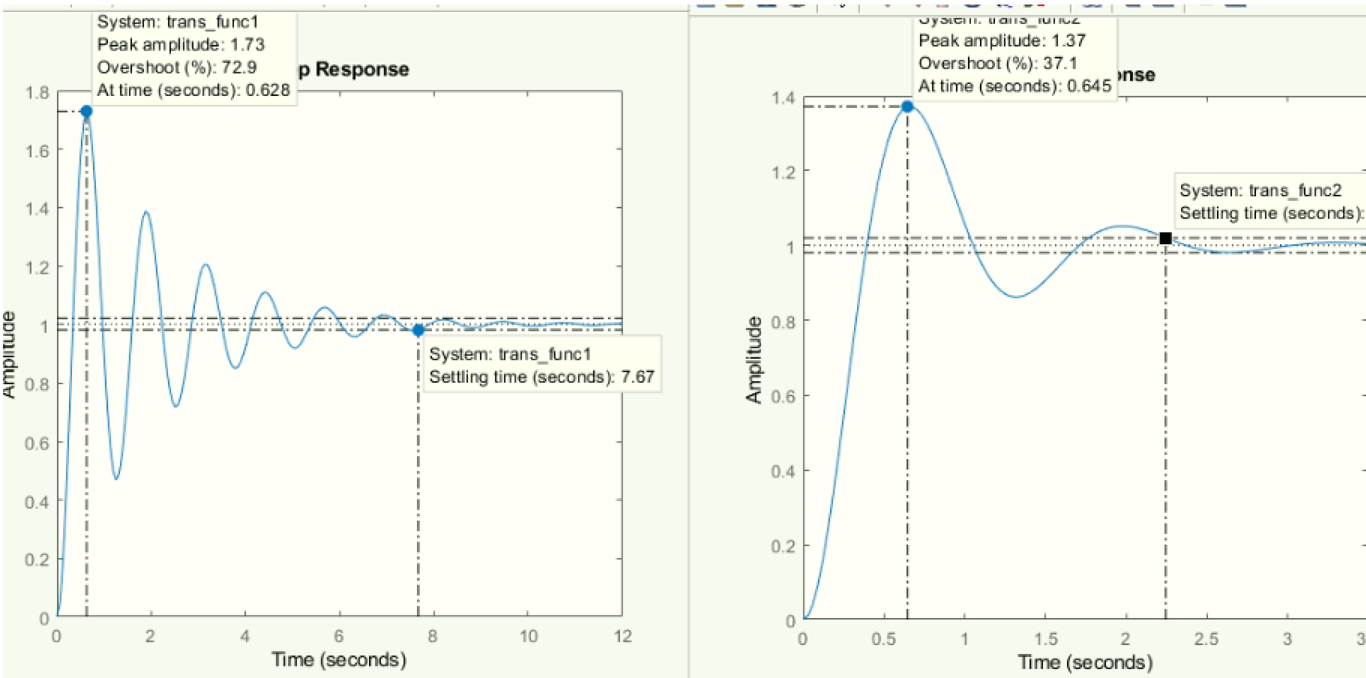


Figure 12: Varying zeta

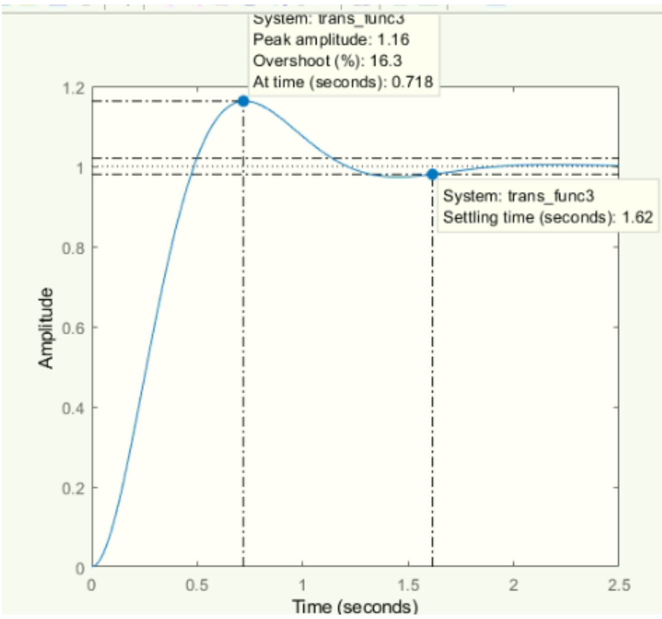


Figure 13: Varying zeta

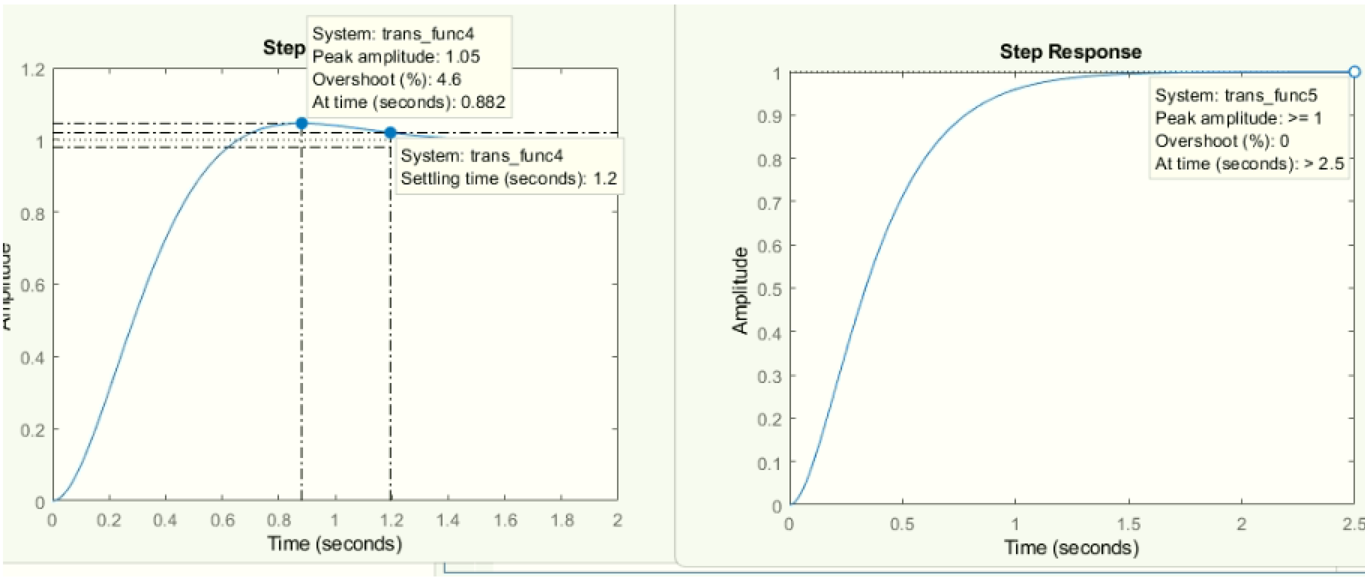


Figure 14: Varying zeta

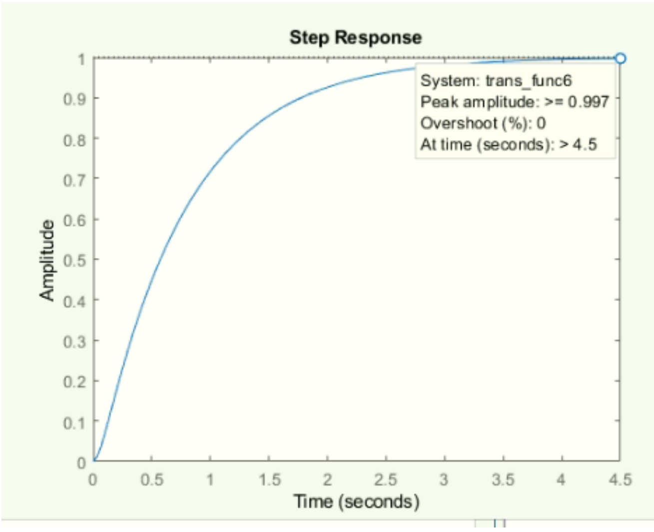


Figure 15: Varying zeta

Code: section c) - addition of poles/zeros

```
clc;
clear;
%% Addition of Zeros
w_n = 5;
s = tf('s');
zeta1 = 0.1;
trans_func1 = w_n^2/(s^2 + 2*w_n*zeta1*s + w_n^2);
stepinfo(trans_func1);
trans_func2 = trans_func1*(s+3);
stepinfo(trans_func2);
trans_func3 = trans_func1*(s+2)*(s+3);
stepinfo(trans_func3);
figure(1)
step(trans_func1);
hold on
step(trans_func2);
hold on
step(trans_func3);
legend('No zero1','1 zero','2 zeros');

clc;
clear;
%% Addition of poles
w_n = 5;
s = tf('s');
zeta1 = 0.1;
trans_func1 = w_n^2/(s^2 + 2*w_n*zeta1*s + w_n^2);
stepinfo(trans_func1);
trans_func2 = trans_func1/(s+3);
stepinfo(trans_func2);
trans_func3 = trans_func1/(s^2+5*s+6);
stepinfo(trans_func3);
figure(1)
step(trans_func1);
hold on
step(trans_func2);
hold on
step(trans_func3);
legend('No pole','1 pole(s+3)','2 pole(s+3)*(s+2)');
```

Results : Addition of zeros:

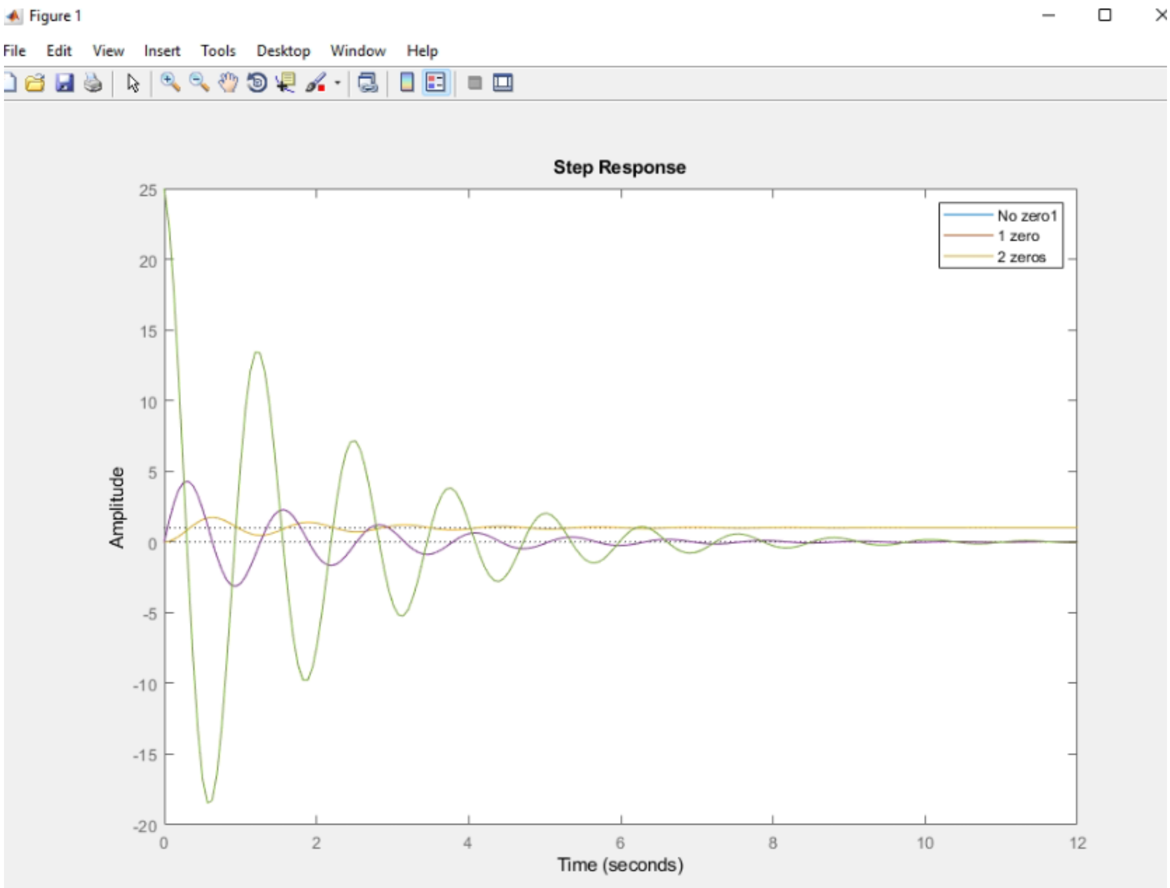


Figure 16: Varying zeros

Analysis: Addition of zeros:

- Addition of zeros does not affect the nature of oscillations, only the magnitude of performance parameters. Oscillations seem subdued on adding negative zeros as can be seen in the plot.
- Adding a zero to the left half plane to the transfer function causes the step response to be faster along with decrease in rise and peak time. Subsequently overshoot increases.
- Adding a zero to the right half plane of transfer functioncauses the step response to become slower along with decrease in overshoot.
- Addition of zeros to the TF tends to cause a pulling of root locus on the left side , making the system more stable.

Results : Addition of poles:

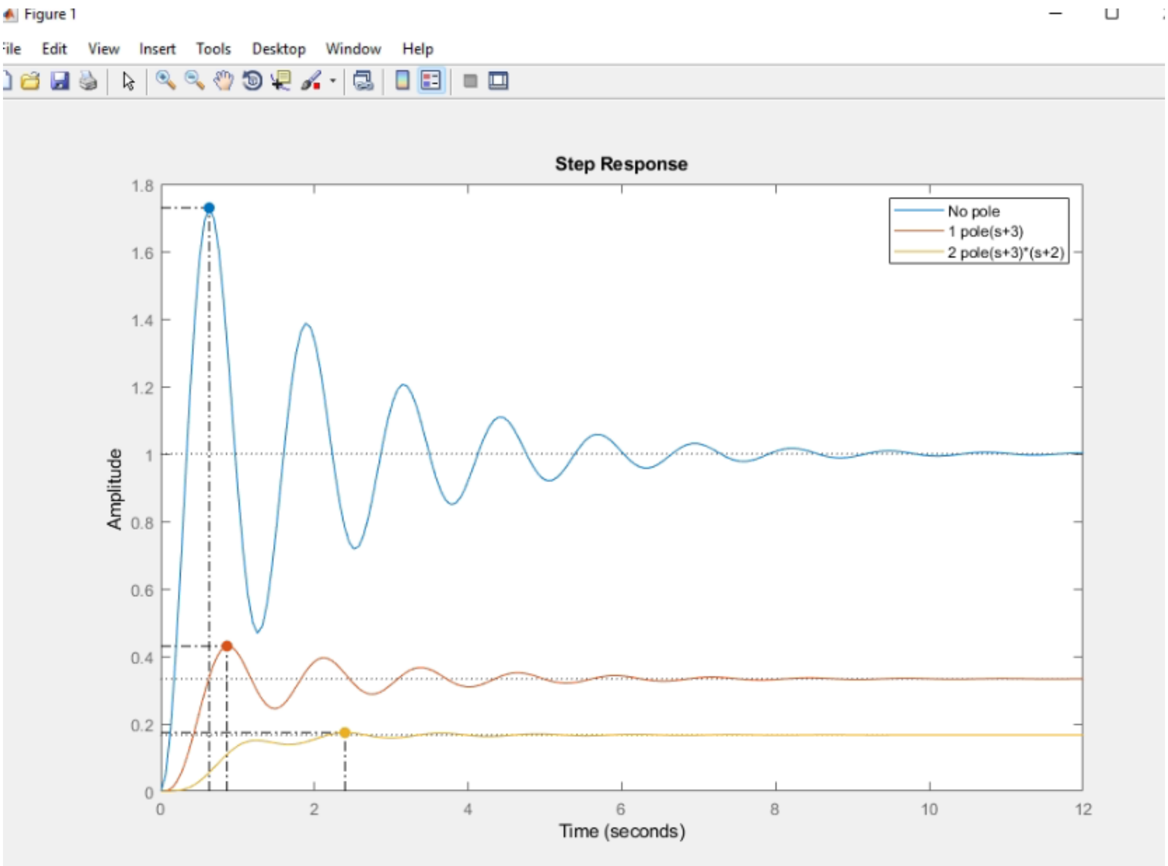


Figure 17: Varying poles

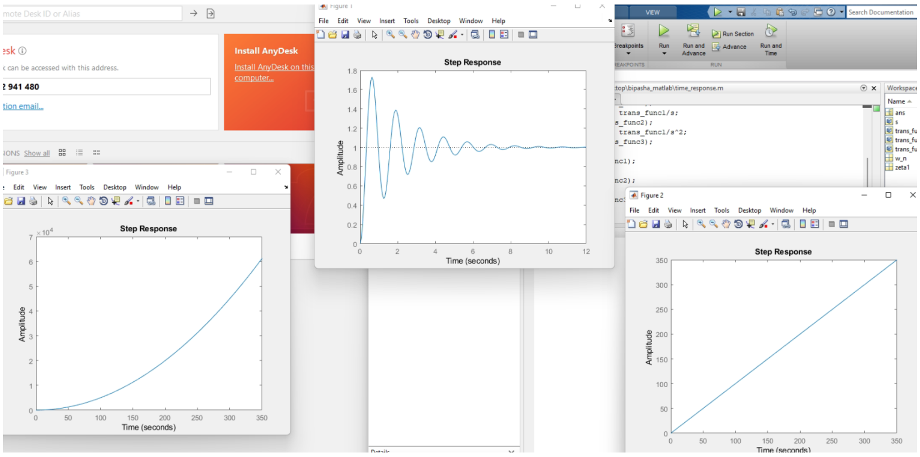


Figure 18: Varying poles only s

Analysis: Addition of poles:

- It is observed that by adding the poles to the transfer function, the step response of the TF becomes increasingly sluggish.
- The poles closest to origin are the systems dominant pole.
- The exponential terms of far away poles die out quick in comparison to dominant poles.
- Adding poles to the TF mmakes the system more unstable as it tends to pull the root locus in the right direction.
- addition of non zero poles tends to still keep the response intact as can been seen in the first plot however addition of poles at origin unexpecptedly overthrows the response thereby giving NaN values for performance parameters as can be seen in the second plot.

Experiment-4:

Problem statement: a) Analyze the effect of P, I, D, PI, PID and prepare the table for all sub-parts individually. b) Comment on the stability, transient and steady state behaviour based on the above prepared tables.

Mathematical formulation: The base transfer function used for analysis is

$$TF = \frac{100}{s^2 + 10s + 100}$$

where wn = 10 and zeta = 0.5

MAIN TF:

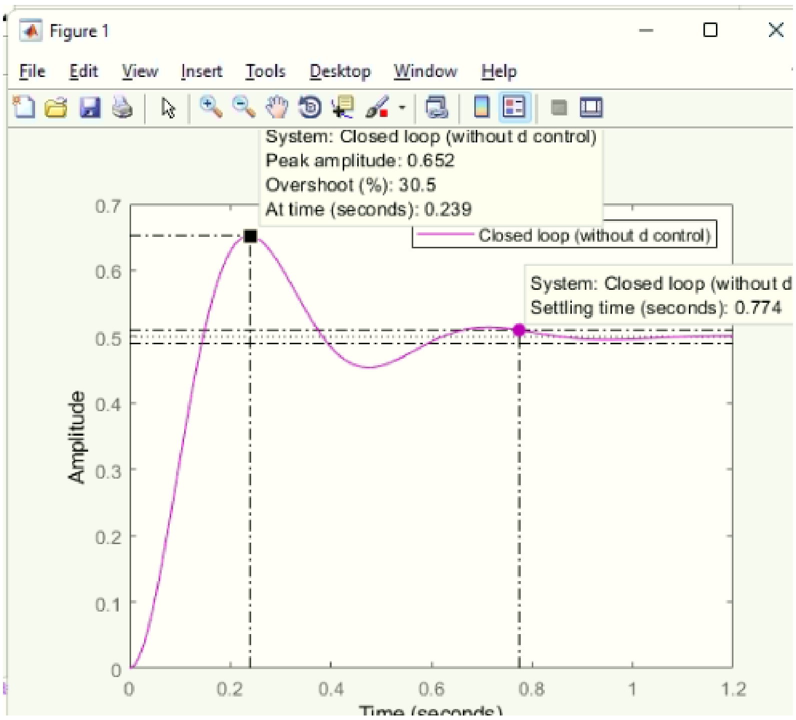


Figure 19: OrgTF

Code:

```
clr;
clear all;
pkg load control;

```

```
% P control
p_contr1 = pid(2);
p_contr2 = pid(3);
p_contr3 = pid(8);
p_contr4 = pid(15);
%p control of 2nd order system
func1 = tf(100,[1,10,100])
fb_func1 = feedback(func1,[1]);

p1 = feedback(func1*p_contr1,[1]);
p2 = feedback(func1*p_contr2,[1]);

p3 = feedback(func1*p_contr3,[1]);
p4 = feedback(func1*p_contr4,[1]);

stepinfo(fb_func1)
stepinfo(p1)
stepinfo(p2)
stepinfo(p3)
stepinfo(p4)

% i control
clear;
clear all;

p_contr1 = pid(0,0.8,0);
p_contr2 = pid(0,5,0);
p_contr3 = pid(0,20,0);
%p control of 2nd order system
func1 = tf(100,[1,10,100]);
fb_func1 = feedback(func1,[1]);

p1 = feedback(func1*p_contr1,[1]);
p2 = feedback(func1*p_contr2,[1]);
p3 = feedback(func1*p_contr3,[1]);

stepinfo(fb_func1)
stepinfo(p1)
stepinfo(p2)
stepinfo(p3)

figure(1)
step(fb_func1,'-m');
legend({'Closed loop (without i control)'});
figure(3)
step(p1,'-r');
hold on
step(p2,'-b');
legend({'ki=0.2','ki=2'});
```

```
figure(2)
step(p3,'-g');
legend({'ki=20'});

% d control
clear;

p_contr1 = pid(0,0,0.2);
p_contr2 = pid(0,0,2);
p_contr3 = pid(0,0,20);
%p control of 2nd order system
func1 = tf(100,[1,10,100]);
fb_func1 = feedback(func1,[1]);

p1 = feedback(func1*p_contr1,[1]);
p2 = feedback(func1*p_contr2,[1]);

p3 = feedback(func1*p_contr3,[1]);

stepinfo(fb_func1)
stepinfo(p1)
stepinfo(p2)
stepinfo(p3)

figure(1)
step(fb_func1,'-m');
hold on
step(p1,'-r');
legend({'Closed loop (without d control)','kp=0.2'});
figure(2)
step(p2,'-b');
hold on
step(p3,'-g');
legend({'kd=2','kd=20'});

% pi
clear;

p_contr1 = pid(5,0.2,0);
p_contr2 = pid(5,0.02,0);
p_contr3 = pid(5,10,0);
p_contr5 = pid(7,0.2,0);
p_contr4 = pid(3,0.2,0);
%p control of 2nd order system
func1 = tf(100,[1,10,100]);
fb_func1 = feedback(func1,[1]);

p1 = feedback(func1*p_contr1,[1]);
p2 = feedback(func1*p_contr2,[1]);
```

```
p4 = feedback(func1*p_contr4,[1]);
p5 = feedback(func1*p_contr5,[1]);
p3 = feedback(func1*p_contr3,[1]);


stepinfo(p1)
stepinfo(p2)
stepinfo(p3)
stepinfo(p4)
stepinfo(p5)
figure(1)
step(fb_func1,'-m');
hold on
step(p1,'-r');
hold on
step(p2,'-b');
hold on
step(p3,'g');
legend({'Closed loop (without pi ...
    control)','kp,ki=5,0.2','kp,ki=5,0.02','kp,ki=5,10'});
figure(2)
step(fb_func1,'-m');
hold on
step(p1,'-r');
hold on
step(p4,'-g');
hold on
step(p5,'-b');
legend({'Closed loop (without pi ...
    control)','kp,ki=5,0.2','kp,ki=3,0.2','kp,ki=7,0.2'});

%pid
pid1 = pid(5,8,0.1);
pid2 = pid(5,10,0.06);
pid3 = pid(5,13,0.1);
pid4 = pid(5,10,0.3);


%p control of 2nd order system
func1 = tf(100,[1,10,100])
fb_func1 = feedback(func1,[1]);
stepinfo(fb_func1)
p1= feedback(func1*pid1,[1]);
stepinfo(p1)
p2= feedback(func1*pid2,[1]);
stepinfo(p2)
p3= feedback(func1*pid3,[1]);
stepinfo(p3)
p4= feedback(func1*pid4,[1]);
```

```
stepinfo(p4)
```

Results : P control:

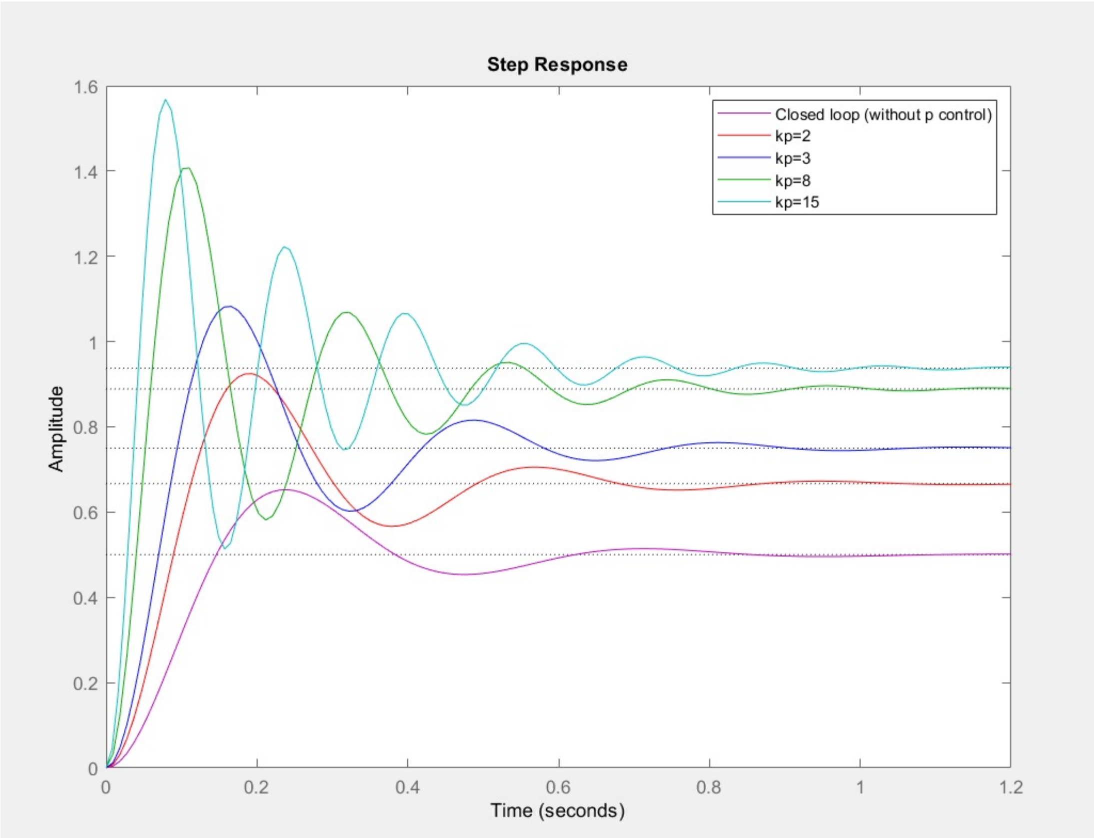


Figure 20: P control

4.1 Results : I control:

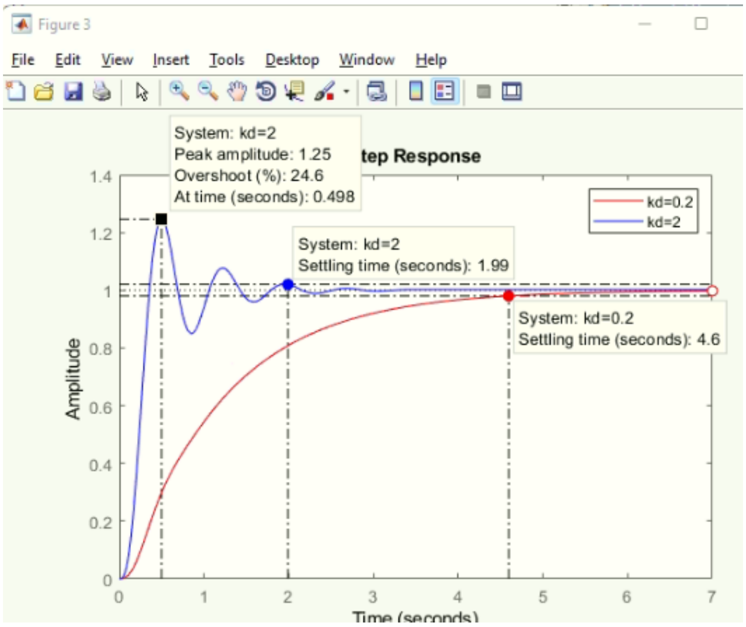


Figure 21: I control 1

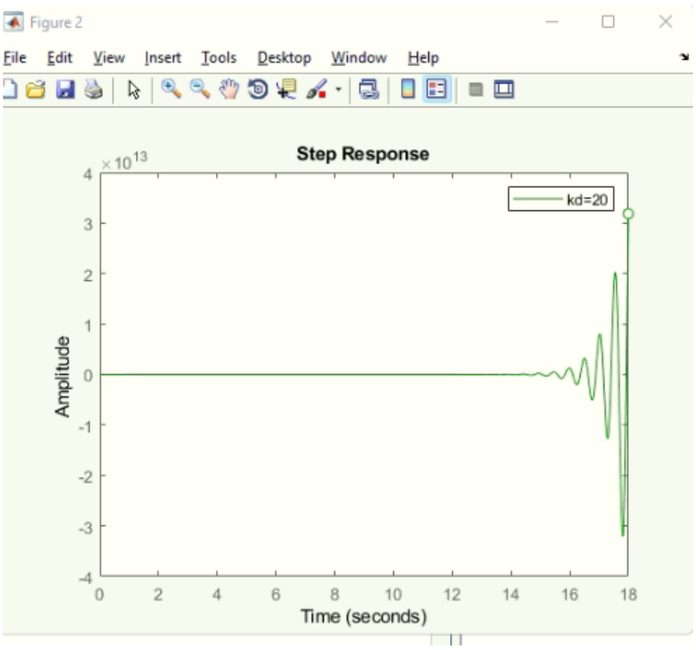


Figure 22: I control 2

<u>Results : D control</u>:

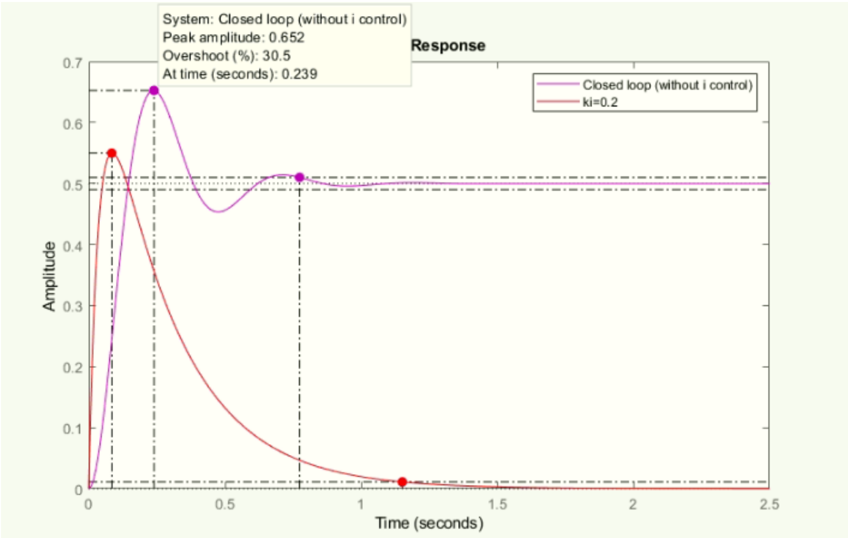


Figure 23: D control 1

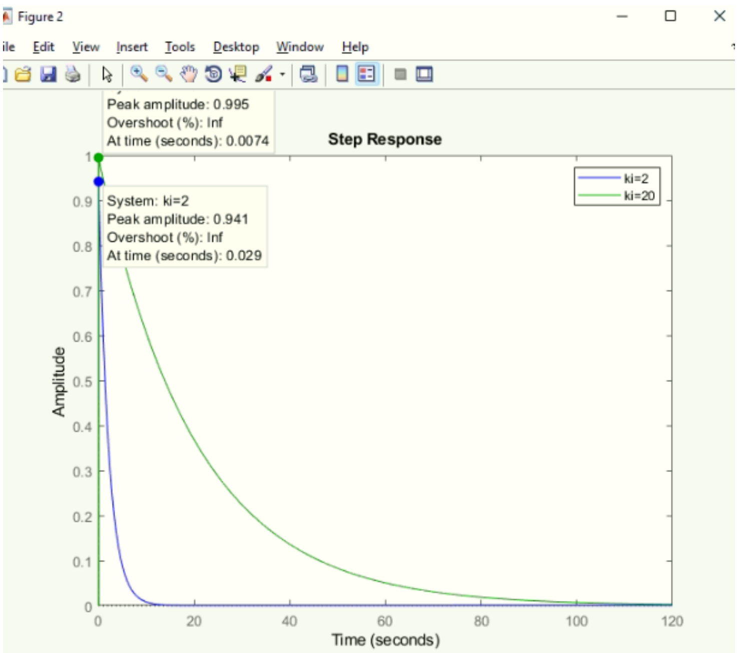


Figure 24: D control 2

<u>Results : PI control</u>:

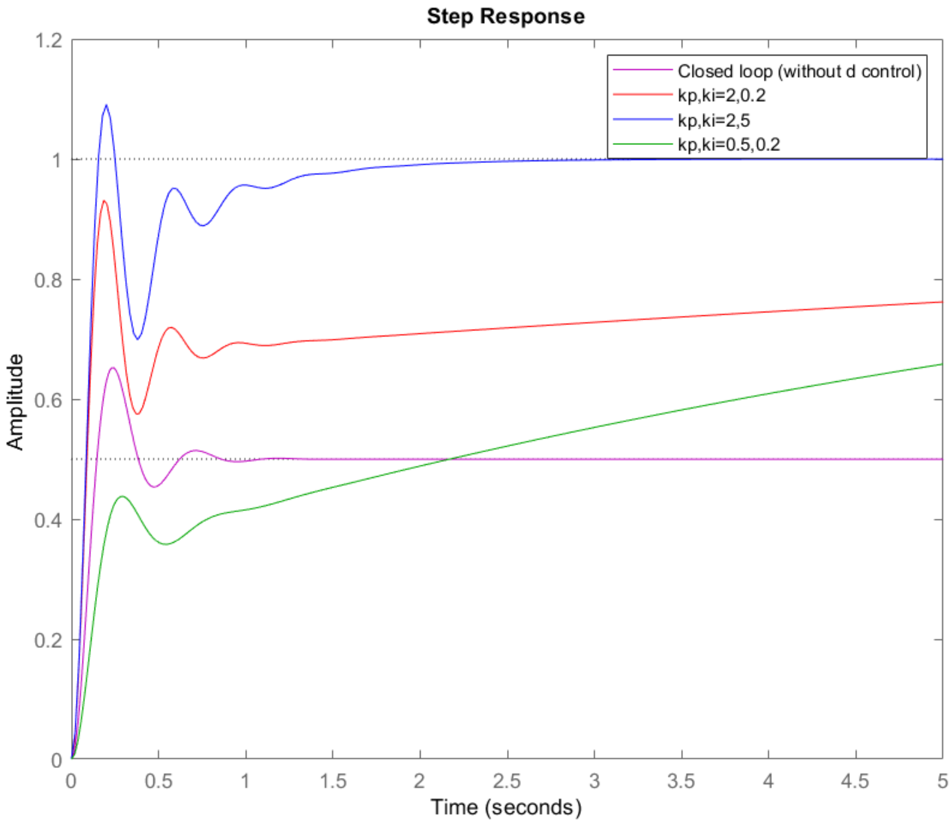


Figure 25: PI Control

Results : PID control:

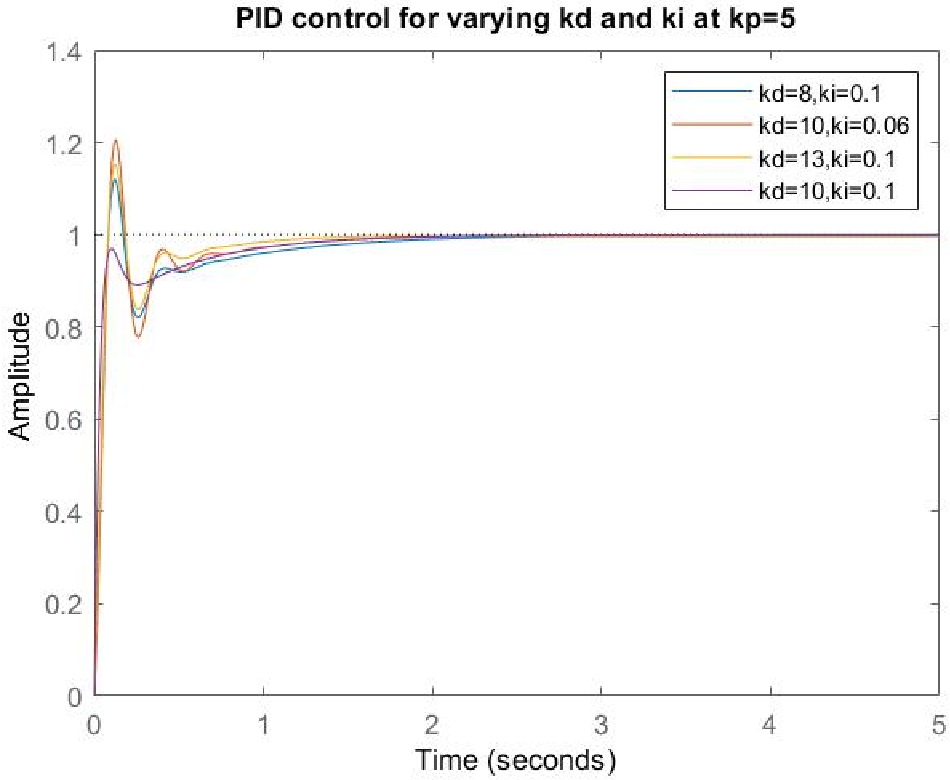


Figure 26: PID control keeping P = 5 constant

P control(Kp)	Tr (s)	Tp (s)	Mp (%)	Ts (s)
2	0.0756	0.1934	38.69	0.787
3	0.0634	0.1658	44.32	0.778
8	0.0395	0.1105	58.31	0.7643
15	0.0288	0.0785	67.29	0.7327

D control(Kd)	Tr (s)	Tp (s)	Mp (%)	Ts (s)
0.2	0	0.08	Inf	1.1516
2	0	0.029	Inf	8.23
20	0	0.0074	Inf	78.63

I control(Ki)	Tr (s)	Tp (s)	Mp (%)	Ts (s)
0.2	2.488	8.42	0	4.59
5	0.2099	0.4976	24.62	1.99
20	NaN	Inf	NaN	NaN

PI control					
Kp	Ki	Tr (s)	Tp (s)	Mp (%)	Ts (s)
2	5	0.1043	0.2	9.1	1.59
2	0.2	0.132	94.18	0	42.07
0.5	0.2	14.06	74.29	0	26.09

PID control		
Kp	Ki	kd
5	0.1	8
5	0.06	10
5	0.1	13
5	0.1	10

Figure 27: P, I , D analysis table

Results : P,I.D analysis table:

Analysis of P control:

- We see that as Kp value increases wn value increases and zeta value decreases. a result we see that there is a decrease in peak time but an increase in peak overshoot which is not desirable.

Analysis D control:

- We obsrve that Kd resists change in the system oscillation vby maintaining zeta value. However there is increase in steady state error but it is not 0 as we desire.

- As kd value increases there a decrease in settling time .
- D-control is unaware where the the set-point is, hence only D control is not affective
- D-only controls do not exist

Analysis I Control:

- I control removes any deviations that exist.
- I controlls are slower in their response time and also lead to decreases in stability.
- I-only controls exists only in very limited cases as there again is not any set point.

Analysis PI Control:

- PI has a fast response time compared to I control due to addition of the kp i.e proportinal control.
- There is an increase in rise time andit also stops the system from fluctuating.
- I part contributes in decreasing peak overshoot and bringing steady state error to 0 and returns system to set point.

Analysis PID Control:

- Kp allows a reduction in rise and peak time thereby givng fast response.
- Ki helps in brnging steady state error to 0.
- Kd helps in keeping oscillations the same i.e it preserves the oscillatory nature by maintaining zeta.

4.2 Conclusions:

- P-only controls system speed but increases steady state response & peak overshoot
- I-only controller found to be unstable due to absense of set-point
- D-only controller id also unstable as it has no set point, hence doesnt exist
- PI controller has faster respponse than I controller as well as increases rise time and return to set-point.

- PID Controller has th best fast response ,0 steady state error minimum peak overshoot and lowsettling time.

Experiment-5:

Problem statement: Write a program to evaluate Routh Stability Criterion using MATLAB. Also include the special cases.

Mathematical formulation: The equation under consideration is

$$3x^4 + 5x^3 + 3x^2 + 4x + 1$$

Then the input array becomes [3 5 3 4 1]

Code:

```
%%routh stabiliity criteria
clear;
clc;
close all;

CharEqn = input('Enter Characteristic equation array [an ...
    an-1...a0]: ');
CharLen = length(CharEqn);

i = mod(CharLen,2);
if i==0
    % Routh matrix in case of even
    a=zeros(1,(CharLen/2));
    b=zeros(1,(CharLen/2));
    for i=1:(CharLen/2)
        b(i)=CharEqn(2*i);
        a(i)=CharEqn((2*i)-1);
    end
else
    % Routh matrix in case of odd
    newMatrix=[CharEqn 0];
    a=zeros(1,((CharLen+1)/2));b=[zeros(1,((CharLen-1)/2)),0];
    for i=1:((CharLen+1)/2)
        b(i)=newMatrix(2*i);
        a(i)=newMatrix((2*i)-1);

    end
end
% Remaining rows
CharLen1=length(a);
RouthMat=zeros(CharLen,CharLen1);
epsilon = 0.0001;
RouthMat(1,:)=a;
RouthMat(2,:)=b;
```

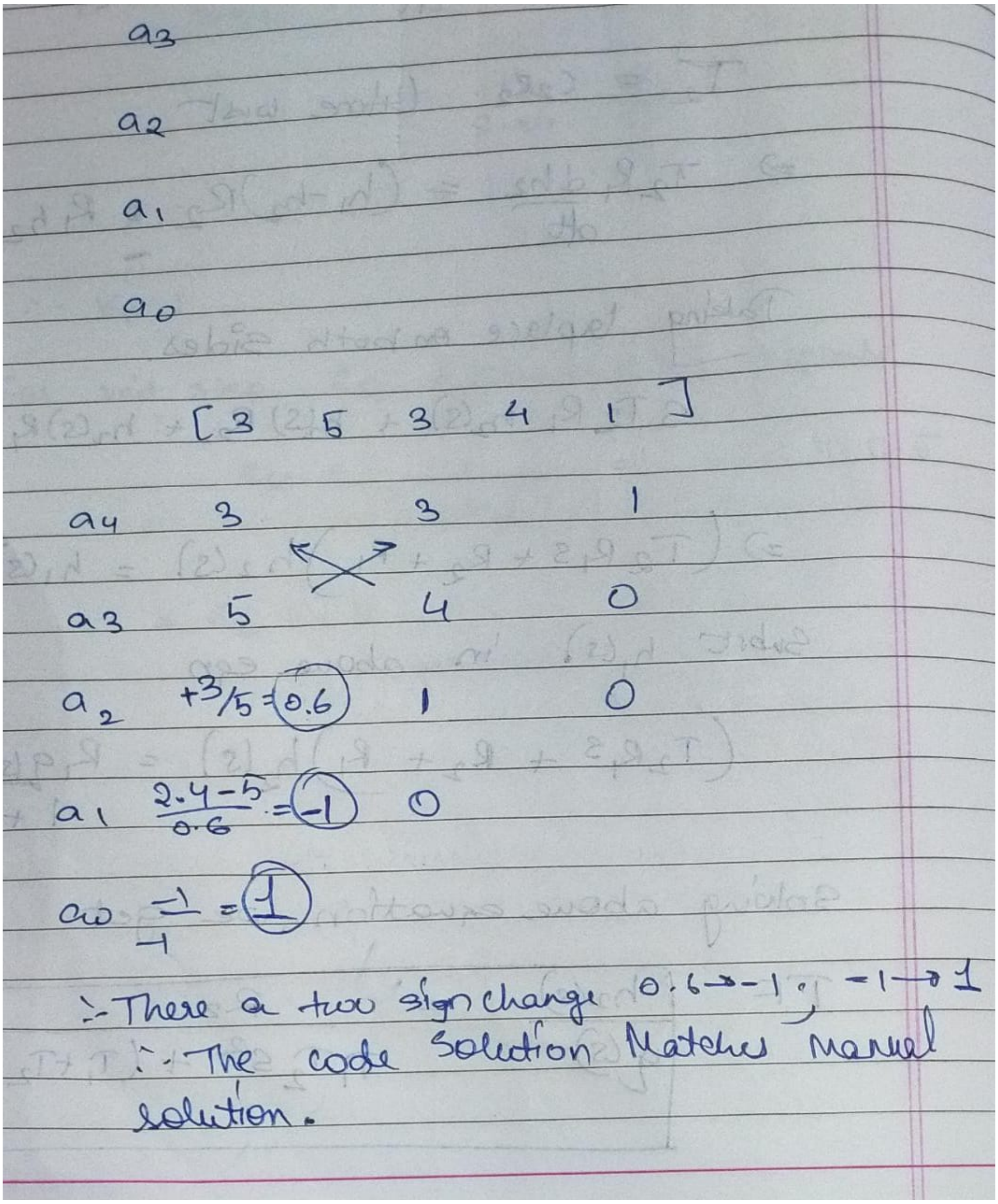
```
for i=3:CharLen
    %In case all rows are zero
    if RouthMat(i-1,:) == 0
        order = (CharLen - i);
        c11 = 0;
        c22 = 1;
        for j = 1:CharLen1 - 1
            RouthMat(i-1,j) = (order - c11) * RouthMat(i-2,c22);
            c22 = c22 + 1;
            c11 = c11 + 2;
        end
    end
    %  compute each element of the table
    for j=1:CharLen1-1
        RouthMat(i,j)=-(1/RouthMat(i-1,1))*det([RouthMat((i-2),1) ...
            RouthMat((i-2),(j+1));RouthMat((i-1),1) ...
            RouthMat((i-1),(j+1))]);
    end
    %Incase 0 in first column
     if RouthMat(i,1) == 0
        RouthMat(i,1) = epsilon;
    end

end
disp('Calculated Routh Matrix is :')
disp(RouthMat)
% check the stablity of system
if RouthMat(:,1)>0
    disp('System is found to be Stable')
else
    disp('System is found to be Unstable');
end
```

Results:

```
Command Window
Enter Characteristic equation array [an an-1...a0]: [3 5 3 4 1]
Calculated Routh Matrix is :
    3.0000    3.0000    1.0000
    5.0000    4.0000         0
    0.6000    1.0000         0
   -4.3333         0         0
    1.0000         0         0
System is found to be Unstable
>> |
```

Figure 28: Routh table result

Manual Solution: below



a_3

a_2

a_1

a_0

[3 5 3 4 1]

a_4	3	3	1
a_3	5	4	0
a_2	$+3/5 = 0.6$	1	0
a_1	$\frac{2.4-5}{0.6} = -1$	0	
a_0	$\frac{-1}{-1} = 1$		

∴ There a two sign change $0.6 \rightarrow -1$; $-1 \rightarrow 1$

∴ The code solution Matches manual solution.

Figure 29: Manually verified routh criteria

Analysis Conclusions:

- Hence we were able to build a Routh table from created function
- The evaluated characteristic equation had 2 sign changes and hence had 2 roots in the +ve axis,, as a result the characteristic equation had 2 real positive roots making the system unstable.

Hence we analysed the second order differential equation of motion of simple pendulum using ODE solver.

Experiment-6:

Problem statement: Do the stability analysis using root locus plot and find the range of gain K for stable system. Also find gain value of marginal stability (Hints: Refer Exp 5) Consider a process controlled by a proportional controller as shown in Figure given above

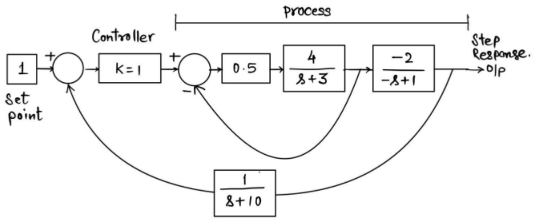


Figure 30: Exp6

Code:

```
clc;
clear all;
close all;
% Block reduction
a_1 = tf([0.5],[1]);
b_1 = tf([4],[1 3]);
c_1 = tf([-2],[-1 1]);
h_1 = tf([1],[1 10]);

K_p = 1; % Defining Kp
gg = series(K_p,series(feedback(series(a_1,b_1),1),c_1));
system = feedback(gg,h_1)

flaggg = 1;

% Finding the lower limit of Kp for the system to be stable
while flaggg == 1
        system = feedback(series(K_p,series(feedback
                ...(series(a_1,b_1),1),c_1)),h_1);
        poles = pole(system);
        if real(poles(1)) ≤ 0 && real(poles(2)) ≤ 0 && ...
            real(poles(3)) ≤ 0
            lower_value = K_p;
            flaggg = 0;
        end
```

```
        K_p = K_p + 1;
end

% Finding the upper limit of Kp for the system to be stable
while real(poles(1)) ≤ 0 && real(poles(2)) ≤ 0 && real(poles(3)) ...
    ≤ 0
    K_p = K_p + 1;
    system = feedback(series(K_p,series(feedback...
            (series(a_1,b_1),1),c_1)),h_1);
    poles = pole(system);

end

disp(lower_value);disp(K_p);
rlocus(system)
```

Results: On verifying the stability by using function in exp 5 for k = 135 we see that the system is now stable

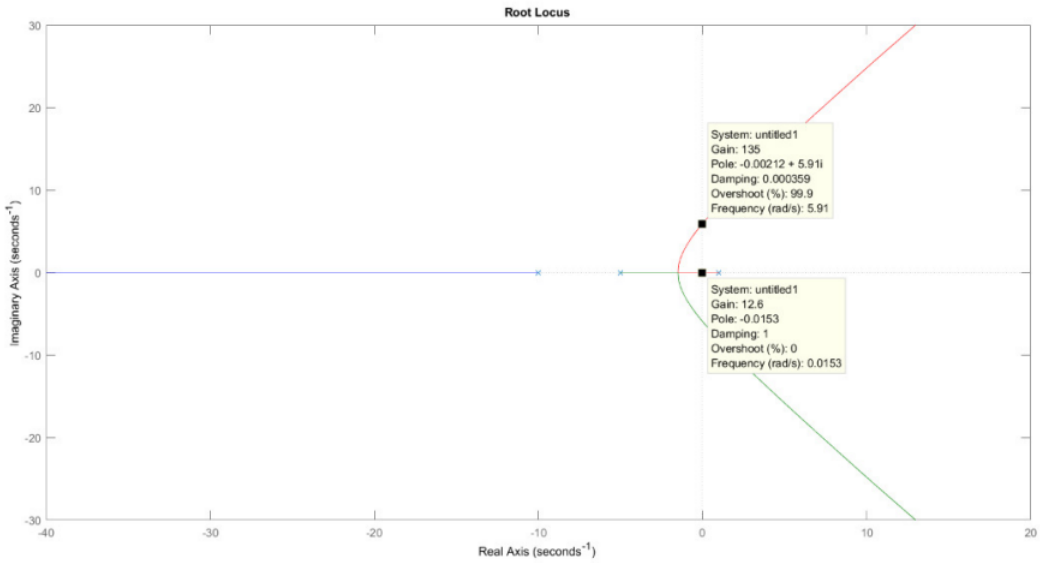
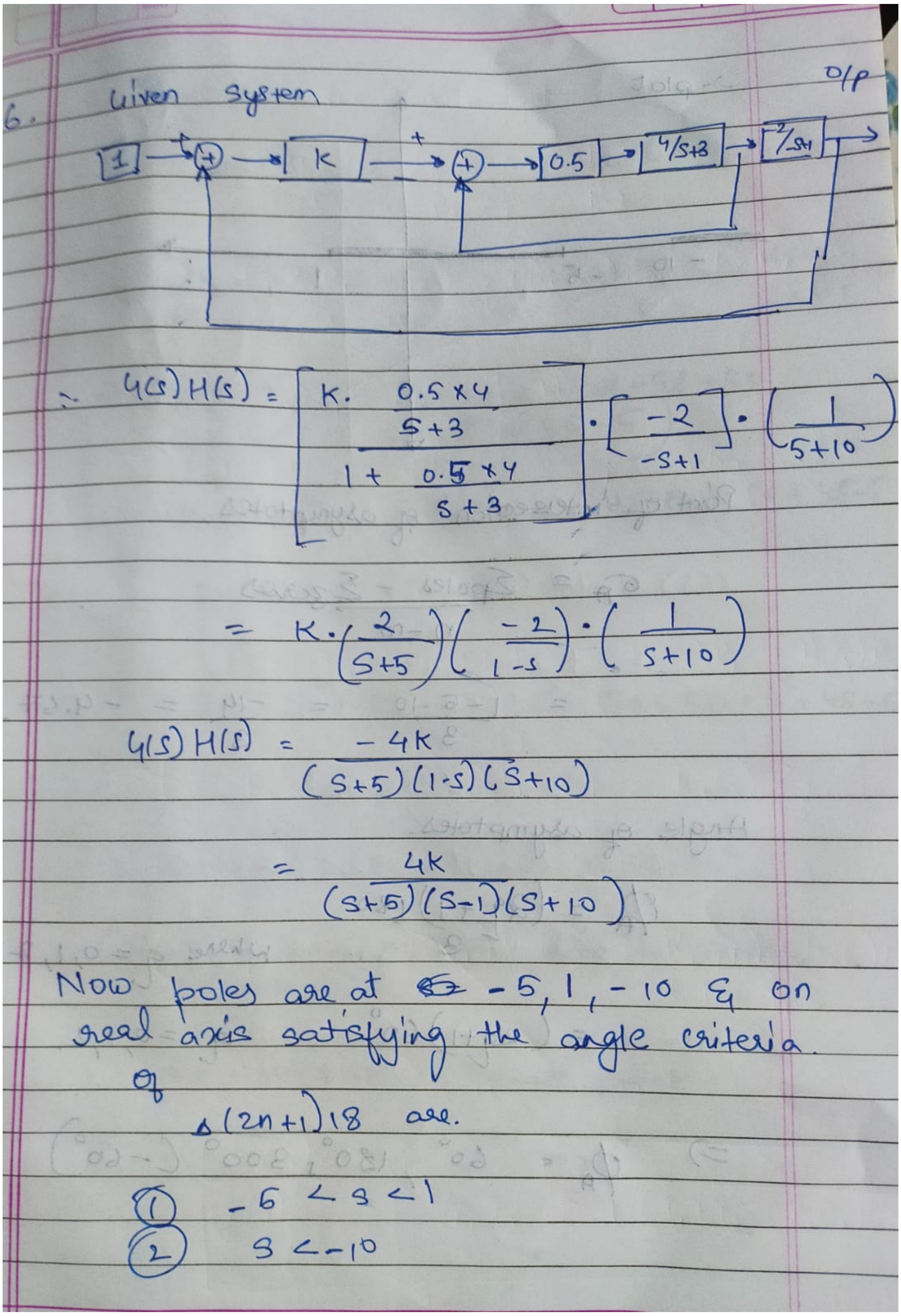


Figure 31: Routh table result

```
Command Window
Enter Characteristic equation array [an an-1...a0]: [1 14 35 490]
Calculated Routh Matrix is :
     1.0000    35.0000
    14.0000   490.0000
     0.0001          0
   490.0000          0
System is found to be Stable
>> |
```

Figure 32: Routh table result

Manual Solution:



6. Given system

$$\therefore G(s)H(s) = \left[\frac{K \cdot \frac{0.5\times 4}{s+3}}{1+\frac{0.5\times 4}{s+3}}\right]\cdot\left[\frac{-2}{-s+1}\right]\cdot\left(\frac{1}{s+10}\right)$$

$$= K\cdot\left(\frac{2}{s+5}\right)\left(\frac{-2}{1-s}\right)\cdot\left(\frac{1}{s+10}\right)$$

$$G(s)H(s) = \frac{-4K}{(s+5)(1-s)(s+10)}$$

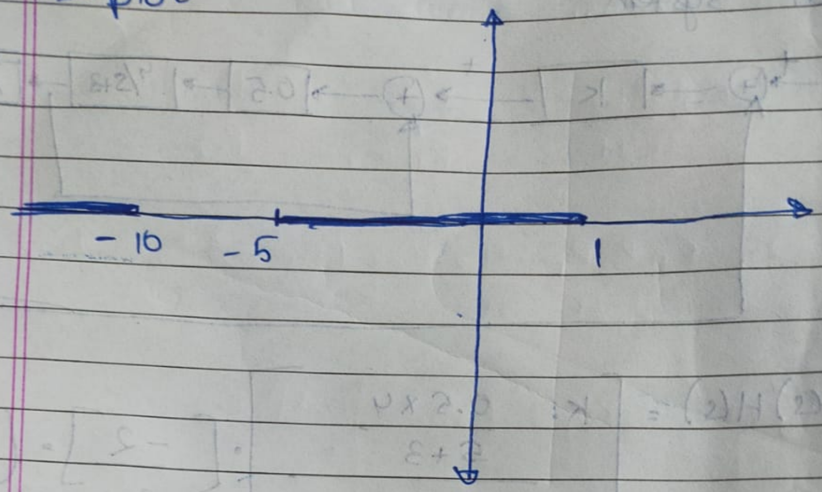
$$= \frac{4K}{(s+5)(s-1)(s+10)}$$

Now poles are at $-5, 1, -10$ & on real axis satisfying the angle criteria of $\pm(2n+1)18$ are.

1. $-6 < s < 1$
2. $s < -10$

Figure 33: Motion of simple pendulum

∴ plot



Point of intersection of asymptotes,

$$\sigma_A = \frac{\sum poles - \sum zeros}{n-m}$$

$$= \frac{1-5-10}{3} = \frac{-14}{3} = -4.67.$$

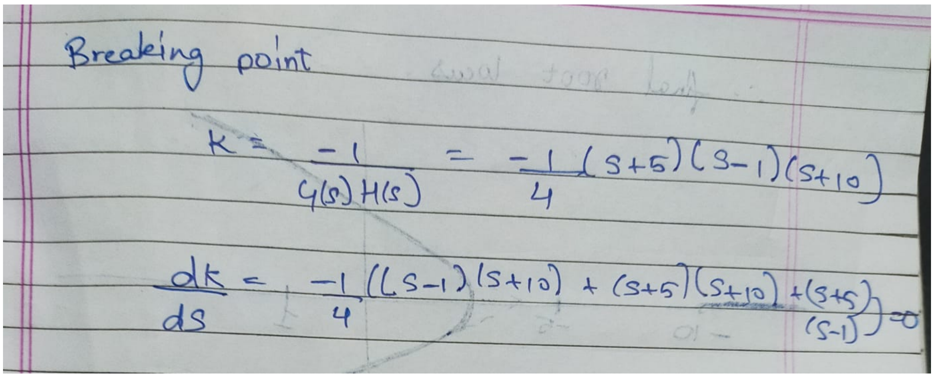
Angle of asymptotes.

$$\phi_A = \frac{(2q+1)180}{3} \quad \text{where } q = 0,1,2.$$

$$= (2q+1)60^\circ \quad q = 0,1,2.$$

$$\Rightarrow \phi_A = 60^\circ, 180^\circ, 300^\circ \ (-60^\circ)$$

Figure 34: Motion of simple pendulum

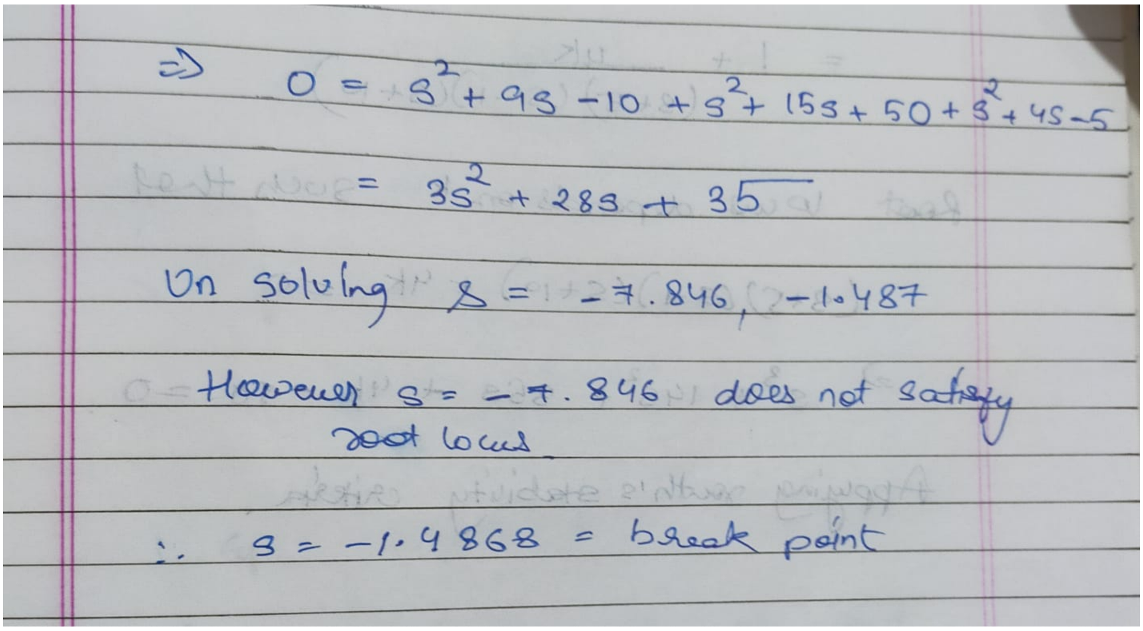


Breaking point

$$K = \frac{-1}{G(s)H(s)} = \frac{-1}{4}(s+5)(s-1)(s+10)$$

$$\frac{dK}{ds} = \frac{-1}{4}\left((s-1)(s+10) + (s+5)(s+10) + (s+5)(s-1)\right) = 0$$

Figure 35: Motion of simple pendulum



$$\Rightarrow \quad 0 = s^2 + 9s - 10 + s^2 + 15s + 50 + s^2 + 4s - 5$$

$$= 3s^2 + 28s + 35$$

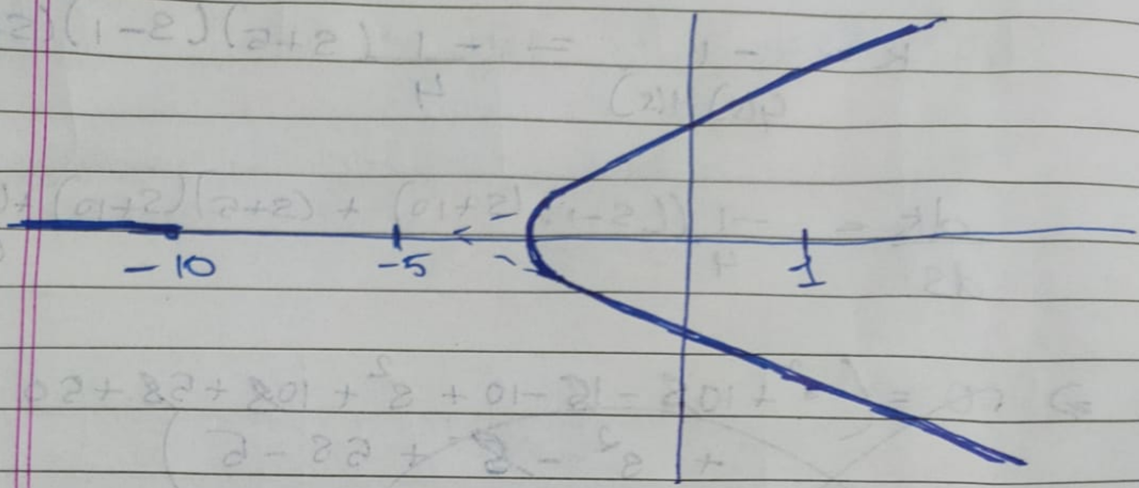
On solving $s = -7.846, -1.487$

However $s = -7.846$ does not satisfy root locus

$\therefore$ $s = -1.4868$ = break point

Figure 36: Motion of simple pendulum

∴ final root locus.



Now char eqn at close loop system

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

$$= 1 + \frac{4K}{(s+5)(s-1)(s+10)}$$

Root locus depends on k such that

$$(s+5)(s-1)(s+10) + 4K = 0$$

$$\Rightarrow s^3 + 14s^2 + 35s + 4K - 50 = 0$$

Applying routh's stability criteria.

Figure 37: Motion of simple pendulum

a^3	1	35
a^2	14	$4k-50$
a^1	$\frac{14\times35-(4k-50)}{14}$	0
a_0	$14k-50$	

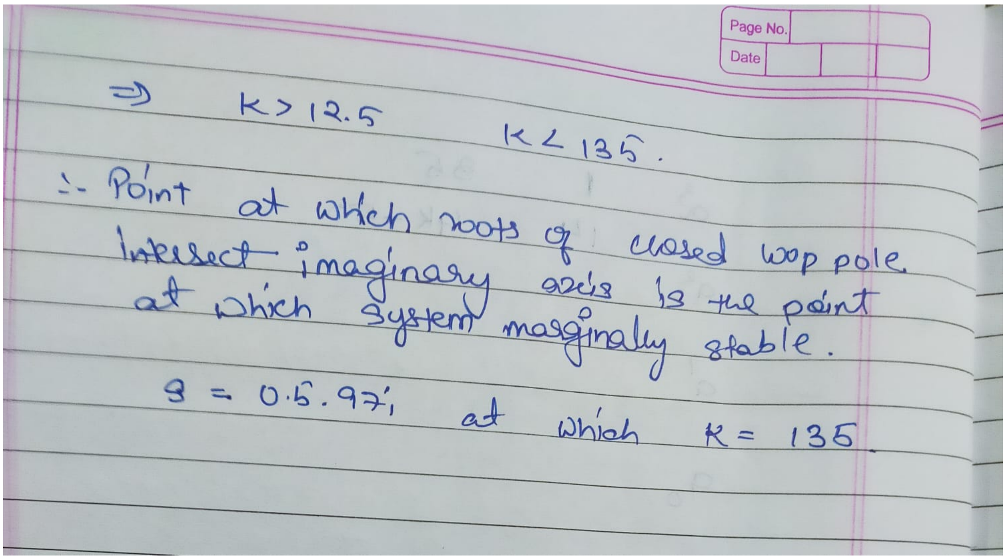
> for stability we want

$$\frac{14\times35-4k+50}{14} > 0$$

& $14k-50 > 0$.

$\Rightarrow \quad k > \frac{50}{4} \quad \& \quad k < \frac{14\times35+50}{4}$

Figure 38: Motion of simple pendulum



⇒ $K > 12.5$ $K < 135$.

∴ Point at which roots of closed loop pole intersect imaginary axis is the point at which system marginally stable.

$s = 0.5.97j$, at which $K = 135$

Figure 39: Motion of simple pendulum

Analysis Conclusions:

- Hence we were able to build a Routh table from created function
- The evaluated characteristic equation had 2 sign changes and hence had 2 roots in the +ve axis,, as a result the characteristic equation had 2 real positive roots making the system unstable.

Hence we analysed the second order differential equation of motion of simple pendulum using ODE solver.