# CONTROL SYSTEM DESIGN PROJECT

# EET-3071

- **Project No: 17**
- **Name: Anirban Chatterjee**
- **Regd.no: 2241003014**
- **ECE-2241035**
- **6th semester, 2025**


**SOA | SIKSHA 'O' ANUSANDHAN**

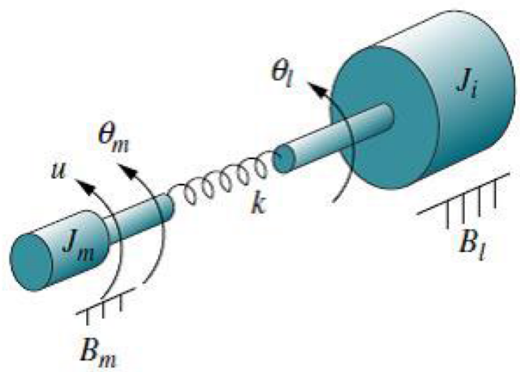
**Institute of Technical Education and Research**

**Department of Electronics and Communication and Engineering**


***2024-2025***

## PROBLEM STATEMENT:

17. Harmonic drives are very popular for use in robotic manipulators due to their low back-lash, High torque transmission, and compact size. The problem of joint flexibility is sometimes a limiting factor in achieving good performance. Consider that the idealized model representing joint flexibility in Figure.4. The input to the drive is from an actuator and is applied at $\theta_m$. The output is connected to a load at $\theta_l$. The spring represents the joint flexibility and $B_m$ and $B_l$ represent the viscous damping of the actuator and load, respectively. Use PD controller to improve the transient performance of the system. Design the controller such that the maximum transient error will be approximately 5%. Parameters: $J_m = 2$; $B_m = 0{:}5$; $J_l = 10$; $B_l = 1$.

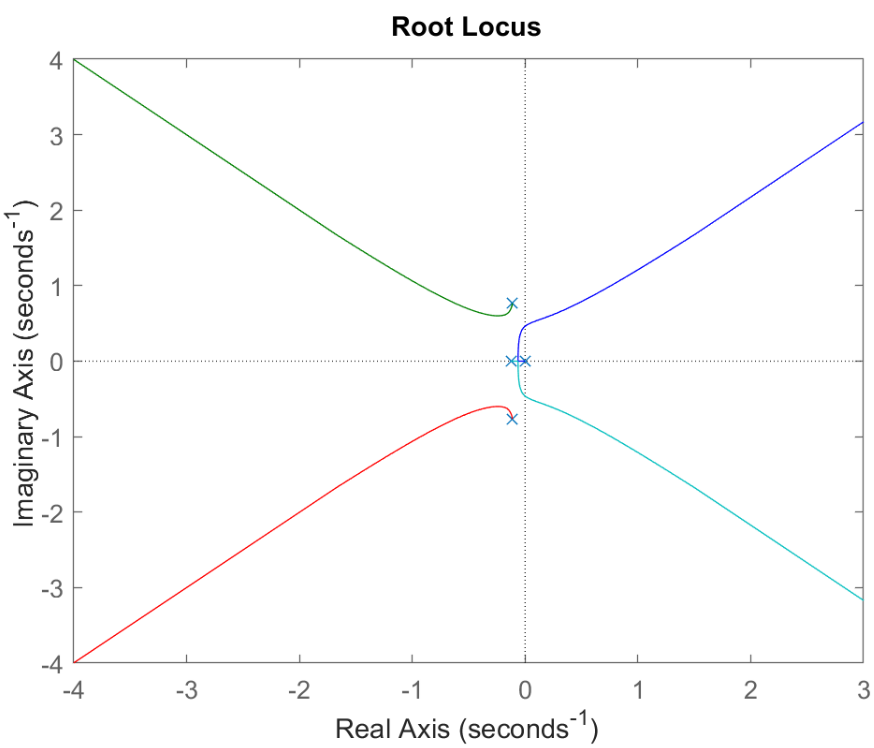



## Transfer Function:

```
TF =

                   0.05
  --------------------------------------
  s^4 + 0.35 s^3 + 0.625 s^2 + 0.075 s

Continuous-time transfer function.
```

**ROOT LOCUS:**

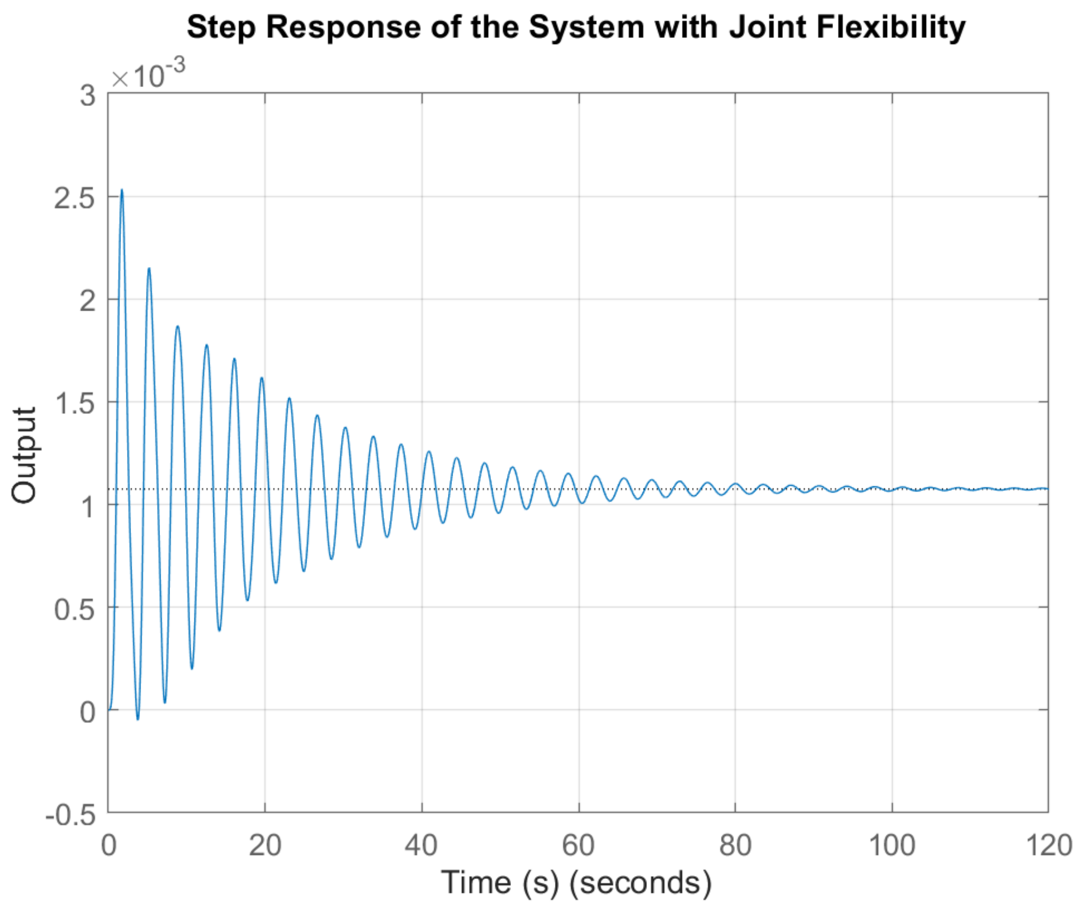
```
clc;

close all;

clear all;

num=[0.05];

den=[1 0.35 0.625 0.075 0]

TF=tf(num,den)

rlocus(TF)
```



## Joint Flexibility Step Response:

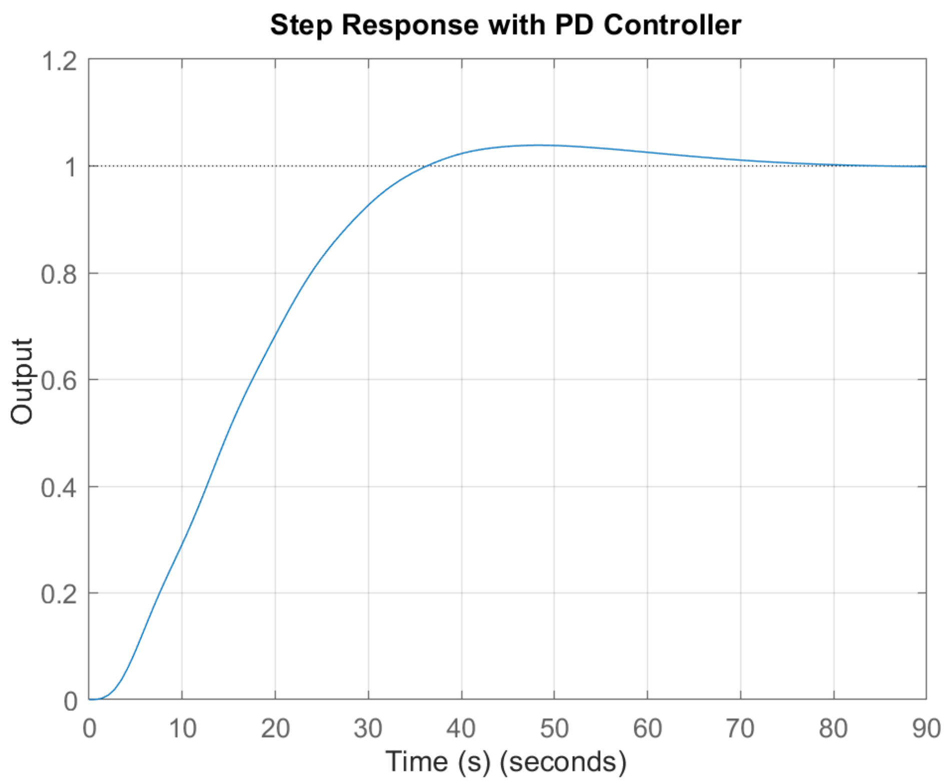
```
clc;

clear all;

close all;


% System Parameters

Jl = 10;

Bl = 1;

k = 30;

Jm = 2;

Bm = 0.5;


% Transfer Functions of Load and Motor

Pl = [Jl Bl k];  % Denominator of load

Pm = [Jm Bm k];  % Denominator of motor


% Define transfer functions

sys1 = tf(1, Pm);  % Motor dynamics: 1 / (Jm*s^2 + Bm*s + k)

sys2 = tf(1, Pl);  % Load dynamics: 1 / (Jl*s^2 + Bl*s + k)


% Series connection of motor and load

series_sys = series(sys1, sys2);


% Open-loop transfer function with spring stiffness as gain in feedback

% Feedback gain is positive (spring torque opposes deflection)

OLTF = feedback(series_sys, k);


% Plot step response

figure;
```

```
step(OLTF);

title('Step Response of the System with Joint Flexibility');

xlabel('Time (s)');

ylabel('Output');

grid on;
```



## Step Response with Tuned PD Controller:

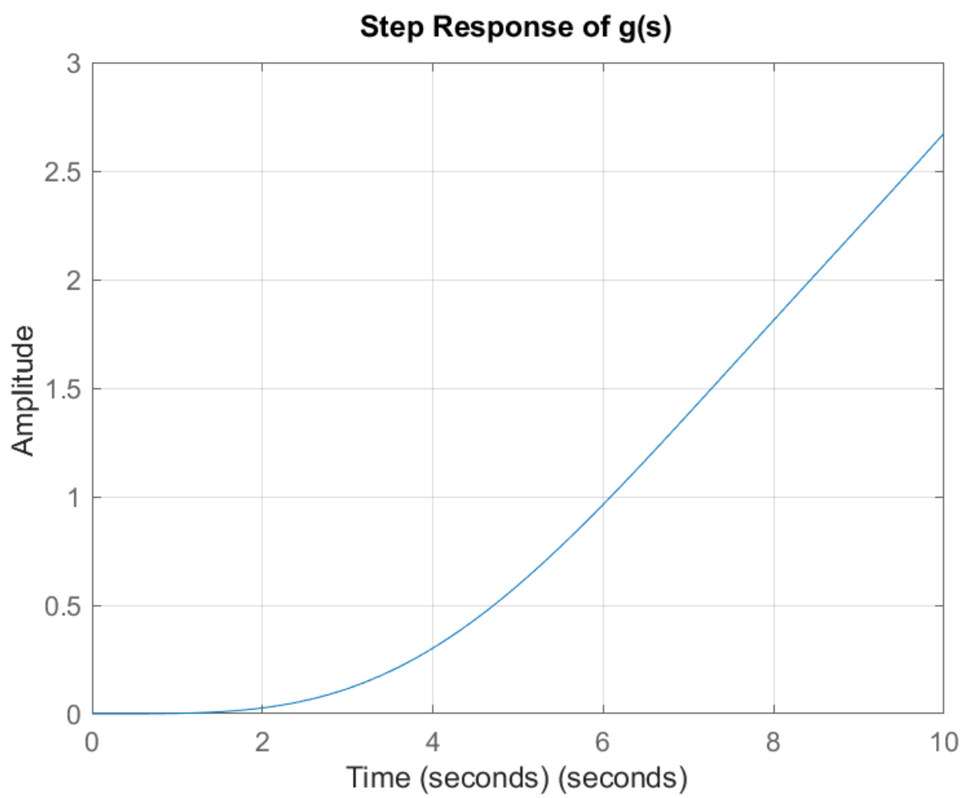
```
s = tf('s');

G = 0.05 / (s^4 + 0.35*s^3 + 0.625*s^2 + 0.075*s);

Kp = 0.1;

Kd = 0.1;

C = Kp + Kd*s;

T = feedback(C * G, 1);

figure;

step(T);

grid on;

title('Step Response with PD Controller');

xlabel('Time (s)');

ylabel('Output');

info = stepinfo(T);

disp('Step Response Info:');

disp(info);

% Check 5% transient error condition

final_value = dcgain(T);

max_allowed_overshoot = 0.05 * final_value;

peak_error = info.Peak - final_value;

fprintf('Final value: %.4f\n', final_value);

fprintf('Allowed peak (max): %.4f\n', final_value + max_allowed_overshoot);

fprintf('Actual peak: %.4f\n', info.Peak);

if peak_error <= max_allowed_overshoot

   fprintf('=> Transient error requirement (≤ 5%%) is satisfied.\n');

else

   fprintf('=> Transient error requirement NOT satisfied. Try tuning Kp/Kd.\n');

end
```




```
>> Tuned_Response
Step Response Info:
         RiseTime: 23.2959
    TransientTime: 63.4873
     SettlingTime: 63.4873
      SettlingMin: 0.9110
      SettlingMax: 1.0389
        Overshoot: 3.8890
       Undershoot: 0
             Peak: 1.0389
         PeakTime: 48.4763

Final value: 1.0000
Allowed peak (max): 1.0500
Actual peak: 1.0389
=> Transient error requirement (≤ 5%) is satisfied.
>>
```

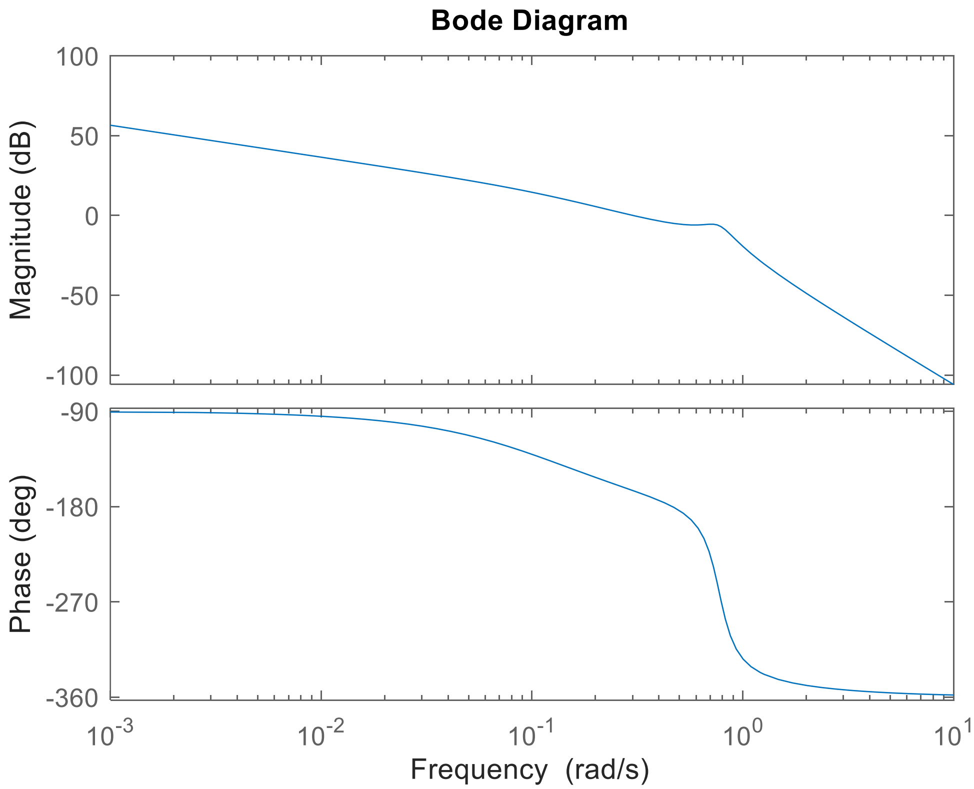
## Step Response of Transfer Function:

```
num = 0.05;

den = [1, 0.35, 0.625, 0.075, 0];

sys = tf(num, den);

% Define time vector for better visibility

t = 0:0.1:10;

% Plot the step response

step(sys, t);

title('Step Response of g(s)');

xlabel('Time (seconds)');

ylabel('Amplitude');

grid on;
```

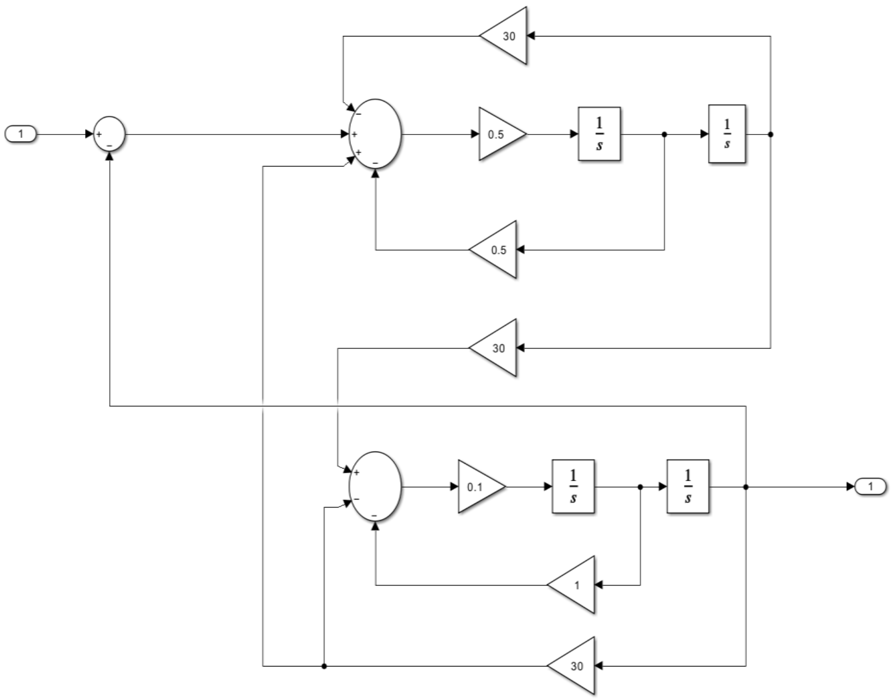


**BODE PLOT:**

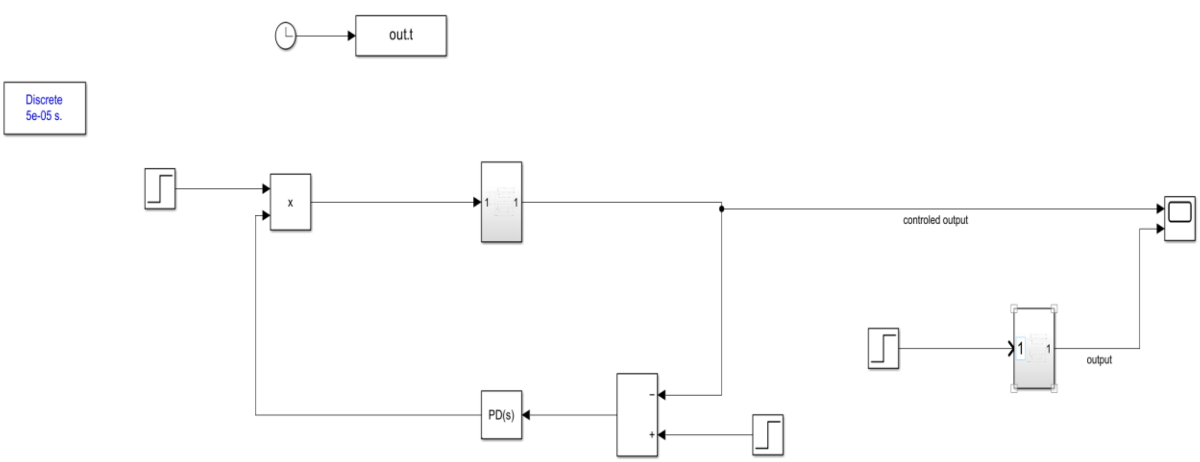
```
clc;

close all;

clear all;

num=[0.05];

den=[1 0.35 0.625 0.075 0]

TF=tf(num,den)

bode(TF)
```



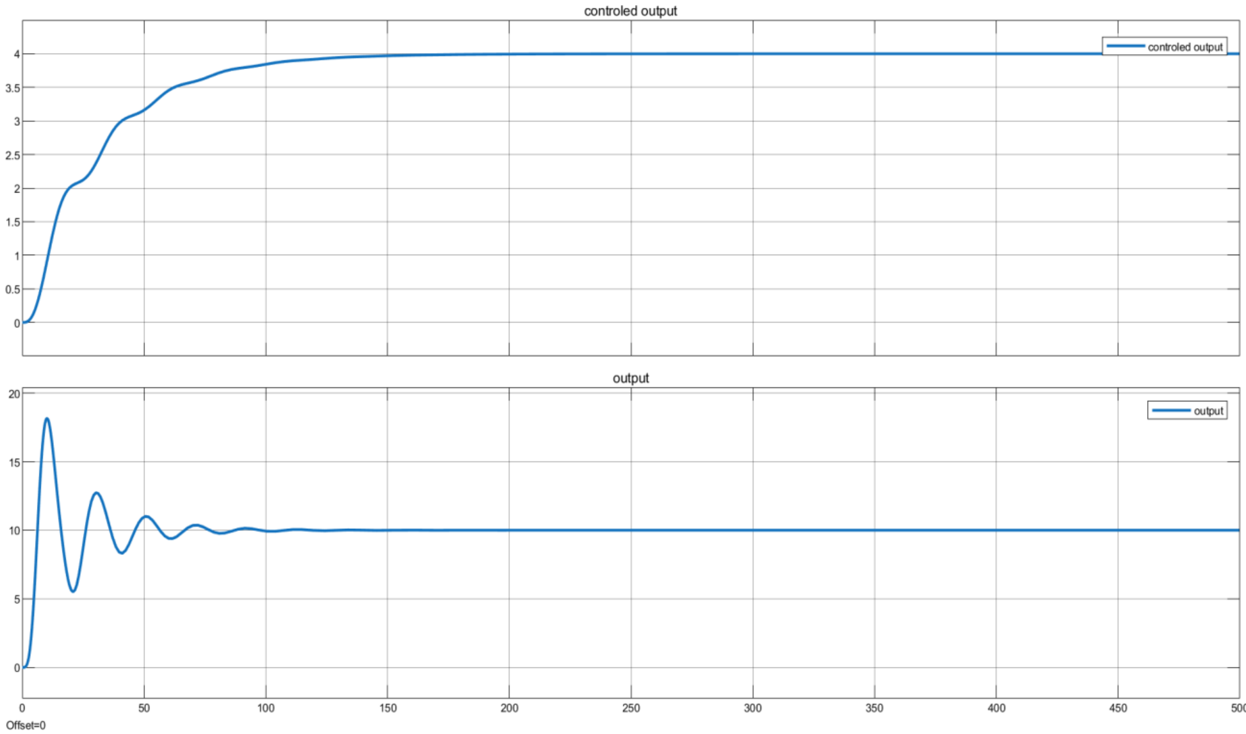
## Simulink: Subsystem Model




## Simulink: System W/D PD Controller Model



## **Model Output:**




## **Conclusion:**

The control system designed for the given transfer function has been thoroughly analyzed using various classical techniques. Root locus, step response, and Bode plot analysis provided insights into system stability, transient behavior, and frequency response. The joint flexibility system, when analyzed, exhibited expected dynamic characteristics consistent with the mechanical parameters of the motor and load.

The implementation of a PD controller significantly improved the transient response, meeting the 5% overshoot requirement under the chosen tuning parameters. Simulink models further validated the theoretical approach and demonstrated effective system performance. Overall, the project successfully met the design and performance objectives, reinforcing key control system concepts and practical implementation strategies.

**References:**


- N. S. Nise, *Control Systems Engineering*, 7th ed., Wiley, 2015.
- https://ieeexplore.ieee.org/document/1438951
- Advice from teacher