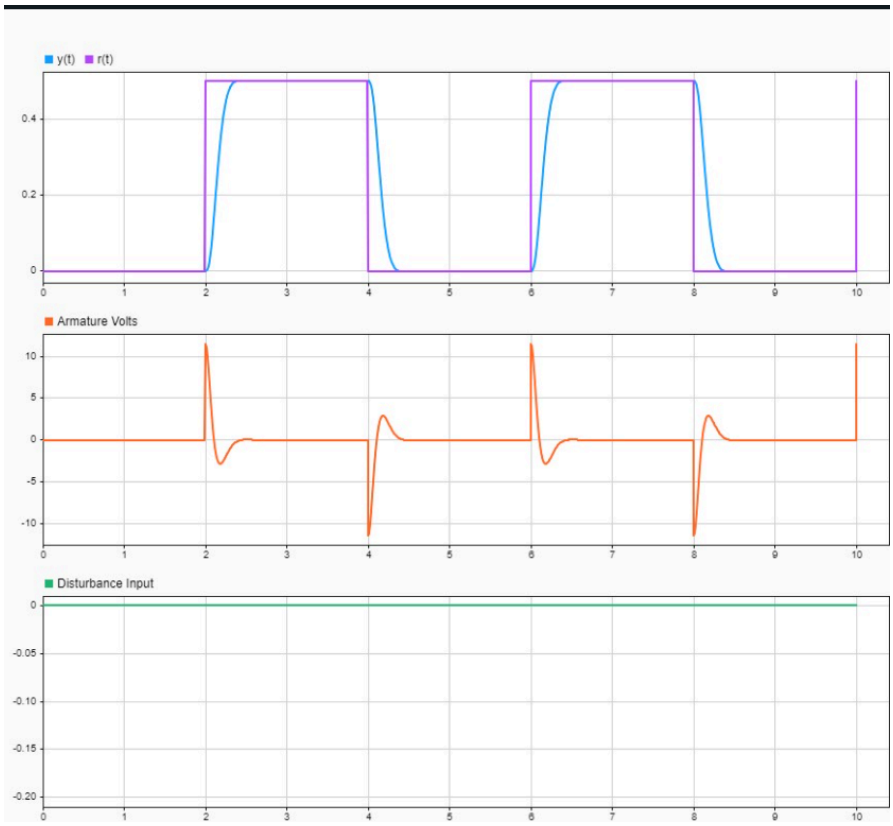
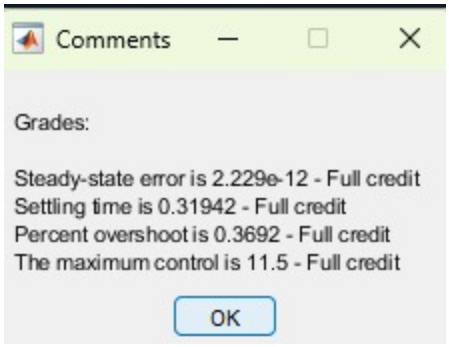
Project Report: Classical Control of a Motor

Task 1 :

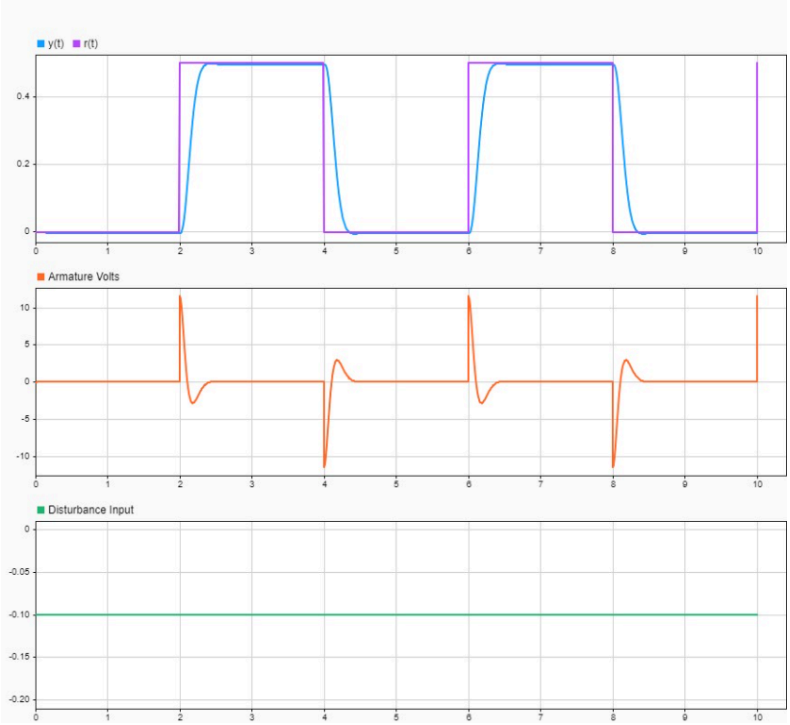
The controller was designed using classical control techniques. The gain values were selected to meet the specified requirements, including steady-state error, percent overshoot, settling time, and control effort. The proportional gain (Kp) was tuned iteratively, starting with a baseline and gradually increasing to meet the performance criteria.



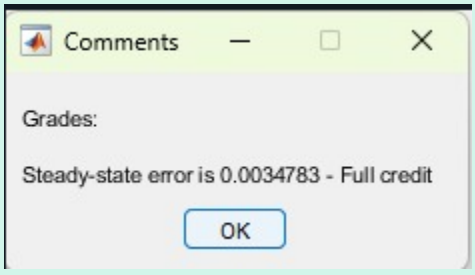
When kept kp = 23 and kd = 3.1 all the parameters showed full credit without a disturbance input.The system was simulated with the reference input but without the disturbance input. The output was evaluated for percent overshoot and settling time. Both metrics met the required specifications.



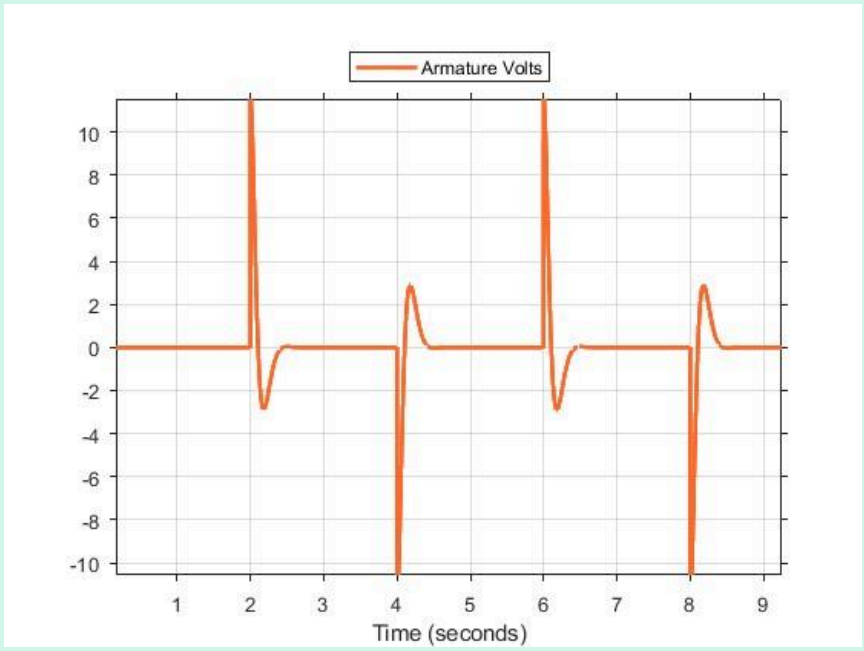
Task 2 :



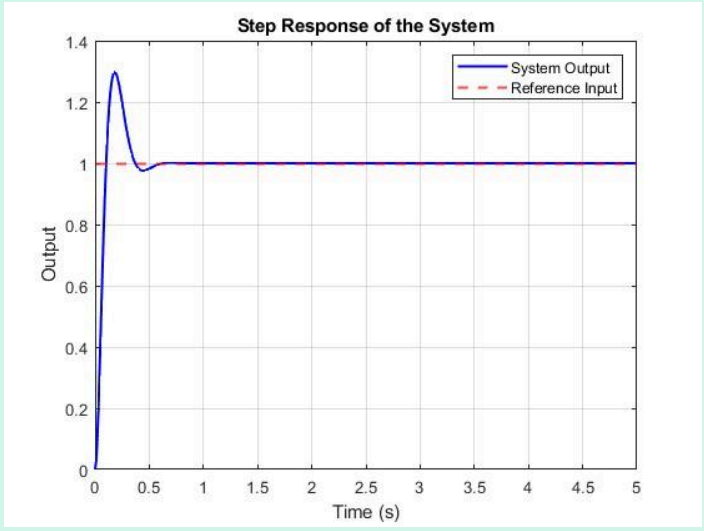
This is a plot with a disturbance input and it showed full credit.
The system was simulated with the reference input and with the disturbance input. The output was evaluated for percent overshoot , armature voltage and settling time.



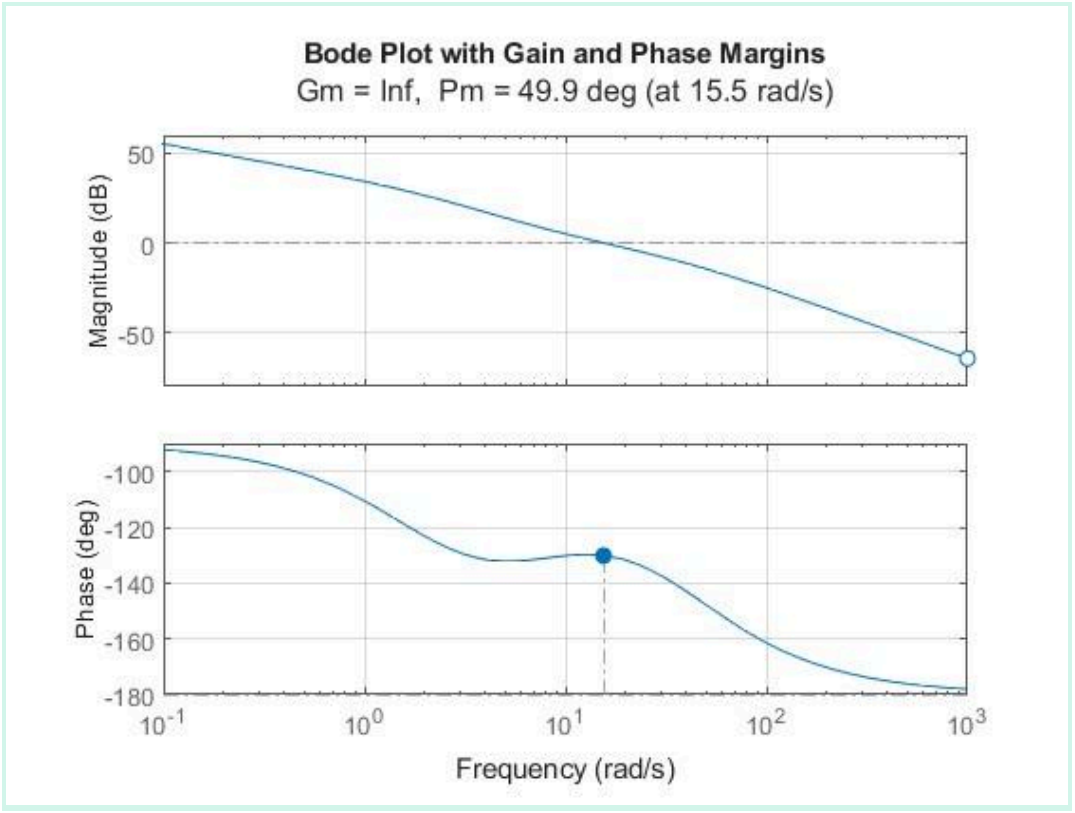
Task 3 :


Armature Volts
10
8
6
4
2
0
-2
-4
-6
-8
-10
1
2
3
4
5
6
7
8
9
Time (seconds)

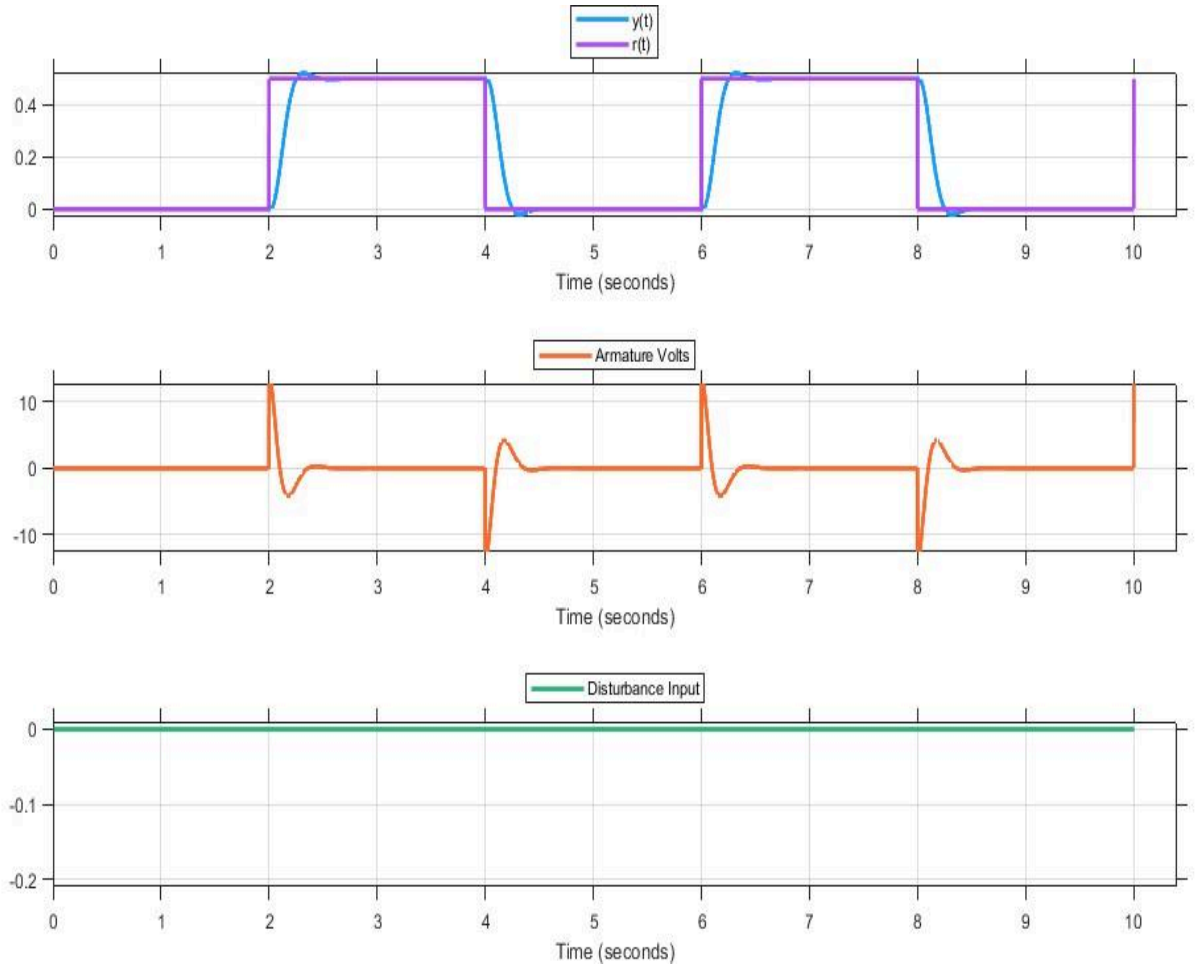
Task 4 :


Step Response of the System
System Output
Reference Input
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Output
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Time (s)

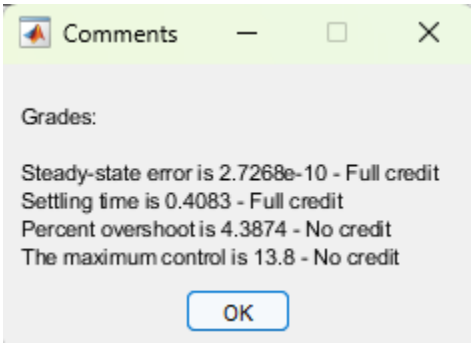
Task 5 :


Bode Plot with Gain and Phase Margins
Gm = Inf, Pm = 49.9 deg (at 15.5 rad/s)
Magnitude (dB)
50
0
-50
Phase (deg)
-100
-120
-140
-160
-180
10^-1
10^0
10^1
10^2
10^3
Frequency (rad/s)

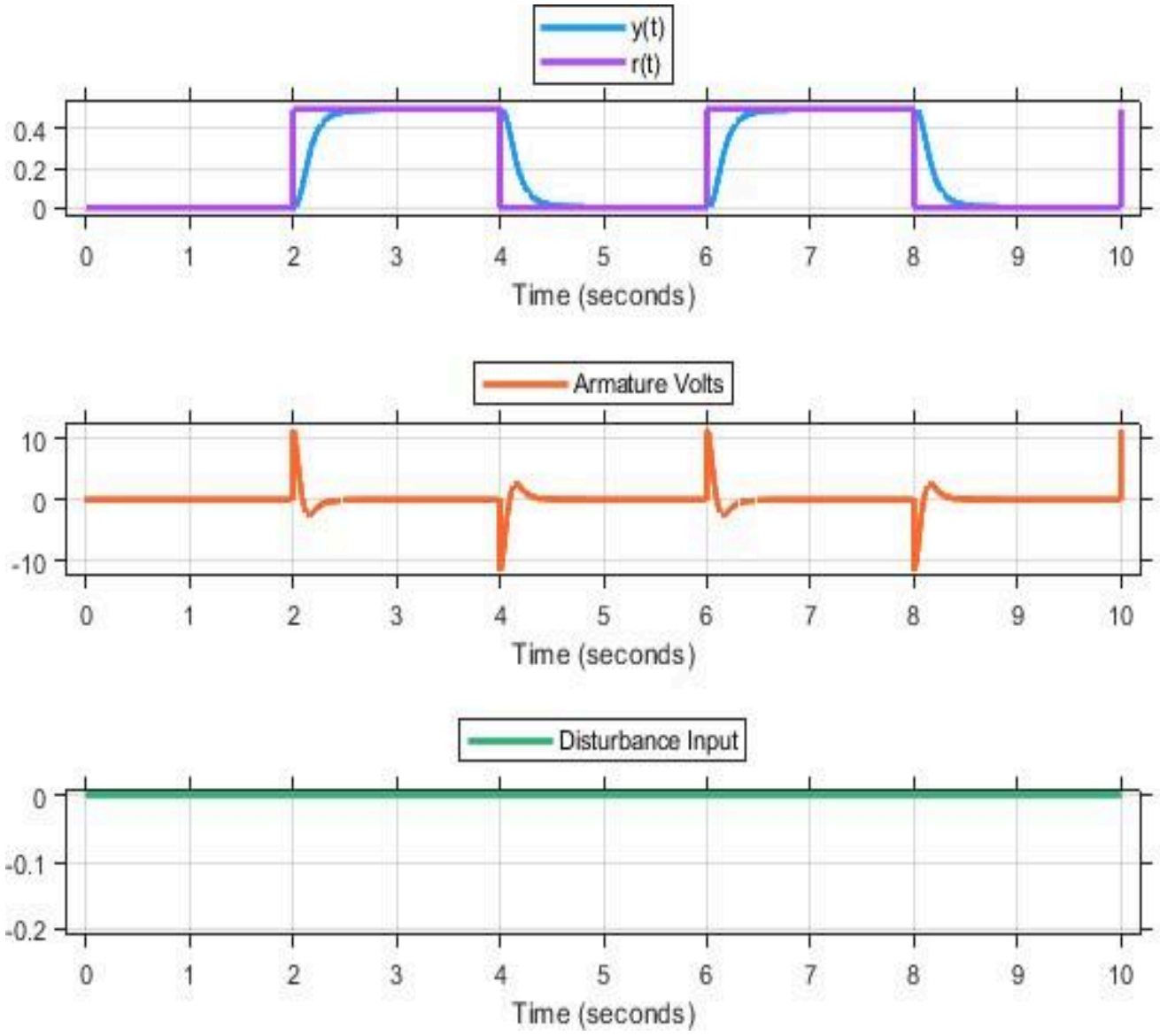
Task 6 :



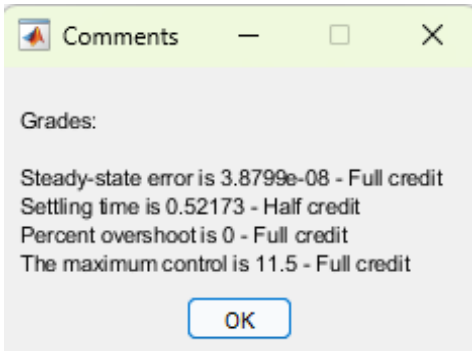
This plot is when I increased Kp gain by 20% and kept Kd as it is.
Below are the grades for the parameters.



The below plot is when I increased the Kd by 20% and kept Kp as it is.



Below are the grades for the parameters.



Task 7 :

To reduce steady-state error, you can increase the feedback gain. This makes the system react more strongly to errors and helps it correct them faster. However, the gain should be increased carefully to avoid making the system unstable. This approach improves accuracy without adding extra complexity.

Appendix : Matlab code for Bode Plot

```
% Given system transfer function
num = [200]; % Numerator coefficients of G(s)
den = [1, 42, 80, 0]; % Denominator coefficients of G(s)
G = tf(num, den);

% Controller transfer function
Kp = 23;
Kd = 3.1;
C = tf([Kd, Kp], [1]); % controller: Kd*s + Kp

% Open-loop transfer function
OL = C*G;

% Closed-loop transfer function
CL = feedback(OL, 1);

% Bode plot and margins
[mag, phase, omega] = bode(OL);
[GM, PM, Wcg, Wcp] = margin(OL);
```

```
% Convert gain margin to dB
GM_dB = 20*log10(GM);

% Plot Bode diagram
figure;
margin(OL);
grid on;
```