Assignment 03

SAURAV KUMAR

18 Feb, 2024

1 Overview of the Assignment

1.1 Aim of the Assignment

QUESTION 1: In this problem, we are calculating the gain margin and its trend and delay margin using the Matlab Nyquist plot.

Question 2: In this problem, we are designing the PID controller and tuning the given transfer characteristic.

1.2 Methods

1. MATLAB has been used for plotting purposes. Nyquist Plots and bode plots have been plotted using Matlab in this assignment.

2. We are using the Matlab and PID controller along with the hand written analytic calculations.

2 Question 1

2.1 Part 1

Yes, the overall close loop of the transfer function at T=0 is stable. We can see from the Nyquist plot characteristics.

Gain Margin =3.53 dB.

Phase Margin = 2.12 degree.
delay margin = 0.0251 s.

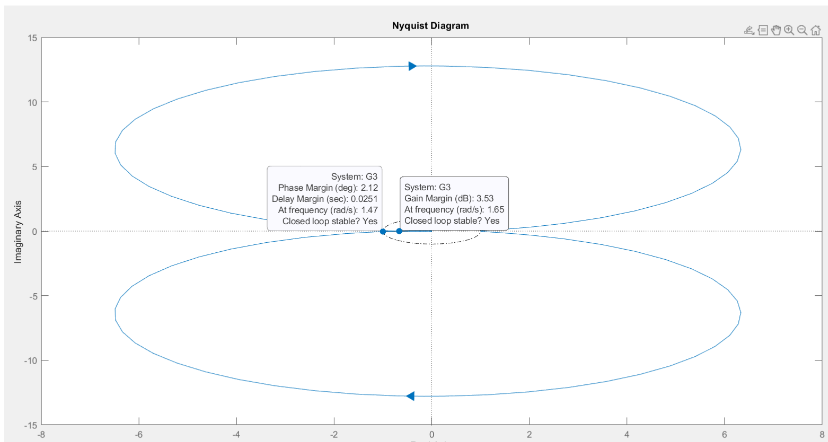


Figure 1: stability margins at T=0

2.2 Part 2

Gain Margins (GM):
GM=2.7 dB, at T=0.5
GM=1.95 dB, at T=1
GM=1.59 dB, at T=1.25

The best gain margin among given three temperature value is 2.7 dB, thus T=0.5 possess best stability margin.

Matlab Code:

```
clc
clear all

s = tf('s');

%defining the given values
R=1;
```

```
L=12.25;
C= 0.075;
td = 0.05 ;
T0 = 5;
T=0;
%T=0.5;
%T=1;
%T=1.25;

% trasnsfer function of plant
NUM1 = [1];
DEN1 = [L*C, R*C, 1];
G1= tf(NUM1, DEN1);

% trasnsfer function of sensor
G2 = (1+T/T0)*exp(-s*td);

%combined trasfer function
G3 = G1*G2;

%plotting the nyquist plot
nyquist(G3)
```

2.3 part3

Gain Margins (GM), Phase Margin(PM), Delay Margin(DM):
GM=3.53 dB, PM=2.12 degree, DM = 0.0251 s, at T=0
GM=2.7 dB, PM=1.58 degree, DM = 0.0183 s, at T=0.5
GM=1.95 dB, PM=1.11 degree, DM = 0.0126 s, at T=1
GM=1.59 dB, PM=0.903 degree, DM = 0.0101 s, at T=1.25

We can clearly note that stability margins are decreasing with increase in the temperature values. Thus, it is reasonable to conclude that increasing operating ambient temperatures tend to push the given closed-loop feedback control system toward instability.

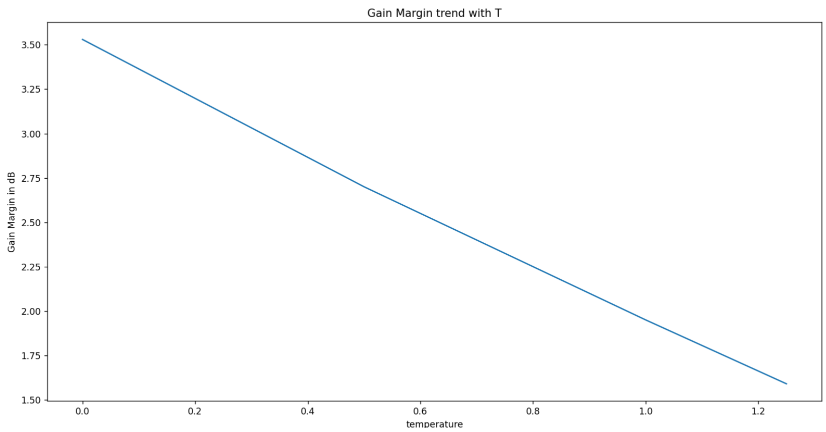


Figure 2: Gain Margins Trend

2.4 Part 4

At T=0, Delay Margin = 0.0251 s.

This means 0.0251 s is the maximum delay that can be accommodated, after which the feedback control system fails if it has to operate at T = 0.

The trend is decreasing with increase in temperature as we can see from the data of stability margins in part 3.

3 Quesion 2

3.1 part 1

PID coefficients that damps step disturbance in 2 second:
Kp = 43
Ki = 48
Kd = 12.4

Matlab Code used:

```
clc
clear all
```

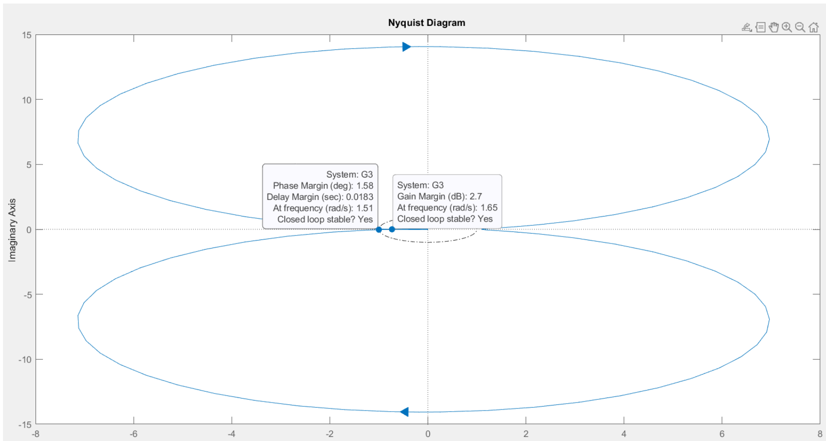


Figure 3: stability margins at T=0.5

```
%defining our PID
Kp = 43;
Ki = 48;
Kd = 12.4;

s = tf('s');
C = Kp + Ki/s + Kd*s ;

%Given trasfer funtion
NUM1 = [1];
DEN1 = [1, 3.6, 9];
G1= tf(NUM1, DEN1);

% close loop trasfer function R to C
G2 =G1*C;
T = feedback(G2, 1);

%plotting step response
step(T)
```

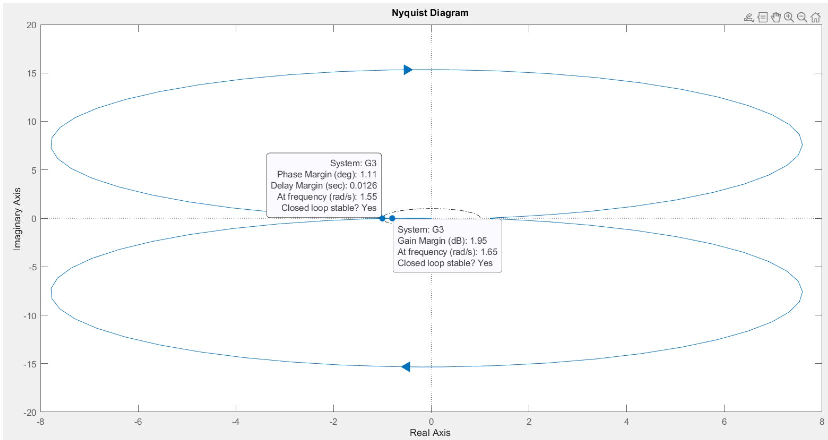


Figure 4: stability margins at T=1

3.2 Part 2

Manual tuning:
Kp: 43
Ki: 48
Kd: 12.4

Ziglar Nicholas Method:
Kp: 61.22
Ki: 4.234
Kd: 0.059

PID auto tuner:
Kp: 18.2470
Ki: 35.0773
Kd: 2.3730

Matlab Code used:

```
clc
clear all

%defining our PID
```

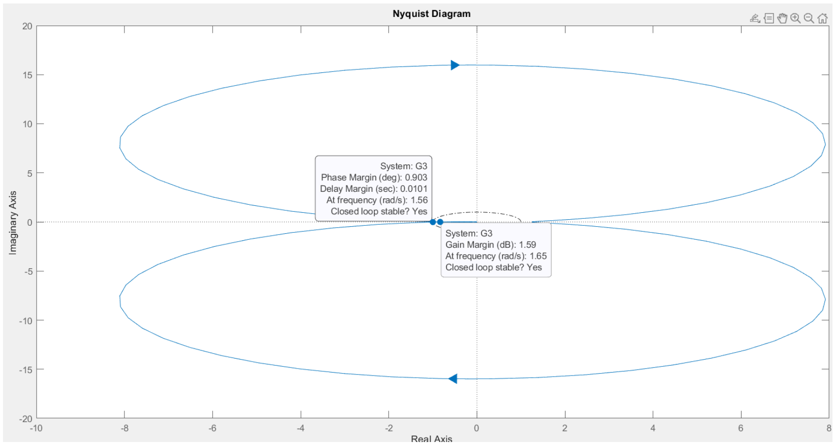


Figure 5: stability margins at T=1.25

```
Kp = 43;
Ki = 48;
Kd = 12.4;

s = tf('s');
C = Kp + Ki/s + Kd*s ;

%Given trasfer funtion
NUM1 = [1];
DEN1 = [1, 3.6, 9];
G= tf(NUM1, DEN1);

% D to C tranfer function
G1 = feedback(G, C);
%step(G1);

% R to C tranfer func
G2 = feedback(G*C, 1);
%step(G2);

% Ziglar Nicolas tuning

% Obtain the step response
```

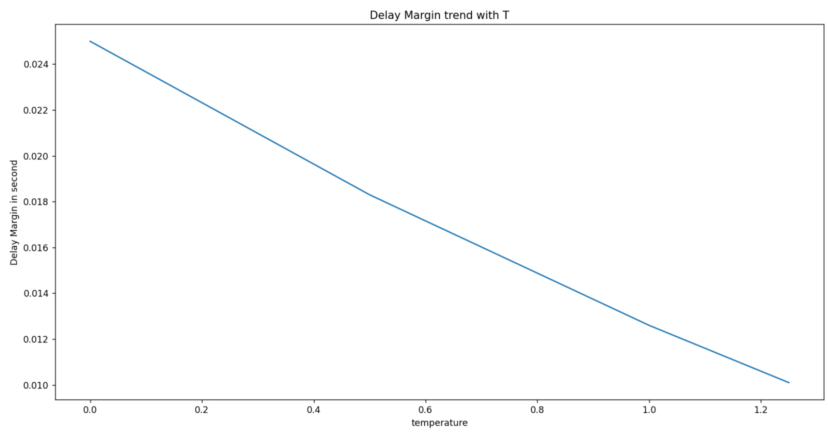


Figure 6: Delay Margins Trend

```
time = 0:0.01:20;                % Time vector
[y_out, time] = step(G, time);   % Simulate the response to the step input

% point of maximum slope
delta_y = diff(y_out) ./ diff(time);          % Calculate the slope
[max_slope, max_slope_index] = max(delta_y);  % Find the maximum slope

%time and output value for the maximum slope
max_slope_time = t(max_slope_index);
max_slope_output = y(max_slope_index);

%slope of the tangent at the point of maximum slope
tangent_slope = max_slope;

%intercepts of the tangent line
y_intercept = max_slope_output - tangent_slope * max_slope_time;
x_intercept = -y_intercept / tangent_slope;

% Plot the step response and tangent line
plot(time, y_out);
hold on;
plot(max_slope_time, max_slope_output, 'ro');  % Mark the point of max slope
plot([0, max_slope_time + 2], [y_intercept, max_slope_output +
```

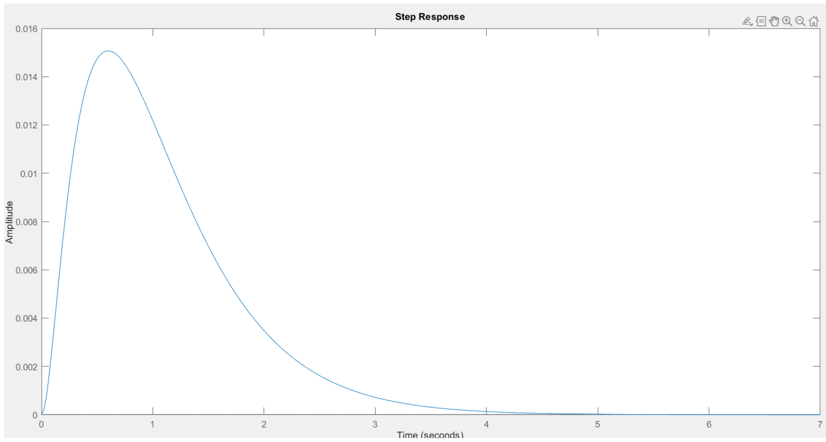


Figure 7: damping of step disturbance in 2 second

```
tangent_slope * (max_slope_time + 2)], 'g--');  % Plot the tangent line
hold off;
xlabel('Time');
ylabel('Response');
title('Step Response with Tangent at Max Slope');

% Display the intercepts
fprintf('L: %.4f\n', x_intercept);
fprintf('a: %.4f\n', -y_intercept);

a = 0.0196;
L = 0.1181;

Kp = 1.2/a;
Kd = 1/(2*L);
Ki = 0.5*L;

% PID Controller
C = Kp + Kd*s + Ki/s;

% R to C tranfer function
G3 = G*C/(1+G*C);
step(G3);
```

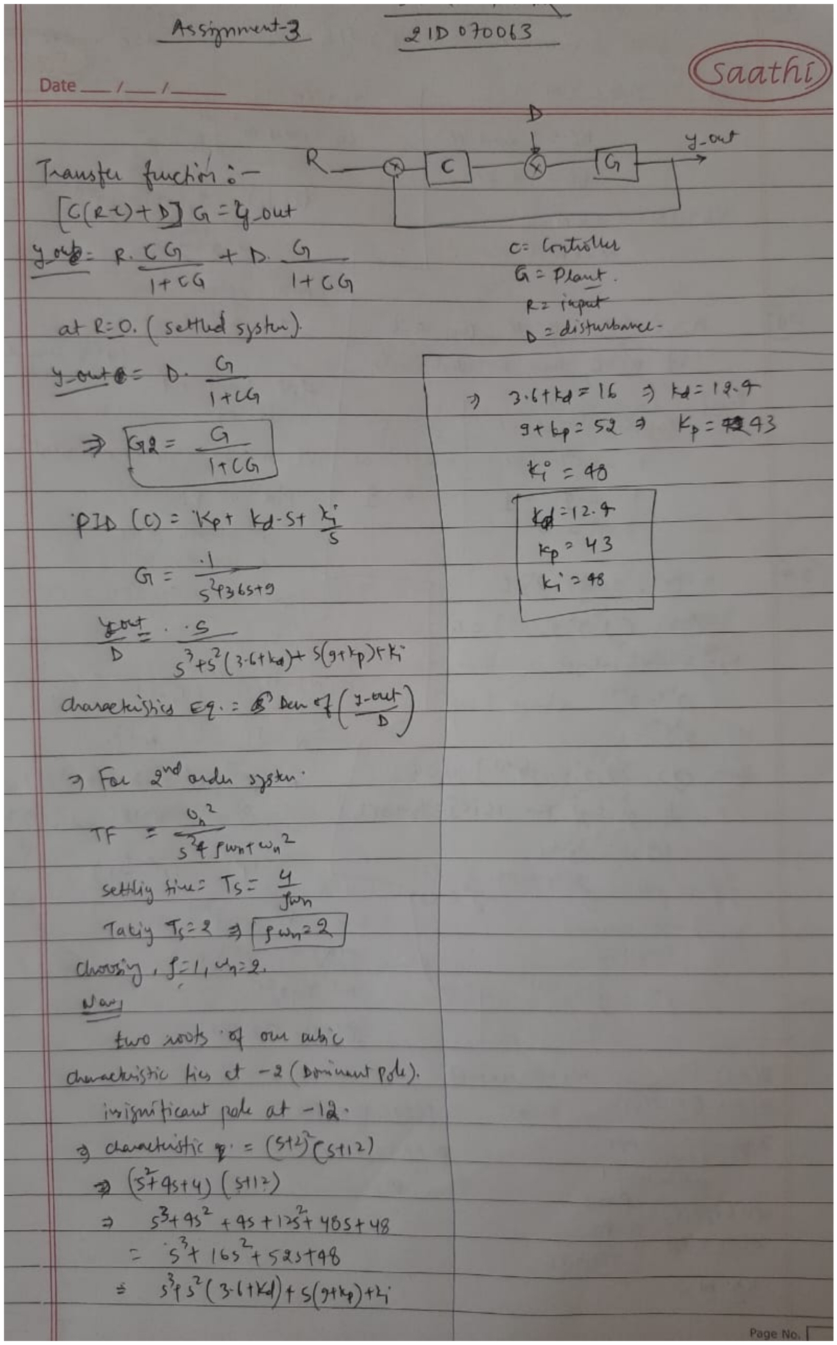
```
% D to C tranfer function
G4 = G/(1+G*C);
%step(G4);
%}

% auto tuning
% Auto-tune the PID controller
[PID_controller,~,info] = pidtune(G,'pid');

% Display the tuned PID controller gains
disp('Tuned PID Controller Gains:');
disp(PID_controller);

% Display tuning information
disp('Tuning Information:');
disp(info);

Kp = get(PID_controller,'Kp');
Kd = get(PID_controller,'Kd');
Ki = get(PID_controller,'Ki');

% PID Controller
C = Kp + Kd*s + Ki/s;

% R to C tranfer funcion
G5 = G*C/(1+G*C);

%step(G5);



% D to C tranfer function
G6 = G/(1+G*C);

step(G6);
```

3.3 Part 3

Observations:
1. Response to Unit-Step Disturbance:
The significant overshoot suggests that the system might be underdamped. Settling to zero in about 2.5 seconds indicates that the system responds relatively quickly, which is desirable. However, settling to zero within 2.5 seconds might be accompanied by overshoot, suggesting a trade-off between settling time and overshoot.

2.Response to Unit-Ramp Disturbance:
Settling to about 0.025 suggests that there might be some steady-state error in the system's response to the unit-ramp disturbance. System is at least marginally stable.

Reducing the effect:
1. The system appears to be at least marginally stable , especially considering the significant overshoot in the response to the unit-step disturbance.
2.To improve stability and reduce overshoot, we might need to adjust the controller parameters or consider a different controller design. Additionally, addressing the steady-state error in response to the unit-ramp disturbance might require further tuning or adjustments to the controller.



Assignment-3

21D070063

Date __/__/__

Transfer function :-

$[C(R-Y)+D]\,G = y\_out$

$y\_out = R \cdot \frac{CG}{1+CG} + D \cdot \frac{G}{1+CG}$

C = Controller
G = Plant.
R = input
D = disturbance.

at R=0. (settled system).

$y\_out = D \cdot \frac{G}{1+CG}$

$\Rightarrow \boxed{G_2 = \frac{G}{1+CG}}$

$PID\ (C) = K_p + K_d \cdot s + \frac{K_i}{s}$

$G = \frac{1}{s^2 + 3.6s + 9}$

$\frac{y\_out}{D} = \frac{s}{s^3 + s^2(3.6+K_d) + s(9+K_p) + K_i}$

characteristics Eq. = Den of $\left(\frac{y\_out}{D}\right)$

$\Rightarrow$ For 2nd order system.

$TF = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$

settling time = $T_s = \frac{4}{\zeta\omega_n}$

Taking $T_s = 2 \Rightarrow \boxed{\zeta\omega_n = 2}$

Choosing, $\zeta = 1$, $\omega_n = 2$.

Now,

two roots of our cubic characteristic lies at -2 (Dominant pole).

insignificant pole at -12.

$\Rightarrow$ characteristic eq. $= (s+2)^2(s+12)$

$\Rightarrow (s^2+4s+4)(s+12)$

$\Rightarrow s^3 + 4s^2 + 4s + 12s^2 + 48s + 48$

$= s^3 + 16s^2 + 52s + 48$

$= s^3 + s^2(3.6+K_d) + s(9+K_p) + K_i$

$\Rightarrow 3.6 + K_d = 16 \Rightarrow K_d = 12.4$

$9 + K_p = 52 \Rightarrow K_p = 43$

$K_i = 48$

$$\boxed{\begin{aligned} K_d &= 12.4 \\ K_p &= 43 \\ K_i &= 48 \end{aligned}}$$

Page No.

Figure 8: Analytical calculations q2

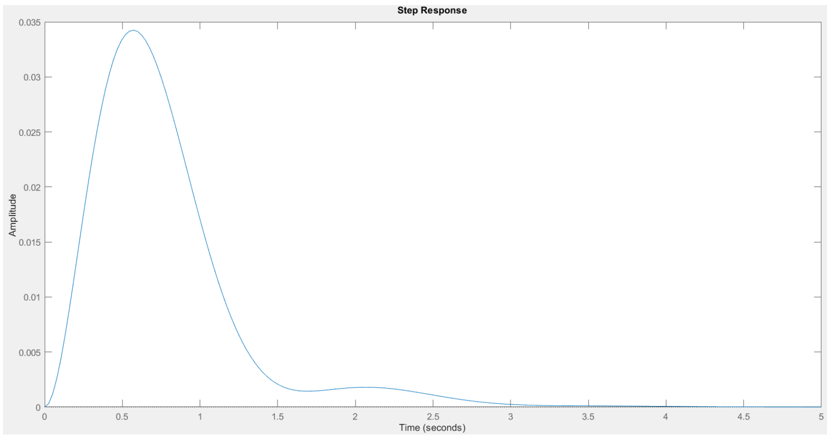


Figure 9: step response q2part3

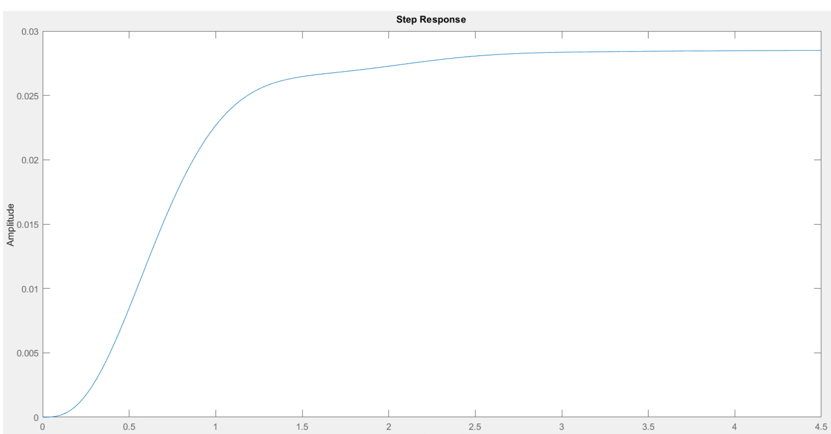


Figure 10: ramp response q2part3