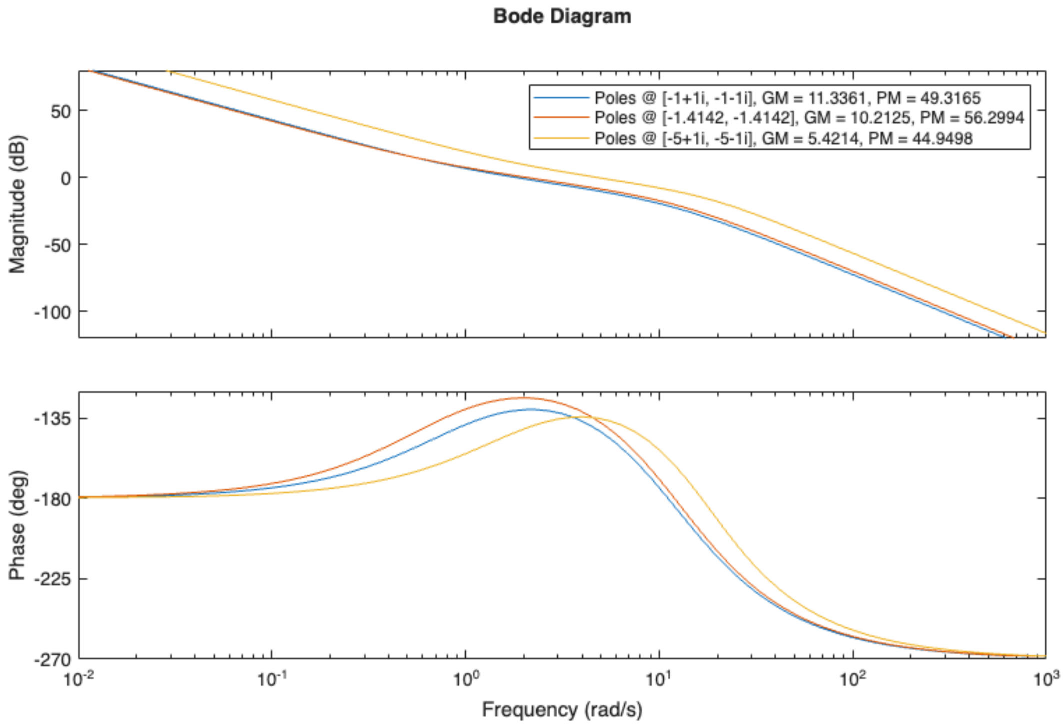
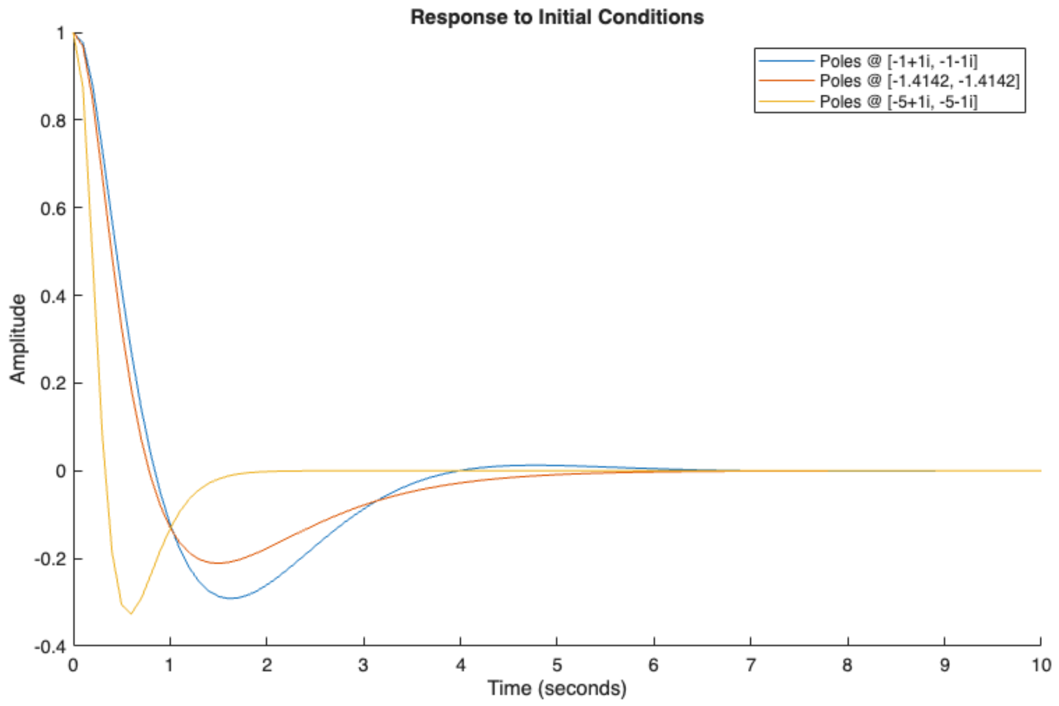
Table of Contents

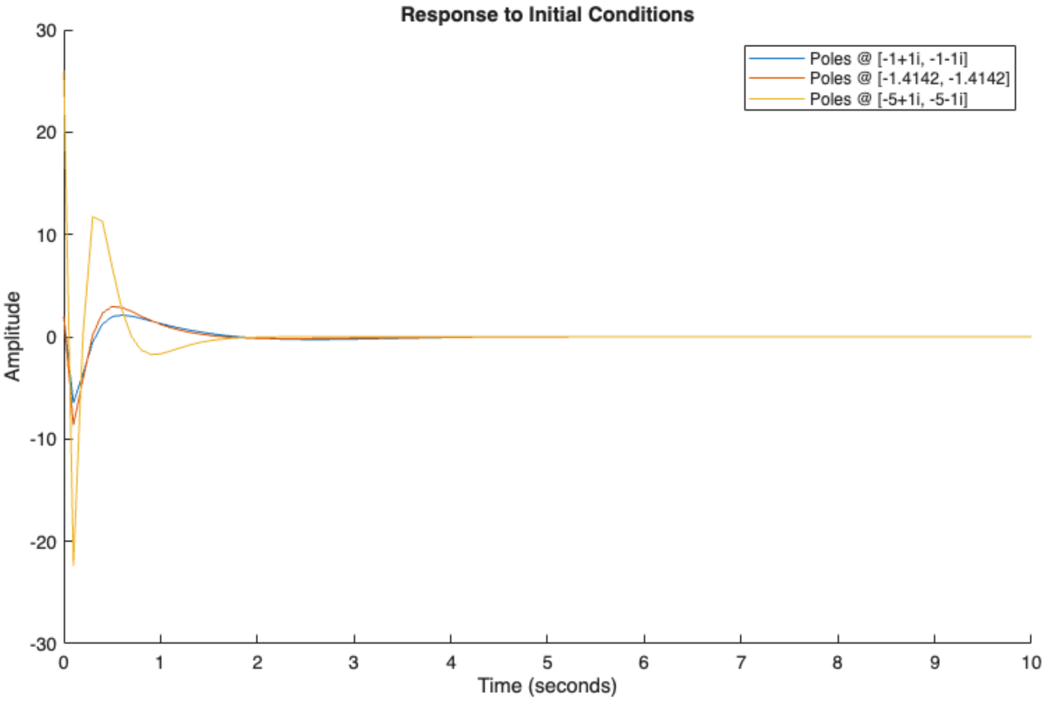
```
close all; clear; clc;
```

Problem 1

```
A = [0 1; 0 0];
B = [0; 1];
C = [1 0];
D = [0];

state = ss(A, B, C, D);

k_desired_sets = [-1+1i, -1-1i; -sqrt(2), -sqrt(2); -5+1i -5-1i];
l_desired_set = [-10, -10];
z = struct();
u = struct();
bode_legend_string = {};


figure;
hold on;
t = [0:0.1:10];
for j = 1:length(k_desired_sets)

    curr_poles = k_desired_sets(j,:);
    K = acker(A, B, curr_poles);
    L = acker(A', C', l_desired_set)';

    At = [A-B*K             B*K;
        zeros(size(A))    A-L*C];

    Bt = [B;
        zeros(size(B))];

    Ct = [C zeros(size(C))];

    new_sys = ss(At, Bt, Ct, D);

    k_r = -1/(Ct*inv(At)*Bt);
    final_tf = k_r*new_sys;
    z.("poles_" + string(j)) = initial(feedback(ss(A,B,C,D)*ss(A-B*K-
L*C,L,K,0), 1), [1;0;0;0], t);

    u.("poles_" + string(j))= initial(feedback(ss(A-B*K-
L*C,L,K,0),ss(A,B,C,D)), [1;0;0;0],t);
    bode(ss(A-B*K-L*C,L,K,0)*ss(A,B,C,D));
    [Gm_ss, PM_ss] = margin(ss(A-B*K-L*C,L,K,0)*ss(A,B,C,D));
    bode_legend_string = horzcat(bode_legend_string, "Poles @ [" +
string(curr_poles(1)) + ", " + string(curr_poles(2)) + "], GM = " +
string(Gm_ss) + ", PM = " + string(PM_ss));
end
legend(bode_legend_string);
hold off;

%
struct_fields = string(fieldnames(z));

z_fig = figure();
hold on;
title("Response to Initial Conditions");
xlabel("Time (seconds)");
ylabel("Amplitude");

u_fig = figure();
hold on;
title("Response to Initial Conditions");
xlabel("Time (seconds)");
ylabel("Amplitude");

bode_legend_string = {};

for field = 1:length(struct_fields)
    figure(z_fig);
    plot(t, z.(struct_fields(field)))

    figure(u_fig);
    plot(t, u.(struct_fields(field)))


    bode_legend_string = horzcat(bode_legend_string, "Poles @ [" +
string(k_desired_sets(field, 1)) + ", " + string(k_desired_sets(field, 2)) +
"]");
end

figure(z_fig);
legend(bode_legend_string);

figure(u_fig);
legend(bode_legend_string);

% When keeping the damping ratio the same pushing your pole further in the
% LHP decreased gain margin and phase margin but if you increase your
% damping ratio the phase margin increases EVEN if your gain margin is
% decreasing due to a more negative pole
```


Bode Diagram
Magnitude (dB)
50
0
-50
-100
Poles @ [-1+1i, -1-1i], GM = 11.3361, PM = 49.3165
Poles @ [-1.4142, -1.4142], GM = 10.2125, PM = 56.2994
Poles @ [-5+1i, -5-1i], GM = 5.4214, PM = 44.9498
Phase (deg)
-135
-180
-225
-270
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (rad/s)


Response to Initial Conditions
Poles @ [-1+1i, -1-1i]
Poles @ [-1.4142, -1.4142]
Poles @ [-5+1i, -5-1i]
Amplitude
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
1
2
3
4
5
6
7
8
9
10
Time (seconds)

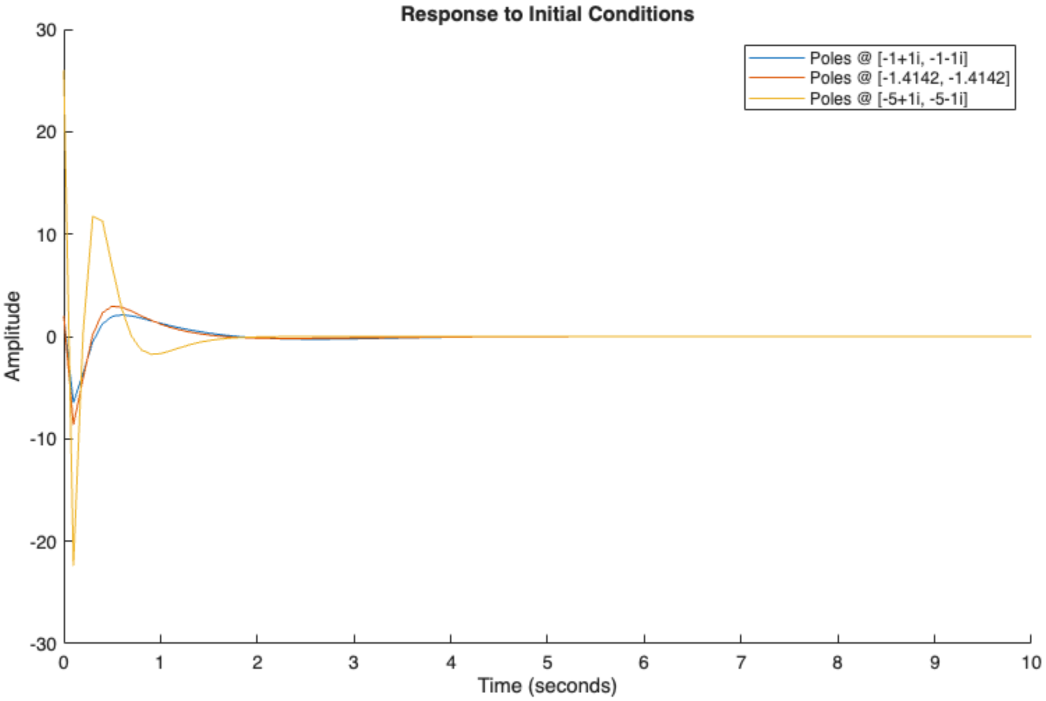


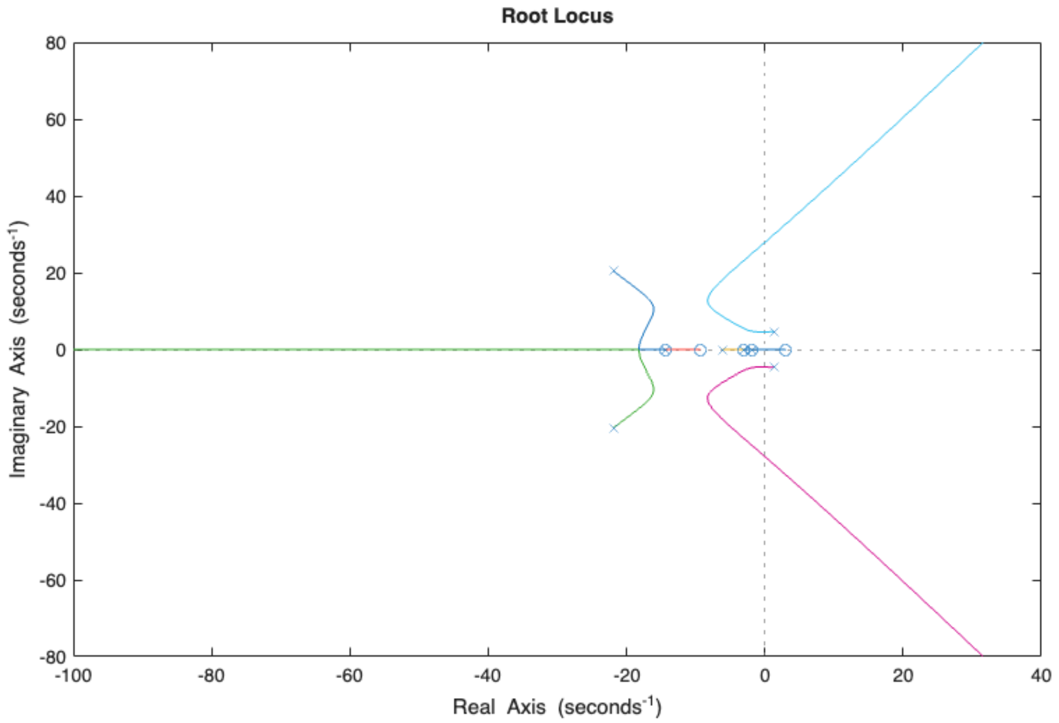
Problem 2

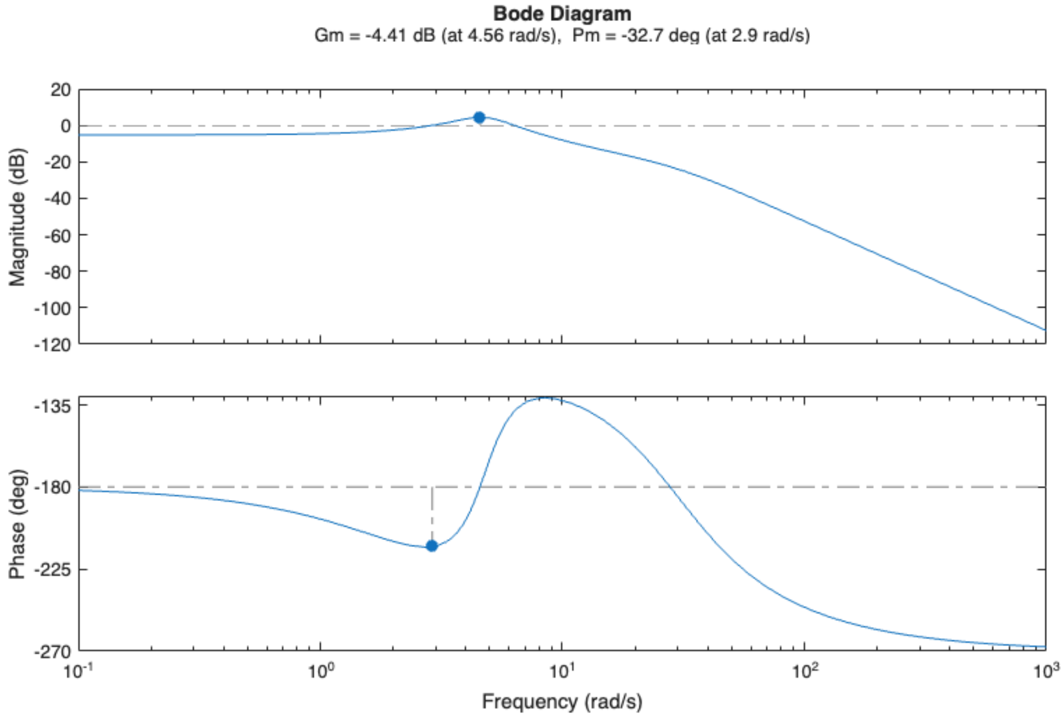
```
%2a
v0 = 5;
[M, C, K0, K2] = matricies();
[a_mat,b_mat,c_mat,d_mat] = generate_state_space_matricies(M, C, K0, K2, v0);
sys = ss(a_mat, b_mat, c_mat, d_mat);
Q = [0 0 0 0;
    0 1 0 0;
    0 0 0 0;
    0 0 0 0];
R = 1;
[K,S,P] = lqr(sys,Q,R);
eig(a_mat - b_mat*K)

% 2b
sigma_d = 0.1;
sigma_n = 10^-3;
mu_d = 0;
mu_n = 0;

sigma_d = sigma_d^2 - mu_d^2;
sigma_n = sigma_n^2 - mu_n^2;

[kalmf,L,P] = kalman(sys, sigma_d, sigma_n, 0);
eig(a_mat - L*c_mat)

% 2c
```

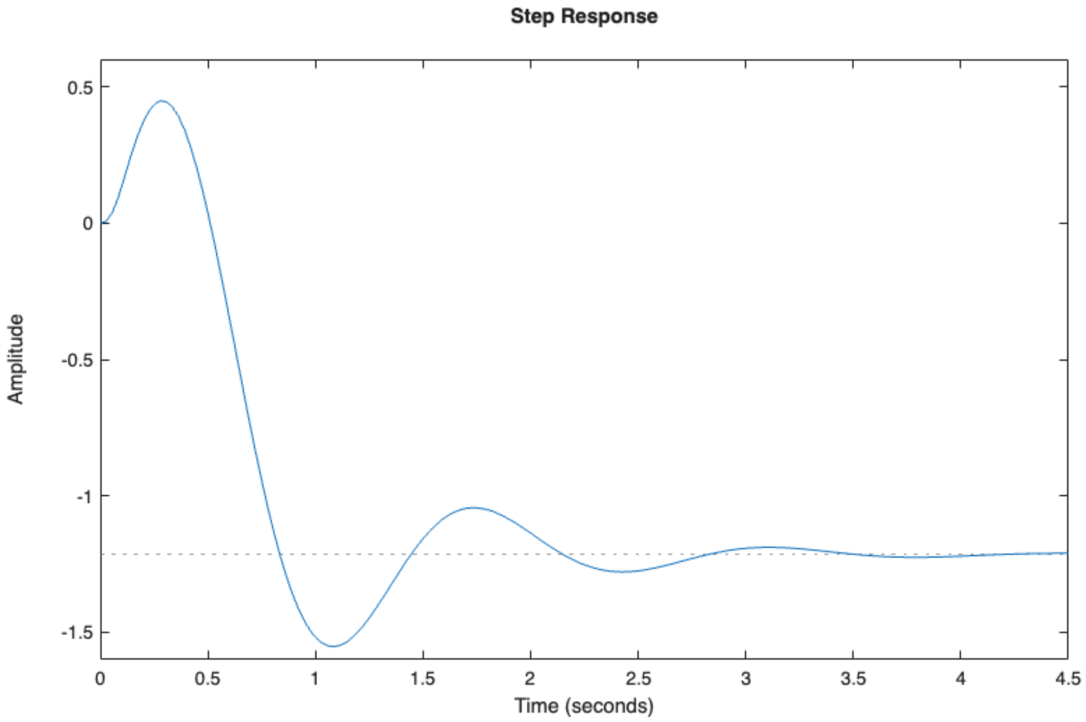
```
controller_d_mat = 0;
controller_ss = ss(a_mat-b_mat*K-L*c_mat, L, K, controller_d_mat);
figure;
rlocus(controller_ss*sys);
figure;
margin(controller_ss*sys);
figure;
step(feedback(controller_ss*sys,1));
```

```
ans =

 -14.2878 + 0.0000i
  -1.3416 + 4.6039i
  -1.3416 - 4.6039i
  -1.8470 + 0.0000i


ans =

 -20.7331 +18.8414i
 -20.7331 -18.8414i
  -2.8548 + 0.0000i
  -3.2991 + 0.0000i
```




Root Locus
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
80
60
40
20
0
-20
-40
-60
-80
-100
-80
-60
-40
-20
0
20
40

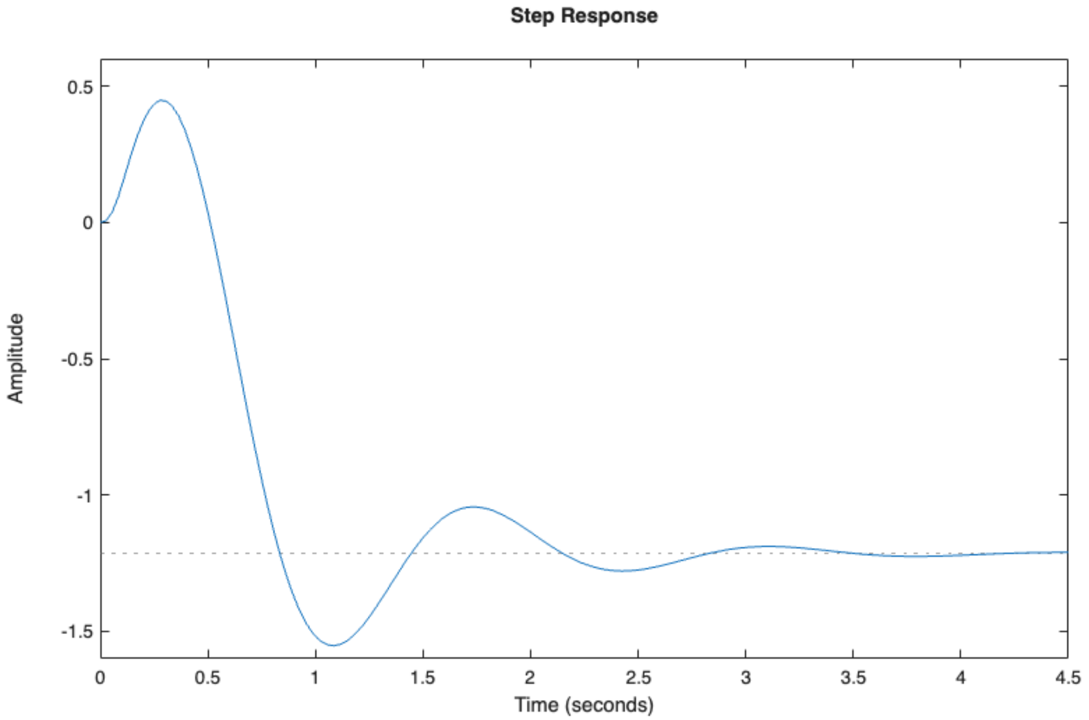

Bode Diagram
Gm = -4.41 dB (at 4.56 rad/s), Pm = -32.7 deg (at 2.9 rad/s)
Magnitude (dB)
20
0
-20
-40
-60
-80
-100
-120
Phase (deg)
-135
-180
-225
-270
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (rad/s)

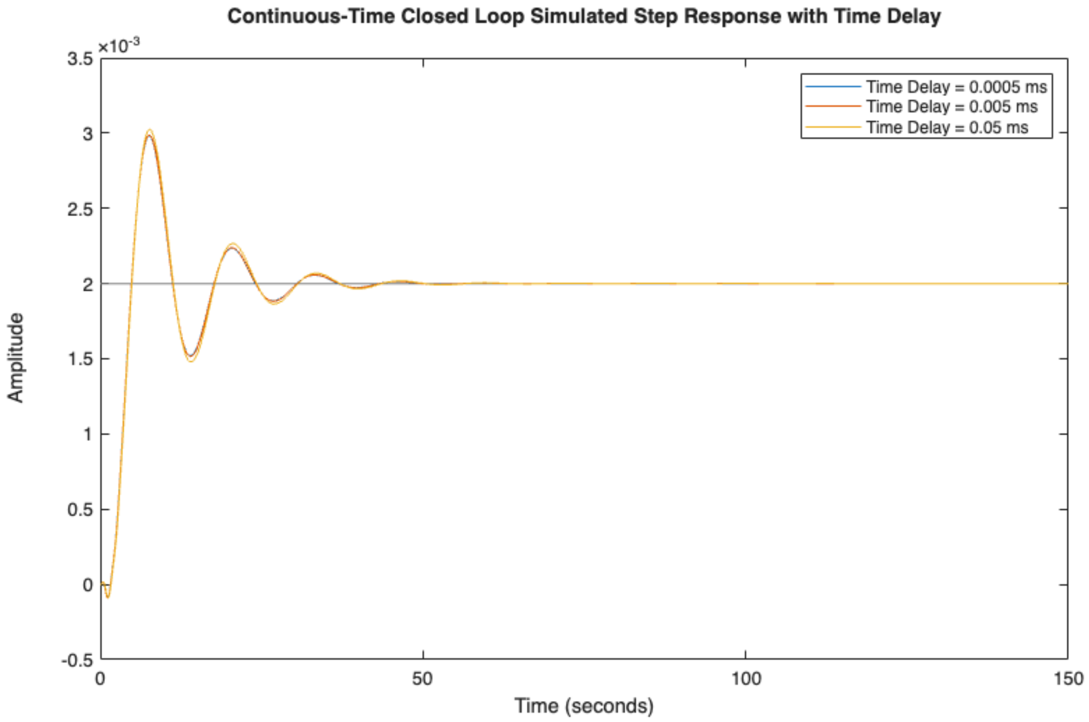


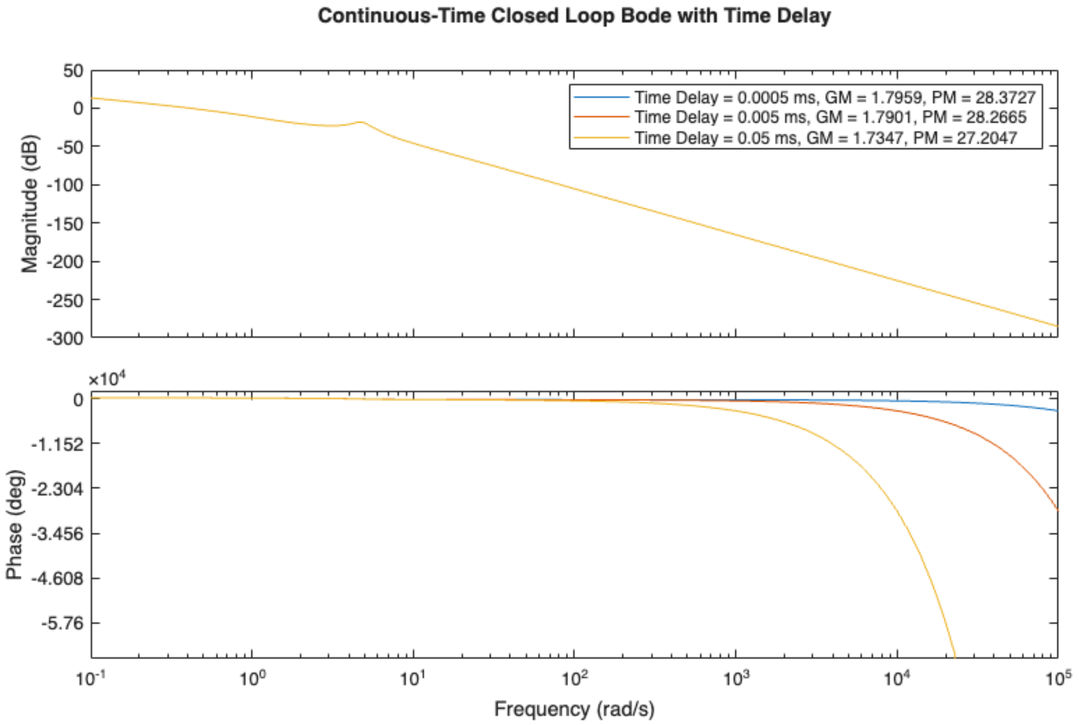
Problem 3

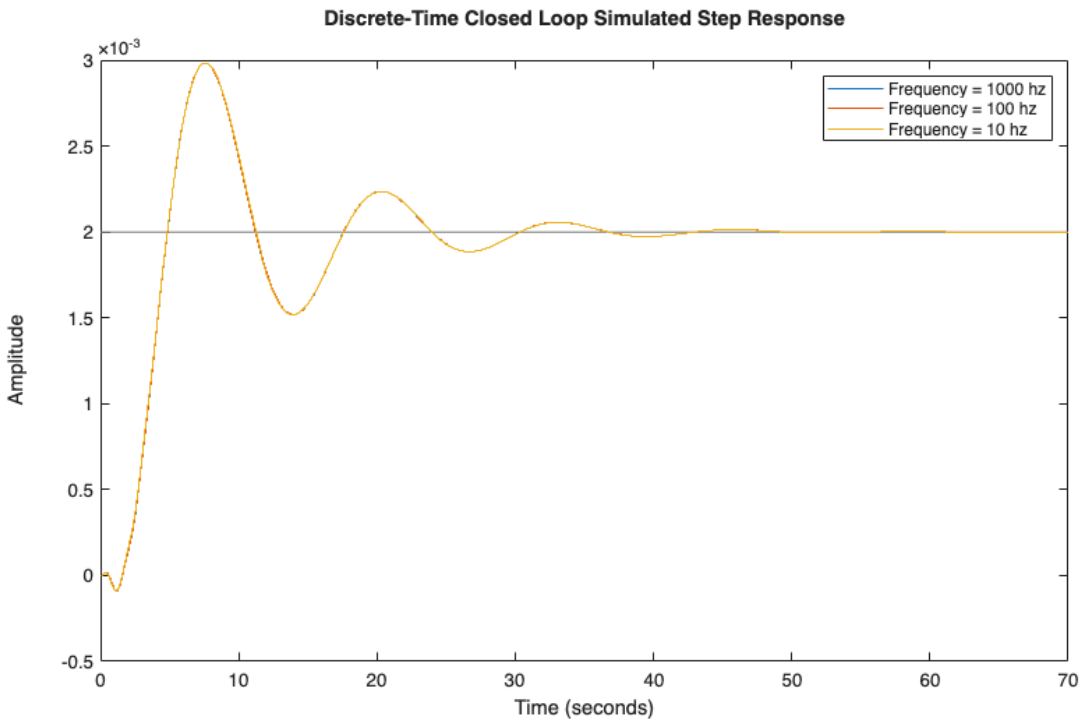
```
% a
s = tf('s');
t = 0:0.01:150;
u = 0.002*ones(size(t));

P_tf = tf(sys);
C_prime = 5; % from sisotool
P_prime_tf = feedback(P_tf, C_prime);
z = 2.1797;
p = [0 -1];
k = 0.6886; %0.1 adjusted to decrease overshoot
C_final = zpk(z, p, k);
P_final = P_prime_tf;

td = [0.5 5 50]*10^-3;

step_legend_string = {};
bode_legend_string = {};

step_fig = figure();
hold on;
bode_fig = figure();
hold on;


for i = 1:length(td)
```

```
    time_delay = exp(-s*td(i));
    open_loop_tf = P_final*C_final*time_delay;
    closed_loop_tf = feedback(open_loop_tf, 1);

    figure(step_fig);
    lsim(closed_loop_tf, u, t);

    figure(bode_fig);
    bode(open_loop_tf);
    [Gm, Pm] = margin(open_loop_tf);

    step_legend_string = horzcat(step_legend_string, "Time Delay = " +
string(td(i)) + " ms");
    bode_legend_string = horzcat(bode_legend_string, "Time Delay = " +
string(td(i)) + " ms, GM = " + string(Gm) + ", PM = " + string(Pm));

end

figure(step_fig);
title("Continuous-Time Closed Loop Simulated Step Response with Time Delay");
legend(step_legend_string);

figure(bode_fig);
legend(bode_legend_string);
title("Continuous-Time Closed Loop Bode with Time Delay");

% b/c
step_fig = figure();
hold on;
bode_fig = figure();
hold on;


step_legend_string = {};
bode_legend_string = {};


Fs = [1e3 100 10];
for i = 1:length(Fs)
    F = Fs(i);
    Ti = 1/F;

    Cd = c2d(C_final, Ti, 'tustin');
    Pd = c2d(P_final, Ti, 'tustin');

    t = 0:Ti:70;
    u = 0.002*ones(size(t));
    discrete_ol = (Cd)*(Pd);
    discrete_cl = feedback(discrete_ol, 1);

    figure(step_fig);
    lsim((discrete_cl), u, t);

    figure(bode_fig);
    bode(discrete_ol);
    [Gm, Pm] = margin(discrete_ol);

    step_legend_string = horzcat(step_legend_string, "Frequency = " +
string(Fs(i)) + " hz");
    bode_legend_string = horzcat(bode_legend_string, "Frequency = " +
string(Fs(i)) + " hz, GM = " + string(Gm) + ", PM = " + string(Pm));
end


figure(step_fig);
title("Discrete-Time Closed Loop Simulated Step Response");
legend(step_legend_string);

figure(bode_fig);
legend(bode_legend_string);
title("Discrete-Time Closed Loop Bode");

% The two plots are effectively the same which makes sense the time delays
% make it so that the system is effectively not responding for that period
% of time which is a similar dynamic to a discrete system where the system
% receives updates at a specified interval
```

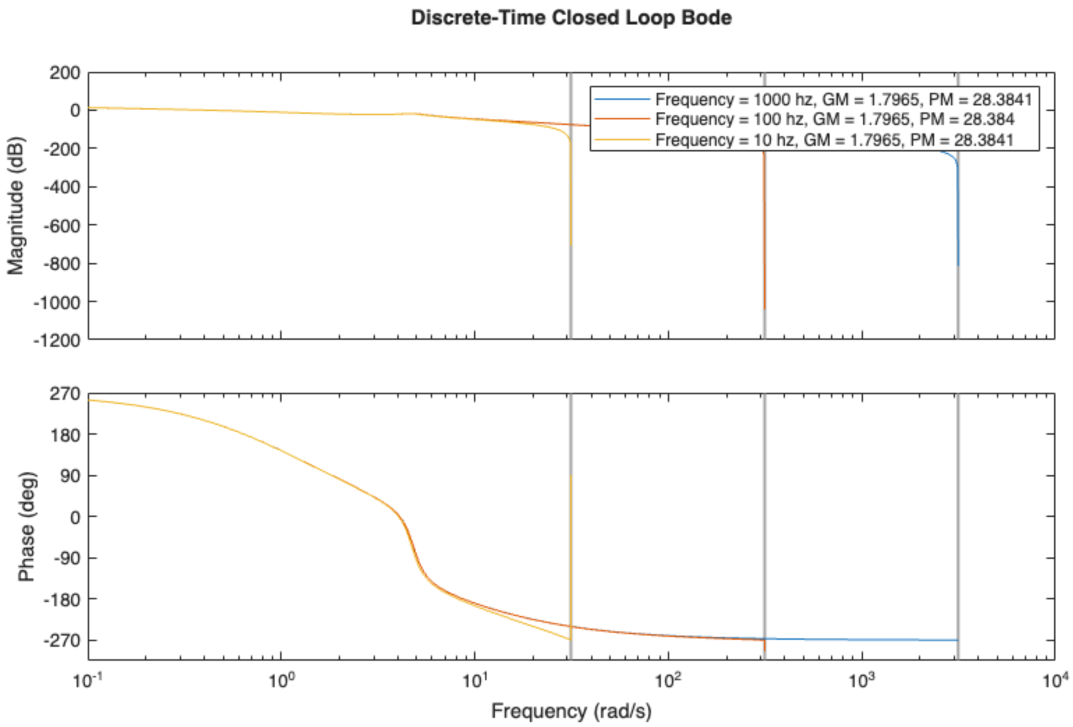



Continuous-Time Closed Loop Simulated Step Response with Time Delay
×10^-3
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
Amplitude
0
50
100
150
Time (seconds)
Time Delay = 0.0005 ms
Time Delay = 0.005 ms
Time Delay = 0.05 ms


Continuous-Time Closed Loop Bode with Time Delay
50
0
-50
-100
-150
-200
-250
-300
Magnitude (dB)
Time Delay = 0.0005 ms, GM = 1.7959, PM = 28.3727
Time Delay = 0.005 ms, GM = 1.7901, PM = 28.2665
Time Delay = 0.05 ms, GM = 1.7347, PM = 27.2047
×10^4
0
-1.152
-2.304
-3.456
-4.608
-5.76
Phase (deg)
10^-1
10^0
10^1
10^2
10^3
10^4
10^5
Frequency (rad/s)



Discrete-Time Closed Loop Simulated Step Response



Discrete-Time Closed Loop Bode

Published with MATLAB® R2025a