Table of Contents


```
%{
MECH 7710
Homework 0
Matt Boler
%}

clc; clear all; close all;
```

Part 1

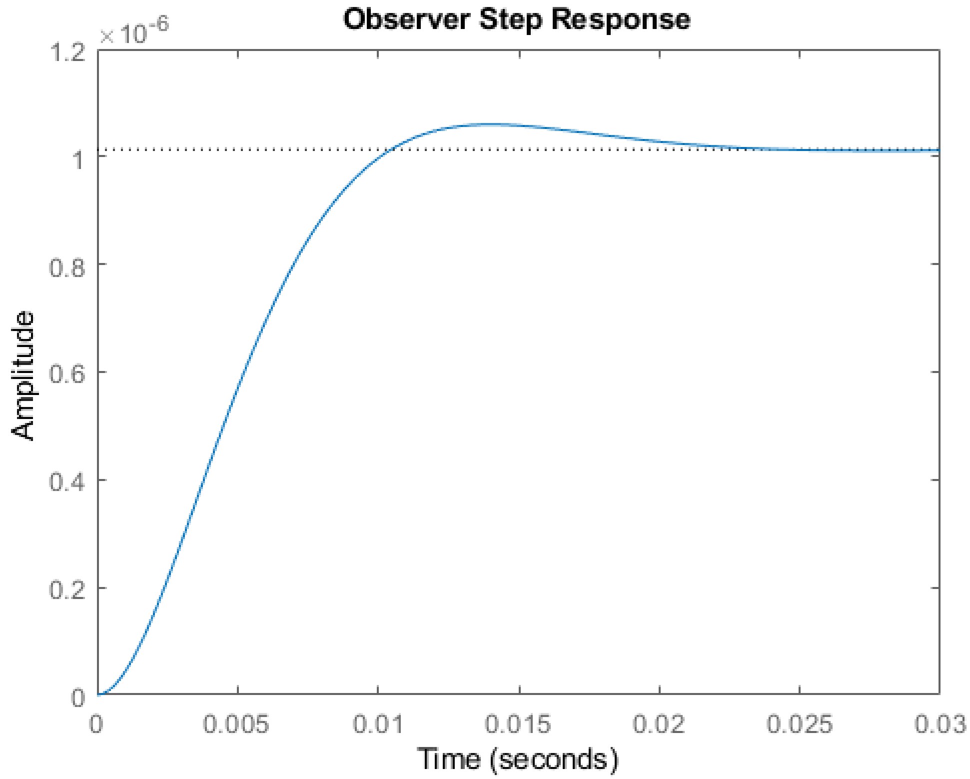
```
J = 10; % kg*m^2
b = 1;% N*m*s/rad

% Derive the differential equation
% u = J*theta_dd + b*theta_d

% Convert the system to state space
% states = [theta; theta_d];

A = [0, 1; ...
    0, -b/J];

B = [0; 1/J];

C = [1, 0];

D = 0;

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

% What are the eigenvalues of the system?
eig_sys = eig(A)


eig_sys =

         0
   -0.1000
```

Part 2

Design an observer for the above system

```
% Show that the system is observable
dimension = 2;
rank(obsv(A, C)) == dimension

% Design L st the error dynamics have:
wn_obs = 50*2*pi; % rad/s
zeta_obs = 0.7;

sigma_obs = zeta_obs * wn_obs;
wd_obs = wn_obs * sqrt(1 - zeta_obs^2);

s_des_obs = [-sigma_obs + 1i*wd_obs, ...
    -sigma_obs - 1i*wd_obs];

L = place(A', C', s_des_obs)';

sys_obs = ss(A-L*C, B, C, 0);
figure(1);
step(sys_obs);
title('Observer Step Response');
```

```
ans =

  logical

   1
```



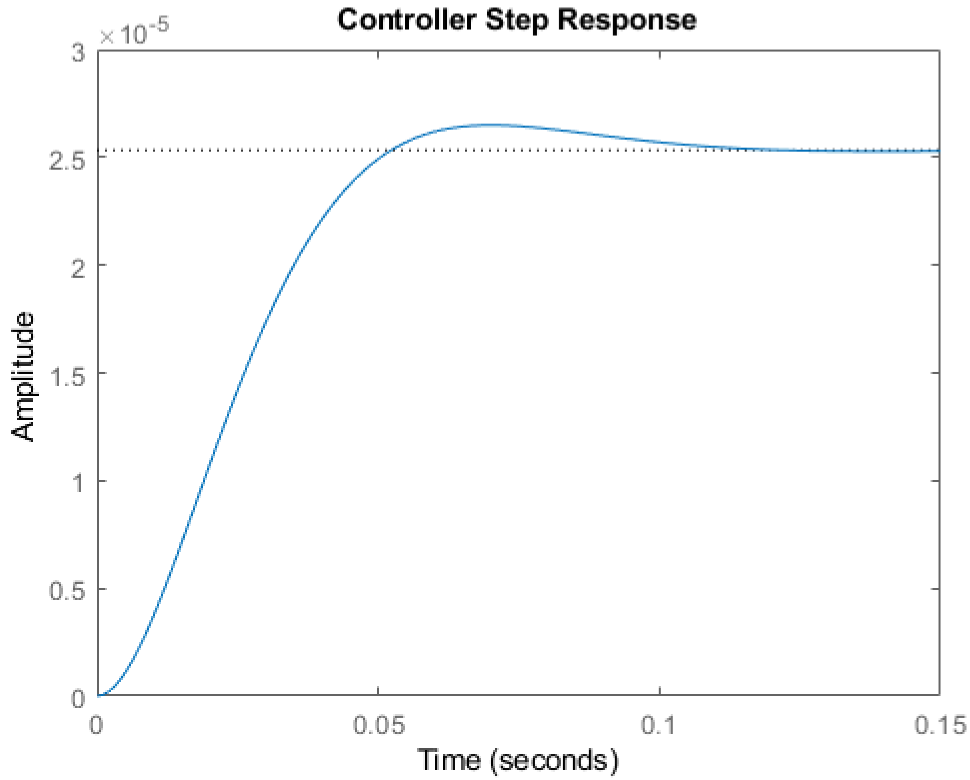
Part 3

Design a state feedback controller for the table

```
% Show the table is controllable
rank(ctrb(A, B)) == dimension

% Design K st the system has:
wn_con = 10*2*pi; % rad/s
zeta_con = 0.7;

sigma_con = zeta_con * wn_con;
wd_con = wn_con * sqrt(1 - zeta_con^2);

s_des_con = [-sigma_con + 1i*wd_con, ...
    -sigma_con - 1i*wd_con];

K = place(A, B, s_des_con);

sys_con = ss(A-B*K, B, C, 0);
figure(2);
step(sys_con);
title('Controller Step Response');
```

```
ans =
```

```
logical

 1
```

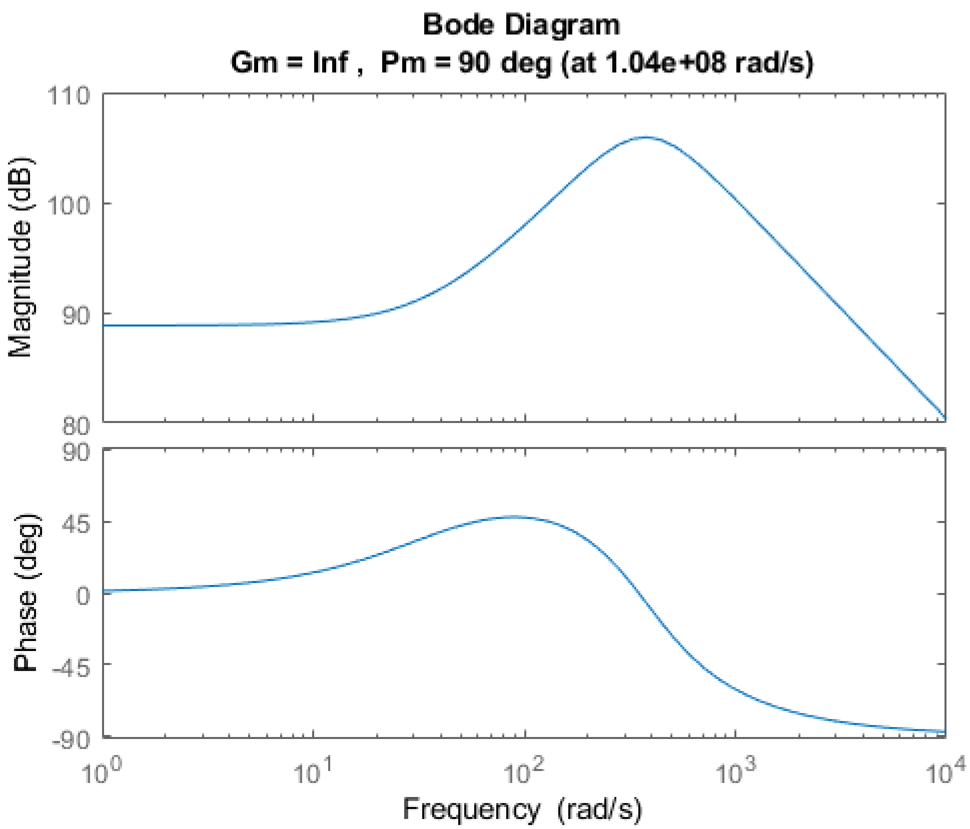


Part 4

Solve for the equivalent compensator

```
A_comp = A - B*K - L*C;
B_comp = L;

% Large number of sources say this should be -K instead, why?
C_comp = K;
s = tf('s');

% NOTE: A_comp has VERY poor condition number
compensator = C_comp * inv(s*eye(dimension) - A_comp) * B_comp;
compensator = minreal(compensator, 0.001);

% What kind of control system does it resemble?

% Ignoring numerical errors in p-z cancellation, the compensator has a
 pair
```

```
% of stable complex poles and a stable real zero, which looks kind of
 like
% a 2nd-order low pass filter with a phase bump

% What can you say about the 'robustness' of the controller?

% Gain + Phase margins shown below:
figure(3)
margin(compensator);

% The controller itself has poor margins, but we don't really look at
% controllers independent of the plant so that's rather
 inconsequential.
```



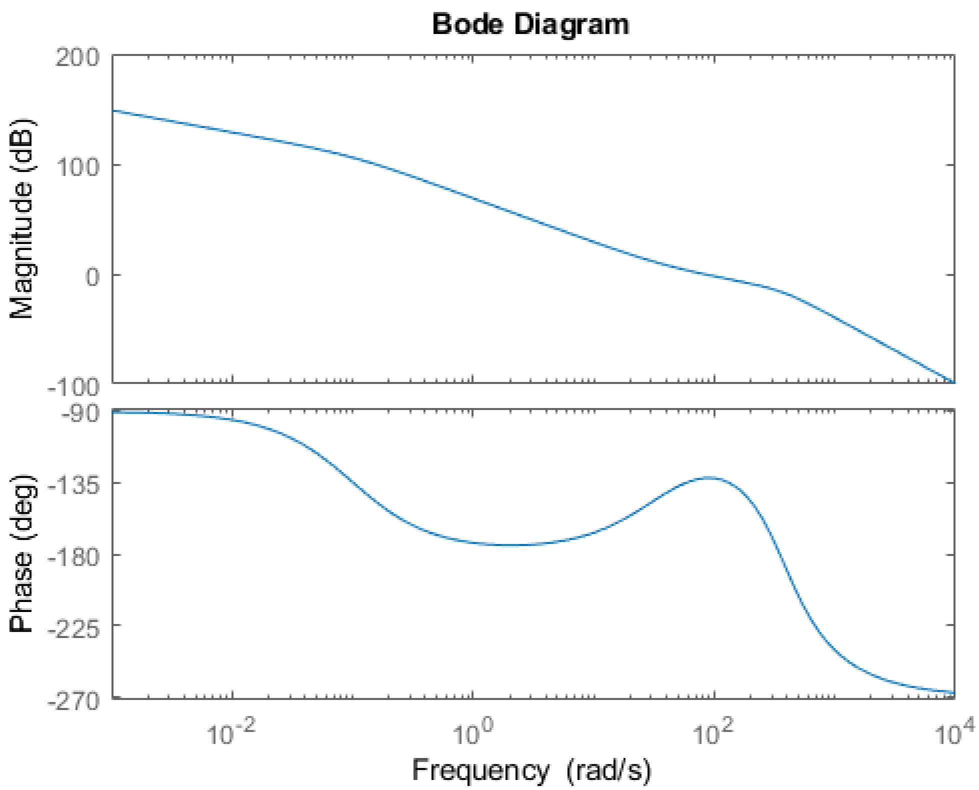
Part 5

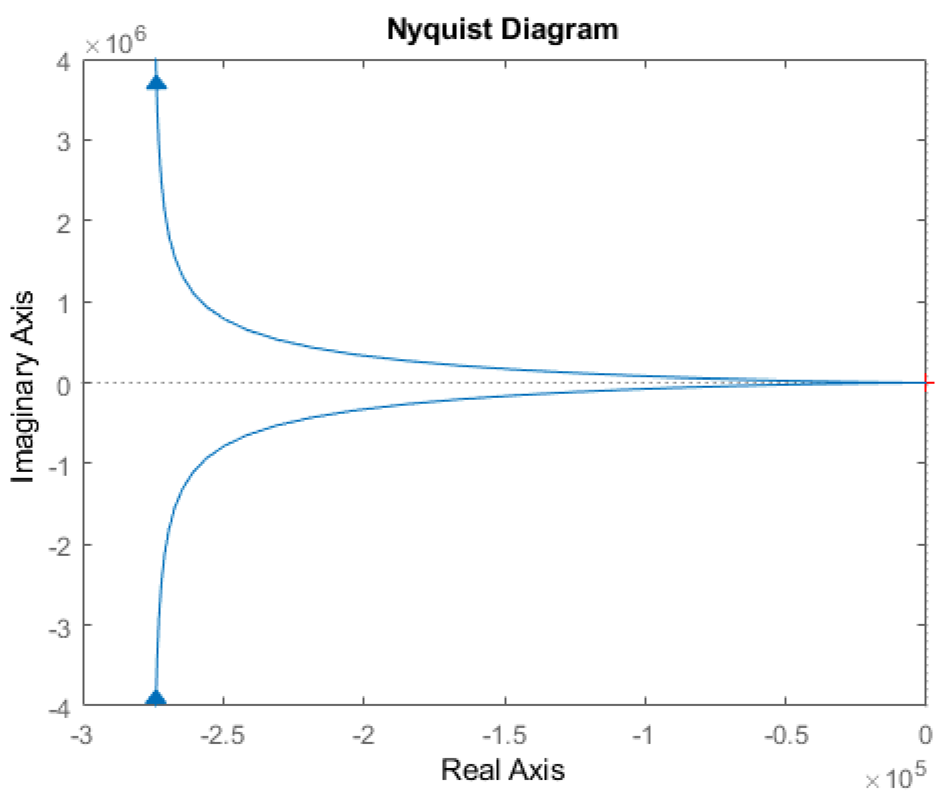
Calculate the closed-loop transfer function

```
[num, den] = ss2tf(A, B, C, D);
sys_tf = tf(num, den);

fp = sys_tf * compensator;

sys_cl = fp/(1 + fp)

% Provide Bode and Nyquist plots for the closed-loop system
```

```
figure(4);
bode(fp)

figure(5);
nyquist(fp);
```

```
sys_cl =


    1.04e07 s^5 + 5.879e09 s^4 + 1.676e12 s^3 + 5.52e13 s^2 +
 5.503e12 s


 --------------------------------------------------------------------------


  s^8 + 1055 s^7 + 5.609e05 s^6 + 1.595e08 s^5 + 2.586e10 s^4 +
 1.68e12 s^3


                                               + 5.52e13 s^2 +
 5.503e12 s


Continuous-time transfer function.
```


Bode Diagram
Magnitude (dB)
200
100
0
-100
Phase (deg)
-90
-135
-180
-225
-270
10^{-2}
10^{0}
10^{2}
10^{4}
Frequency (rad/s)


Nyquist Diagram
$\times 10^{6}$
Imaginary Axis
4
3
2
1
0
-1
-2
-3
-4
-3
-2.5
-2
-1.5
-1
-0.5
0
Real Axis
$\times 10^{5}$

Part 6

Design the controller in the discrete domain with a 1kHz sample rate

```
% Discretize the state space model. What are the eigenvalues?
dt = 0.001;
sys_d = c2d(sys, dt);
eig_sys_d = eig(sys_d.A)

% Design L to provide the same response as problem 2
s_des_obs_d = exp(dt * s_des_obs);
L_d = place(sys_d.A', sys_d.C', s_des_obs_d)';

% Design K to provide the same response as problem 3
s_des_con_d = exp(dt * s_des_con);
K_d = place(sys_d.A, sys_d.B, s_des_con_d);

% Where are the closed-loop estimator and controller poles?
eig_d = [eig(sys_d.A - L_d * sys_d.C); eig(sys_d.A - sys_d.B * K_d)]

% Solve for the equivalent compensator transfer function
z = tf('z', dt);

[num, den] = ss2tf(sys_d.A, sys_d.B, sys_d.C, sys_d.D);
sys_tf_d = tf(num, den, dt);

A_comp_d = sys_d.A - sys_d.B*K_d - L_d*sys_d.C;
B_comp_d = L_d;

% Large number of sources say this should be -K instead, why?
C_comp_d = K_d;

compensator_d = C_comp_d * inv(z*eye(dimension) - A_comp_d) *
 B_comp_d;
compensator_d = minreal(compensator_d, 0.001)

fp_d = sys_tf_d * compensator_d;
sys_cl_d = fp_d / (1 + fp_d);
```

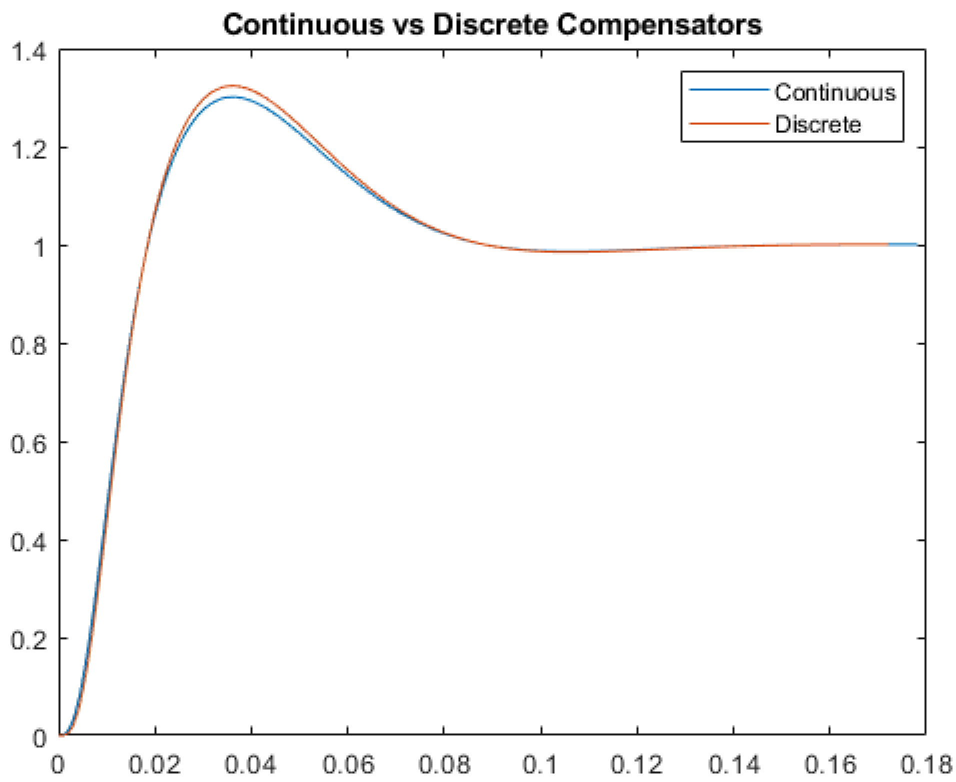
```
eig_sys_d =

    1.0000
    0.9999


eig_d =

   0.7825 + 0.1786i
   0.7825 - 0.1786i
   0.9560 + 0.0429i
   0.9560 - 0.0429i
```

```
compensator_d =

  8.451e04 z - 8.152e04
  ---------------------
  z^2 - 1.477 z + 0.594

Sample time: 0.001 seconds
Discrete-time transfer function.
```

Part 7

Compare continuous and discrete response using equivalent compensator Plot response on a single graph

```
[y_c, t_c] = step(sys_cl);
[y_d, t_d] = step(sys_cl_d);

figure(6);
plot(t_c, y_c, t_d, y_d);
legend('Continuous', 'Discrete');
title('Continuous vs Discrete Compensators');
```



Published with MATLAB® R2018b