

Project 2
Course: Automatic Control Systems – ASEN 5114-001 – Spring
2025
Professor: Dale Lawrence
Teaching Assistant: Anantha Dhruva

Steve Gillet

May 4, 2025

This project will build on Project 1, applying state space modeling and design methods to control the attitude of a spacecraft mockup model. A state space model for the spacecraft will be obtained by fitting the empirical frequency response model. A controller will be designed using state variable feedback and state observation methods, then implemented in a closed loop simulation. Here however, the controller will be in state-space form, rather than the transfer function form from Project 1.

The project will be carried out using the same teams as for Project 1, and a single report will be submitted by each team. The report should include an introduction, sections for each of the following parts, and a conclusion pointing out key lessons learned or difficulties. All team members should contribute approximately equally to the conduct of the project and the write-up, and a short description of member contributions is required. The report is due at 11:59 pm on Monday May 5.

1)

[20 pts] Use the empirical frequency response data taken during Project 1 to fit an analytical transfer function model relating input reaction wheel torque (in mN – m) to output body angular deflection (in rad). Overplot the empirical and analytic model frequency responses in Bode plot format. Try to match the damping in the empirical data, and modify the dynamics to capture the phase at low frequencies seen in the empirical data. Treat the dynamics around 3 Hz and above as unmodeled. Using this analytic transfer function model to create a minimal state space representation of the system.

I used the same technique and code that I used in the previous project, basically capturing the dynamics using second-order polynomials of the form $(s^2 + 2\zeta\omega_n s + \omega_n^2)$ to capture the resonances and antiresonances and a couple of extra poles to capture the magnitude and phase drop-off.

Here is the code:

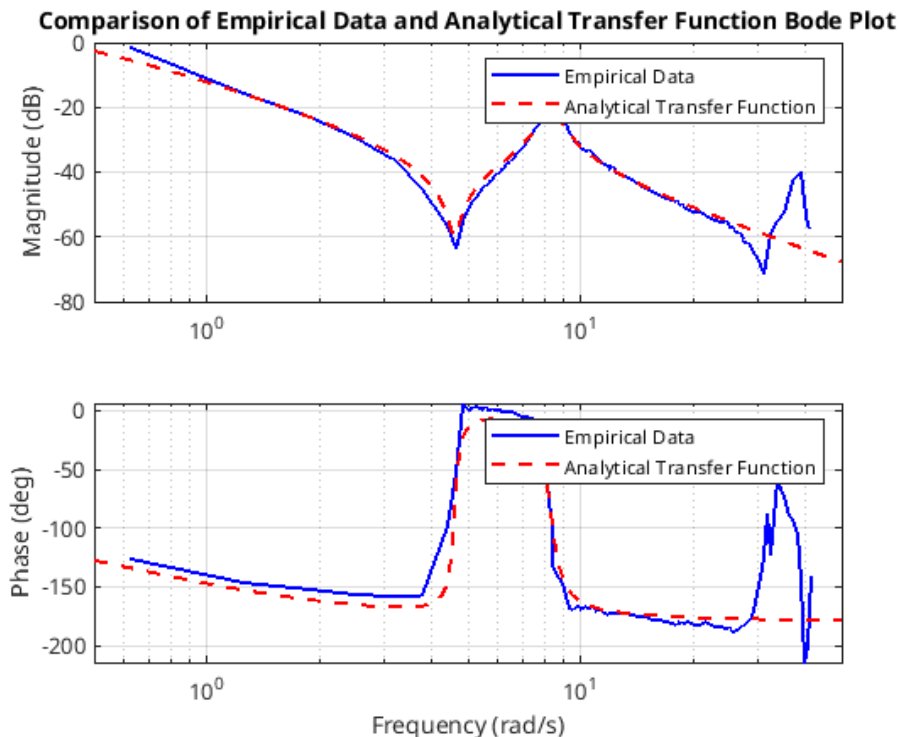
```
1  dataTable = readtable('data.xlsx', 'VariableNamingRule', 'preserve');
2  data = table2array(dataTable);
3
4  [uniqueFreq, ia] = unique(data(:,1));
5  n = size(uniqueFreq,1);
6  pResponse = zeros(n,3);
7
8  for i = 1:n
9      idx = ia(i);
10     pResponse(i,1) = 20*log10(data(idx,2)) - 20*log10(data(i,1));
11     pResponse(i,2) = data(idx,3)*180/pi - 90;
12     pResponse(i,3) = data(idx,1)*2*pi;
13 end
14
15 K = 1;
16 zetaZ = 0.05;
17 zetaP = 0.1;
18 omegaZ = 4.60118;
19 omegaP = 8.34686;
20
```

```

21 num = K*[1 zetaZ*omegaZ omegaZ^2];
22 den = [1 zetaP*omegaP omegaP^2 0];
23 pole = [1 0.65];
24 den = conv(den, pole);
25 estTF = tf(num,den);
26
27 % Generate Bode plot data for the transfer function
28 [mag, phase, w] = bode(estTF);
29 mag = squeeze(mag); % Convert to 1D array
30 phase = squeeze(phase); % Convert to 1D array
31
32 % Plot empirical data and transfer function Bode plot on the same figure
33 clf; figure(1);
34
35 % Magnitude plot
36 subplot(2,1,1);
37 semilogx(pResponse(:,3), pResponse(:,1), 'b', 'LineWidth', 1.5); % Empirical data in
    blue
38 hold on;
39 semilogx(w, 20*log10(mag), 'r--', 'LineWidth', 1.5); % Transfer function in red dashed
40 grid on;
41 ylabel('Magnitude (dB)');
42 title('Comparison of Empirical Data and Transfer Function Bode Plot');
43 legend('Empirical Data', 'Transfer Function');
44 xlim([0.5, 50]);
45
46 % Phase plot
47 subplot(2,1,2);
48 semilogx(pResponse(:,3), pResponse(:,2), 'b', 'LineWidth', 1.5); % Empirical data in
    blue
49 hold on;
50 semilogx(w, phase, 'r--', 'LineWidth', 1.5); % Transfer function in red dashed
51 grid on;
52 xlabel('Frequency (rad/s)');
53 ylabel('Phase (deg)');
54 legend('Empirical Data', 'Transfer Function');
55 xlim([0.5, 50]);

```

Here is the output:



I then used the 'ss2tf' Matlab function to get the state space realization from the transfer function I

had developed.

The resulting state space matrices look like this:

- **A matrix:**

$$A = \begin{bmatrix} -1.4847 & -70.2126 & -45.2855 & 0 \\ 1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix}$$

- **B matrix:**

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

- **C matrix:**

$$C = [0 \quad 1.0000 \quad 0.2301 \quad 21.1709]$$

- **D matrix:**

$$D = [0]$$

2)

[20 pts] Determine suitable closed loop pole locations so that the closed loop tracking bandwidth (-3 dB) is at least 1 Hz. Design a state variable feedback controller to place closed loop poles in these locations. Plot the closed loop tracking frequency response (rad/rad), and the closed loop frequency response from reference input to plant input (mN – m/rad). Plot the (negative) loop gain of this control system using the analytic plant model in both a Bode and Nyquist format. Comment on the stability margins of this design relative to those obtained in project 1.

I got the initial open loop poles using the 'eig' function on the A matrix:

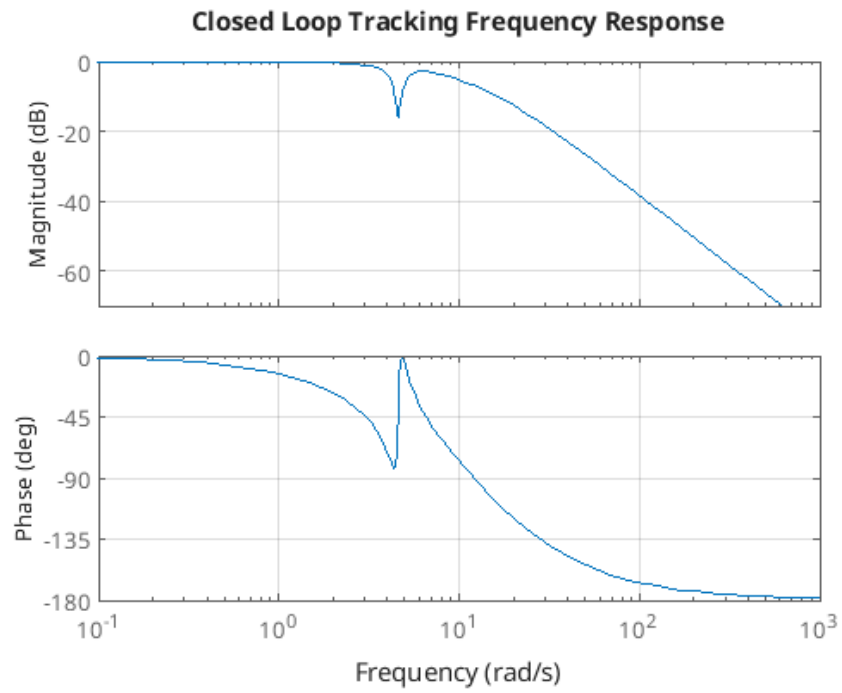
$$\begin{bmatrix} 0.0000 + 0.0000i \\ -0.4173 + 8.3364i \\ -0.4173 - 8.3364i \\ -0.6500 + 0.0000i \end{bmatrix}$$

I then decided to use a similar strategy as in the first project to create some pole/zero cancellation to reduce the effects of the resonances, however this time I wanted to only partially cancel them to make sure I wouldn't create instability anywhere else so I placed the dominant poles at $-0.615 \pm 4.5997i$, just to the left of the analytical transfer function zeros. Then I placed the other two poles at -10 and -12 so that they decayed quickly. I then calculated the K matrix using 'place' and the F matrix using $F = [C(A + BK)^{-1}B]^{-1}$ to get good tracking.

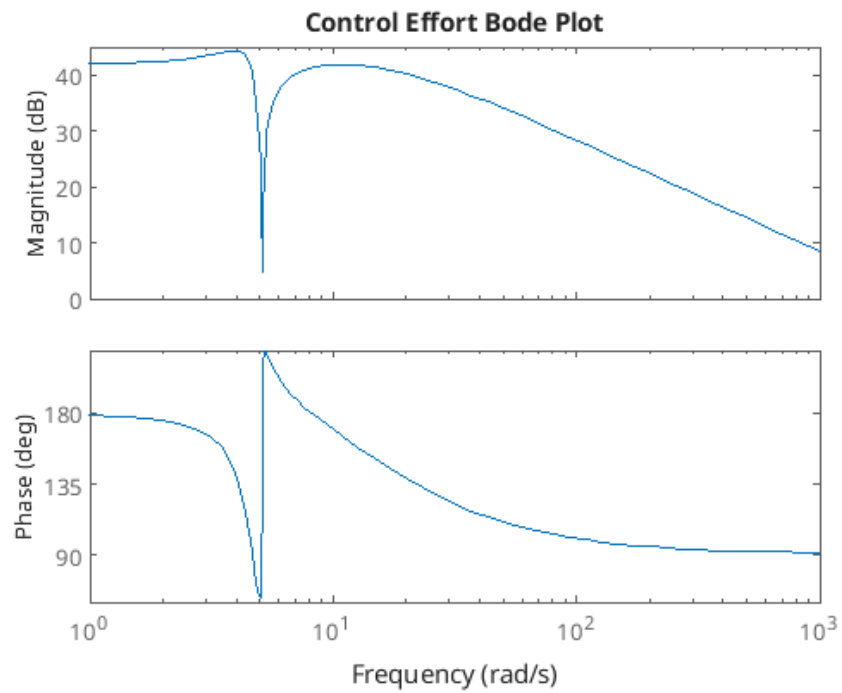
```

1 [A,B,C,D] = tf2ss(num, den);
2
3 disp(eig(A));
4
5 desiredPoles = [-0.615 + 4.5997i, -0.615 - 4.5997i, -10, -12];
6 K = place(A, B, desiredPoles);
7 F = inv(C*inv(-A+B*K)*B);
8
9 sysCL = ss(A-B*K, B*F, C, D);
10 figure(2);
11 bode(sysCL);
12 title('Close Loop Tracking Frequency Response');
13 grid on;
```

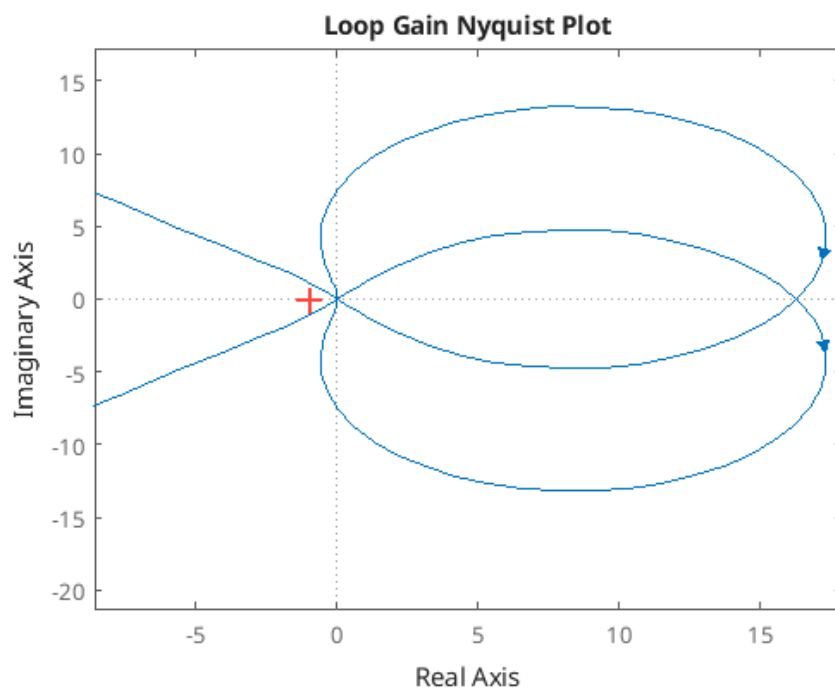
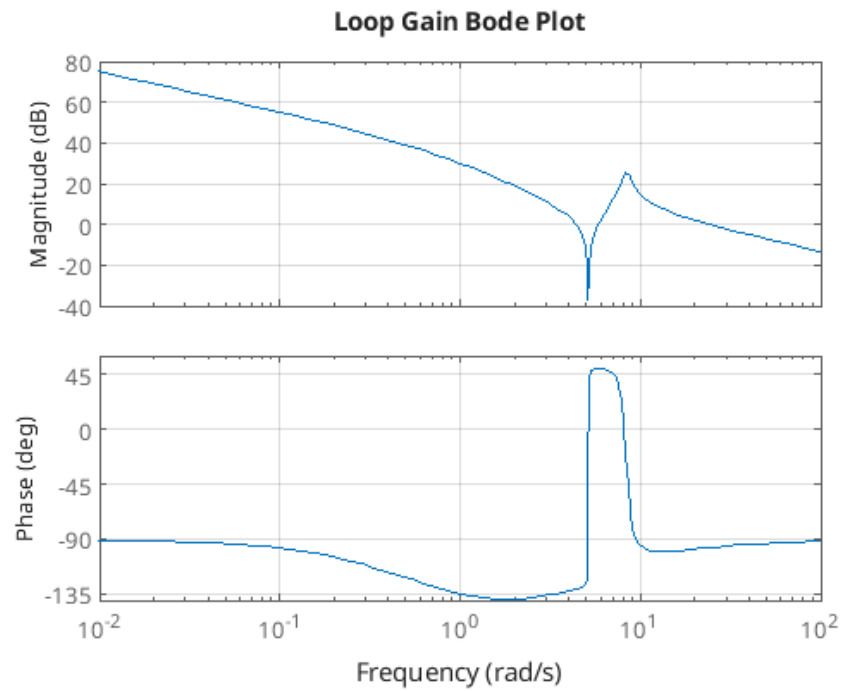
Then I plotted the closed loop frequency response:



The closed loop frequency response from reference input to plant input:



The (negative) loop gain Bode and Nyquist plots:

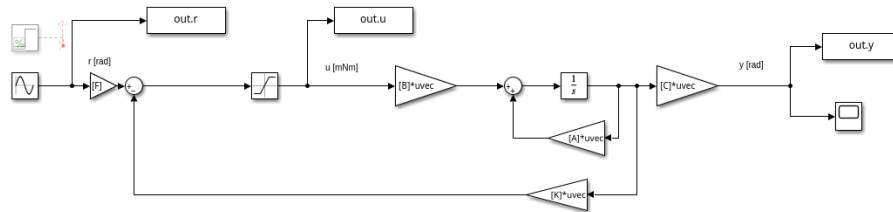


So the stability margin here is not as good as it was in the previous project, here the phase margin is 49 and the gain margin is infinite for the loop gain. In the previous project the phase margin was 108 degrees with infinite gain margin but in that case I had used a compensator to completely cancel the resonances so while the closed loop behavior would have been better, there would be instabilities elsewhere.

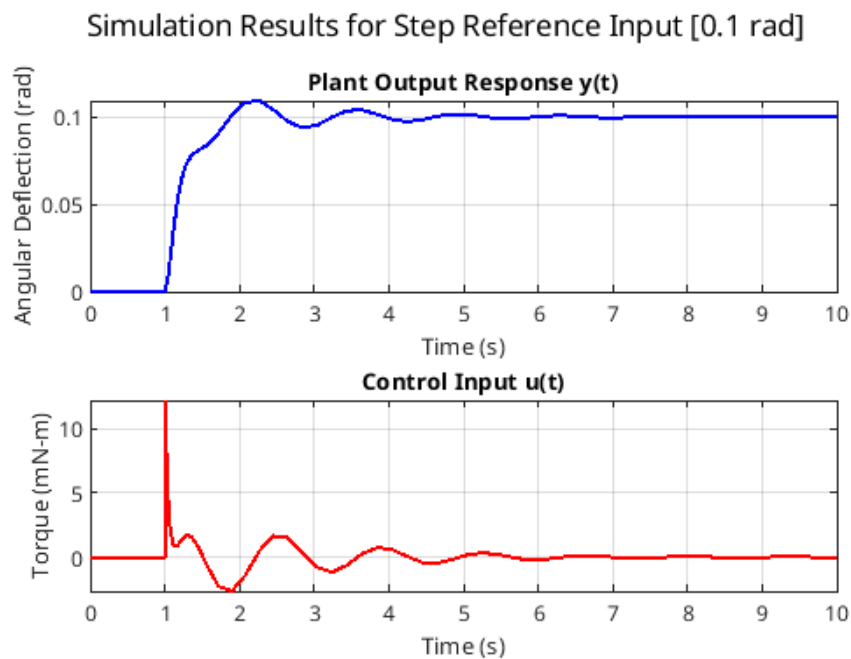
3)

[20 pts] Simulate the state variable feedback controller design in part 2 including a saturation at the plant input corresponding to $\pm 67 \text{ mN} \cdot \text{m}$ limits on the reaction wheel motor actuator. Use sinusoidal reference inputs of 0.1 and 1.0 Hz, at amplitudes small enough to avoid actuator saturation. Also use a step input of height 0.1 rad. Show plant output and plant input responses. Compare them with the frequency responses from part 2.

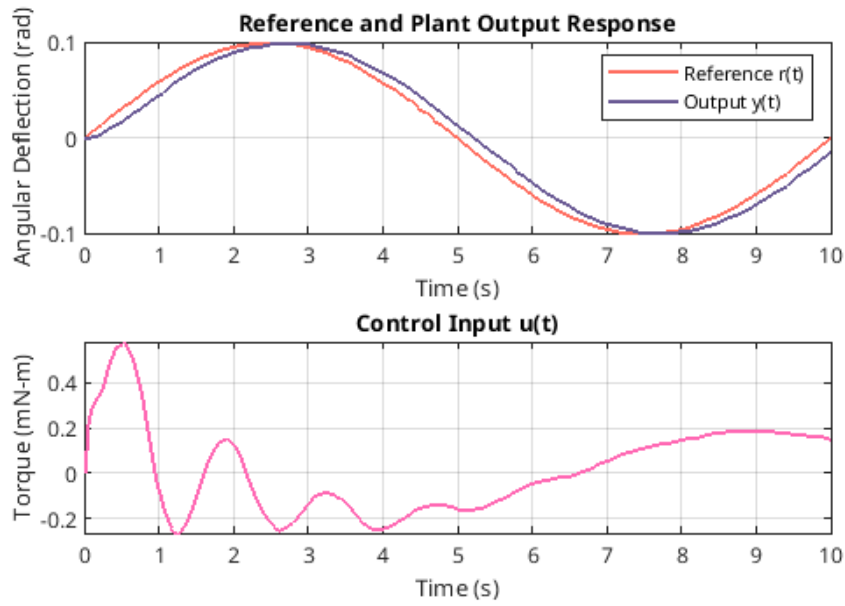
I created the simulation using the Simulink extension of Matlab, I built the system using gains and integrators as you can see in the block diagram below:



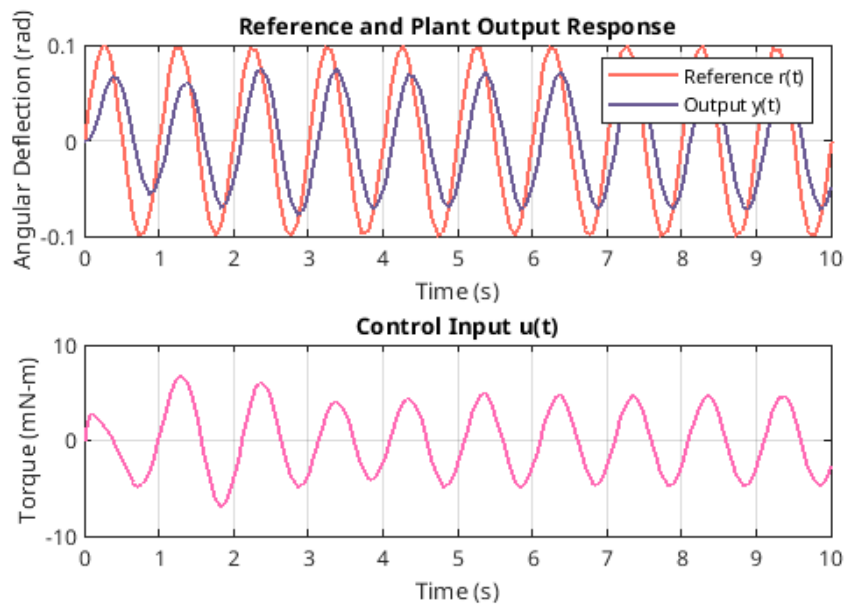
Then I simulated a step reference input of 0.1 rad and a sine reference input of 0.1 rad at 0.1 and 1 Hz and plotted the results:



Simulation Results for Sine Reference Input [0.1 rad at 0.1 Hz]



Simulation Results for Sine Reference Input [0.1 rad at 1 Hz]



You can see that the tracking is pretty close at 0.1 Hz and matches up with the 70% amplitude and 45 degree phase lag at 1 Hz that we would expect from the closed loop frequency response calculated earlier.

4)

[20 pts] Design a full state observer so that state observation errors have settling times (5 percent) of 3 sec. or less. Plot the (negative) loop gain (loop cut at plant input) for the combined observer/state estimate feedback controller, in both Bode and Nyquist formats, using both the analytic and empirical model for the plant. Comment on the stability margins of your combined observer/state feedback controller design. Redesign the observer, if necessary, to retain 40 degrees of phase margin and 10 dB of gain margin.

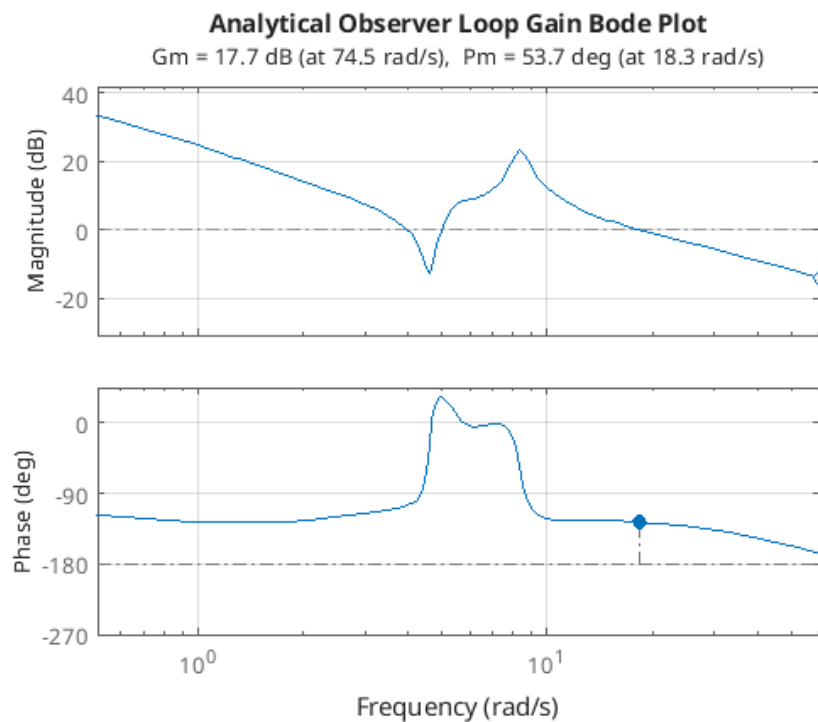
In order to design the observer, I just started by multiplying the closed loop poles by 5 as a rule of thumb and then calculated the settling time using the $t_s = \frac{3}{a}$ approximation and got about 1 second for the dominant poles which fits the requirements. Then I used the 'place' function to get the observer gain matrix L and created and plotted the results using the following code:

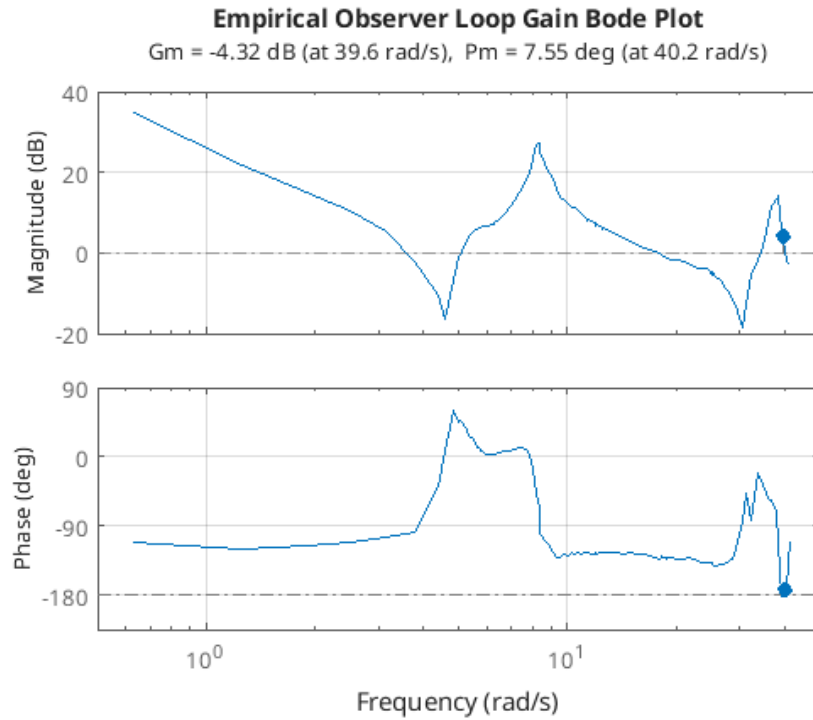
```

1 scalingFactor = 5;
2 desiredObserverPoles = [scalingFactor*-0.615 + 4.5997i, scalingFactor*-0.615 - 4.5997i,
   scalingFactor*-10, scalingFactor*-12];
3
4 disp(desiredObserverPoles);
5
6 L = place(A', C', desiredObserverPoles)';
7
8 obsSys = ss(A-B*K-L*C, L, -K, 0);
9 analG = ss(A, B, C, D);
10
11 obsLGsys = -obsSys*analG;
12 figure(9);
13 margin(obsLGsys);
14 title('Analytical Observer Loop Gain Bode Plot');
15 grid on;
16 xlim([0.5, 50]);
17
18 figure(10);
19 nyquist(obsLGsys);
20 title('Analytical Observer Loop Gain Nyquist Plot');
21
22 empG = frd(10.^(pResponse(:,1)/20) .* exp(1j * deg2rad(pResponse(:,2))), pResponse(:,3))
   ;
23
24 empObsLGsys = -obsSys*empG;
25 figure(11);
26 margin(empObsLGsys);
27 title('Empirical Observer Loop Gain Bode Plot');
28 grid on;
29 xlim([0.5, 50]);
30
31 figure(12);
32 nyquist(empObsLGsys);
33 title('Empirical Observer Loop Gain Nyquist Plot');

```

The results are below:



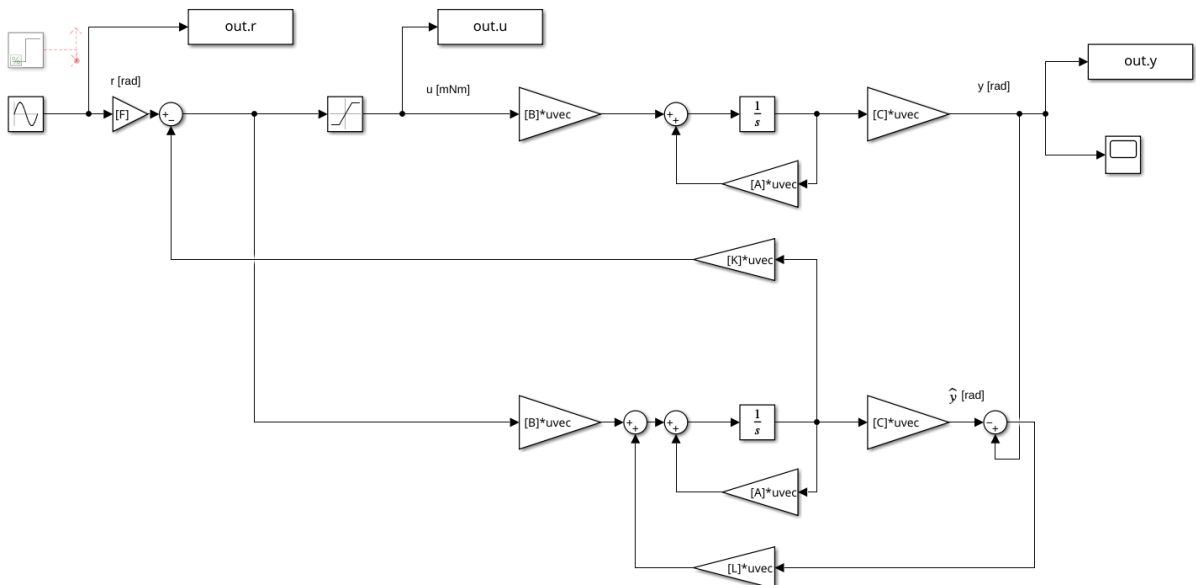


You can see the analytical and empirical observer systems both line up pretty well and the phase and gain margin meet the requirements.

5)

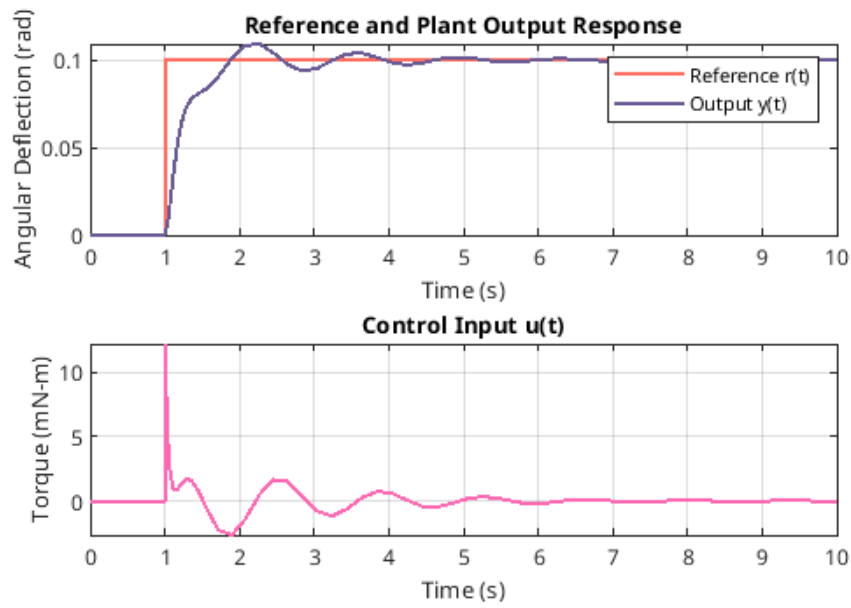
[20 pts] Simulate the combined observer/state feedback control system, using the same reference inputs as in part 3. Comment on the effects of adding an observer in your state feedback control design.

I started by creating the observer system in Simulink, using the system before but adding an observer path and connecting K to \hat{x} instead of x .

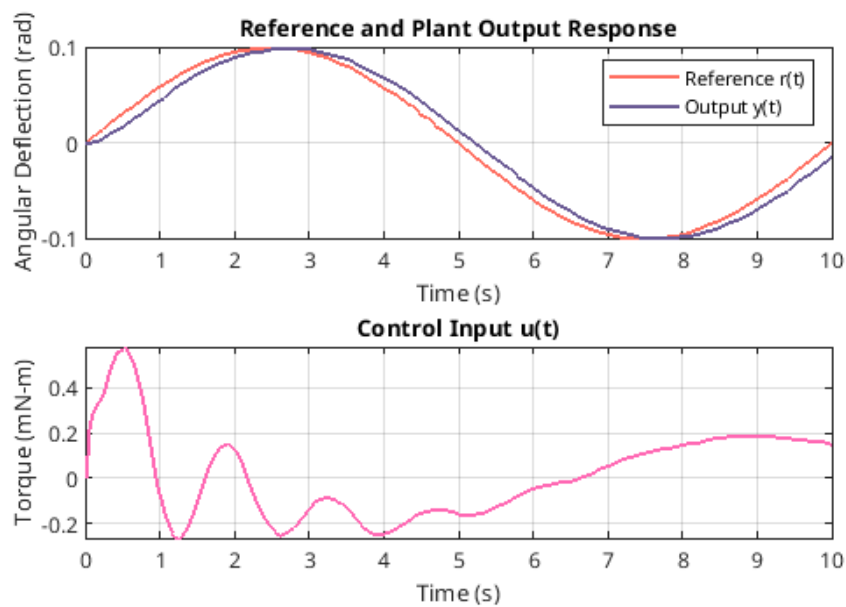


The I simulated and plotted the response to the sine and step inputs.

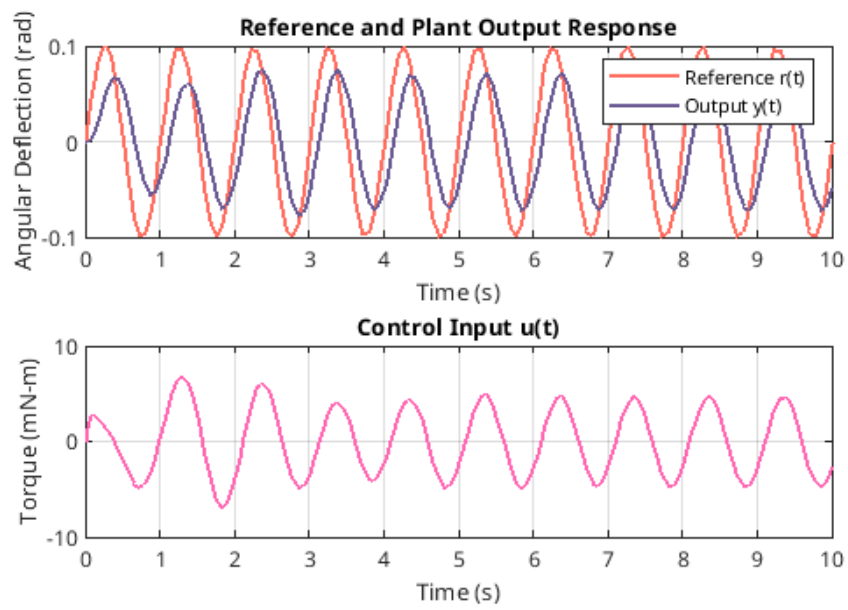
Simulation Results for Sine Reference Input [0.1 rad at 0.1 Hz]



Simulation Results for Sine Reference Input [0.1 rad at 0.1 Hz]



Simulation Results for Sine Reference Input [0.1 rad at 0.1 Hz]



You would expect the frequency response to be exactly the same as without the observer since the states are known in this case except that you would expect some initial tracking error as the observer states converge to the real states. This is consistent with the results.