Digital Control of a Maneuvering Submarine

ELE/MCE 503 Final Project
Fall 2019
Due Thursday, December 19

1 Introduction

The motion of a submarine is influenced by the angles of several control surfaces (inputs) and the goal is to achieve desired motion along several degrees of freedom (outputs). Thus, submarine control requires a multiple-input multiple-output (MIMO or multivariable) control system. The mathematical description of a submarine is a set of nonlinear differential equations, or equivalently, a nonlinear state-space model. It is customary to linearize the model output about an operating point such as a constant-velocity trajectory, and to control deviations from this operating point. It may be necessary to design several linear control systems, each for a different velocity, and put them together with a gain-scheduling algorithm as is done in [3]. In this project, we consider only the design of a single linear multivariable digital tracking system.

2 Description of the Plant

The material from this section is taken from [1,2]. The linearized state-space model for the submarine is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

where the state variables are:

$x_1 =$ forward velocity, u, ft/sec
$x_2 =$ lateral velocity, v, ft/sec
$x_3 =$ vertical velocity, w, ft/sec
$x_4 =$ roll rate, p, deg/sec
$x_5 =$ pitch rate, q, deg/sec
$x_6 =$ yaw rate, r, rad/sec
$x_7 =$ roll angle, degrees
$x_8 =$ pitch angle, degrees.

The inputs are:

$u_1 =$ bow/fairwater planes, degrees
$u_2 =$ rudder deflection, degrees
$u_3 =$ port stern plane deflection, degrees
$u_4 =$ starboard stern plane deflection, degrees

The outputs are:

$y_1 =$ roll angle, degrees
$y_2 =$ pitch angle, degrees
$y_3 =$ yaw rate, deg/sec
$y_4 =$ depth rate, ft/sec.

The numerical values for the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are given in the Matlab function `plant_param.m`, which is available on the course website. The modeled submarine is about 400 ft long. The system variables are indicated in the following figure:

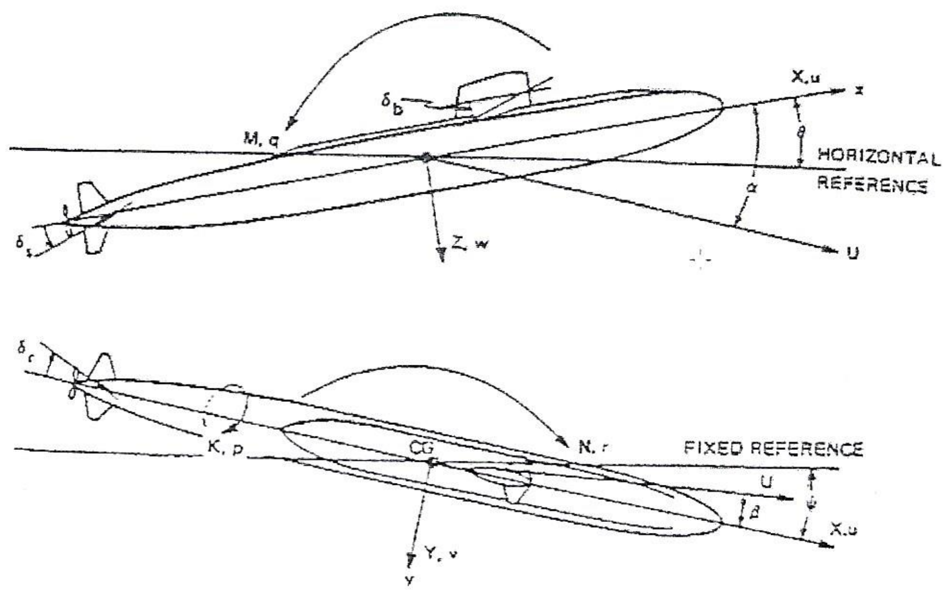


Figure 1: Sketch showing positive directions of axes, angles, velocities, forces, and moments. From [2].

3 Tracking System Design

We begin with the design and analysis of a full-state feedback tracking system and then consider using an observer to estimate the plant state vector from the input and output signals of the plant. **The sampling interval for this project is** `T=0.1`.

3.1 Full-State Feedback

A tracking system is to be designed to follow step commands for each of the four plant outputs. Thus, the discrete-time additional dynamics must have an eigenvalue equal to 1 on each of the four tracking errors. This results in the additional dynamics block consisting of four parallel digital integrators, `phia=eye(4)`, `gammaa=eye(4)`.

It is not uncommon for high-order control systems to exhibit more than one settling time. That is, different signals settle in different amounts of time. The control systems encountered in class this semester were all designed using a single settling time, T_S, which governed all of the state variables. In this project, most of the closed-loop poles may be chosen to have a setting time `Ts1`. However, to avoid actuator saturation, at least one of the closed-loop poles must be chosen with a longer settling time, `Ts2`. For the particular submarine considered in this project, realistic values for the two settling times are:

$$T_{S_1} = 25 \text{ sec} \quad \text{and} \quad T_{S_2} = 85 \text{ sec}. \tag{1}$$

Consider the following choice of closed-loop poles:

```
cp=-0.1969+j*0.3130
spoles=[-0.5049 -0.4511 cp conj(cp) s7/Ts1 s1/Ts2].
```

1. **Explain how the choice of sampling interval given at the beginning of this section agrees with the rule-of-thumb given in class for choosing the sampling interval of a digital tracking system.**

2. **Explain how the choice of `spoles` given above agrees with the information in the *Rules for Selecting Pole Locations* handout. For each entry in the `spoles` vector explain which rule is being used.**

3. **Call `tsd` two times, once with `place` and a second time with `rfbg`. Examine the stability robustness of each system and explain which of these systems has adequate robustness to be useful as a real-world submarine control system. Provide a printout of your Matlab code.**

3.2 Observer-Based

[Unless otherwise specified, use the feedback gain matrix calculated using `rfbg` for the observer-based tracking systems in this section.] The pole-placement approach to the calculation of observer gains amounts to choosing the observer pole locations. However, when the observer uses more than one measured plant output, there are an infinite number of observer gain matrices that result in the specified observer pole locations. A given pole-placement program selects a particular gain matrix from this infinite set and that selection has an influence on the stability margins of the resulting observer-based tracking system. For this project, the following choice of observer pole locations has been found to work well:

```
opoles = [-.0384 -0.251 s6/(Ts1/5)].
```

4. **Explain how this choice of observer pole locations agrees with the information in the *Rules for Selecting Pole Locations* handout. For each entry in the `opoles` vector explain which rule is being used.**

$\delta_1 = 0.4784$

$\delta_2 = 0.5337$

5. **Using the given `opoles` vector, calculate observer gain matrices using `place` and using a new function `obg_ts`, which is available on the course web site. Compare the stability robustness bounds for the resulting observer-based tracking systems. Compare these with each other and with the robustness bounds for the state-feedback tracking system from Part 2. Note that the function `rb_tsob` calculates the stability robustness bounds for any observer-based tracking system. Provide a printout of your Matlab code.**

6. **Draw a block diagram of the complete observer-based tracking control system used for this project. Show all equations used to implement the digital tracking system, A/D and D/A converters, and a block containing the hardware plant.**

4 Simulations

The performance of the observer-based tracking system is demonstrated by simulating a combined maneuver in which step commands for each plant output are applied simultaneously at $t = 5$ sec. The commands are as follows: roll angle (y_1) is to be maintained at 0 deg, pitch (y_2) is commanded to 1 deg, yaw rate (y_3) is commanded to 1 deg/sec, and depth rate (y_4) is commanded to -0.5 ft/sec.

The simulations for this section may be obtained using the Simulink model `project_sim.slx`, which is available on the course web site. Note that the plots may be obtained with the plotting script `ts_dobp.m`. The simulations are to be performed for 200 sec. Put two separate graphs on each Matlab plot.

7. **Consider the observer-based tracking system designed using `rfbg` and `obg_ts` with `spoles` and `opoles` given above. Compare the plant outputs and inputs with those shown in Figs. 2 and 3, which were obtained using an LQG/LTR control system in [1,2].**

Suppose you were to do this project only with the standard pole-placement tool available in Matlab, which is the `place` function:

8. **Calculate the feedback and observer gains using the `place` function with the given vectors for `spoles` and `opoles`. Compare robustness bounds of this system with those of the observer-based tracking system designed using the new functions `rfbg` and `obg_ts`. Provide a printout of your Matlab code.**

9. **Simulate the tracking system obtained using only `place`. Compare the output and input plots with those of the observer-based tracking system designed using `rfbg` and `obg_ts`. Do the simulation results for the `place` tracking system give any cause for concern? Is this tracking system suitable for hardware testing?**

References

[1] R.J. Martin, L. Valavani, and M. Athans, "Multivariable Control of a Submersible using the LQG/LTR Design Methodology," in *Proc. American Control Conference,* pp. 1313-1324, June 1986.

[2] R.J. Martin, "Multivariable Control System Design for a Submarine Using Active Roll Control," Engineers Thesis, MIT, 1985. Available on line at `http://dspace.mit.edu/bitstream/handle/1721.1/15276/13511288.pdf?sequence=1`

[3] K.A. Lively, "Multivariable Control System Design for a Submarine," Engineers Thesis, MIT, 1984. Available on line at `http://dspace.mit.edu/bitstream/handle/1721.1/15355/12192029.pdf?sequence=1`

Corey Murphy

MCE 571

12/19/19

Digital Control of a Maneuvering Submarine

Editor - C:\Users\corey\Documents\MATLAB\FinalProject503.m

FinalProject503.m

```
% Corey Murphy %
% MCE 503 %
% Digital Control of a Maneuvering Submarine %

load sroots

A=[-3.8269e-02 -2.1964e-02 -2.7533e-03 -3.3173e-04 2.0734e-03 5.5394e-02 0 5.1285e-06
    1.1417e-03 -1.5939e-01 -3.3786e-05 -2.3578e-02 2.8353e-03 -2.6860e-01 2.2745e-03 -2.5914e-05
    -4.7476e-04 1.3910e-03 -9.6526e-02 -2.7949e-02 2.1163e-01 7.6140e-04 0 1.3221e-04
    1.3945e-02 -6.6430e-01 -8.0931e-02 -4.3452e-01 -2.5262e-01 -2.1920e-02 -1.6030e-01 1.8264e-03
    7.1418e-05 -2.5929e-04 7.8117e-02 -1.1406e-02 -4.0815e-01 -7.7327e-04 0 -2.4985e-03
    -1.5782e-03 -1.1622e-01 3.4035e-04 -8.0011e-03 2.2809e-03 -3.8201e-01 2.5893e-04 -2.9501e-06
    0 0 0 1 1.1328e-02 -1.0538e-01 -4.9352e-10 -1.2635e-02
    0 0 0 0 9.9427e-01 1.0689e-01 1.2494e-02 0];
B=[-1.2666e-03 -1.5279e-03 9.8625e-05 9.8625e-05
    0 6.0491e-02 -3.6976e-03 3.6976e-03
    -2.5204e-02 -3.8763e-08 -2.1483e-02 -2.1483e-02
    0 6.3847e-02 2.6060e-01 -2.6060e-01
    1.3873e-02 7.3256e-07 -2.9781e-02 -2.9781e-02
    0 -8.7846e-02 -4.2094e-04 4.2094e-04
    0 0 0 0
    0 0 0 0];
C=[0 0 0 0 0 0 .1 0
    0 0 0 0 0 0 0 .1
    0 0 0 0 -1.0749e-01 9.9984e-01 4.6827e-09 1.3316e-03
    1.0539e-01 -1.0629e-01 9.8873e-01 0 0 0 2.7119e-02 -8.4843e-01];

Ts1=25;
Ts2=85;
T=0.1;

% Full-State Feedback (Problem #3) %

cp=-0.1969+i*0.3130;
spoles=[-0.5049 -0.4511 cp conj(cp) s7/Ts1 s1/Ts2];
zpoles=exp(T*spoles);
[phi,gamma]=c2d(A,B,T);
phia=eye(4);
gammaa=eye(4);

[K1,K2,delta1,delta2]=tsd(phi,gamma,C,phia,gammaa,zpoles,T,'place')
[K1,K2,delta1,delta2]=tsd(phi,gamma,C,phia,gammaa,zpoles,T,'rfbg')

% Observer-Based (Problem #5)%

opoles=[-0.0384 -0.251 s6/(Ts1/5)];
zopoles=exp(T*opoles);

L=place(phi',C',zopoles)'
[L,delta1,delta2]=obg_ts(phi,gamma,C,phia,gammaa,K1,K2,zopoles,T)
[delta1,delta2]=rb_tsob(phi,gamma,C,phia,gammaa,K1,K2,L,T)

%project_sim
ts_dobp
```

Problem #1:

$$T = \min\left(\frac{T_{s1}}{20(n+q)}, \frac{\pi}{5\beta_{max}}\right) \rightarrow n+q = 8+4 = 12$$

$T_{s1} = 25$

$\beta_{max} = 0.313$

$$\frac{T_{s1}}{20(n+q)} = \frac{25}{20(12)} = 0.104 \checkmark$$

→ $T = 0.1$, agrees with rule of thumb.

$$\frac{\pi}{5\beta_{max}} = \frac{\pi}{5(0.313)} = 2.007 \times$$

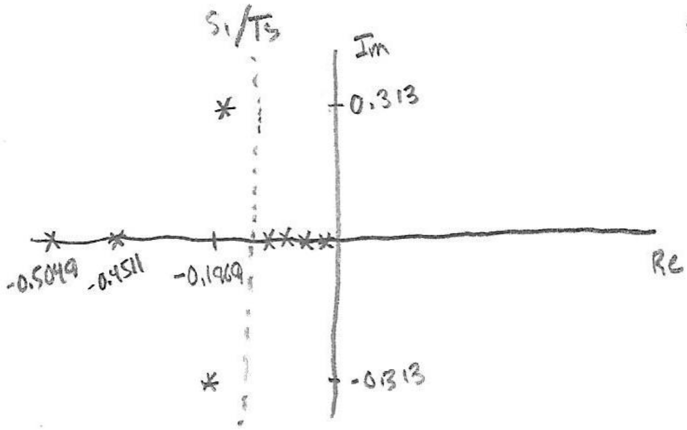
The sampling interval of T = 0.1 agrees with the rule-of-thumb given in class. In order to choose a sampling interval for a digital tracking system, the formula $T = \min(Ts1/[20(n+q), \pi/5\beta_{max}]$ must be considered. Plugging in 0.313 for β_{max} yields a sampling time of T = 2.007. Plugging in 25 for Ts1, 8 for n and 4 for q yields a sampling time of T = 0.104, the smaller value of the two formulas. This value matches the original given sampling time of T = 0.1, confirming that it does in fact agree with the rule-of-thumb.

Problem #2

2. $eig(A) = -0.1969 \pm 0.313i, -0.5049, -0.4511, -0.0714, -0.0142, -0.0429, -0.0407$

$s_1/T_s = -0.1848$

$spoles = [-0.5049 \quad -0.4511 \quad -0.1969 \pm 0.313i \quad \frac{s_7}{T_{s1}} \quad \frac{s_1}{T_{s2}}]$



The first step when determining the spoles for this problem would be to determine the eigenvalues of the A matrix. There are eight eigenvalues for this matrix, six with negative real parts and two containing negative real parts and imaginary parts. The eigenvalues of -0.5049 and -0.4511 are chosen as spoles as they are negative real parts to the left of s1/Ts1 (s1/Ts1 = -0.1848). These poles are also considered sufficiently damped plant poles. This is rule #2 from regulator poles. The eigenvalues -0.1969+0.313i and -0.1969-0.313i are also chosen as spoles. These poles are also considered sufficiently damped plant poles. Despite being complex, no added damping is required because they are to the left of s1/Ts1 (regulator rule #2). The four remaining eigenvalues are all negative real parts but remain to the right of s1/Ts1, meaning these will instead be accounted for by adding them to the Bessel cluster. Because this is an observer-based tracking system, n+q = 8+4 = 12 spoles will be required. One closed-loop poles must be chosen with a longer settling time of Ts2 to avoid actuator saturation. This pole is accounted for by scaling s1/Ts2. The remaining poles are accounted for by using scaled Bessel, s7/Ts1 (regulator rule #1).

Problem #3

place

K1 =

-7.2980	1.3033	-6.5174	-2.2193	27.6763	1.6926	-1.0742	16.2011
6.1534	0.2754	-0.9930	0.0301	0.0297	-2.6667	-0.1934	0.3971
0.8798	-0.9477	-8.5341	1.5119	-2.5616	1.8097	0.5455	2.9357
1.1801	1.8376	-8.3536	0.0008	-1.8826	2.6120	-0.0882	2.7949

K2 =

-0.2285	1.2368	0.1197	-0.1350
-0.0318	-0.0663	-0.2527	-0.0312
0.2071	-0.6976	0.0189	-0.1863
-0.0345	-0.7727	0.0080	-0.1750

delta1 =

0.2324

delta2 =

0.2489

rfbg

delta values:

0.4165	0.5386
0.4180	0.5483
0.4774	0.6248
0.5177	0.6548

K1 =

47.4060	-1.7602	-21.8209	0.0274	27.1351	-4.8188	-0.6127	25.2090
27.1003	-1.2350	-2.6111	0.1767	2.4614	-3.9844	-0.1973	2.1608
-69.9461	3.9405	-4.4437	0.5373	-5.7534	5.6405	0.2283	0.2971
-44.4292	5.2394	-0.3599	-0.3151	-3.3479	3.8440	-0.1895	-2.3096

K2 =

-0.1115	0.3913	-0.6218	-0.5300
-0.0045	0.0997	-0.5200	-0.0223
0.0488	-0.6609	0.7424	-0.1792
-0.0834	-0.3300	0.4780	0.0837

delta1 =

0.5177

delta2 =

0.6548

The stability robustness bounds that were determined using the place command would not be considered acceptable for hardware testing. With delta1 = 0.2324 and delta2 = 0.2489, these bounds fall well below the acceptable hardware testing target of 0.5. The stability robustness bounds that were determined using rfbg were much better than the previous bounds. With delta1 = 0.5177 and delta2 = 0.6548, both bounds are greater than 0.5 and would be considered adequate to be used in a real-world submarine control system.

Problem #4

-0.0384 and -0.251 were both determined to be zeros and resided in the left half plane, making them a good choice for opoles (observer poles rule #2). The remaining 6 poles would be determined using normalized Bessel poles scaled by the desired observer settling time (observer poles rule #1). In this case, the observer settling time is targeted to be five times faster than the original settling time of Ts1 = 25. This makes Tso1 = 5. These normalized Bessel poles are represented by s6/(Ts1/5) or s6/Tso1.

Problem #5

rfbg and place with rb_tsob

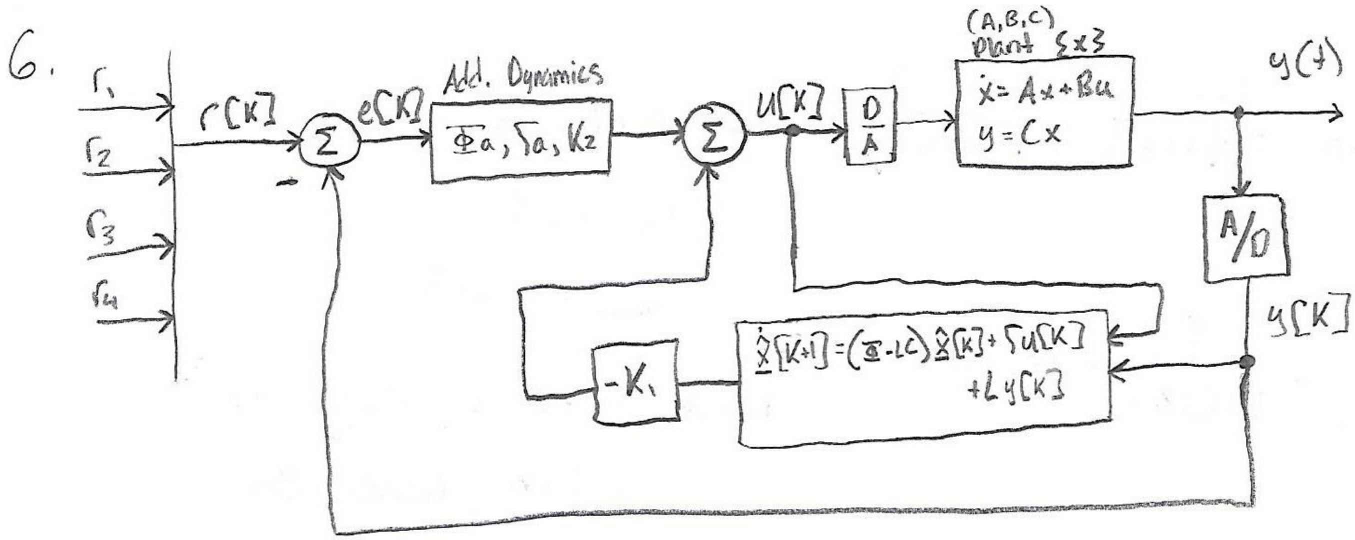
```
L =

    0.0772    0.1875    0.0126    0.0112
   -0.3441    0.1279   -0.0295   -0.0026
   -0.2318    1.3222   -0.0191    0.1248
    1.6826   -0.8642    0.0110    0.0214
    0.2798    1.4022    0.0235    0.0132
   -0.0439    0.2132    0.0949    0.0308
    1.7118   -0.2893    0.0029    0.0211
    0.6541    1.8057    0.0344    0.0001


delta1 =

    0.4373


delta2 =

    0.4320
```

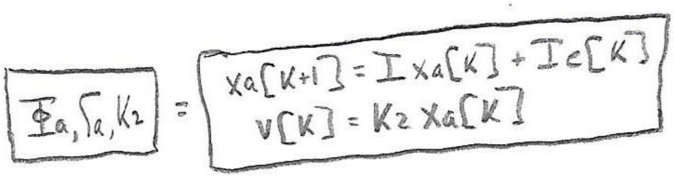
rfbg and obg_ts

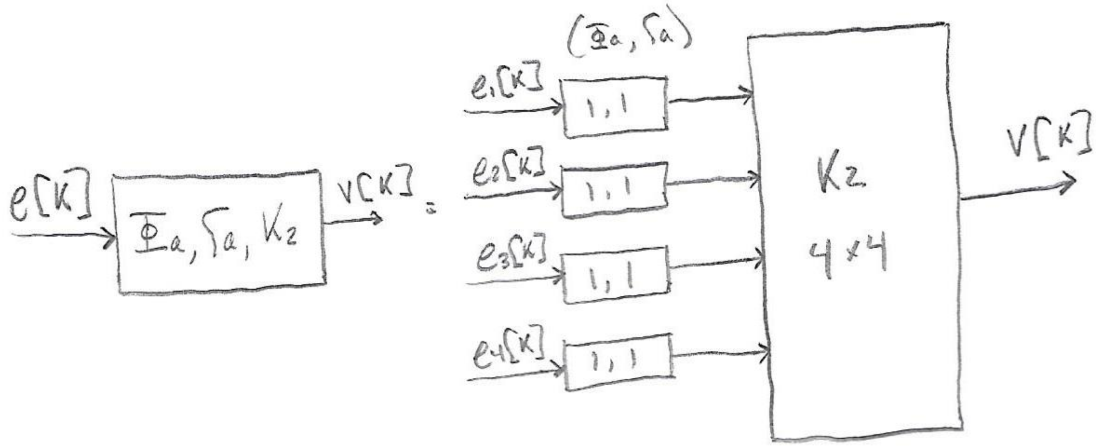
```
delta values:
0.4631   0.5440
0.4577   0.5532
0.4404   0.5627
0.4863   0.5952

L =

   -0.0994   -1.6430    0.0323   -0.0736
   -0.1000    0.0171   -0.0941    0.0632
    0.1590    2.8059   -0.0282    0.1829
   -0.0454  -10.6364    0.1951   -0.6528
    0.1717    3.3283    0.0012    0.0763
   -0.0676   -0.7830    0.1030   -0.0804
    0.5606  -13.3501    0.3598   -0.7174
    0.0860    2.8054   -0.0539    0.0199


delta1 =

    0.4863


delta2 =

    0.5952
```

The stability robustness bounds that were determined using the place command were delta1 = 0.4373 and delta2 = 4320. These bounds were not ideal when compared with the bounds determined using the obg_ts function but weren't terrible. They were very close to the acceptable delta value of 0.5. The bounds determined using the obg_ts function were delta1 = 0.4863 and delta2 = 0.5952. Delta1 is nearly at an acceptable value while delta2 is well above 0.5. The robustness bounds determined for the state-feedback tracking systems in part two were superior for the rfbg function but not nearly as good for the place function.

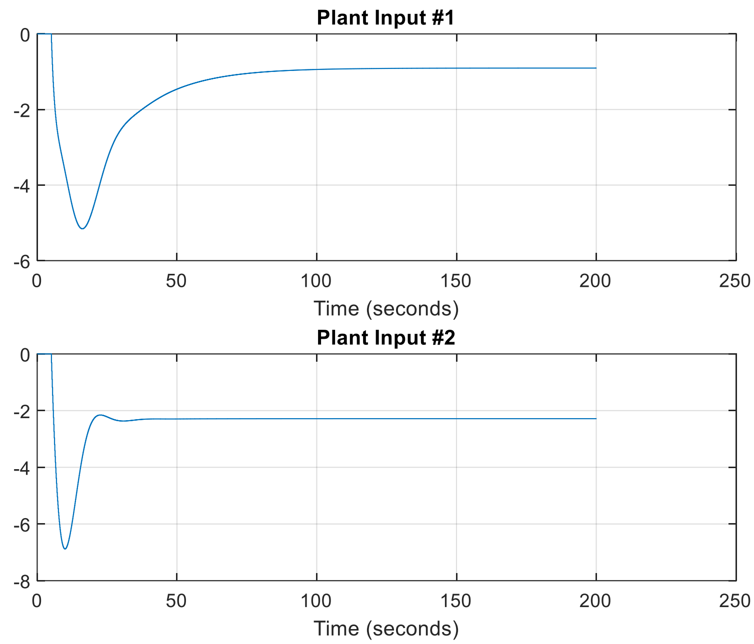
Problem #6

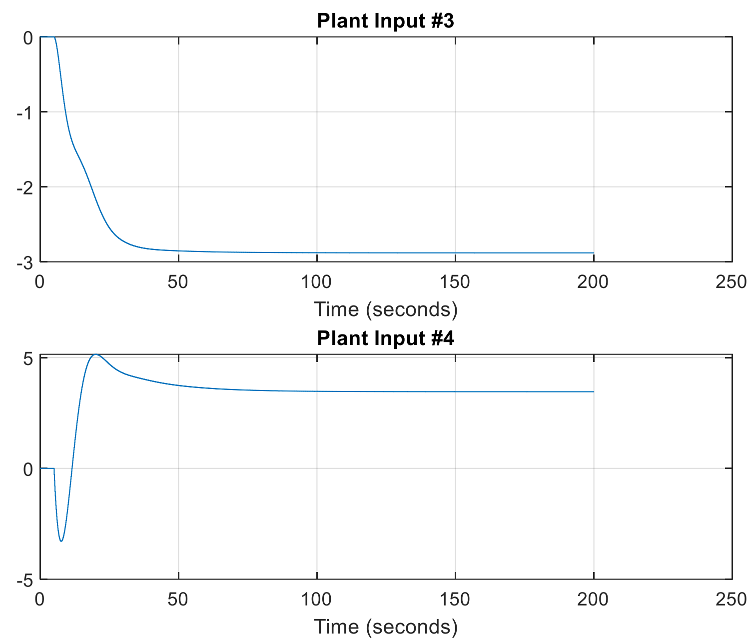

6.
r_1
r_2
r_3
r_4
r[K]
e[K]
Add. Dynamics
Φ_a, Γ_a, K_2
u[K]
D/A
(A,B,C)
Plant {x}
$\dot{x} = Ax + Bu$
$y = Cx$
y(t)
A/D
y[K]
$-K_1$
$\hat{x}[K+1] = (\Phi - LC)\hat{x}[K] + \Gamma u[K] + Ly[K]$


Φ_a, Γ_a, K_2
=
$x_a[K+1] = I x_a[K] + I e[K]$
$v[K] = K_2 x_a[K]$

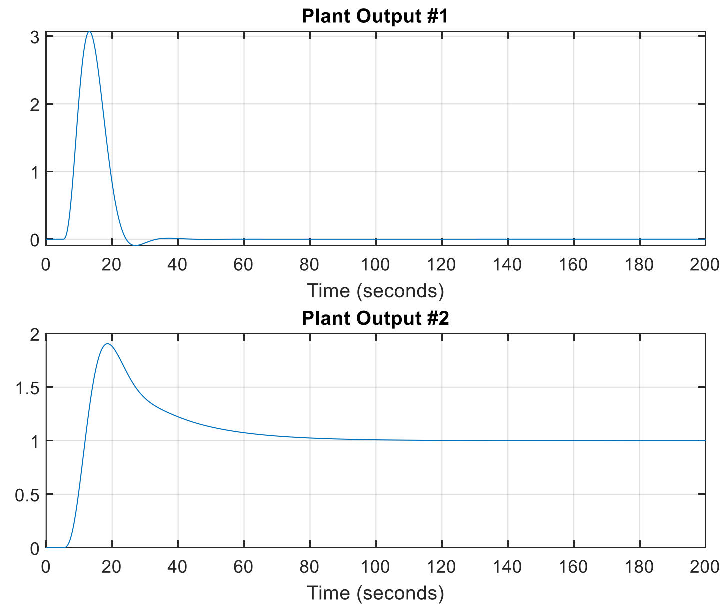

e[K]
Φ_a, Γ_a, K_2
v[K]
=
(Φ_a, Γ_a)
$e_1[K]$
$e_2[K]$
$e_3[K]$
$e_4[K]$
1, 1
1, 1
1, 1
1, 1
K_2
4×4
v[K]

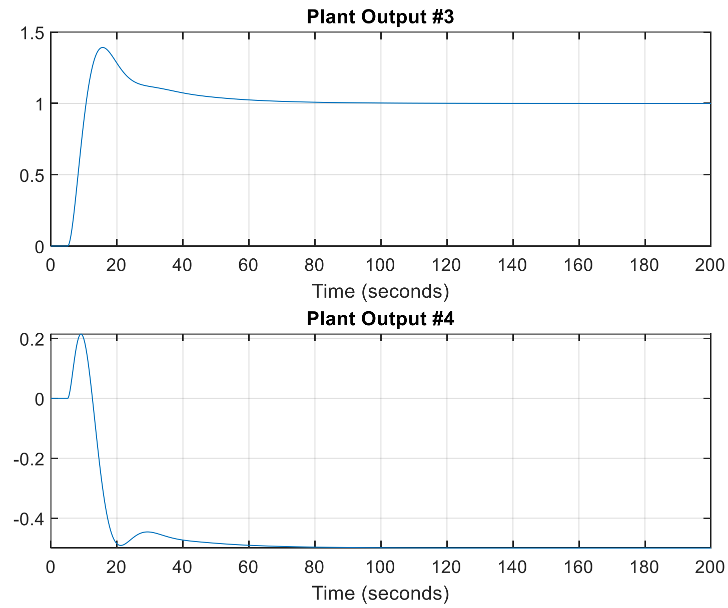
Problem #7





The graph of input #1 from the above results has an initial negative overshoot unlike the input #1 graph from figure 3 where an initial positive overshoot is displayed. The graph of input #2 from above looks very similar to the graph of input #2 in figure 3. The graph of input #3 from above also looks similar to the graph of input #3 from figure 3, but the graph above appears to have slightly less oscillation. The graph of input #4 has a small initial negative overshoot where the graph of input #4 from figure 3 has a small positive overshoot.





The graph of output #1 above only has an initial positive overshoot while the graph of output #1 from figure 2 has an initial negative overshoot and positive overshoot. The graph of output #2 only has an initial positive overshoot before settling. The graph of output #2 from figure 2 has a small oscillation after its initial overshoot before settling. The graph of output #3 above looks very similar to the graph of output #3 from figure 2. The graph of output #4 also looks similar to the graph of output #4 from figure 2 with perhaps slightly less oscillations after the initial negative overshoot.

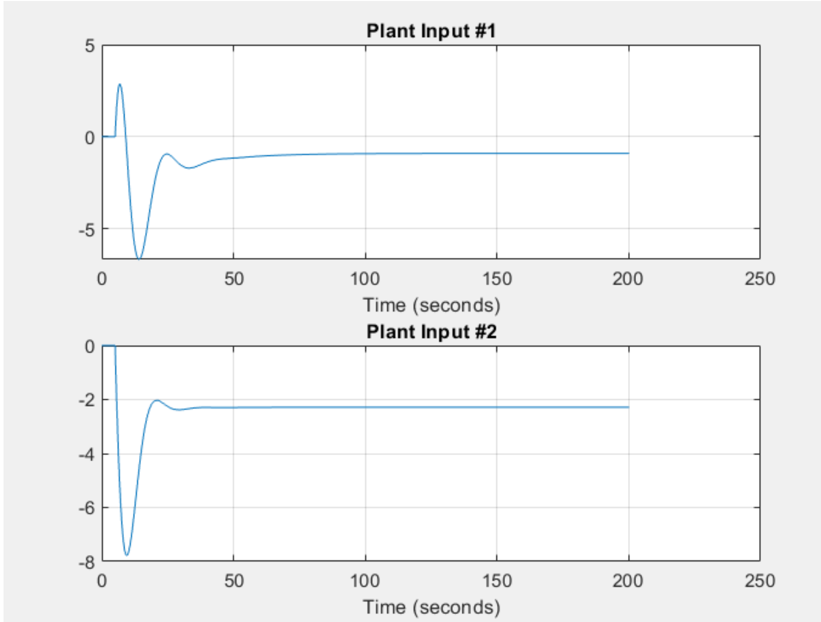
Problem #8

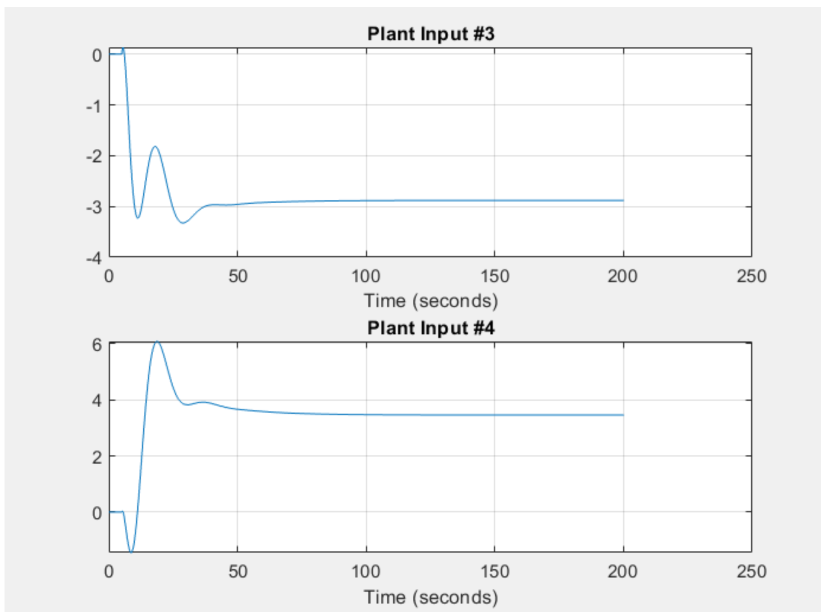
```
K1 =

   -7.2980    1.3033   -6.5174   -2.2193   27.6763    1.6926   -1.0742   16.2011
    6.1534    0.2754   -0.9930    0.0301    0.0297   -2.6667   -0.1934    0.3971
    0.8798   -0.9477   -8.5341    1.5119   -2.5616    1.8097    0.5455    2.9357
    1.1801    1.8376   -8.3536    0.0008   -1.8826    2.6120   -0.0882    2.7949

K2 =

   -0.2285    1.2368    0.1197   -0.1350
   -0.0318   -0.0663   -0.2527   -0.0312
    0.2071   -0.6976    0.0189   -0.1863
   -0.0345   -0.7727    0.0080   -0.1750

delta1 =

    0.2324

delta2 =

    0.2489
```

```
L =

    0.0772    0.1875    0.0126    0.0112
   -0.3441    0.1279   -0.0295   -0.0026
   -0.2318    1.3222   -0.0191    0.1248
    1.6826   -0.8642    0.0110    0.0214
    0.2798    1.4022    0.0235    0.0132
   -0.0439    0.2132    0.0949    0.0308
    1.7118   -0.2893    0.0029    0.0211
    0.6541    1.8057    0.0344    0.0001

delta1 =

    0.2196

delta2 =

    0.2217
```

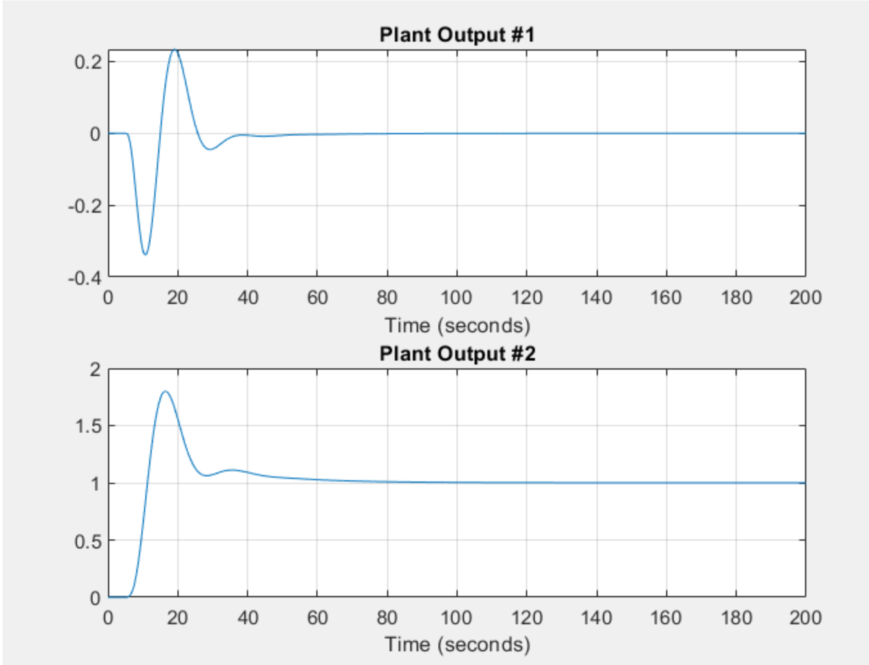
The stability robustness bounds for both observer and feedback using the spoles and opoles are not ideal. All deltas are relatively close to 0.2 which is well below the acceptable value of 0.5. Once again, it appears that the place function is not the best choice for good stability robustness. The observer-based tracking system that was designed using the rfbg function had the best stability robustness bounds at delta1 = 0.5177 and delta2 = 0.6548. The system designed with the obg_ts function had the second-best robustness bounds at delta1 = 0.4784 and delta2 = 0.5337. The observer-based tracking system with by far the worst robustness bounds was the system designed using the place command. This system had bounds of delta1 = 0.2196 and delta2 = 0.2217.

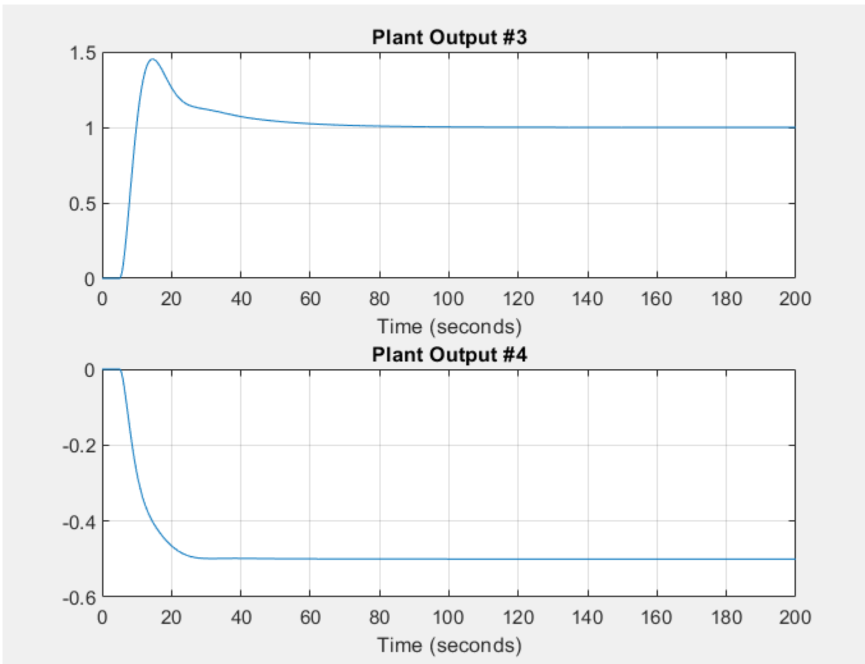
Problem #9





Input #1 appears to settle slightly faster in this system using the place function. The other three inputs for both systems all appear to settle at around the same time. The main difference between the place input graphs and the graphs from the rfbg and obg_ts input graphs is the larger amounts of oscillation that takes place before settling in regards to the system designed using place. This might be cause for concern, but it is difficult to say solely based off of the input graphs.





Many of the place system plant output graphs appear very similar to the output graphs from the rfbg and obg_ts system. The graph of output #1 does appear to have more oscillation than the previous system, but the other three looks just as good if not better. Once again, it is difficult to tell whether or not there should be concern with this system solely based off of its output graphs. However, it is clear from previous problems that the stability robustness of the system using place was not very good when compared with the bounds from the rfbg and obg_ts system. Based on the robustness bounds being less than the recommended 0.5, this control system designed using the place function would not be suitable for hardware testing.