ELE 503 Homework 8

Table of Contents

Noah Johnson

Problem 1

See attached paper.

Problem 2

See attached paper.

Problem 3a (Choosing Ts)

```
load sroots;
A = [0 1 0; -4 -.4 40; 0 0 -4];
B = [0;0;2];
x0=[0.34;0;0]; %Given Initial State
Ts = 4.5; %Given Settling Time

plantPoles = eig(A)
```

```
plantPoles =

  -0.2000 + 1.9900i
  -0.2000 - 1.9900i
  -4.0000 + 0.0000i
```

Note that the system has complex valued poles.

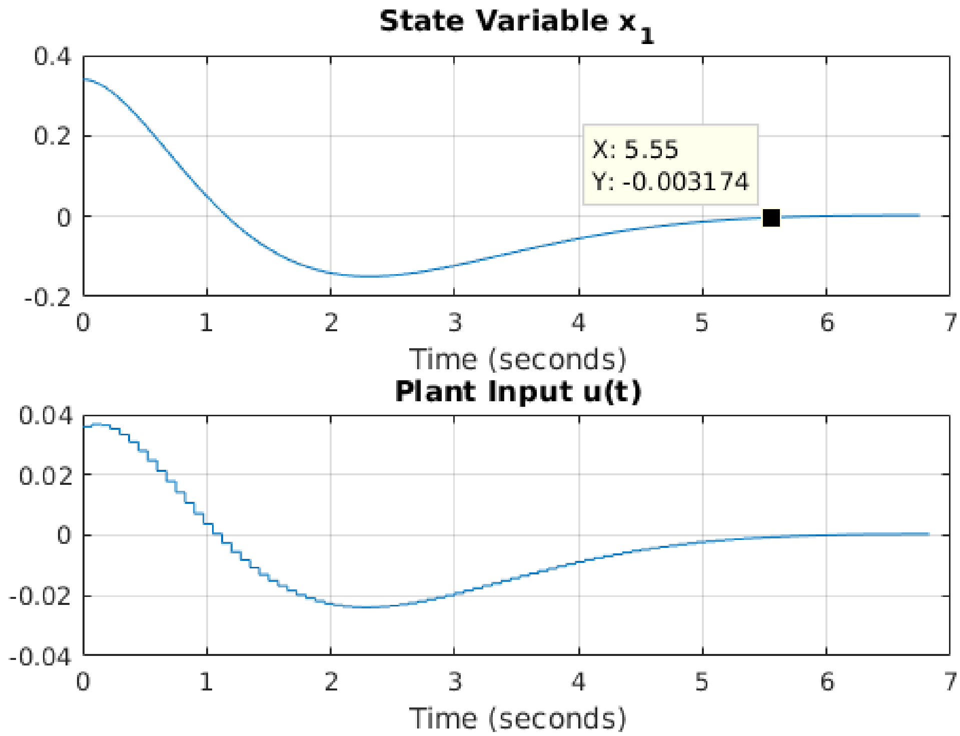
```
beta_max = imag(plantPoles(1));
n = 3;

T = min(pi/beta_max,Ts/(20*n))
```

```
T =

    0.0750
```

Problem 3b.1 (Design of Bessel Pole DSFR)

```
[phi, gamma] = c2d(A,B,T);

sPoles = s3/Ts;
zPoles = exp(T*sPoles);

K = place(phi, gamma, zPoles);
results = sim('reg_dsf','StopTime', '6.75');
tout = results.tout;
u = results.u;
x = results.x;
%run reg_dsfp script, open result
open('BesselPoles.fig')

%Normalized Bessel Poles only
fprintf('------------------- Bessel Poles -------------------\n\n');
dsm(phi,gamma,K)
[del1_1,del2_1] = rb_regsf(phi,gamma,K,T)
```

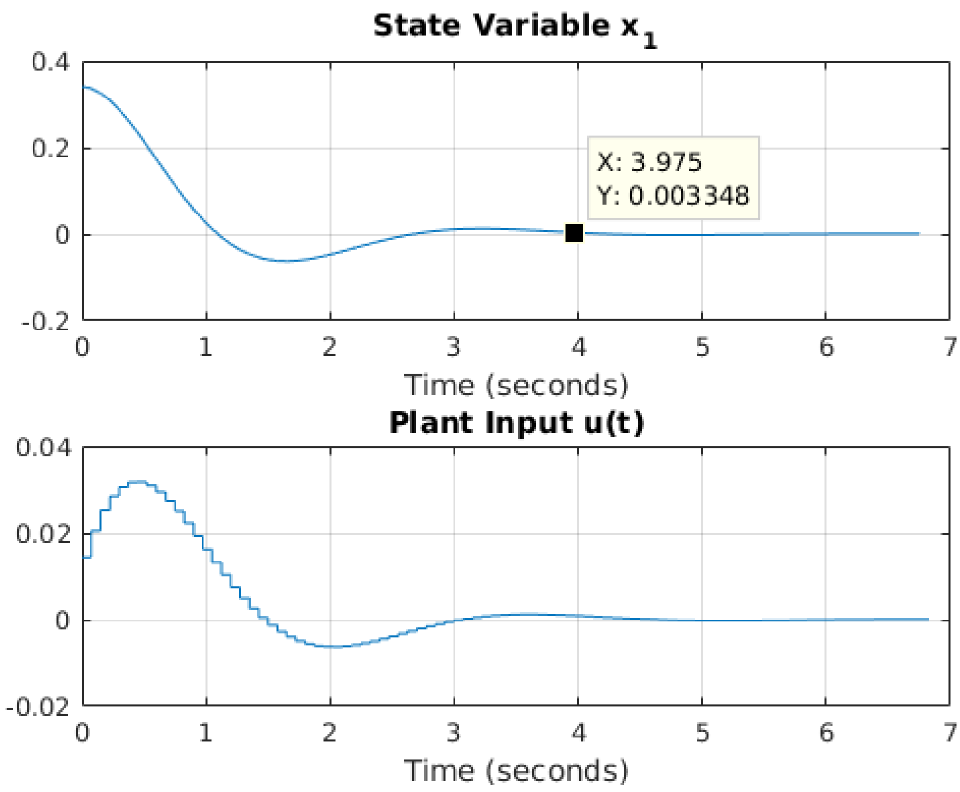
```
------------------- Bessel Poles -------------------

Upper gain margin for input #1 is 1 dB

Lower gain margin for input #1 is -30.1 dB

Phase margin for input #1 is 11 degrees

del1_1 =

    0.1225


del2_1 =

    0.1092
```



Problem 3b.2 (Design of SDPP/ADP DSFR)

```
adp = s1/Ts + j*imag(plantPoles(1));

sPoles = [adp conj(adp) plantPoles(3)];

zPoles = exp(T*sPoles);

K = place(phi, gamma, zPoles);
results = sim('reg_dsf','StopTime', '6.75');
tout = results.tout;
u = results.u;
x = results.x;
%run reg_asfp script, open result
open('SelectedPoles.fig')

% SDPP,ADP
fprintf('------------------- SDPP/ADP Poles -------------------\n\n');
dsm(phi,gamma,K)
[del1_2, del2_2] = rb_regsf(phi,gamma,K,T)

------------------- SDPP/ADP Poles -------------------

Upper gain margin for input #1 is 24.62 dB

Lower gain margin for input #1 is -30.1 dB

Phase margin for input #1 is 90 degrees
```

```
del1_2 =

    1.2102


del2_2 =

    0.9412
```



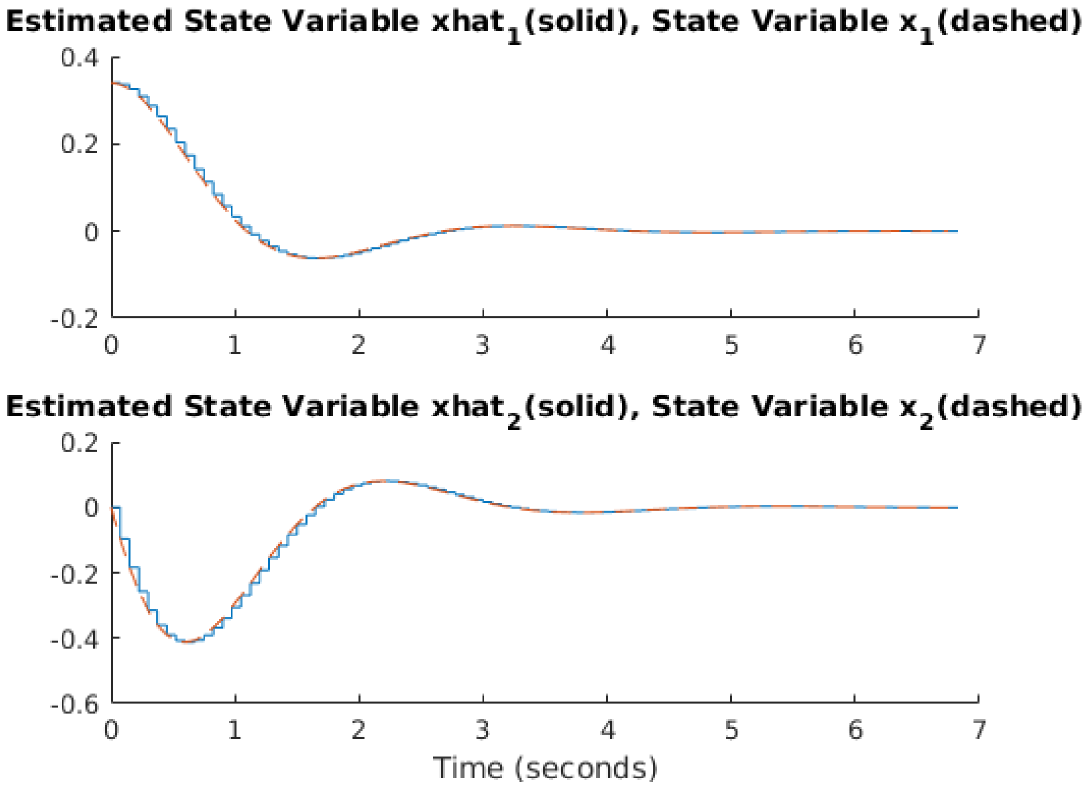
Problem 3b.3 (Analysis)

```
The Bessel pole regulator system gives unacceptable classical and
robustness bounds, I would say it is safe to say that it should never be
considered. By contrast, the second regulator gives fairly robust
stability bounds.
```

Problem 3c (Design of OBR)

```
C = [1 0 0];

soPoles = [adp conj(adp) plantPoles(3)];
zoPoles = exp(T*soPoles);
L = (place(phi',C',zoPoles))';
```

```
results = sim('reg_dob','StopTime', '6.75');
tout = results.tout;
u = results.u;
x = results.x;
xhat = results.xhat;
y = results.y;


% % Load saved figures
% c=hgload('x1vsxhat1.fig');
% k=hgload('x2vsxhat2.fig');
% % Prepare subplots
% figure
% h(1)=subplot(2,1,1);
% title('Estimated State Variable xhat_1(solid), State Variable
 x_1(dashed)')
%
% h(2)=subplot(2,1,2);
% title('Estimated State Variable xhat_2(solid), State Variable
 x_2(dashed)')
% xlabel('Time (seconds)')
%
% % Paste figures on the subplots
% copyobj(allchild(get(c,'CurrentAxes')),h(1));
% copyobj(allchild(get(k,'CurrentAxes')),h(2));

open('xvsxhat.fig')

fprintf('------------------- OBR Stability -------------------\n\n');
dsm(phi,gamma,K)
[del1_2, del2_2] = rb_regob(phi,gamma,C,K,L,T)

------------------- OBR Stability -------------------

Upper gain margin for input #1 is 24.62 dB

Lower gain margin for input #1 is -30.1 dB

Phase margin for input #1 is 90 degrees

del1_2 =

    1.5051


del2_2 =

    0.6785
```



Published with MATLAB® R2017a