John Sigmon
Js85773

HW 4

EE 362K

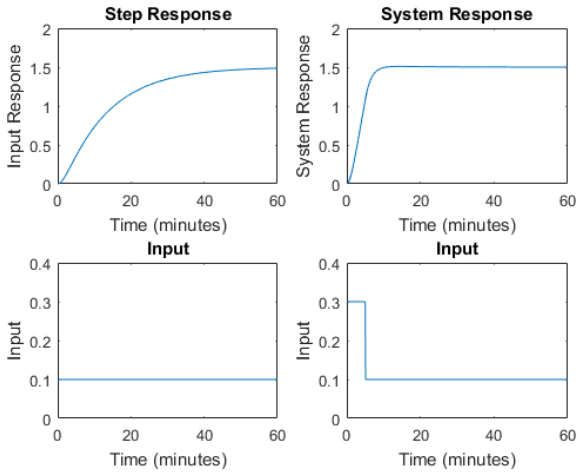
1.)

See attached.

2.)

I scaled my inputs by 1/10 to match the example in the book. (e.g. my step input is 0.1 instead of 1)

```
function problem2
k0 = 0.1;
k1 = 0.1;
k2 = 0.5;
b0 = 1.5;

a = [-k0-k1, k1; k2, -k2];
b = [b0, 0]';
c = [0, 1];
d =  0;
sys = ss(a, b, c, d);

t = 0:0.1:60;
u = ones(1, length(t));
u = u./10;
%%%%Step Response
subplot(2,2,1);
plot(t, lsim(sys, u, t));
xlabel('Time (minutes)');
ylabel('Input Response');
title('Step Response');
xlim([0 60]);
ylim([0 2]);

%%%%%Step Input
subplot(2,2,3);
plot(t, u);
xlabel('Time (minutes)');
ylabel('Input');
title('Input')
xlim([0 60]);
ylim([0 0.4]);


for i = 1: 51
    u(i) = 0.3;
end

%%%%%Pulse Response
subplot(2,2,2);
plot(t, lsim(sys, u, t));
xlabel('Time (minutes)');
ylabel('System Response');
title('System Response');
xlim([0 60]);
ylim([0 2]);

%%%%%Pulse Input
subplot(2,2,4);
plot(t, u);
xlabel('Time (minutes)');
ylabel('Input');
title('Input');
xlim([0 60]);
ylim([0 0.4]);
```



3.)

John Sigmon
Js85773

Find the RCF for system

$$\dot{\bar{z}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & 3 & 5 \end{bmatrix}\bar{z} + \begin{bmatrix} 0 \\ 0 \\ 1/2 \end{bmatrix}u$$

$$\bar{y} = [1\ 0\ 0]\bar{z} + [0]u$$

RCF is determined by the characteristic equation.

$$\det(\lambda I - A) = 0$$

$$\det\begin{bmatrix} -\lambda & 1 & 0 \\ 0 & -\lambda & 1 \\ -2 & 3 & -\lambda - 5 \end{bmatrix} = 0$$

$$-\lambda \cdot \det\begin{bmatrix} -\lambda & 1 \\ 3 & -\lambda - 5 \end{bmatrix} - 1 \cdot \det\begin{bmatrix} 0 & 1 \\ -2 & -\lambda - 5 \end{bmatrix} = 0$$

$$-\lambda\left(-\lambda(-\lambda - 5) - 3\right) - 1 \cdot [0 - (-2)] = 0$$

$$-\lambda^3 + 5\lambda^2 + 3\lambda - 2 = 0$$

RCF for A is:

$$\begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} -5 & -3 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

and b is $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

So, RCF for the system is

$$\dot{\bar{z}} = \begin{bmatrix} -5 & -3 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\bar{z} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\bar{u}$$

$$\bar{y} = [\quad\quad]\bar{z} + [0]\bar{u}$$

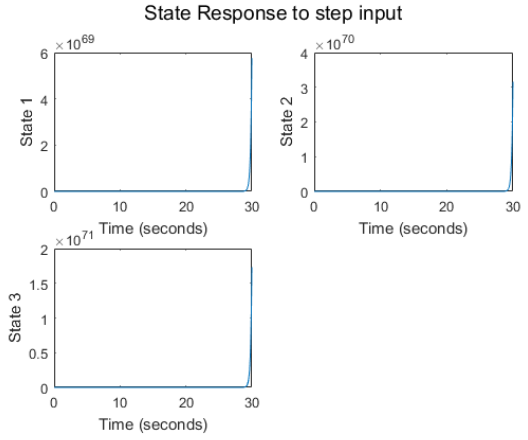
problem3.m

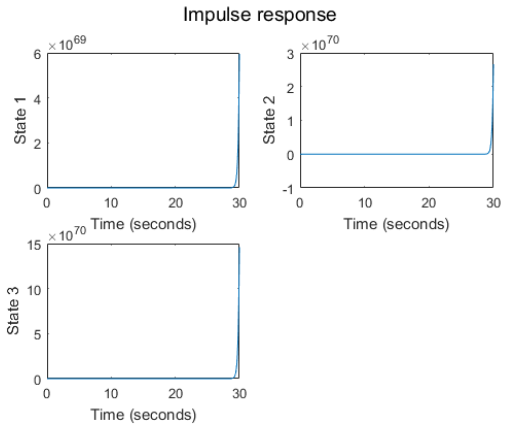
```
function problem3

A = [0,1,0;0,0,1;-2,3,5];
B = [0,0,0.5]';
wr = ctrb(A,B);
reachable = 0;
if (rank(wr) == size(A))
    reachable = 1;
    disp("System is reachable");
else
    disp("System is not reachable");
end
```

```
>> problem3
System is reachable
```

4.)

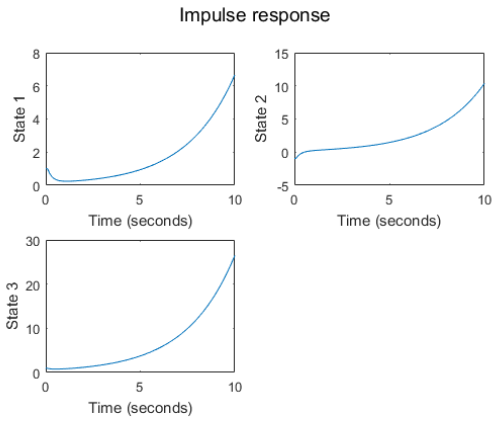
Step response and impulse response for given system.

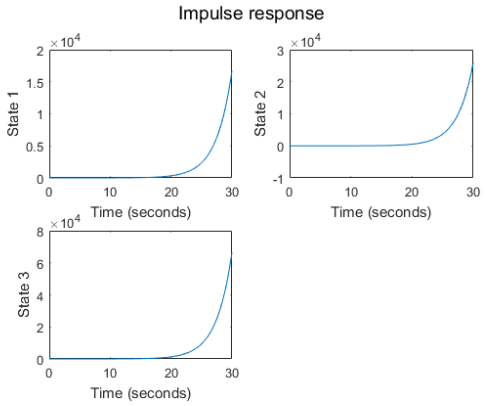


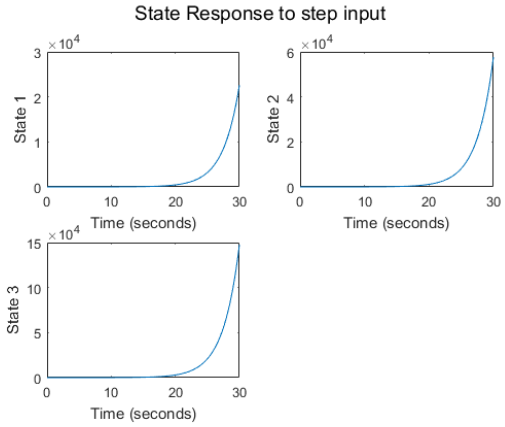


Step response and impulse response for RCF. (10 second span gave more information so I included it)

John Sigmon
Js85773







Code for Step and Impulse Responses.

```m
function xprime =mysystem(t,x)

u = 1;
A = [-5,-3,2;1,0,0;0,1,0;];    %RCF
B = [1,0,0]';                  %RCF
%A = [0, 1, 0; 0, 0 1; -2, 3, 5];
%B = [0, 0, 0.5]';
C = [1,0,0];
D = 0;

xprime = A*x + B*u;
```

```m
function problem4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Part A Step Response
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[t,y] = ode45('mysystem', [0 30], [1 0 1]);

subplot(2,2,1);
plot(t,y(:,1));
xlabel('Time (seconds)');
ylabel('State 1');
subplot(2,2,2);
plot(t,y(:,2));
xlabel('Time (seconds)');
ylabel('State 2');
subplot(2,2,3);
plot(t,y(:,3));
xlabel('Time (seconds)');
ylabel('State 3');
suptitle('State Response to step input');

%}
```

```m
suptitle('State Response to step input');

%}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part B Impulse Response
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A = [0, 1, 0; 0, 0 1; -2, 3, 5];
B = [0, 0, 0.5]';
%A = [-5,-3,2;1,0,0;0,1,0;];    %RCF
%B = [1,0,0]';                  %RCF
C = [1,0,0];
D = 0;
t = 0:0.1:30;
u = zeros(1, length(t));
u(1) = 3;                      %impulse of value 3
sys1 = ss(A, B, C, D);
C = [-1,1,0];
sys2 = ss(A, B, C, D);
C = [0,-1,1];
sys3 = ss(A, B, C, D);
z1 = lsim(sys1, u, t, [1 0 1]);
z2 = lsim(sys2, u, t, [1 0 1]);
z3 = lsim(sys3, u, t, [1 0 1]);
subplot(2,2,1);
plot(t,z1(:,1));
xlabel('Time (seconds)');
ylabel('State 1');
subplot(2,2,2);
plot(t,z2(:,1));
xlabel('Time (seconds)');
ylabel('State 2');
subplot(2,2,3);
plot(t,z3(:,1));
xlabel('Time (seconds)');
ylabel('State 3');
suptitle('Impulse response');
%}
```

John Sigmon
Js85773

5.)

```
>> problem5
wr is
  Columns 1 through 2

         0         0
         0    0.5000
    0.5000    2.5000

  Column 3

    0.5000
    2.5000
   14.0000

wrtilda is
     1     5    28
     0     1     5
     0     0     1

     1
     0
     0

     5     3    -2
     1     0     0
     0     1     0

Transformation matrix is
     0     0     2
     0     2     0
     2     0     0
```

```
function problem5

A = [0, 1, 0; 0, 0, 1; -2, 3, 5];
B = [0; 0; 0.5];
Atilda = [5, 3, -2; 1, 0, 0; 0, 1, 0;];    %RCF
Btilda = [1; 0; 0];                  %RCF

wr = ctrb(A, B);
%wr = [B A*B (A^2)*B];
wrtilda = ctrb(Atilda, Btilda);

disp('wr is')
disp(wr)
disp('wrtilda is')
disp(wrtilda)

%wrtilda = [Btilda Atilda*Btilda (Atilda.^2)*Btilda];
T = wrtilda/wr;
disp(T*B);
disp(T*A/T);
disp('Transformation matrix is');
disp(T);
```

6.)

```
>> problem6
k2/2 - k1/2 - k3/2 + 1

(3*k2)/2 - k1/2 - (9*k3)/2 + 61

3*k2 - k1/2 - 18*k3 + 376

k is:
 32
 60
 30

k2 - k1/2 - 2*k3 + 20

- k1/2 + k2*(1/2 - 5i/2) + k3*(12 + 5i) - (199 - 75i)

- k1/2 + k2*(1/2 + 5i/2) + k3*(12 - 5i) - (199 + 75i)

k is:
 100
  66
  18
```

```
function problem6
syms k1 k2 k3
%part a
f1 = cecalculator(-1);
f2 = cecalculator(-3);
f3 = cecalculator(-6);

[A,b] = equationsToMatrix([f1,f2,f3], [k1,k2,k3]);
k = linsolve(A,b);
disp('k is:')
disp(k)


%part b
f4 = cecalculator(-2);
f5 = cecalculator(-1+5*j);
f6 = cecalculator(-1-5*j);
[A,b] = equationsToMatrix(f4,f5,f6);
k = A\b;
disp('k is:')
disp(k)
```

```
function [f] = cecalculator(lambda)

syms k1 k2 k3;

f = -(lambda.^3) + (lambda.^2)*(5-k3*0.5) + lambda*(3-k2*0.5) - k1*0.5 - 2;
disp(f);
```

John Sigmon
Js85773

Plotting the states turned out to be a little difficult. I ran out of time, and attempted to go based off one of the slides, but couldn't figure out how to do it. Here is what I got done. I compiled the two systems into function files and tried to run ode45 to at least plot the solutions, but got dimension errors in zprime.

```
function zprime1 = mysystem1(t,z)

A = [0,1,0;0,0,1;-2,3,5];
B = [0;0;0.5];
C = [1,0,0];

[k1,k2] = problem6;

yr = 1;
kr = -1/(C/(A-B*k1')*B);

zprime1 = (A - B*k1')*z + B*kr*yr;
```

```
function zprime2 = mysystem2(t,z)

A = [0,1,0;0,0,1;-2,3,5];
B = [0;0;0.5];
C = [1,0,0];

[k1,k2] = problem6;

yr = 1;
kr = -1/(C*inv(A-B*(k2'))*B);

zprime2 = (A - B*k2)*z + B*kr*yr;

t = 0:0.1:30;
plot(t, kr*yr);
```

```
function controller

clear all;
[t,z] = ode45('mysystem1', [0 30], [0 0]);
plot(t, z(:,1));
hold on;
[t1,z1] = ode45('mysystem2', [0 30], [0 0]);
plot(t1, z1(:,1));
hold on;
```

7.)

The general Acker's method works for both complex, as well as real values.

8.)

John Sigmon
Js85773

```
>> problem8
[ bl,     0,     0]
[  0,     0, al*bl]
[  0, al*bl, al*bl]

-al^2*bl^3

The only values of alpha that make wr not full rank are:
 0
 0

The only values of beta that make wr not full rank are:
 0
 0
 0
```

```
function problem8

syms al      %alpha
syms bl      %beta

f(al) = [0,1,0;0,0,1;al,1,1];
g(bl) = [bl;0;0];
wr = [g(bl) f(al)*g(bl) (f(al)^2)*g(bl)];
disp(wr);
disp(det(wr));

S1 = solve(det(wr) == 0, al);
disp('The only values of alpha that make wr not full rank are:')
disp(S1);
S2 = solve(det(wr) == 0, bl);
disp('The only values of beta that make wr not full rank are:')
disp(S2);
```