

---

# Table of Contents

.....	1
Prob 1 .....	1
Prob 2 .....	1
part b .....	2
part c .....	4

```
% Justin Francis
% MEEN 5210, State Space Controls
% Dr. Abbot, U of U
% HW 8, controls and observers
```

```
clc; clear all; close all;
```

## Prob 1

```
%system
A = [[2 1 0 0];[0 2 0 0];[0 0 -1 0];[0 0 0 -1]];
B = [0 1 1 1].';
%check controllability
Co = ctrb(A,B);
isControllable = (rank(Co) == length(A));

Anc = [[-1 0];[0 -1]];
Bnc = [1 1].';

Conc = ctrb(Anc, Bnc);

P = [[1 0];[-1 1]];

A_n = P*Anc\P;
B_n = P*Bnc;

B = [0; 1; B_n];

syms k1 k2 k3 k4 s
K = [k1 k2 k3 k4];
A_bar = A - B*K;
charEqn = det(s*eye(size(A_bar))-A_bar);
```

## Prob 2

```
clc; clear; close all;

%part a
A = [[-3 5];[0 -2]];
B = [1 -1].';
C = [1 0];
D = [0];
```

---

```

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

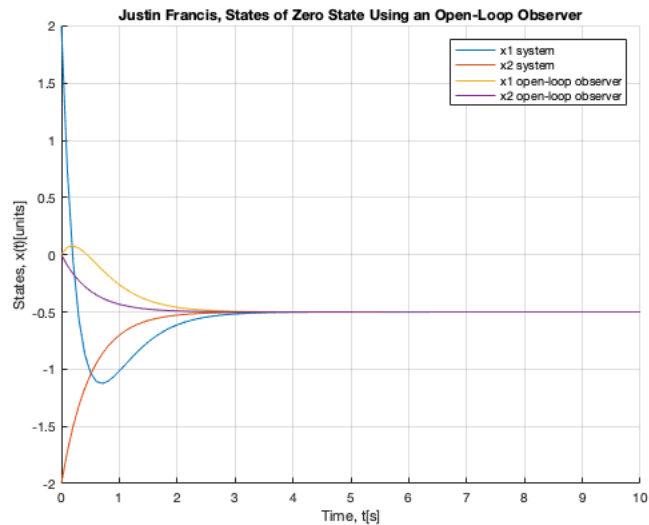
x0 = [0 0].';
x0sys = [2 -2].';

t = linspace(0,10);
u = ones(size(t));

[Y,T,X] = lsim(sys, u, t, x0sys);
[Y0,T0,X0] = lsim(sys, u, t, x0);

figure();
hold on;
plot(t, X);
plot(t, X0);
title('Justin Francis, States of Zero State Using an Open-Loop Observer');
xlabel('Time, t[s]');
ylabel('States, x(t)[units]');
grid();
legend({'x1 system', 'x2 system', 'x1 open-loop observer', 'x2 open-loop observer'});

```



## part b

```

clc;

%find observer gains
desEig1 = -1;
desEig2 = -1;

syms L1 L2 s
l = [L1 L2].';

```

---

```

A_n = (A-l*C);
charEqn = det(s.*eye(size(A_n)) - A_n);

desCharEqn = poly([desEig1 desEig2]);

eqn1 = 5 + L1 == 2;
eqn2 = 2*L1 + 5*L2 + 6 == 1;

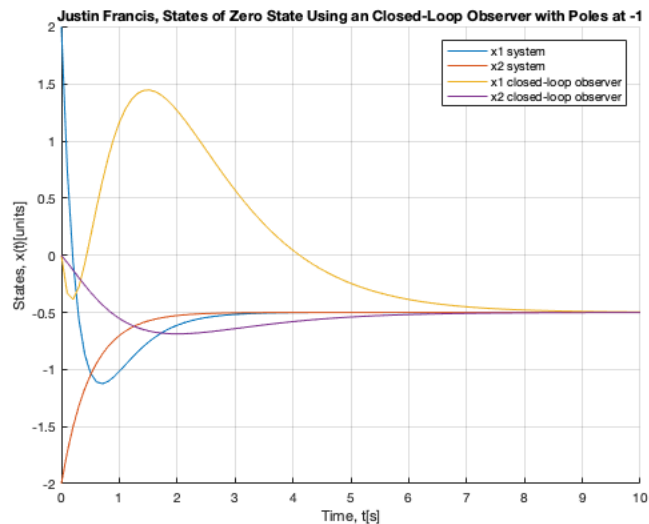
[meatMat,potatoesMat] = equationsToMatrix([eqn1, eqn2], [L1, L2]);
L= double(linsolve(meatMat, potatoesMat));

%find estimated states
A_cOb = (A-L*C); %closed observer matrices
B_cOb = [B L]; %this is how it was recommended to modify the B matrix
closedObserver = ss(A_cOb, B_cOb, C, D);
u_cOb = [u; Y.']; %use the output of the system as a second input to
    the observer sim

[Y_cOb, T_cOb, X_cOb] = lsim(closedObserver, u_cOb, t, x0);

figure();
hold on;
plot(t, X);
plot(t, X_cOb);
title('Justin Francis, States of Zero State Using an Closed-Loop
    Observer with Poles at -1');
xlabel('Time, t[s]');
ylabel('States, x(t)[units]');
grid();
legend({'x1 system', 'x2 system', 'x1 closed-loop observer', 'x2
    closed-loop observer'});

```



---

## part c

```
clc;

%find observer gains
desEig1 = -6;
desEig2 = -6;

syms L1 L2 s
l = [L1 L2].';
A_n = (A-l*C);
charEqn = det(s.*eye(size(A_n)) - A_n);

desCharEqn = poly([desEig1 desEig2]);

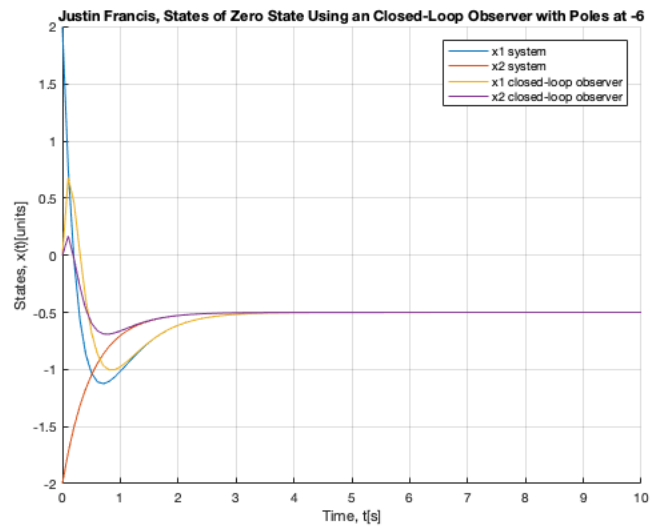
eqn1 = 5 + L1 == desCharEqn(2);
eqn2 = 2*L1 + 5*L2 + 6 == desCharEqn(3);

[meatMat,potatoesMat] = equationsToMatrix([eqn1, eqn2], [L1, L2]);
L= double(linsolve(meatMat, potatoesMat));

%find estimated states
A_cOb = (A-L*C); %closed observer matrices
B_cOb = [B L]; %this is how it was recommended to modify the B matrix
closedObserver = ss(A_cOb, B_cOb, C, D);
u_cOb = [u; Y.']; %use the output of the system as a second input to
    the observer sim

[Y_cOb, T_cOb, X_cOb] = lsim(closedObserver, u_cOb, t, x0);

figure();
hold on;
plot(t, X);
plot(t, X_cOb);
title('Justin Francis, States of Zero State Using an Closed-Loop
    Observer with Poles at -6');
xlabel('Time, t[s]');
ylabel('States, x(t)[units]');
grid();
legend({'x1 system', 'x2 system', 'x1 closed-loop observer', 'x2
    closed-loop observer'});
```



*Published with MATLAB® R2017a*