Putu Andika Eka Putra (Pu an di) F11127807

Assignment 2

Design a compensator for the inverted pendulum on a moving cart.

1. controllability and observability

```
co = ctrb(A,B);
%disp(length(sys.A))

fprintf('rank of system : %i \n',rank(co));
fprintf('oder of system : %i \n',length(A));
if rank(co) < length(A)
    fprintf("system is uncontrolable \n");
else
    fprintf("system is controlable \n");
end

fprintf("\n");


C2= C(2,:);
ob = ctrb(A',C2');

fprintf('c:\n');
disp(C2)
fprintf('rank of system : %i \n',rank(ob));
fprintf('oder of system : %i \n',length(A));
if rank(ob) < length(A)
    fprintf("system is unobservable \n");
else
    fprintf("system is observable \n");
end
```

```
>> ConObs
rank of system : 4
oder of system : 4
system is controlable

c:
     0     1     0     0

rank of system : 4
oder of system : 4
system is observable
>>
```

System is controllable and observable.

2. full-state feedback regulator

```
import varIput.*
varInput

sn = 7;
V1 = [   -(3.3+(0.05*sn))+(3.3+(0.05*sn))*1i;
        -(3.3+(0.05*sn))-(3.3+(0.05*sn))*1i;
        -(8.8+(0.05*sn))+(8.8+(0.05*sn))*1i;
        -(8.8+(0.05*sn))-(8.8+(0.05*sn))*1i;
    ];

fprintf("seat number :%i \n", sn);
fprintf("pole :\n");
disp(V1);
k = acker(A,B,V1);
k2 = place(A,B,V1);
fprintf("K :\n");
disp(k);
fprintf("eigenvalue :\n");
disp(eig(A-B*k));
%disp(k2);
```

```
>> fsfrPole
seat number :7
pole :
  -3.6500 + 3.6500i
  -3.6500 - 3.6500i
  -9.1500 + 9.1500i
  -9.1500 - 9.1500i

K :
 -793.7225 -455.2625 -200.0849 -174.4849

eigenvalue :
  -9.1500 + 9.1500i
  -9.1500 - 9.1500i
  -3.6500 + 3.6500i
  -3.6500 - 3.6500i

>>
```

3. Full-order observer

```
import varInput.*
varInput

sn = 7;
V2 = [   -(33+(0.05*sn))+(33+(0.05*sn))*1i;
        -(33+(0.05*sn))-(33+(0.05*sn))*1i;
        -(88+(0.05*sn))+(88+(0.05*sn))*1i;
        -(88+(0.05*sn))-(88+(0.05*sn))*1i;
    ];



fprintf("seat number :%i \n", sn);
fprintf("pole :\n");
disp(V2);

C2= C(2,:);
L=acker(A',C2',V2);

fprintf("L :\n");
disp(L);
fprintf("eigenvalue :\n");
disp(eig(A-L'*C2));
%disp(k2);
```

```
>> foobPole
seat number :7
pole :
 -33.3500 +33.3500i
 -33.3500 -33.3500i
 -88.3500 +88.3500i
 -88.3500 -88.3500i

L :
   1.0e+07 *

   -0.1466    0.0000   -3.5761    0.0030

eigenvalue :
 -88.3500 +88.3500i
 -88.3500 -88.3500i
 -33.3500 +33.3500i
 -33.3500 -33.3500i
```

4. Compensator(A_{CL})

```
import foobPole.*
import fsfrPole.*

foobPole
fsfrPole

Kd=zeros(1,4);
ACL = [A -B*k; L'*C2 (A-L'*C2-B*k)];
ACLC = A -B*k;

disp(ACL);
disp(eig(ACL));
```

```
-88.3500 +88.3500i
-88.3500 -88.3500i
-33.3500 +33.3500i
-33.3500 -33.3500i
 -9.1500 + 9.1500i
 -9.1500 - 9.1500i
 -3.6500 + 3.6500i
 -3.6500 - 3.6500i
```

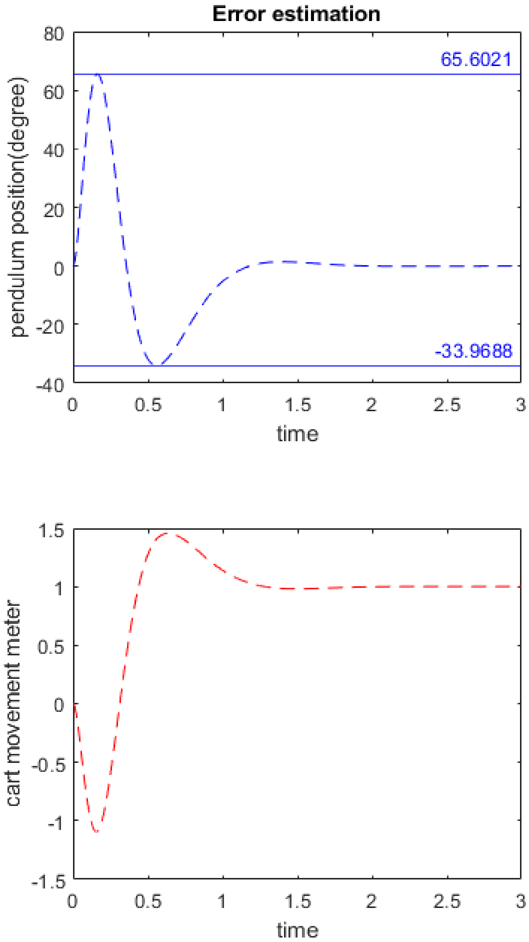
5. Lsim

```
t = 0:0.00005 :3;
n=size(t,2);
Xdt = [0 1 0 0] ;
Xd = repelem(Xdt,n,1);
C2= C(2,:);
C1= C(1,:);
BCL = [B*(k-Kd); B*(k-Kd)];
%sysCL=ss(ACL, BCL,[C1 zeros(1,4)], zeros(1,4));
sysCL=ss(ACL, BCL,[C zeros(2,4)], zeros(2,4));

[y,t,X] = lsim(sysCL,Xd,t');

pendulum = rad2deg(X(1:end, 1));
cart = X(1:end, 2);
```

6. ploting




```
figure('Position', [1, 1, 400, 800])
tiledlayout(2,1)

nexttile
plot(t, pendulum, 'b--');
title('Error estimation')
xlabel("time")
ylabel("pendulum position(degree)")
yline(max(pendulum),'b-',max(pendulum));
yline(min(pendulum),'b-',min(pendulum));

nexttile
plot(t, cart, 'r--');
xlabel("time")
ylabel("cart movement meter")
```

7. Observation

Based on the eigenvalues, The pole placement is correct. And the maximum pendulum swing is 65.6 degrees and cart maximum movement in 1.5 meter. The pendulum swing is quite far but did not exceed 90 degrees. The plant is still inefficient but will not fail and the pendulum eventually stable.