PURVANG LAPSIWALA

UIN = 662689378

Submitted to Dr. Sabri Cetinkunt

University of Illinois at Chicago

Take Home Exam Report of ME512

AUTOMATIC CONTROL OF MECHANICAL SYSTEM

<u>Some References are taken from the Textbook to linearize Non-linear equations and Dynamics of given problem from Chapter 3.</u>

<u>Answer 6.13</u>

For Given Inverted Pendulum Problem, we have been given Non-linear dynamics model of system, and system Parameters. Force applied on system is u (t) and Output (measured variable) is the cart position, x (t).

(a)

Given problem contains Nonlinear equations , we can't proceed for state space formulation of given system so first thing I did for it to work is to Linearize given nonlinear system dynamics at given nominal point.

For Linearization we can use different numerical integration methods such as Euler's method, Runge-kutta method. I choose Euler's method for given problem to make system linearize.

So I have got 2 equation. By rearranging them in proper order, I got matrix A, B, C.

We are interested in position, angle and derivative of this both state. So my interested states are 4 and A matrix size is 4*4.

(b)

Using matlab function "ctrb " and "obsv", we can find controllability and observability matrix. And if this two matrices are full rank, system is controllable otherwise not. And I am getting system as controllable and observable.

(c)

System is Observable if output measurement variable is cart position.

But System is not observable if output measurement variable is angular position.

(d)

Results with Integral action

The full state feedback gain, K, in order to achieve closed loop poles are

K = [-45.0000 -137.5000 -28.5278 -31.1019]

K1 =[-33.3333 -123.8889 -23.3333 -27.2222]

K2 = [-45.0000 -137.5000 -28.5278 -31.1019 -16.6667]

Ki = -16.6667

Without Integral Controller

K = [-33.3333 -123.8889 -23.3333 -27.2222]

K1 =[-33.3333 -123.8889 -23.3333 -27.2222]

K2 = [-45.0000 -137.5000 -28.5278 -31.1019 -16.6667]

Ki = 0

(e)

The full state estimator gain, L, in order to achieve state estimator poles

L =1.0e+04 *

0.0070

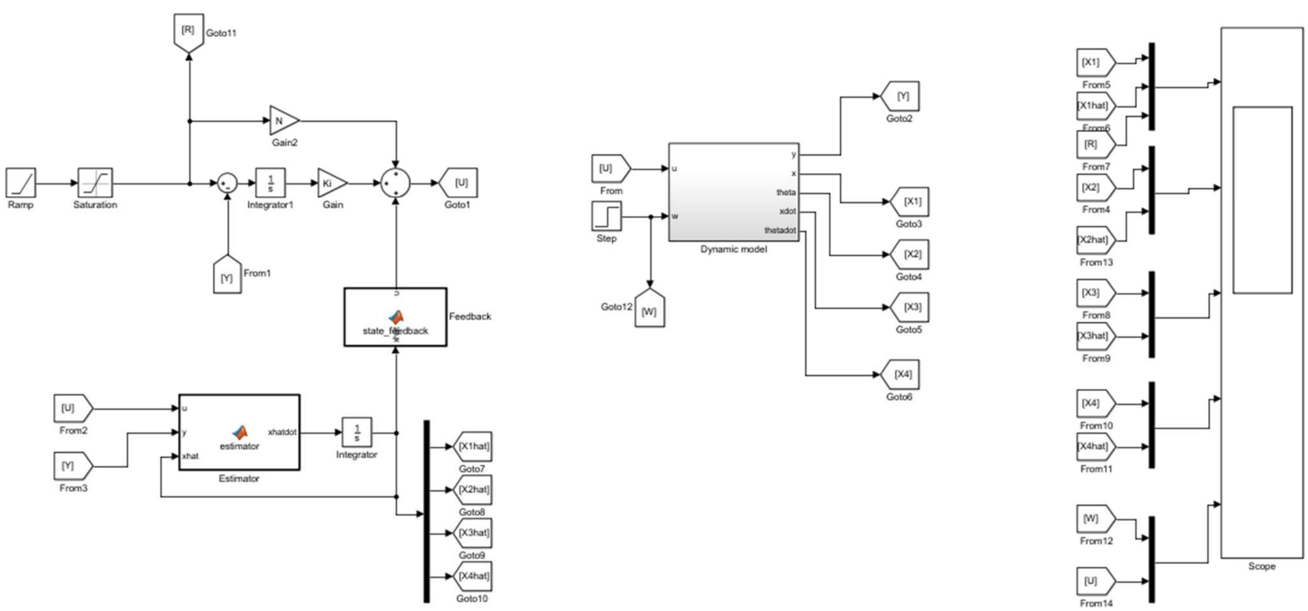
-0.6113

0.2474

-5.1563

(f)

Block diagram of the overall control system and actual values of the control algorithm parameters.



Simulink Model

Matlab Files

Main File

```
clear all
close all
clc

m1=1.0; %mass of cart
m2=1.0; %mass of pendulum
l=1.0;
g=10.0;

x0 = [ 0.0 0.0 0.0 0.0 ]' ; % Initial States
u0 = [ 0.0 ]' ; % Initial Control Force

[A,B] = linearize('invp_dyn', x0, u0); % Linearizing Non linear Characteristic of
Inverted Pendulum
                                       % A and B respresents Dynamics of
                                       % system

C = [ 1.0 0.0 0.0 0.0 ];   % Output measurenment variable is cart position
%C = [ 0.0 1.0 0.0 0.0 ];   % Output measurenment variable is angle and
                            %system will become unObservable


nA=rank(ctrb(A,B))      % Checking Controllability
nB=rank(obsv(A,C))      % Checking Observability
display('As both nA and nB is full rank, system is controllable and Observable')

% Augmented Linear System Dynamics: to include integral control
v0=[0 0 0 0]';
A_a=[A,v0;
     C, 0];  % Including Integral action using Augmented matrix

B_a=[B;
     0];

C_a= [C, 0];

% Controller Design
% Open loop poles: for information purposes
p_ol=eig(A);

% Desired closed loop system poles: for state feedback control without
% integral action
p_c = [ -2+2*1i,  -2-2*1i  -5+5*1i   -5-5*1i ] ;

 % Desired closed loop system poles: for state feedback control with
 % integral action

 p_ca = [ -2+2*1i,  -2-2*1i  -5+5*1i   -5-5*1i  -.5 ] ;

 % Full state feedback gain (without integral action)

 K1 = place (A,B,p_c);

 % Full state feedback gain (with integral action)
```

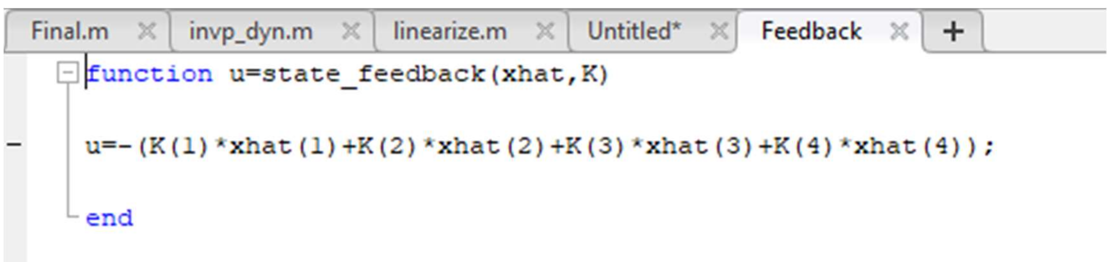
```
 K2= place(A_a,B_a,p_ca);

 choice= 0 ;
while (~((choice==1 || choice==2)))
choice = input('Enter 1 for WITHOUT integral control,\n 2 for WITH integral control
simulation: ');

if (choice==1) % Without Integral Control
    K = K1 ;
    Ki = 0.0 ;
elseif (choice==2)  % With Integral Control
     K = K2(1:4);
    Ki= K2(5);
end

end

p_e = 5. * p_c ;
L=(place(A',C',p_e))';

% Feedforward gain to achieve unity DC gain
N=1/(C*inv(B*K-A)*B);
```

```
%%% Inverted Pendulum Dynamics

% Reference taken From Text book Chapter 3

function xdot=invp_dyn(t,x,u )
m1 = 1.0 ;
m2 = 1.0 ;
l = 1.0 ;
g = 10.0 ;
```

```
J = [ (m1+m2) 0.5*m2*l*cos(x(2))
      0.5*m2*l*cos(x(2)) (1.0/3.0)*m2*(l^2) ] ;

detJ = J(1,1)*J(2,2) - J(1,2)*J(2,1) ;

invJ = [ J(2,2)/detJ -J(2,1)/detJ
        -J(1,2)/detJ J(1,1)/detJ ]' ;

temp = invJ * [ 0.5*m2*l*(x(4)^2)*sin(x(2))+ u
                0.5*m2*g*l*sin(x(2)) ];

            xdot = [ x(3)
                     x(4)
                     temp(1)
                     temp(2) ] ;
return ;
```

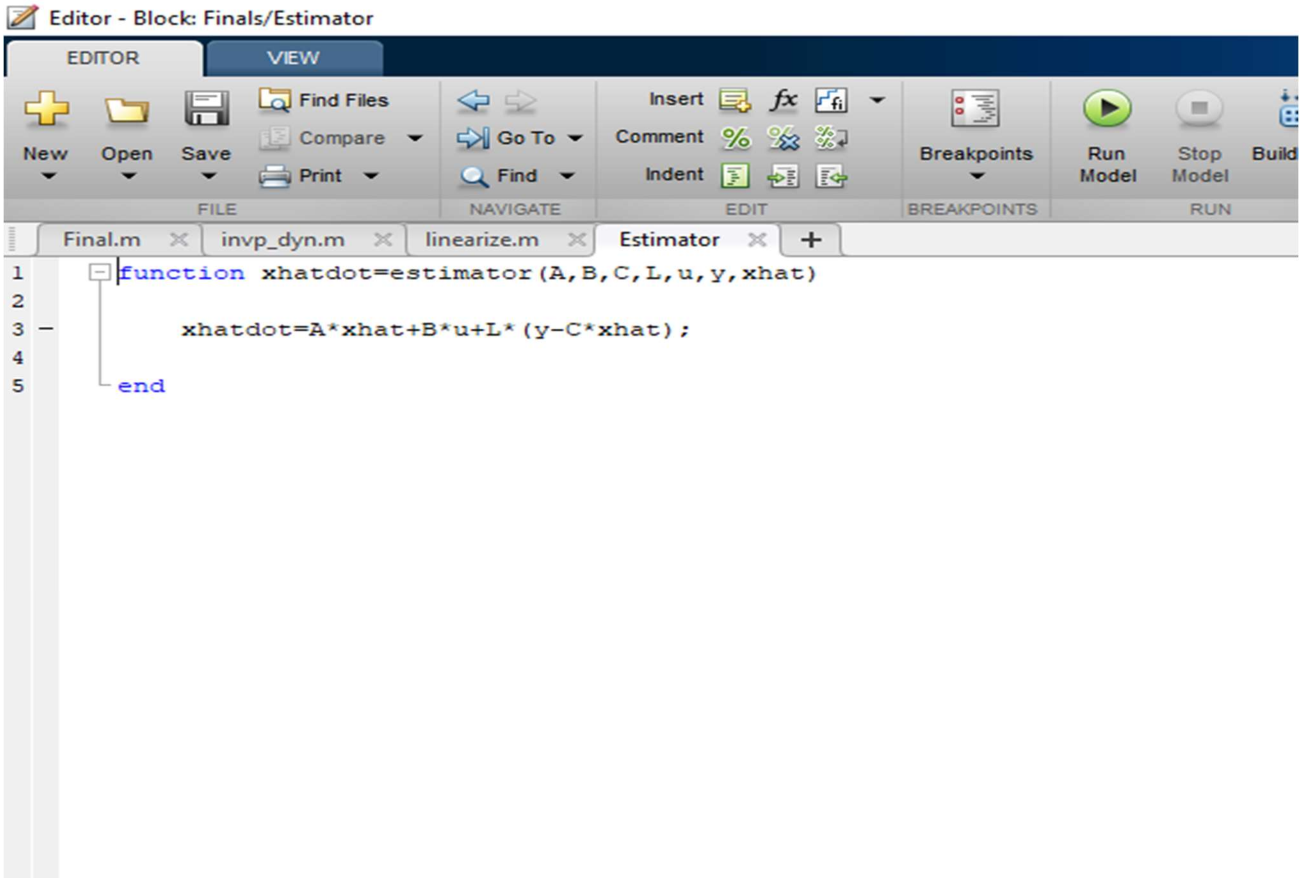
```
%%% Linearize Function

% Reference taken from text book chapter 3

function [A,B] = linearize(Fname, x0, u0)

n = size(x0) ;
m = size(u0) ;
delta = 0.001 ;

x = x0 ;
u = u0 ;
t = 0.0 ;

for j=1:n
    x(j) = x(j) + delta;
    A(:,j) = ( feval(Fname,t,x,u) - feval(Fname,t,x0,u0))/delta ;
    x(j) = x(j) - delta ;
end

for j=1:m
    u(j) = u(j) + delta;
```

```
    B(:,j) = (feval(Fname,t,x,u)-feval(Fname,t,x0,u0))/delta ;
    u(j) = u(j) - delta ;
end
```




```
function u=state_feedback(xhat,K)

u=-(K(1)*xhat(1)+K(2)*xhat(2)+K(3)*xhat(3)+K(4)*xhat(4));

end
```

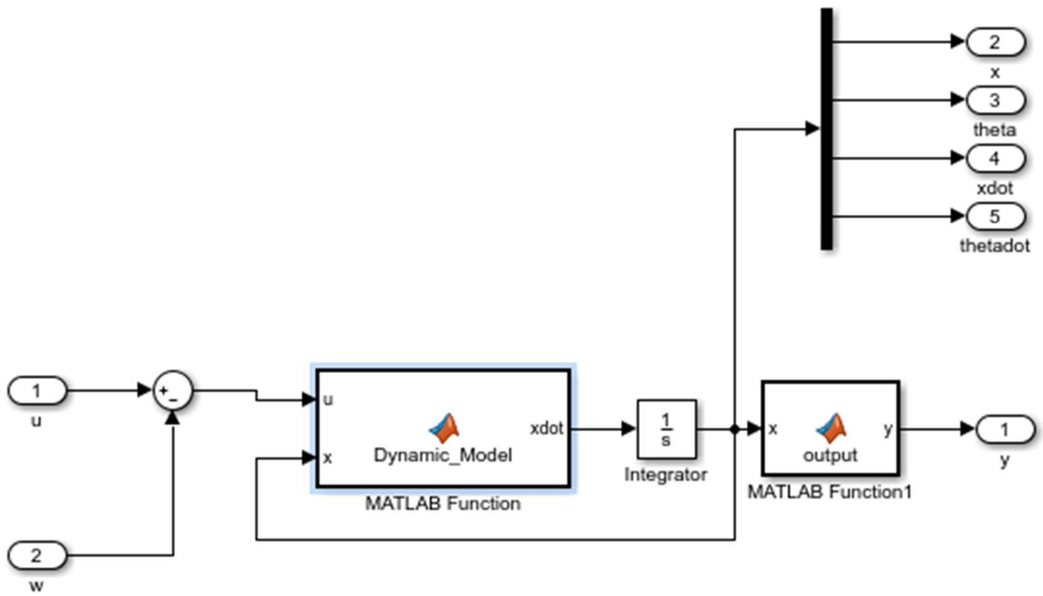
State Feedback Controller



State Estimator

```
function xdot=Dynamic_Model(u,x,A,B)


m1 = 1.0 ;
m2 = 1.0 ;
l = 1.0 ;
g = 10.0 ;

% State Vector: [x(1) x(2) x(3) x(4) ] = [x theta xdot thetadot ]
J = [ (m1+m2) 0.5*m2*l*cos(x(2))
0.5*m2*l*cos(x(2)) (1.0/3.0)*m2*(l^2) ] ;

detJ = J(1,1)*J(2,2) - J(1,2)*J(2,1) ;

invJ = [ J(2,2)/detJ -J(2,1)/detJ
-J(1,2)/detJ J(1,1)/detJ ]' ;

temp = invJ * [ 0.5*m2*l*(x(4)^2)*sin(x(2))+ u
0.5*m2*g*l*sin(x(2)) ];

xdot = [ x(3)
         x(4)
         temp(1)
         temp(2) ] ;

end
```

Plant Dynamics Representing in terms of State Space Model



Derivation of States and Integration for finding original state

g) After swapping the desired pole locations for estimator and state feedback controller, the result is as follows.

K =1.0e+04 *[-2.8125 -2.0958 -0.6899 -0.4660]

K1 =1.0e+04 *[-2.0833 -1.5951 -0.2917 -0.2003]

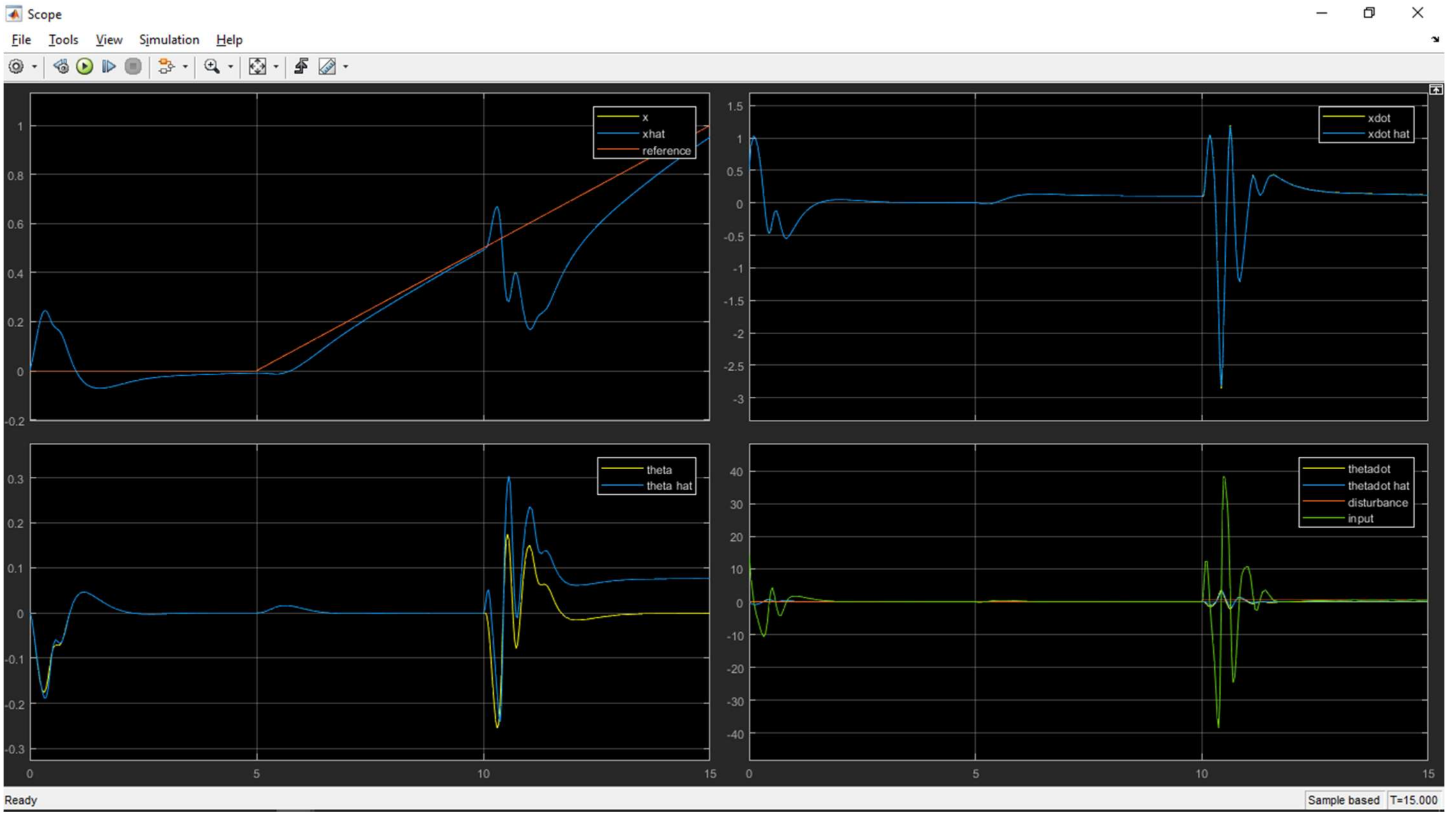
K2 = 1.0e+04 *[-2.8125 -2.0958 -0.6899 -0.4660 -5.2083]

Ki =-5.2083e+04

By Looking at gain, it is clearly said that it is not good choice after swapping because gains become much smaller which leads to system more unstable.

(h)

Simulation of the controller for the following non-zero initial condition using Simulink: x(t) = 0.0, ˙ x(t) = 0, θ(t) = 0.5 , ˙ θ(t0) = 0.0.Result is shown below.



Conclusion

- For First Graph, as we can see in Graph for Given Ramp input signal, measured state variable and Predicted State Variable for position are almost following reference signal. So controller made is working fine. We can interpret as force is continuously increase as ramp function, our position is also increasing trying to stabilize system.
- For Second Graph, Actual measured angular position and predicted angular position is most of time working fine till 10 sec and then values are deviating little from that time. Angle of pendulum becomes stable after applying force which shows system is controllable.
- For Third Graph, Actual Measured and predicted value for velocity of cart is coming same. Slight overshoot is there because of disturbance. But value eventually got stabilized.
- For the last Graph, with constant disturbance application, derivative of angular position deviate for some fraction of time when disturbance applied, and come back to stable position after few seconds.

Answer 7.18

For Given Inverted Pendulum Problem, we have been given Non-linear dynamics model of system, and system Parameters. Force applied on system is u(t) and Output (measured variable) is the cart position, x(t).

(a)

For given non-linear system equation, we can't proceed for state space formulation of given system so first thing I had done to make it work is to Linearize given nonlinear system dynamics at given nominal point.

For Linearization we can use different numerical integration methods such as Euler's method, Runge-kutta method. I choose Euler's method for given problem to make system linearize.

So I have got 2 equation. By rearranging them in proper order, I got matrix A,B,C.

We are interested in position, angle and derivative of this both state. So my interested states are 4 and A matrix size is 4*4.

(b)

Those A, B , C matrices are in continuous domain, so to make it discretize for given sampling period of 0.0001 sec using zero order hold method. After doing this, I got values for Phi, Tau and C. In reality Phi is nothing but A*Tsample approximately and Tau is almost equal to B*Tsample.

(c)

Yes, system is controllable.

(d)

Yes, System is Observable.

(e)

Results with Integral action

The discrete controller full state feedback gain, K, in order to achieve closed loop poles are

With integral control

K =1.0e+03 *[-0.0866 -1.0556 -0.1052 -0.7890]

K_d =[-53.3067 -791.8697 -66.6360 -527.5538]

K_da =1.0e+03 *[-0.0866 -1.0556 -0.1052 -0.7890 -0.0000]

K_i = -0.0027

Without integral control

K =1.0e+05 *[-0.3325 -2.3208 -0.0831 -0.5584]

K_d =1.0e+05 *[-0.3325 -2.3208 -0.0831 -0.5584]

K_da = 1.0e+05 *[-0.5403 -3.7165 -0.6632 -4.4258 -0.0001]

K_i =0

(f)

The discrete controller full state estimator gain, L, in order to achieve state estimator poles

L =

0.0050

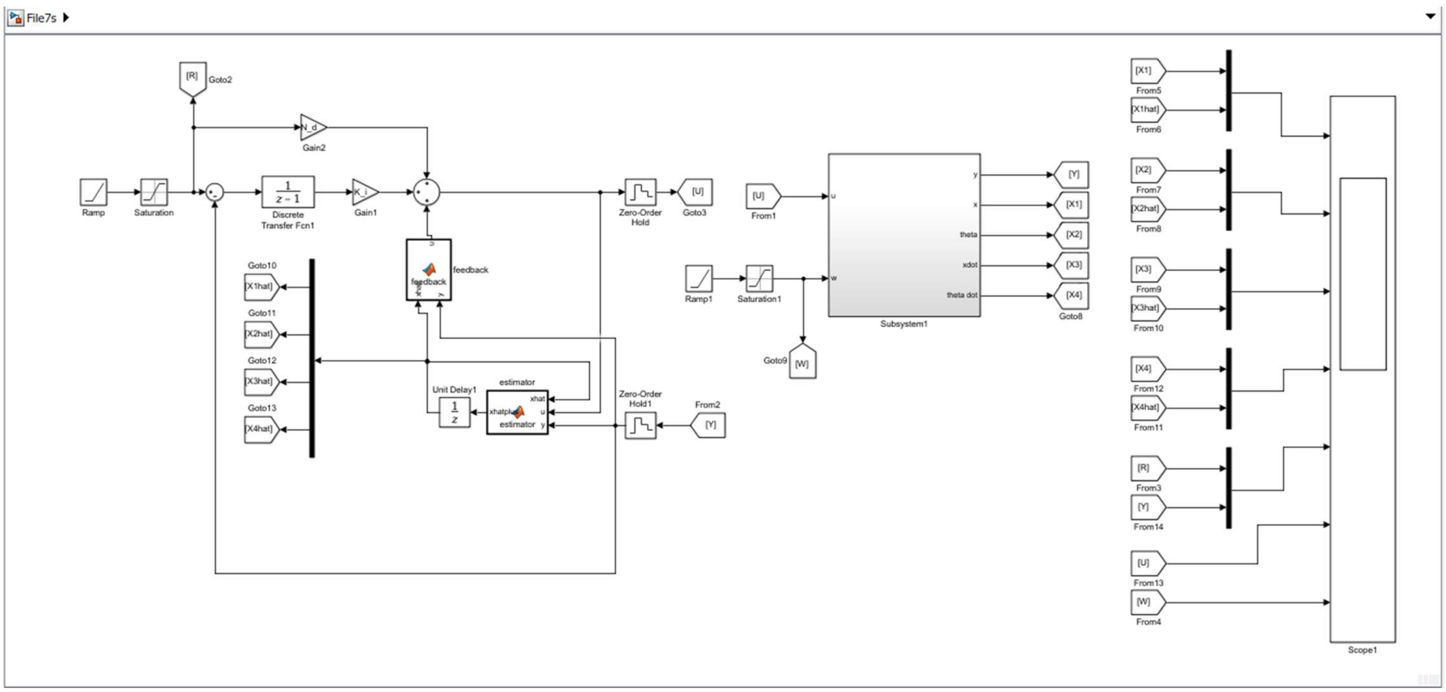
-0.1683

0.1250

-0.7150

(g)

Block diagram of the overall control system with controller parameters; discrete control algorithm, ADC and DAC components.



Simulink File

%%% Main File

```
clear all;
clc;

m1=1;  %mass of cart
m2=1;  %mass of pendulum
l=10;
g=10;


x0 = [ 0.0  0.0  0.0  0.0 ]' ;  % Initial States

u0 = [ 0.0 ]' ;   % Initial Control Force

[A,B] = linearize('dynamic_system', x0, u0);  % Linearizing Non linear Characteristic of
Inverted Pendulum
                                                % A and B respresents Dynamics of
                                                % system
C = [ 1.0 0.0 0.0 0.0 ];   % Output measurenment variable is cart position
%C = [ 0.0 1.0 0.0 0.0 ];   % Output measurenment variable is angle and
                            %system will become unObservable



% Sampling Period:

T_sample = 0.0001 ;

% ZOH equivalent discrete-time LTI model

SYS1 = ss(A,B,C,0) ;
SYS1D = c2d(SYS1,T_sample,'zoh') ;  % Descrete values for A,B,C using sampling time and
zero order hold method
[A_d,B_d,C_d,D_d] =ssdata(SYS1D);   % getting Values


% Controllability and observability of the discrete time system

Wc = ctrb(A_d,B_d) ;  % Checking Controllability
nc = rank(Wc) ;
if nc==4
   disp('Discrete time model: Controllable ');
else
   disp('Discrete time model: Not Controllable') ;
end



Wo = obsv(A_d,C_d) ;   % Checking Observability
no = rank(Wo) ;
if no==4
  disp('Discrete time model: Observable');
else
   disp('Discrete time model: Not Observable') ;
end

%Augmented Linear System Dunamics: to include integral control

v0=[0;0;0;0];
```

```
A_da=[A_d, v0;
      C_d, 1];    % Including Integral action using Augmented matrix

B_da=[B_d;
       0];

C_da=[C_d, 0];

% Desired closed loop eigenvalues;

%p_c = [ -1+j*1  -1-j*1  -4+j*4  -4-j*4 ] ;
p_c = 5*[ -1+j*1  -1-j*1  -4+j*4  -4-j*4 ]
p_cd = exp(p_c*T_sample) ; % desired poles in z-domain
%p_ca = [ -1+j*1 -1-j*1 -4+j*4 -4-j*4 -0.5 ] ;
p_ca = 5*+[ -1+j*1 -1-j*1 -4+j*4 -4-j*4 -0.5 ] ;
p_cad = exp(p_ca*T_sample) ; % desired poles in z-domain

% Desired full-state observer (state estimator) poles:

%p_e =5*p_c;
p_e = [-1+j*1  -1-j*1  -4+j*4  -4-j*4 ];
p_ed = exp(p_e *T_sample) ;

K_d = place(A_d,B_d,p_cd) ;
L_d = (place(A_d',C',p_ed))' ;
K_da = place(A_da, B_da, p_cad) ;




choice= 0 ;
while (~((choice==1 || choice==2)))
 choice = input('Enter \n 1 for without integral control \n 2 for with integral control
simulation: ');
end
if (choice==1)
  K = K_d ;
  K_i = 0.0 ;
elseif (choice==2)
  K=K_da(1:4);
  K_i = K_da(5) ;
end




% Feedforward gain to achieve uniy DC gain
% Feed for9ward gain:

 I = eye(4,4);
 N_d = 1/(C_d*inv(I-A_d+B_d*K)*B_d ) ;
```

%%% Dynamics of System

```
function xdot=dynamic_system(t,x,u)

m1=1.0;
m2=1.0;
l=10.0;
g=10.0;

A = [  (m1+m2)                    0.5*m2*l*cos(x(2))
        0.5*m2*l*cos(x(2))        (1/3)*m2*(l^2)     ] ;

b= [ 0.5*m2*l*(x(4)^2)*sin(x(2))+ u
     0.5*m2*g*l*sin(x(2))];

y=A\b;

xdot=[x(3);x(4);y(1);y(2)];

end
```

%%% Linearization Function

```
function [A,B] = linearize(Fname, x0, u0)


n = size(x0) ;
m = size(u0) ;
delta = 0.001 ;
x = x0 ;
u = u0 ;
t = 0.0 ;

for j=1:n
  x(j) = x(j) + delta;
  A(:,j) = ( feval(Fname,t,x,u) - feval(Fname,t,x0,u0))/delta ;
  x(j) = x(j) - delta ;
end
for j=1:m
  u(j) = u(j) + delta;
  B(:,j) = (feval(Fname,t,x,u)-feval(Fname,t,x0,u0))/delta ;
  u(j) = u(j) - delta ;
```

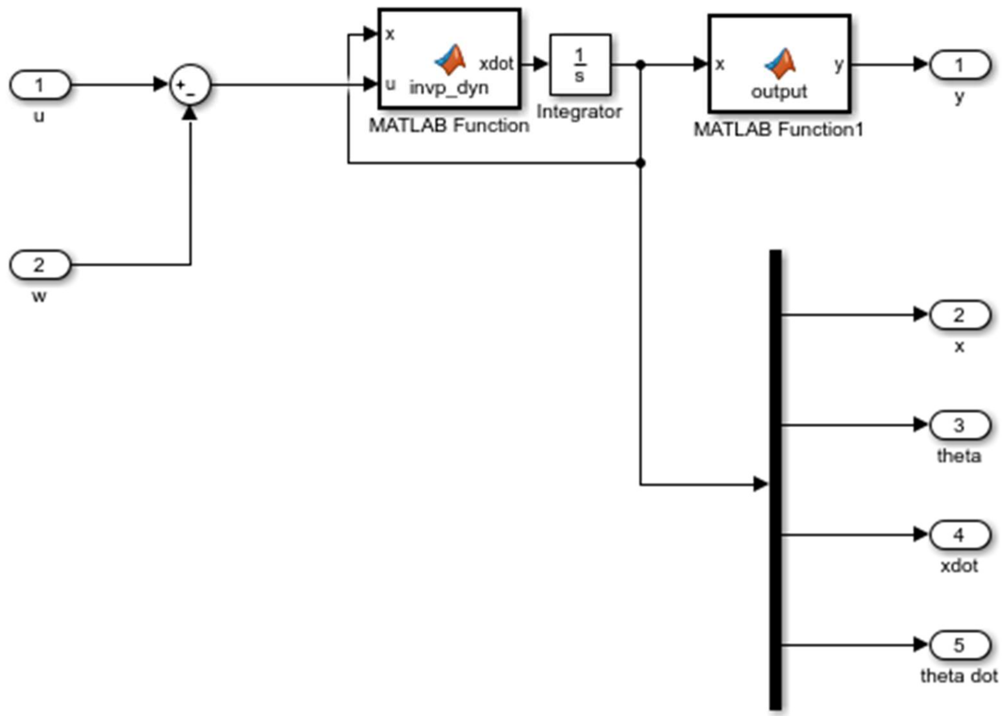
```
end
```

```
function u=state_feedback(xhat,K)

u=-(K(1)*xhat(1)+K(2)*xhat(2)+K(3)*xhat(3)+K(4)*xhat(4));

end
```

Discrete State Feedback Controller

```
function xhatplus=estimator(A_d, B_d, C_d, xhat, u, L_d,y)

    xhatplus= A_d*xhat + B_d*u + L_d*(y - C_d*xhat);
end
```

Discrete State Estimator



Derivative output and integration for original output

g) After swapping the desired pole locations for estimator and state feedback controller, the result is as follows.

K =1.0e+05 *[-0.5403 -3.7165 -0.6632 -4.4258]

K_d = 1.0e+05 *[-0.3325 -2.3208 -0.0831 -0.5584]

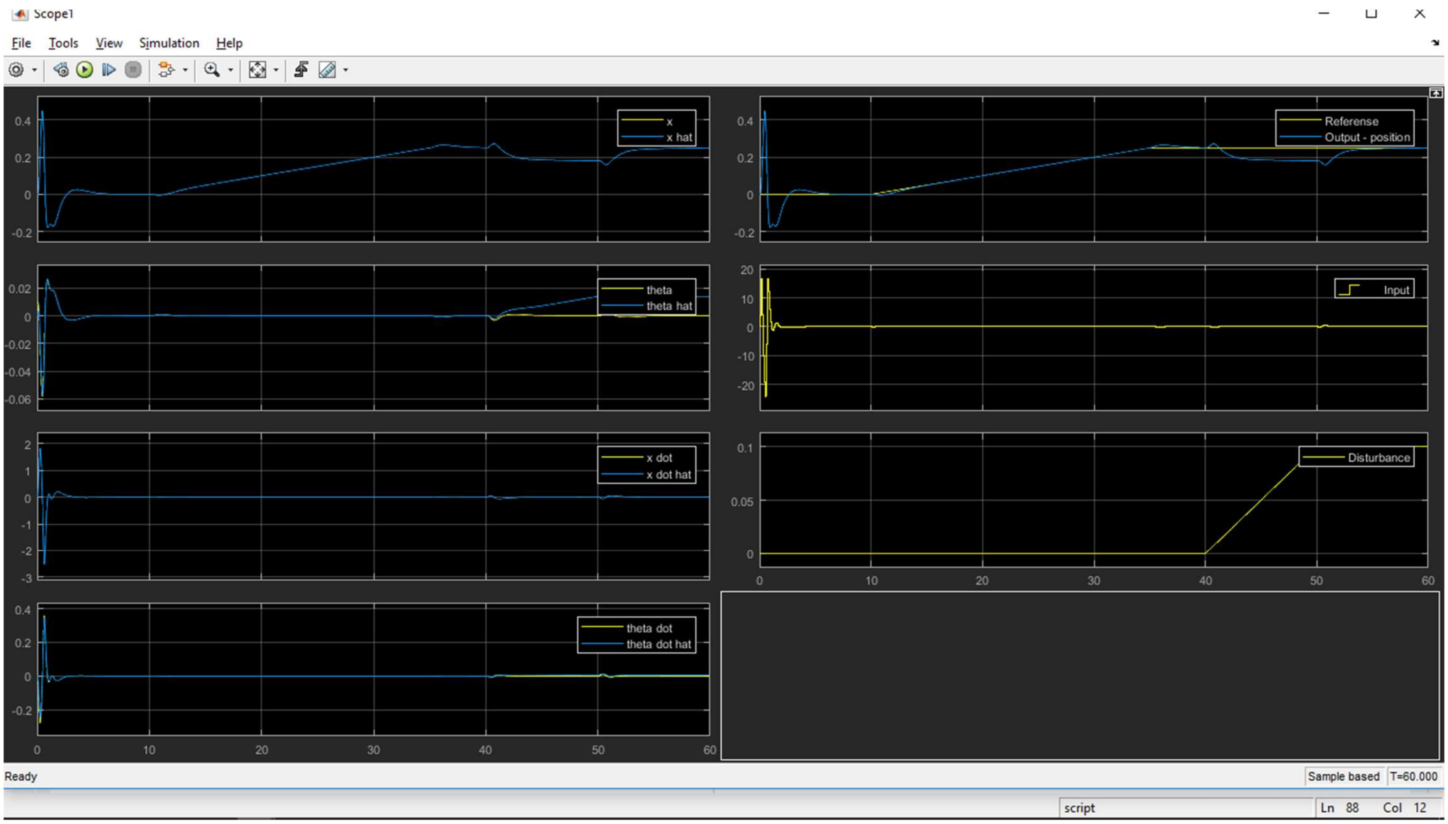
K_da =1.0e+05 *[-0.5403 -3.7165 -0.6632 -4.4258 -0.0001]

K_i = -8.3115

By Looking at gain, It is clearly said that It is not good choice because after swapping gains become much smaller which leads to system more unstable.

(h)

Simulation of the discrete controller for the following non-zero initial condition using Simulink: x(t) = 0.0, ˙ x(t) = 0, θ(t) = 0.5 , ˙ θ(t0) = 0.0.Result is shown below.



Conclusion:

- For discrete system, measured position and estimated position graph is one on another and looks like almost identical result. Cart Position starts from zero as Home position and cart moves forward and backward in order to maintain pendulum stable as initially pendulum is slightly off from 90 degree position and after few seconds, cart maintains almost identical position if no disturbance applied.
- For pendulum as we can see from graph that after slight giggling in initial period, where angle of pendulum is slightly off from 0 degree and in order to achieve it straight and after certain period of time, pendulum becomes stabilized.
- Measured velocity and estimated velocity graph are almost identical which is stabilize after some overshoot. Controller behaving well as predicted.
- Angular velocity graph is also identical which is stabilize after some overshoot. Controller behaving well as predicted.
- By comparing position with our reference desired signal, we can see that it is following perfectly and getting desired response of system.