INDEX

Topic - Controlling an Inverted Pendulum on a Cart using Full State Feedback Controller Using MATLAB

AIM - To control an inverted pendulum on a cart using full state feedback controller using MATLAB.

INTRODUCTION -

The system consists of an inverted pendulum mounted to a motorized cart. The inverted pendulum system is an example commonly found in control system textbooks and research literature. Its popularity derives in part from the fact that it is unstable without control, that is, the pendulum will simply fall over if the cart isn't moved to balance it. Additionally, the dynamics of the system are nonlinear. The objective of the control system is to balance the inverted pendulum by applying a force to the cart that the pendulum is attached to. A real-world example that relates directly to this inverted pendulum system is the attitude control of a booster rocket at takeoff.

The inverted pendulum is a classic problem in dynamics and control theory, and is used as a benchmark for testing control strategies. In our project, we will be controlling an inverted pendulum on a cart using full state feedback, by calculating the state space equations and converting them into controllable canonical form. Full state feedback (FSF), or pole placement, is a method employed in feedback control system theory to place the closed-loop poles of a plant in predetermined locations in the s-plane. Placing poles is desirable because the location of the poles corresponds directly to the eigenvalues of the system, which control the characteristics of the response of the system. However, the system must be controllable in order to implement this method.

SOFTWARE -

The software platform which we are using is MATLAB.

MATLAB is a proprietary multi-paradigm programming language and numerical computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages.

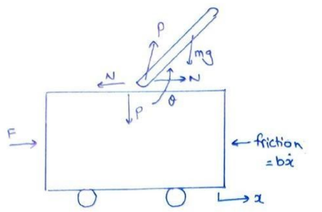
Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

METHODOLOGY -

A. Mathematical Modelling-Differential Form

With respect to mathematical modelling, a differential equation is an equation that contains one or more functions with its derivatives. The derivatives of the function define the rate of change of a function at a point. To obtain the differential form of the inverted pendulum-cart system, we draw the free body diagram of the system and obtain the necessary equations. From the differential equations, we find the transfer function.

The theoretical calculations for obtaining the differential equations and the transfer function are as shown below.



Free Body Diagram of Cart

Equation of Motion →

$$M\ddot{x} + b\dot{x} + N = F \quad \rightarrow (1)$$

Summing the forces in the FBD of pendulum in horizontal direction:

$$N = m\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta \quad \rightarrow (2)$$

$$(M+m)\ddot{x} + b\dot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \quad \rightarrow (3)$$

Sum of the forces perpendicular to pendulum:

$$P\sin\theta + N\cos\theta - mg\sin\theta = ml\ddot{\theta} + m\ddot{x}\cos\theta \rightarrow (4)$$

$$-Pl\sin\theta - Nl\cos\theta = I\ddot{\theta} \quad \rightarrow (5)$$

$$(I+ml^2)\ddot{\theta} + mgl\sin\theta = -ml\ddot{x}\cos\theta \quad \rightarrow (6)$$

→ Let ϕ represent the deviation of pendulum position →

$$\cos\theta = \cos(\pi+\phi) \approx -1 \quad \rightarrow (7)$$

$$\sin\theta = \sin(\pi+\phi) \approx -\phi \quad \rightarrow (8)$$

$$\dot{\theta}^2 = \dot{\phi}^2 \approx 0 \quad \rightarrow (9)$$

After substituting the above approx. into our non linear governing eqn:

$$(I+ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x} \rightarrow (10)$$

$$(M+m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \rightarrow (11)$$

TRANSFER FUNCTION :→

$$(I+ml^2)\phi(s)s^2 - mgl\phi(s) = mlX(s)s^2 \rightarrow (12)$$

$$(M+m)X(s)s^2 + bX(s)s - ml\phi(s)s^2 = U(s) \rightarrow (13)$$

$$X(s) = \left[\frac{I+ml^2}{ml} - \frac{g}{s^2}\right]\phi(s) \rightarrow (14)$$

$$(M+m)\left[\frac{I+ml^2}{ml} - \frac{g}{s^2}\right]\phi(s)s^2 + b\left[\frac{I+ml^2}{ml} - \frac{g}{s^2}\right]\phi(s)s$$
$$- ml\phi(s)s^2 = U(s) \rightarrow (15)$$

Transfer function:

$$\frac{\phi(s)}{U(s)} = \frac{\frac{mls^2}{q}}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s}$$

$$q = \left[(M+m)(I+ml^2) - (ml)^2\right]$$

$$P_{pend}(s) = \frac{\phi(s)}{U(s)} = \frac{\frac{ml}{q}s}{s^3 + \frac{b(I+ml^2)}{q}s^2 - \frac{(M+m)mgl}{q}s - \frac{bmgl}{q}} \quad \left[\frac{rad}{N}\right]$$

Transfer function with cart position $X(s)$ as output:

$$P_{cart}(s) = \frac{X(s)}{U(s)} = \frac{\frac{(I+ml^2)s^2 - gml}{q}}{s^4 + \frac{b(I+ml^2)}{q}s^3 - \frac{(M+m)mgl}{q}s^2 - \frac{bmgl}{q}s}$$

$$\left[\frac{m}{N}\right]$$

EQUATIONS:

$$(20) \leftarrow \frac{d^3\phi}{dt^3} + \frac{b(I+ml^2)}{q}\frac{d^2\phi}{dt^2} - \frac{(M+m)mgl}{q}\frac{d\phi}{dt} - \frac{bmgl}{q}\phi(t) = \frac{ml}{q}\frac{du}{dt}$$

$$(21) \leftarrow \frac{d^4x}{dt^4} + \frac{b(I+ml^2)}{q}\frac{d^3x}{dt^3} - \frac{(M+m)mgl}{q}\frac{d^2x}{dt^2} - \frac{bmgl}{q}\frac{dx}{dt} = \frac{(I+ml^2)}{q}\frac{d^2u}{dt} - gml\, u(t$$

B. State Space Modelling

In control engineering, a state-space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations or difference equations. The linearized equations of motion can also be represented in state-space form. Since we are dealing with a nonlinear system, the final state space model is made up of complicated terms, which would be time consuming to solve. In order to simplify it, we substitute the expression for inertia of motion of a rod. As seen below, the A matrix is simplified greatly, and the B matrix also changes accordingly.

From previous differential Equations are

$$(I+ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x} \quad -(1)$$

$$(M+m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \quad -(2)$$

Let the state space variables be x, $\dot{x}$, ϕ and $\dot{\phi}$

The output variable y ~~is~~ x and ϕ

From 1:

$$\ddot{x} = \frac{(I+ml^2)\ddot{\phi} - mgl\phi}{ml}$$

Substituting $\ddot{x}$ in 2

$$(M+m)\left[\frac{(I+ml^2)\ddot{\phi} - mgl\phi}{ml}\right] + \cancel{mgl}\, b\dot{x} - ml\ddot{\phi} = u$$

$$\frac{[I(M+m) + Mml^2]\ddot{\phi}}{ml} + \cancel{ml}\ddot{\phi} - (M+m)g\phi + b\dot{x} - \cancel{ml}\ddot{\phi} = u$$

$$\frac{[I(M+m) + Mml^2]\ddot{\phi}}{ml} = -b\dot{x} + (M+m)g\phi + u$$

$$\ddot{\phi} = \frac{-bml\dot{x}}{I(M+m)+Mml^2} + \frac{(M+m)mgl\phi}{I(M+m)+Mml^2} + \frac{uml}{I(M+m)+Mml^2} \quad (3)$$

Substituting value of $\ddot{\phi}$ in $\ddot{x}$

$$\ddot{x} = \frac{(I+ml^2)}{ml}\left[\frac{-bml\dot{x} + (M+m)mgl\phi + uml}{I(M+m)+Mml^2}\right] - \frac{mgl\phi}{ml}$$

$$= \frac{-b(I+ml^2)\dot{x}}{I(M+m)+Mml^2} + \frac{mgl\phi}{ml}\left[\frac{(I+ml^2)(M+m)}{I(M+m)+Mml^2} - 1\right] + \frac{u(I+ml^2)}{I(M+m)+Mml^2}$$

$$\ddot{x} = \frac{-b(I+ml^2)\dot{x}}{I(M+m)+Mml^2} + \frac{mgl\phi}{ml}\left(\frac{m^2l^2}{I(M+m)+Mml^2}\right) + \frac{u(I+ml^2)}{I(M+m)+Mml^2}$$

From 3 and 4

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-b(I+ml^2)}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-bml}{I(M+m)+Mml^2} & \frac{(M+m)mgl}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + 0$$

For uniform mass distribution of the pendulum

$$I = \frac{1}{3} m l^2$$

Where, I is rod inertia
m is mass of rod
l is distance from the rod axis rotation center to the mass center

Thus, the equation can be simplified as →

$$\left(\frac{1}{3}ml^2 + ml^2\right)\ddot{\phi} - mgl\phi = m\ddot{x}$$

$$\therefore \ddot{\phi} = \frac{3g}{4l}\phi + \frac{3}{4l}\ddot{x}$$

$$\therefore \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{3g}{4l} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{3}{4l} \end{bmatrix} u$$

C. Controllable Canonical Form(CCF)

For any given system, there are essentially an infinite number of possible state space models that will give the identical input/output dynamics. Thus, it is desirable to have certain standardized state space model structures: these are canonical forms. The controllable canonical form arranges the coefficients of the transfer function denominator across one row of the A matrix. The controllable canonical form is useful for the pole placement controller design technique.

We assume the initial values of the parameters such as mass of cart, mass of the pendulum, coefficient of friction of the cart, etc. We use the concepts of similarity transformation in MATLAB to find the controllable canonical form of the given system, as shown below.

```
M = 1.096; %Mass of the cart
m = 0.109; %Mass of the Pendulum
b = 0.1; %Coefficient of friction of the cart
I = 0.0034; %MOI of the pendulum
g = 9.8; %Accelaration due to gravity
l = 0.25; %Length to pendulum center of mass
```

```
p = I*(M+m)+M*m*l^2;
```

```
%From state space model calculated manually
A = [0      1              0           0;
     0      0              0           0;
     0      0              0           1;
     0    0       (3*g)/4*l  0];
B = [     0;      1;           0;         4*l];
C = [1 0 0 0;     0 0 1 0];
D = [0;0];
sys=ss(A,B,C,D)
```

```
 %Controllable canonical form calculation
syms s
expand (det(s*eye(4)-A)); % Characteristic equation
```

```
w=[0 -1.8375 0 1;-1.8375 0 1 0;0 1 0 0;1 0 0 0];
```

```
 %Controllable canonical form calculation
syms s
expand (det(s*eye(4)-A)); % Characteristic equation
```

```
w=[0 -1.8375 0 1;-1.8375 0 1 0;0 1 0 0;1 0 0 0];
```

```
%Conversion to CCF
cm=[B A*B A^2*B A^3*B]; %Controllabbility matrix
p=cm*w; %Transformation matrix
% Using similarity transformation
af=inv(p)*A*p
bf=inv(p)*B
cf=C*p
df=D
```

D. Controller Design

Controller design refers to techniques for controlling the modes of a system using any controllable device in the system. We call the previous MATLAB file and calculate the eigenvalues. One of the values is on the right half of the s-plane, therefore this system is inherently unstable. Therefore, we need to add an additional control law to stabilize it. We calculate the time constant, settling time, and calculate the dominant and non dominant poles of the system.

To calculate the non dominant poles, we referred to a research paper and used an additional parameter known as the separation factor, as seen below. We now obtain the new state space equations and plot them to get the controller design plot. However, we observe that the values in the graph are in the order of 10^(-3), and therefore no conclusions can be drawn from this graph. In order to rectify this, we go for the integral design next.

```
sympref('FloatingPointOutput',true);
state_space_modelling;   %Calling the state space model file
z=eig(A)    % One of the open loop poles is in the right half of the plane
% showing that the system is unstable
```

```
wn = 4.66;
zeta=0.858;
tau = 1/(zeta*wn)   % Time constant
ts = 4/(wn*zeta)    % Settling time using design parameters
p1 = -(wn*zeta)+wn*sqrt((0.858^2)-1)     % Dominant pole calculation
p2 = -(wn*zeta)-wn*sqrt((0.858^2)-1)
sep_factor=2;
a=sep_factor*zeta*9.3;
b=(sep_factor*wn)^2;
syms s
eqn = s^2+a*s+b == 0;
y=solve(eqn,s);     % Non-dominant poles calculation
p3=y(1)
p4=y(2)
```

```
% Controller Design
states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'r'};
outputs = {'x'; 'phi'};

sys_cl = ss(A-B*k,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);

t = 0:0.01:5;
r =0.2*ones(size(t));
[y,t,x]=lsim(sys_cl,r,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,2),'plot');
set(get(AX(1),'Ylabel'),'String','cart position (m)')
set(AX(1),'YLim',[-0.00300 0.00150])
set(AX(1),'YTick',-0.003:0.0005:0.0015)
set(get(AX(2),'Ylabel'),'String','pendulum angle (radians)')
set(AX(2),'YLim',[-0.00100 0.00175])
set(AX(2),'YTick',-0.001:0.0005:0.0018)
title("Step Response with Controller-Based -Feedback Control")
```

Desired characteristic equation →

$$s^4 + 23.9554 s^3 + 236.1935 s^2 + 1.0412e{+}03\, s + 1.8863\, e{+}03$$

$|sI - A + BK|$ →

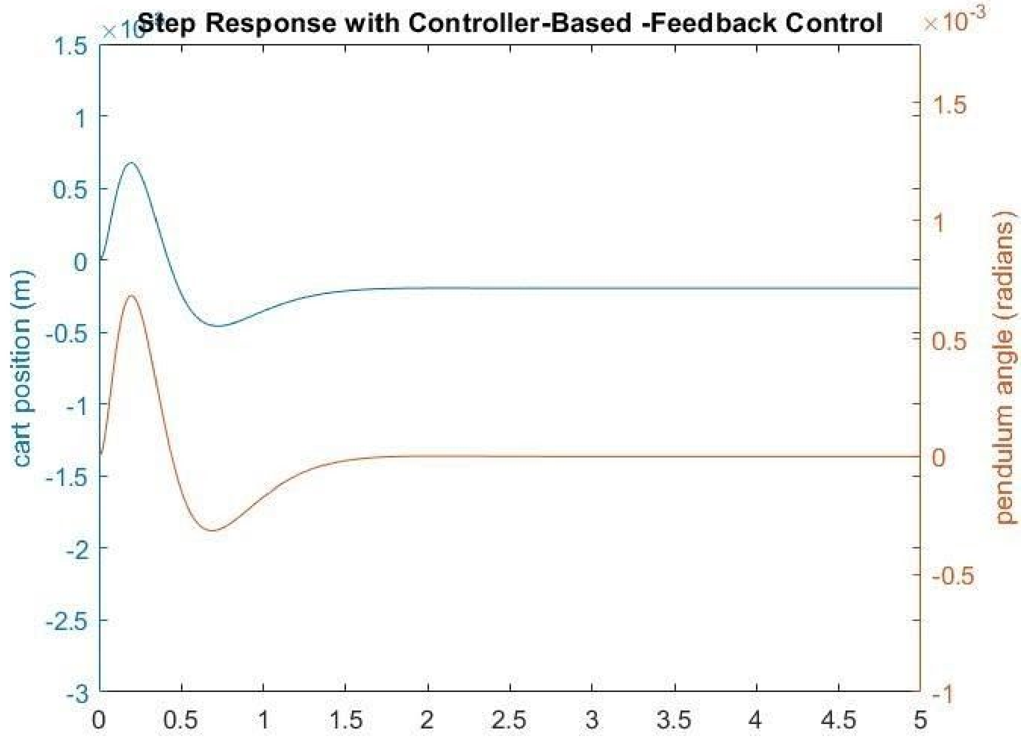
$$s^4 + (k_2 + k_4) s^3 + (k_1 + k_3 - 1.8375) s^2 - 1.8375 k_2 s - 1.8375 k_1$$

On comparing, $k_1 = -1027.42$

$k_2 = -566.911$

$k_3 = +1265.2575$

$k_4 = 589.91$



E. Integral Control

To correct the errors in the previous output, we apply integral control. Integral control detects and corrects trends in error over time. We use a special function “rscale” which provides pre-compensation and corrects the previous errors.

Given a system and the feedback matrix K,the function rscale(sys,K) or rscale(A,B,C,D,K) finds the scale factor N which will eliminate the steady-state error to a step reference for a continuous-time, single-input system with full-state feedback using the schematic below:

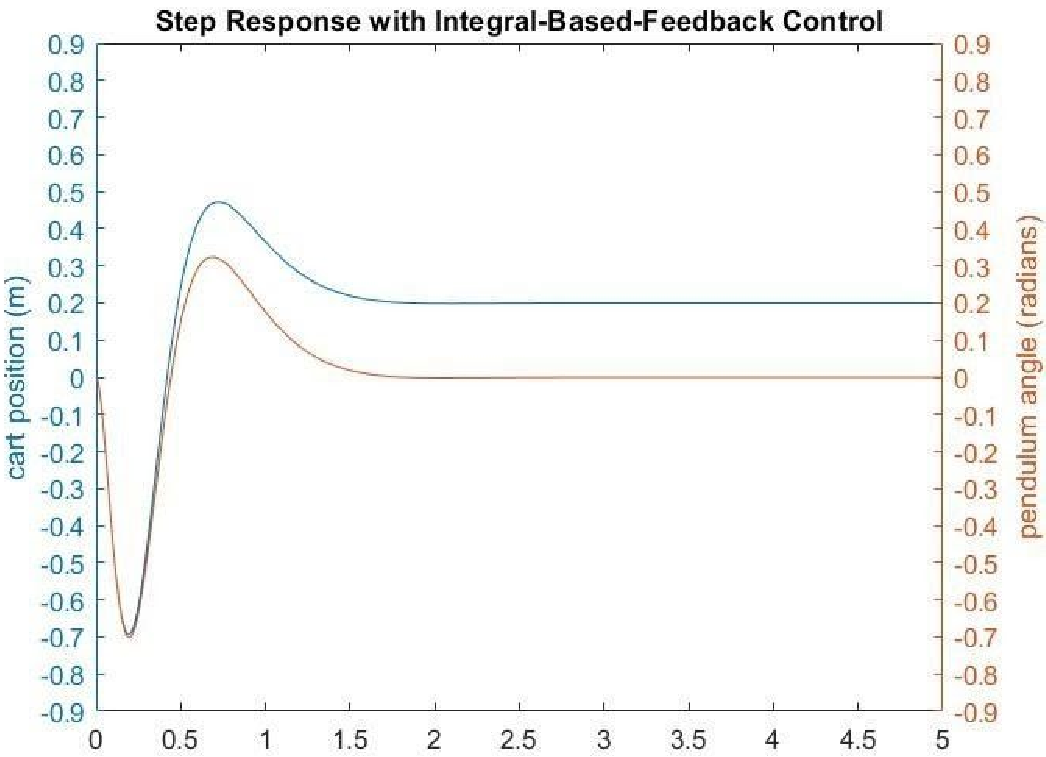
```
                             /---------\
 R           +          u   |  .        |
 ---> N --->() ----> | X=Ax+Bu |--> y = Cx ---> y
             -|              \---------/
              |                       |
              |<---- K <----|
```

Therefore the new matrices are obtained and plotted in MATLAB. We observe that the error has been sufficiently corrected in the new output.

```
% Integral Design
Cn = [1 0 0 0];
sys_ss = ss(A,B,Cn,0);
Nbar = rscale(sys_ss,k)
states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'r'};
outputs = {'x'; 'phi'};

sys_cl = ss(A-B*k,B*Nbar,C,D,'statename',states,'inputname',inputs,'outputname',outputs);

t = 0:0.01:5;
r =0.2*ones(size(t));
[y,t,x]=lsim(sys_cl,r,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,2),'plot');
set(get(AX(1),'Ylabel'),'String','cart position (m)')
set(AX(1),'YLim',[-0.9 0.9])
set(AX(1),'YTick',-0.9:0.1:0.9)
set(get(AX(2),'Ylabel'),'String','pendulum angle (radians)')
set(AX(2),'YLim',[-0.9 0.9])
set(AX(2),'YTick',-0.9:0.1:0.9)
title("Step Response with Integral-Based-Feedback Control")
```



F. Observer Design

In control theory, a state observer is a system that provides an estimate of the internal state of a given real system, from measurements of the input and output of the real system. If a system is observable, it is possible to fully reconstruct the system state from its output measurements using the state observer.

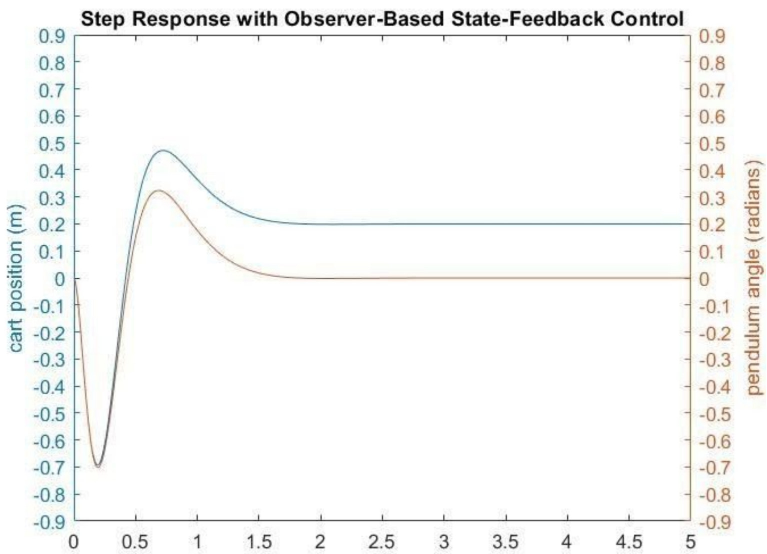
We use a specialised MATLAB function to place the poles, and subsequently construct a new state space model using the control law we developed. We plot the new matrices to get our final observer design step response with state feedback control.

```
P=[-40 -41 -42 -43];
L = place(A',C',P)';
Ace = [(A-B*k) (B*k);
       zeros(size(A)) (A-L*C)];
Bce = [B;
       zeros(size(B))];
Cce = [C zeros(size(C))];
Dce = [0;0];

states = {'x' 'x_dot' 'phi' 'phi_dot' 'e1' 'e2' 'e3' 'e4'};
inputs = {'r'};
outputs = {'x'; 'phi'};

sys_est_cl = ss(Ace,Bce,Cce,Dce,'statename',states,'inputname',inputs,'outputname',outputs);

t = 0:0.01:5;
r = 0.2*ones(size(t));
[y,t,x]=lsim(sys_est_cl,r,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,2),'plot');
set(get(AX(1),'Ylabel'),'String','cart position (m)')
set(get(AX(2),'Ylabel'),'String','pendulum angle (radians)')
title('Step Response with Observer-Based State-Feedback Control')
```



CONCLUSION -

In conclusion, it is clear that the inverted pendulum is a system with many variations that render it a fundamental control problem. It is used as a benchmark for testing control strategies.In this project, we have successfully controlled settling time of the inverted pendulum-cart system. From the observer design, we conclude that the system takes 1.5 seconds to settle. The cart has a positive displacement of 0.2 meters. The inverted pendulum has a maximum negative displacement of -0.7 radians (-40 degrees) and a positive displacement of 0.3 radians (17 degrees), before it ultimately settles to 0.Thus, the inverted pendulum is a system that helps engineers test the efficacy of new control methods, and works as a bridge between theoretical approaches and their application to real life problems.

FUTURE SCOPE -

Further work can be done in the control of inverted pendulum systems by applying nonlinear system control techniques such as the Linear Quadratic Regulation method. By applying such methods we can further control other control and optimize other parameters such as rise time and steady state error as well. The settling time can also be further controlled.

ACKNOWLEDGEMENTS

Firstly, we would like to express our heartfelt and sincere gratitude to our teacher and mentor. Dr. Arun N, for giving us the opportunity to carry out our project under his guidance. We wish to express our gratitude for his competent guidance, supervision and intelligent attention to our work. We always felt comfortable going to him if an experiment wasn't working, and knew he could help us figure out something else to try. The discussion sessions with him were very informative and helped us a lot in thinking clearly about our work. We thank him for keeping us motivated and inspiring us at the low tides.

We also take immense pleasure in thanking the School of Electrical Engineering for providing us the opportunity to carry out this project work. We are sure this tenure will surely help us in our future endeavours.

We would also like to extend our sincere gratitude to our team for their unconditional love, blessings and total support and for always encouraging us to do our best .This project would never have been possible without them.

REFERENCES

[1].Roshdy, Amr & zheng, Lin & Mokbel, Hany & Tongyu, Wang. (2012). Stabilization of Real Inverted Pendulum Using Pole Separation Factor. 10.2991/mems.2012.107.

[2].Razmjooy, Navid & A., Madadi & Alikhani, Hamid & Mohseni, Mona. (2014). Comparison of LQR and Pole Placement Design Controllers for Controlling the Inverted Pendulum. journal of worldd electrical engineering and technology.

[3]].Juan C. Martínez-García,2 Alberto Soria-López,2 and José de Jesús Rubio"On the Stabilization of the Inverted-Cart Pendulum Using the Saturation Function Approach"10.1155/2011/856015(2011)

[4].C. Aguilar Ibanez, J. C. Martinez Garcia, A. Soria Lopez, J. d. J. Rubio and M. S. Suarez Castanon, "Stabilization of the Inverted Cart-Pendulum System with Linear Friction," in IEEE Latin America Transactions, vol. 16, no. 6, pp. 1650-1657, June 2018, doi: 10.1109/TLA.2018.8444162.

[5].Siradjuddin, Indrazno & Amalia, Zakiyah & Erfan, ROHADI & Setiawan, Budhy & Setiawan, Awan & Putri, Ratna & Yudaningtyas, Erni. (2018). State-feedback control with a full-state estimator for a Cart-Inverted Pendulum System. International Journal of Engineering and Technology. 7. 203-209. 10.14419/ijet.v7i4.44.26985.