Linear Inverted Pendulum Report

Thomas Dragonetti - 3323

Joel Giordano - 5182

Lottie Jones - 5182

Norman Strickland - 3323

MECA 482- Control System Design

December 25, 2019

I. **Introduction**

"Linear Inverted Pendulum" is an experiment that consists of a straight beam mounted to a hinge (forming an inverted pendulum) on a DC servo motor driven cart, which travels along a single axis rack and pinion gantry. The goal of the experiment is to control the motor's position on the gantry using linear, or nonlinear, control algorithms and balance the inverted pendulum beam upright within a specified angle range. The system uses two encoders, one which measures the position of the cart on the linear rack and pinion gantry, and another to sense the angle of the inverted pendulum beam. A real life example of everyday technology utilizing a very similar model would be the Segway, a self-balancing electric vehicle. In the case of the Segway, the cart and gantry are replaced by a platform with motorized wheels on either side, and the inverted pendulum is represented by the human operator standing atop the platform. We will be creating a mathematical model of this system with a designed control system and simulating the results using SimuLink software.

II. Modeling

A. System Model:

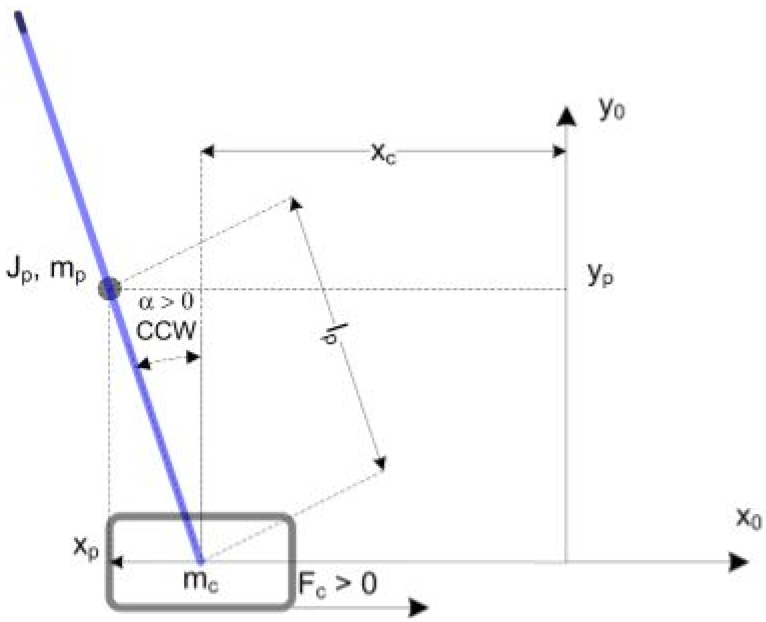

Figure 1. Linear Inverted Pendulum Schematic

B. Math Model:

After analyzing Figure 1, it is clear that there are 2 degrees of freedom. This means that four equations of motion should come from the system. By studying the system in Figure 1, the following equations can be written.

$$\ddot{x}_c = \frac{1}{J_T}\left(-\left(J_p + M_p l_p^2\right) B_{eq} \dot{x}_c - M_p l_p B_p \dot{\alpha} + M_p^2 l_p^2 g\alpha + \left(J_p + M_p l_p^2\right) F_c\right)$$

The second equation of motion can be found below.

$$\ddot{\alpha} = \frac{1}{J_T}\left(-\left(M_p l_p B_{eq}\right) \dot{x}_c - \left(J_{eq} + M_p\right) B_p \dot{\alpha} + \left(J_{eq} + M_p\right) M_p l_p g\alpha + M_p l_p F_c\right)$$

In these equations, J_T and J_{eq} are symbolizing the equations below.

$$J_T = J_{eq} J_p + M_p J_p + J_{eq} M_p l_p^2 \qquad J_{eq} = M_c + \frac{n_g K_g^2 J_m}{r_{mp}^2}$$

For this system, a state needs to be defined. X_c and $\boldsymbol{\alpha}$ are two of the equations of motion, so their derivatives will be the next two equations of motion. For now, the state is defined as follows.

$$x^T = \begin{bmatrix} x_c \; \alpha \; \dot{x}_c \; \dot{\alpha} \end{bmatrix}$$

Therefore, the following definitions can be made.

$$\dot{x}_1 = x_3 \qquad \dot{x}_2 = x_4 \qquad x_c = x_1 \qquad \alpha = x_2 \qquad \dot{x}_c = x_3 \qquad \dot{\alpha} = x_4$$

Using these definitions, the last two equations of motion can be found.

$$\dot{x}_3 = \frac{1}{J_T}\left(-\left(J_p + M_p l_p^2 \right) B_{eq} x_3 - M_p l_p B_p x_4 + M_p^2 l_p^2 g x_2 + \left(J_p + M_p l_p^2 \right) u \right)$$

$$\dot{x}_4 = \frac{1}{J_T}\left(-\left(M_p l_p B_{eq} \right) x_3 - \left(J_{eq} + M_p \right) B_p x_4 + \left(J_{eq} + M_p \right) M_p l_p g x_2 + M_p l_p u \right)$$

The input equation for state space representation can be seen below.

$$\dot{x} = Ax + Bu$$

The matrices for A and B can be filled in using the four equations of motion outlined previously. These matrices are as follows below.

$$A = \frac{1}{J_T}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & M_p^2 l_p^2 g & -\left(J_p + M_p l_p^2 \right) B_{eq} & -M_p l_p B_p \\ 0 & \left(J_{eq} + M_p \right) M_p l_p g & -M_p l_p B_{eq} & -\left(J_{eq} + M_p \right) B_p \end{bmatrix}$$

$$B = \frac{1}{J_T}\begin{bmatrix} 0 \\ 0 \\ J_p + M_p l_p^2 \\ M_p l_p \end{bmatrix}$$

Now the matrices for the output equation need to be calculated. The general equation can be seen below for the output.

$$y = Cx + Du$$

The matrices for C and D can be seen below.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$m = 0.38kg$ $k = 0$ $b = 0$ DOF: 2 (Angle of the Pendulum, position of the cart)

Max Motor Speed: 6000 rpm

Pendulum mass (kg):

Medium: 0.127 Long: 0.23

Cart Travel: 81.4 cm= 0.814 m

III. Controller Design and Implementation

A. Appendix- MATLAB Code(s)

drawcartpend_bw.m

```
function drawcartpend(y,m,M,L)
x = y(1);
th = y(3);

% kinematics
% x = 3;        % cart position
% th = 3*pi/2;   % pendulum angle

% dimensions
% L = 2;  % pendulum length
W = 1*sqrt(M/5);  % cart width
H = .5*sqrt(M/5); % cart height
wr = .2; % wheel radius
mr = .3*sqrt(m); % mass radius

% positions
% y = wr/2; % cart vertical position
y = wr/2+H/2; % cart vertical position
w1x = x-.9*W/2;
w1y = 0;
w2x = x+.9*W/2-wr;
w2y = 0;

px = x + L*sin(th);
py = y - L*cos(th);

plot([-10 10],[0 0],'w','LineWidth',2)
```

```
hold on
rectangle('Position',[x-W/2,y-H/2,W,H],'Curvature',.1,'FaceColor',[1 0.1 0.1],'EdgeColor',[1 1 1])
rectangle('Position',[w1x,w1y,wr,wr],'Curvature',1,'FaceColor',[1 1 1],'EdgeColor',[1 1 1])
rectangle('Position',[w2x,w2y,wr,wr],'Curvature',1,'FaceColor',[1 1 1],'EdgeColor',[1 1 1])

plot([x px],[y py],'w','LineWidth',2)

rectangle('Position',[px-mr/2,py-mr/2,mr,mr],'Curvature',1,'FaceColor',[.3 0.3 1],'EdgeColor',[1 1
1])

% set(gca,'YTick',[])
% set(gca,'XTick',[])
xlim([-5 5]);
ylim([-2 2.5]);
set(gca,'Color','k','XColor','w','YColor','w')
set(gcf,'Position',[10 900 800 400])
set(gcf,'Color','k')
set(gcf,'InvertHardcopy','off')

% box off
drawnow
hold off
```

poleplace_cartpend.m

```
%clear all, close all, clc

m = 1;
M = 5;
L = 2;
g = -10;
d = 1;

s = 1; % pendulum up (s=1)
```

```
A = [0 1 0 0;
    0 -d/M -m*g/M 0;
    0 0 0 1;
    0 -s*d/(M*L) -s*(m+M)*g/(M*L) 0];

B = [0; 1/M; 0; s*1/(M*L)];
eig(A)

rank(ctrb(A,B))  % is it controllable

%%  Pole placement

% p is a vector of desired eigenvalues
% p = [-.01; -.02; -.03; -.04]; % not enough
p = [-.3; -.4; -.5; -.6];  % just barely
p = [-1; -1.1; -1.2; -1.3]; % good
p = [-2; -2.1; -2.2; -2.3]; % aggressive
p = [-3; -3.1; -3.2; -3.3]; % aggressive
% p = [-3.5; -3.6; -3.7; -3.8]; % breaks
K = place(A,B,p);
% K = lqr(A,B,Q,R);

tspan = 0:.00015:15;
if(s==-1)
    y0 = [0; 0; 0; 0];
    [t,y] = ode45(@(t,y)cartpend(y,m,M,L,g,d,-K*(y-[4; 0; 0; 0])),tspan,y0);
elseif(s==1)
    y0 = [-3; 0; pi+.1; 0];
%     [t,y] = ode45(@(t,y)cartpend(y,m,M,L,g,d,-K*(y-[1; 0; pi; 0])),tspan,y0);
    [t,y] = ode45(@(t,y)cartpend(y,m,M,L,g,d,-K*(y-[1; 0; pi; 0])),tspan,y0);
else
end
```

```
for k=1:100:length(t)
    drawcartpend_bw(y(k,:),m,M,L);
end
```

cartpend.m

```
function dy = cartpend(y,m,M,L,g,d,u)

Sy = sin(y(3));
Cy = cos(y(3));
D = m*L*L*(M+m*(1-Cy^2));

dy(1,1) = y(2);
dy(2,1) = (1/D)*(-m^2*L^2*g*Cy*Sy + m*L^2*(m*L*y(4)^2*Sy - d*y(2))) + m*L*L*(1/D)*u;
dy(3,1) = y(4);
dy(4,1) = (1/D)*((m+M)*m*g*L*Sy - m*L*Cy*(m*L*y(4)^2*Sy - d*y(2))) - m*L*Cy*(1/D)*u
+.01*randn;
```

B. References

[1] Quanser.com, Linear servo base unit with Inverted Pendulum, Retrieved by 10-17-2019 from https://www.quanser.com/products/linear-servo-base-unit-inverted-pendulum/#overview