Linear Inverted Pendulum

Due: 12/25/2019

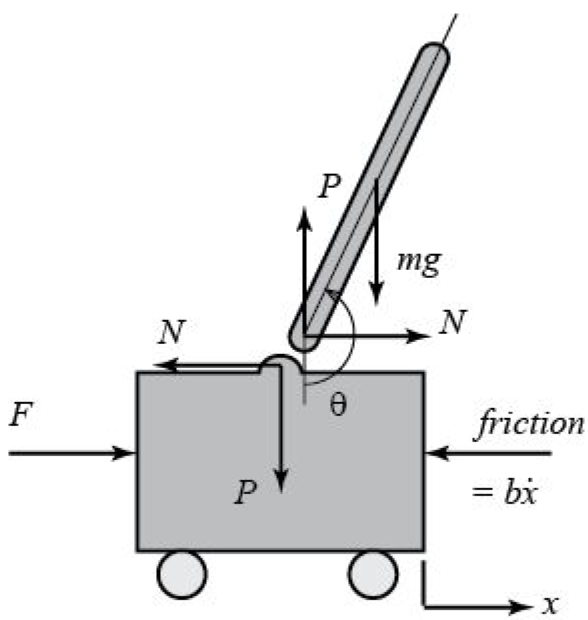
Luis Arevalo (5182)
Jordan Bailey Fletcher (5182)
Esmeralda Hernandez (5182)
Daniel Sousa (5182)

Introduction

In 1965 the first pendulum was invented by Christiaan Huygens. This linear inverted pendulum consists of a cart which moves linearly, left to right, in a way such that the pendulum stays upright. This is done by maintaining the pendulum's center of mass above its pivot point. In order to accomplish this, the team will need to mathematically relate the position of the cart 'X' and the angle of the pendulum 'Θ' in order to create an algorithm for the desired control system.

We used the website from MIT which explained exactly how the pendulum functions and its control system. From this pdf, we were able to obtain our equations in order to do the transfer function and state space.

By looking through this document of the inverted pendulum, the reader should expect to learn the background of the pendulum, such as who invented it, when, and where. The figure below shows all the angles and forces that acts on it, which we also show the equations for it. Through MATLAB, we code the transfer function and state space. Also through MATLAB, we coded different functions in order to stimulate the cart and how it works.



With the notation x – cart position, θ – pendulum angle and F – applied force, the system can be described with the c

$$(M+m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta = F \qquad (1)$$

$$l\ddot{\theta} + \ddot{x}\cos\theta - g\sin\theta = -f_\theta\dot{\theta} \qquad (2)$$

where M and m denotes the cart and pendulum mass, respectively, l the pendulum length, g the gravitational constant and f_θ the friction coefficient for the link where the pendulum is attached to the cart. A short derivation of the equations can be found in Appendix B.

With the following state variables

$$x_1 = x, \quad x_2 = \dot{x}, \quad x_3 = \theta, \quad x_4 = \dot{\theta}$$

and some calculations, this yields the state space equations

$$\begin{aligned} \dot{x}_1 &= \dot{x} = x_2 \\ \dot{x}_2 &= \ddot{x} = \frac{-mg\sin x_3\cos x_3 + mlx_4^2\sin x_3 + f_\theta m x_4\cos x_3 + F}{M + (1-\cos^2 x_3)m} \\ \dot{x}_3 &= \dot{\theta} = x_4 \\ \dot{x}_4 &= \frac{(M+m)(g\sin x_3 - f_\theta x_4) - (lmx_4^2\sin x_3 + F)\cos x_3}{l(M + (1-\cos^2 x_3)m)} \end{aligned} \qquad (3)$$

Modeling

*all calculations were done on matlab, published it and inserted pictures from our solutions.

Transfer Function

Code for transfer function:

```
M = 0.5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;
q = (M+m)*(I+m*l^2)-(m*l)^2;
s = tf('s');

P_cart = (((I+m*l^2)/q)*s^2 - (m*g*l/q))/(s^4 + (b*(I + m*l^2))*s^3/q - ((M + m)*m*g*l)*s^2/q - b*m*g*l*s/q);

P_pend = (m*l*s/q)/(s^3 + (b*(I + m*l^2))*s^2/q - ((M + m)*m*g*l)*s/q - b*m*g*l/q);

sys_tf = [P_cart ; P_pend];

inputs = {'u'};
outputs = {'x'; 'phi'};

set(sys_tf,'InputName',inputs)
set(sys_tf,'OutputName',outputs)

sys_tf
```

Results from code:

```
sys_tf =

                   4.182e-06 s^2 - 0.0001025
x:  ----------------------------------------------------------
     2.3e-06 s^4 + 4.182e-07 s^3 - 7.172e-05 s^2 - 1.025e-05 s

                          1.045e-05 s
phi:  -----------------------------------------------------
       2.3e-06 s^3 + 4.182e-07 s^2 - 7.172e-05 s - 1.025e-05
```

State Space

Code for State-Space

```
M = .5;
m = 0.2;
b = 0.1;
I = 0.006;
g = 9.8;
l = 0.3;

p = I*(M+m)+M*m*l^2; %denominator for the A and B matrices

A = [0      1              0           0;
     0 -(I+m*l^2)*b/p  (m^2*g*l^2)/p   0;
     0      0              0           1;
     0 -(m*l*b)/p       m*g*l*(M+m)/p  0];
B = [     0;
     (I+m*l^2)/p;
          0;
        m*l/p];
C = [1 0 0 0;
     0 0 1 0];
D = [0;
     0];

states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'u'};
outputs = {'x'; 'phi'};

sys_ss = ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs)
```

```
sys_ss =

  A =
                  x    x_dot      phi  phi_dot
   x              0        1        0        0
   x_dot          0  -0.1818    2.673        0
   phi            0        0        0        1
   phi_dot        0  -0.4545    31.18        0

  B =
              u
   x          0
   x_dot  1.818
   phi        0
   phi_dot  4.545

  C =
              x    x_dot      phi  phi_dot
   x          1        0        0        0
   phi        0        0        1        0

  D =
        u
   x    0
   phi  0
```

```
sys_tf =

           1.818 s^2 + 1.615e-15 s - 44.55
   x:  --------------------------------------
       s^4 + 0.1818 s^3 - 31.18 s^2 - 4.455 s

                4.545 s - 1.277e-16
   phi:  ----------------------------------
         s^3 + 0.1818 s^2 - 31.18 s - 4.455
```

Controller Design and Simulations

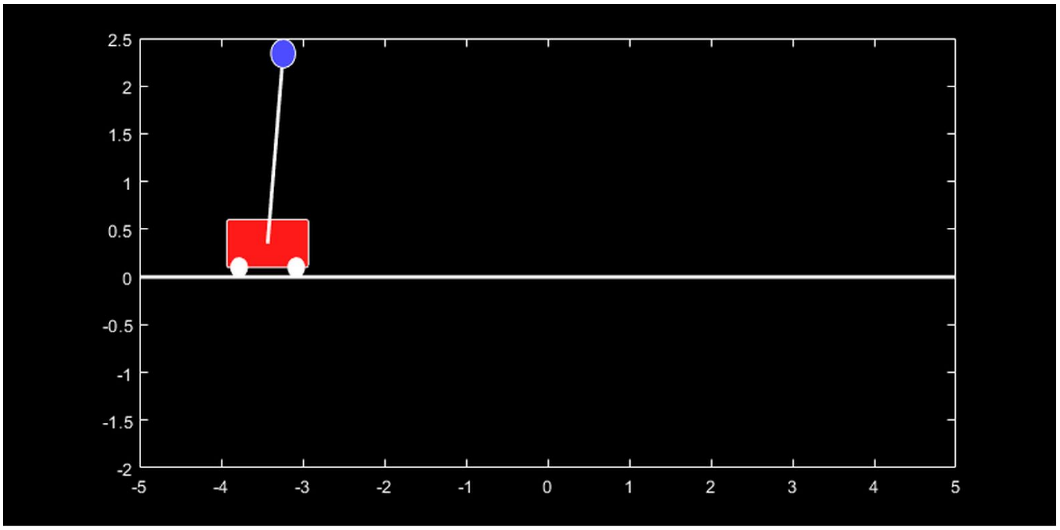
Function 1:

```
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')
```

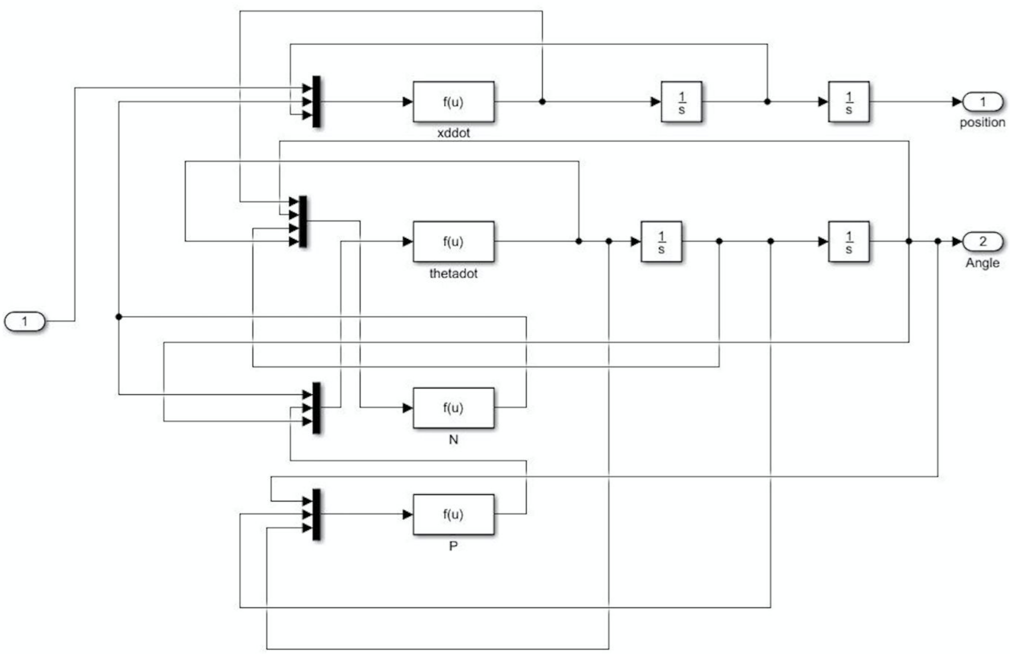
```
set(gcf,'InvertHardcopy','off')

% box off
drawnow
hold off
```

Script for the function 1:

```
clear all, close all, clc

m = .001;
M = 5;
L = 2;
g = -10;
d = 1;

tspan = 0:.1:30;
y0 = [0; 0; pi; .1];
[t,y] = ode45(@(t,y)cartpend(y,m,M,L,g,d,0),tspan,y0);

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



Block Diagram



Checklist

http://web.mit.edu/klund/www/papers/UNP_pendulum.pdf