Swing up and stabilization of a pendulum system via energy method.

Thein Win

ELE 592 Report

Contents

1 Introduction

This report analyzes the swinging up and stabilization of a pendulum system with friction by using an energy control method. The energy method utilized in this report is described in the Journal Paper “Swinging up a pendulum by energy control” by KJ Astrom & K Furuta. By using this energy control method we can simplify the control system and allow for a more robust design. However the affects of frictional losses as the pendulum swing are not considered in the Journal Paper. The objective of this report is to design a pendulum system and take into account the loss of energy due to frictional forces. The pendulum control system will be split into two major sections, the swing-up section and the stabilization section. In the swing-up section the energy method is used, with considerations for friction included, as a non-linear controller to swing up the pendulum to its inverted vertical position. Once the pendulum is at or near the inverted position, the stabilization section will take control and drive the system to stabilize at that inverted position using a linear state-feedback controller.

2 Equations of Motion

The system we will be considering is a single pendulum mounted on a motor driven cart. The cart is placed along a finite track and the pendulum is free to rotate 360° along axis of the track. As the cart accelerated along the track, the pendulum will swing according to the direction of the acceleration. The pendulum-cart system is driven by a motor that actuates a lead screw which moves the cart. The pendulum system has the following properties: the pendulum mass of m, a moment of inertia with respect to the pivot point as J, the angle between the vertical and the pendulum to be θ, the angle is positive in the clockwise direction, the acceleration of gravity to be g, and the acceleration of the pivot to be u, the acceleration is positive if in the direction of the positive x-axis. Using Newton's law of motion we can derive that the equation of motion for this pendulum system to be:

$$J\ddot{\theta} - mgl\sin\theta + mul\cos\theta = 0 \tag{1}$$

The equation of motion given in the paper is based on the assumption that friction has been neglected. But for our controller we need account for losses due to friction. But before we add the friction term let's rearrange our Eq. (1) to reduce the number of variables. Currently we have six variables in the equation, most of these variables we can eliminate by combining some of the terms. To do this we will borrow a technique shown in “Control System Design A State-Space Approach” by Richard Vaccaro. Let us create a new variable called A, where $A = {}^{mgl}/_{J}$. Eq. (1) is rewritten with this new variable as:

$$\ddot{\theta} - A\sin\theta + \frac{A}{g}u\cos\theta = 0 \tag{2}$$

The equation of motion, Eq. (2) only contain one variable, A. The value for A can be experimentally calculated relatively easily. It is found by letting the pendulum swing freely and finding the natural frequency of the pendulum in radians per second. The square of frequency is the value of A.

To account for frictional losses we will add an additional term to the equation of motion giving us Eq. (3). Where α is the coefficient of friction that will compensate for the loss in energy as the pendulum swings up. The friction expression will be a function of α , $\dot{\theta}$, and the sgn function. The coefficient of friction is chosen to be a default value of, $\alpha = 0.04$.

$$\ddot{\theta} - A\sin\theta + \frac{A}{g}u\cos\theta = -\alpha\dot{\theta}^2 sgn(\dot{\theta}) \tag{3}$$

The equation of motion derived in Eq. (3) has two state variables, the angle θ and the rate of change of the angle $\dot{\theta}$. To create our plant equation let us say that $x_1 = \theta$ and $x_2 = \dot{\theta}$. Using this relationship we can rewrite Eq. (3) to get a second-order plant:

$$\begin{aligned} \dot{x}_1 &= \dot{\theta} = x_2 \\ \dot{x}_2 &= \ddot{\theta} = -\alpha\dot{\theta}^2 sgn(\dot{\theta}) + A\sin\theta - \tfrac{A}{g}u\cos\theta \end{aligned} \tag{4}$$

The pendulum cart system will have an initial condition of $\theta = \pi$ and $\dot{\theta} = 0.17$. When $\theta = \pi$, the pendulum is hanging freely on the bottom vertical position. The $\dot{\theta} = 0.17$ is mimicking the behavior of when the pendulum is lightly flicked, giving it the initial energy it needs to start swinging. When the pendulum has swing up and $\theta = 0$ our pendulum will be in the inverted pendulum position.

3 Non-Linear System

To design the non-linear system we will break it down into two sections. The first section will be our energy controller, this will be our control theory that will determine how much pivot acceleration we need to swing the pendulum up to the inverted vertical position. We will be using the lyapanov method to determine a control theory that will drive our system to the desired state, the inverted vertical position. The second section will consider our physical components we have, i.e. power amplifier, motor, and cart-pendulum system. We need a way to control the voltage input to the motor to get the desired distance for the cart to travel to get the required acceleration from the energy controller.

3.1 Energy Controller

To design the controller we need an energy equation that describes the pendulum system. The energy of the system is described in the Journal Paper "Swinging up a pendulum by energy method" as shown in Eq. (5). It is defined to be zero when the pendulum is in the inverted vertical position.

$$E = \frac{1}{2}\dot{x}_2^2 + A\left[cos(x_1) - 1\right] \tag{5}$$

By controlling the energy of the pendulum, we can swing the pendulum up to the inverted position by giving it the energy that corresponds to that position. As the pendulum swings up it will pass through the unstable equilibrium point at the inverted position. A state-feedback linear regular will then be used to catch and stabilize the pendulum to this position. To perform the energy control of the system we need to first the relationship between the acceleration of the pivot, u, and the rate of change of the energy of the pendulum, $\frac{dE}{dt}$. Taking the derivative of the energy equation, Eq. (5), with respect to time we find:

$$\frac{dE}{dt} = \dot{E} = \dot{x}_2\ddot{x}_2 - A\sin(x_1)x_2 \tag{6}$$

With the energy equation and our plant equations we can design a controller that will swing the pendulum up to the vertical inverted position. A control method can be obtained by using the lyapunov method with the lyapunov function $V = {}^{E^2}\!/_2$. To design this controller we need an expression of the control law (equation for u). To find our control law we need to choose an expression that will ensure that the derivative of the lyapanouv function, $\dot{V} = E\dot{E}$, is negative definite. By substituting Eq. (4) , Eq. (5) and Eq. (6) we can find that the derivative of the lyapanouv function is found to be:

$$\dot{V} = E\left[-\alpha x_2^3 sgn(x_2) - \frac{A}{g}x_2\cos(x_1)u\right] \tag{7}$$

We pick an arbitrary expression that we know is negative definite and equate it to $\dot{V}$, Eq. (8) and then solve for u ,Eq. (9), our control law.

$$-E^2\dot{\theta}^2(\frac{A}{g})^2\cos^2(\theta)K = \dot{V} = E\left[-\alpha x_2^3 sgn(x_2) - \frac{A}{g}x_2\cos(x_1)u\right] \tag{8}$$

$$u = \frac{A}{g}KEx_2\cos(x_1) - \frac{g}{A}\alpha x_2^2\frac{sgn(x_2)}{\cos(x_1)} \tag{9}$$

3.2 Motor Control

We have our control law u, which is the acceleration of the pivot of the pendulum, but we need to use that acceleration to drive the motor-driven cart. Recall that our physical system is a pendulum-cart system that is connected to a motor that is driving a lead screw which in turn drives the cart. And the pendulum is hanging off from the cart and is free to rotate. When a voltage is applied to the motor it turns the lead screw and drives the cart a certain distance which creates a pivot acceleration on the pendulum. We need to find a way to calculate the voltage and distance we want the cart to move based on the pivot acceleration we need to swing the pendulum up to the desired position. Based on the equipment available at the University of Rhode Island the transfer function for the relationship between the voltage applied to the motor and the distance the cart travels can be described as:

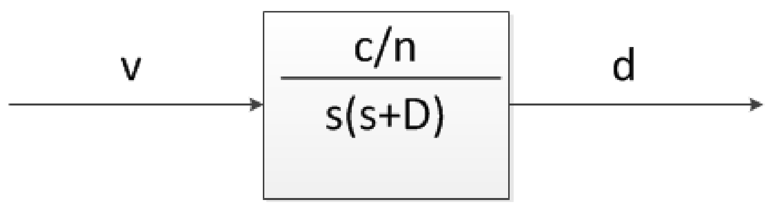

Fig. 1: Motor Transfer Function

Where v is the voltage applied to the motor power amplifier and D is the displacement of the cart in meters. The value n is the transduction ratio of the lead screw. The pivot acceleration needs to be multiplied by n to account for the transduction ratio. A state-space model of the cart is shown in Eq. (10), where x_3 is the motor angular position in radians and $x_4$is the motor angular velocity in radians per second.

$$\begin{aligned} \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -Cx_4 + Dv \end{aligned} \tag{10}$$

The angular acceleration of the cart,$\dot{x}_4$, is divided by n, plugging this value in for u into Eq. (4) and include Eq. (10) we get our fourth order non-linear plant Eq. (11).

$$\begin{array}{cc} \dot{x}_1 = x_2 & x_1 = pendulum\,position \\ \dot{x}_2 = A\sin x_1 - \frac{A}{g}\cos x_1 \left[\frac{-Cx_4+Dv}{n}\right] - \alpha x_2^2 sgn(x_2) & x_2 = pendulum\,velocity \\ \dot{x}_3 = x_4 & x_3 = motor\,position \\ \dot{x}_4 = -Cx_4 + Dv & x_4 = motor\,velocity \end{array} \tag{11}$$

This plant take the voltage, v, as the input to the overall cart-pendulum system. The energy controller calculates u as a function of $x_1$and x_2. The cart acceleration must be translated to a voltage to get our final close-loop feedback system. We will utilize a phase-lead compensator with pre-filter design shown in “A Control System Approach” to design our motor control. The motor control will translate the pivot acceleration from the energy controller to the the desired input voltage, v.

We add a compensator and use real double roots that would give us critical damping. This would give us a new overall transfer function as shown in Eq. (12) Where C is the zero compensator that we have chosen, and C' is an arbitrary value chosen to be greater than C. We define $C = 25$ and $C' = 28$. Since we are trying to achieve critical damping we are looking for the function to have the form $(s+p)^2$. We can solve for the overall transfer function shown in Fig. (3.2), giving us Eq. (12). Using this we can get expression for $K = \frac{(C'/2)^2}{D}$ and $C' = 2P$.

$$\frac{H}{1+H} = \frac{KD}{s^2 + c's + KD} \tag{12}$$

The input signal we are converting is the pivot acceleration and the output of our function is the motor angular position. We can convert the pivot acceleration to angular acceleration by dividing by n and

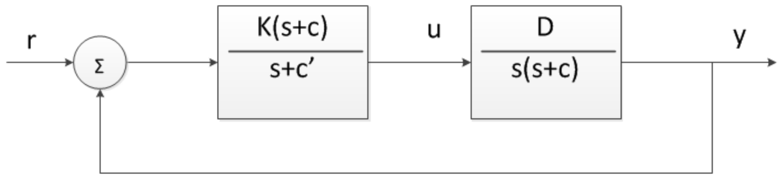

Fig. 2: Compensator & Pre-Filter Transfer Function

integrating twice to get angular position, as shown in Fig. (3) The values we need for the pre-filter can be found using the compensator transfer function, the coefficient of the Eq. (12).

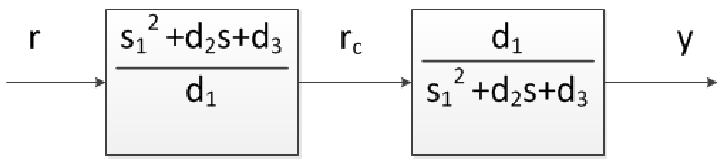

Fig. 3: Pre-Filter Transfer Function

$$r_c = \frac{1}{d_1}\ddot{r} + \frac{d_2}{d_1}\dot{r} + \frac{d_3}{d_1}r \tag{13}$$

From the transfer function and Eq. (13) we can derive expression for the two constants, $C_1 = {}^4/_{C'^2}$and $C_2 = {}^4/_{C'}$.The pre-filter is an improper transfer function unlike the compensator. With this we can now take our pivot acceleration and control the pendulum with the voltage input directly feeding into the cart-pendulum plant shown in Eq. (11).

4 State Feedback Linear Regulator

The non-linear control system will swing up the pendulum from the down vertical position to the inverted vertical position. A state-feedback linear regulator is needed to stabilize the pendulum in the inverted vertical position. To create the linear regulator from the non-linear plant, Eq. (11) must be linearized about the equilibrium point. The pendulum must be stabilize in the inverted position, this is where the equilibrium point for the linear system will be. Recall that $\theta = 0$ when the pendulum is in the inverted position. At this position we want to stabilize the pendulum thus the equilibrium points for our 4-th order system will be:

$$x_e = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{14}$$

Using the equilibrium points, the system is translated to the z- coordinate system,but since the equilibrium point of the system is zero, the translation is a direct substitution of variables. The input voltage is also translated in a similar method, $v = w$.

$$\begin{gathered} \dot{z}_1 = z_2 \\ \dot{z}_2 = A\sin z_1 - \tfrac{A}{g}\cos z_1 \left[\tfrac{-Cz_4 + Dw}{n}\right] - \alpha z_2^2 sgn(z_2) \\ \dot{z}_3 = z_4 \\ \dot{z}_4 = -Cz_4 + Dw \end{gathered} \tag{15}$$

A linear function has the form $\dot{z} = Az + Bw$, where z is the new state variables and w is the input. To linearize Eq. (15) the expression for A and B must be calculated. We can find the A matrix by taking the partial derivative for each of the state variable function with respect to z,. Similarly we can find the B matrix by taking the partial derivative for each of the state variable function with respect to the input w.

$$A = \frac{\partial f}{\partial z} \mid_{z=0,w=0} \qquad B = \frac{\partial f}{\partial w} \mid_{z=0,w=0} \tag{16}$$

Recall that:

$$\frac{\partial f}{\partial z} = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \frac{\partial f_1}{\partial z_3} & \frac{\partial f_1}{\partial z_4} \\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \frac{\partial f_2}{\partial z_3} & \frac{\partial f_2}{\partial z_4} \\ \frac{\partial f_3}{\partial z_1} & \frac{\partial f_3}{\partial z_2} & \frac{\partial f_3}{\partial z_3} & \frac{\partial f_3}{\partial z_4} \\ \frac{\partial f_4}{\partial z_1} & \frac{\partial f_4}{\partial z_2} & \frac{\partial f_4}{\partial z_3} & \frac{\partial f_4}{\partial z_4} \end{bmatrix} \qquad \frac{\partial f}{\partial w} = \begin{bmatrix} \frac{\partial f_1}{\partial w} \\ \frac{\partial f_2}{\partial w} \\ \frac{\partial f_3}{\partial w} \\ \frac{\partial f_4}{\partial w} \end{bmatrix} \tag{17}$$

Using Eq. (16) and Eq. (17) the expressions for the A and B matrix are found to be:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ A & 0 & 0 & \frac{Ac}{gn} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -c \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ -\frac{Ac}{gn} \\ 0 \\ D \end{bmatrix}$$

Now that we have obtained both the A and B matrix we have created our linearize system based on the non-linear plant. But our goal is to create a linear regular so we must find the feedback gain value that will drive our system to the equilibrium point, meaning we need find k (the gain). To calculate the gain value there is a MATLAB function called "place" which will calculate the gain value based on the values of the A, B matrix and the spoles for the system. To find the gain, we need spoles, which are the pole placement of the system that will have adequate stability to bring the system to the equilibrium point. We have previously calculated the A and B matrices already but we do not have the spoles for the system. Picking the spoles is very important as they determine whether your linear regulator will have adequate stability margins. To pick the correct spoles we need to first inspect the plant poles of the linear system, the eigenvalues of the A matrix.

$$PlantPoles = \begin{bmatrix} 4.7749 \\ -4.7749 \\ 0 \\ -25 \end{bmatrix}$$

From inspection the plant poles does not include any complex poles but we have one sufficiently damped pole. Sufficiently damped poles are poles that are smaller than the first Bessel function divided by the settling time. For our pole selection we have assumed a settling time of 1 second. This mean that we have two plant poles that is a sufficiently damped poles, -25 and -4.7749. Since we have one sufficiently damped pole and no complex poles, we will use normalized Bessel poles for the remainder of the spoles, thus our final spoles matrix is:.

$$spoles = \begin{bmatrix} \frac{s_2}{T_s} & -4.7749 & -25 \end{bmatrix}$$

With the spoles, A and B expression we can use the MATLAB Place function to calculate our gain matrix k:

$$k = [-47.1371 - 9.8718 - 0.0488 - 0.0409]$$

The feedback gain will stabilize the system to the equilibrium points. Now we can designed a linear regulator that will bring the pendulum to the inverted vertical position.

5 Region of Attraction

Recall that the system is using a non-linear control to swing up and then a linear regulator is used to stablize the pendulum at the inverted vertical position. A region of attraction model is used to switch from the non-linear controller to the linear regulator. The region of attraction is the region in which the linear regulator will be able t drive the system to the equilibrium point. The region of attraction is defined as the largest invariant set M in D, such that $M = \left\{x : X^{'}PX \leqq L^{*}\right\}$. Where D is the domain in which the lyapanov function choosen $V(x)$ is positive definite and $\dot{V}(x)$ is negative definite for all values of X in a given radius R. Note that this $\dot{V}(x)$ is not the same as the one specified in section 3.1. This $\dot{V}(x) = 2\dot{x}'P[x_1(x); x_2(x)]$. Using this method it was found that the region of attraction of the fourth order model is too small for practical use, thus a second order estimation of the region of attraction is used instead.

The process for finding the region of attraction begins with linearzing the second order plant and finding its feedback gain. This is the same steps as shown in Section 4. The only difference here is that we choose our spoles to be $s2/T_s$. Using the resulting feedback gain we solve for a general form of the lyapunov equation, P, using the MATLAB function "lyap". With the lyapunov equation and the feedback gain we find the largest value for R such that $\dot{V}(x)$is negative definte inside a sphere of radius R. This means that the equilibrium point at the inverted vertical position is locally asymptotically stable. It is local because the system will converge only when the values for our state variables, $x_1$and x_2 are within the radius R. Using the "find_r.m" shown in Fig.(4) the radius R was found to be 0.8506. With this we can now define our region of attraction as when $X^{'}PX \leqq L^{*}$, where $L^{*} = minimum\ Eigenvalue(P) \cdot R^2$.

```
function r=find_r(P,K)
%P is matrix defining Lyapunov function V=x^TPx
%K is linear state feedback gain matrix
%r is the radius of the largest circle in which dot{V} is neg. def.
%see the function mVdot below
Rmax=10;
Rmin=.1;
r1=1;
while mVdot(r1,P,K)>0 & r1>Rmin
    r1=r1/2;
end
r2=r1;
while mVdot(r2,P,K)<0 & r2<Rmax
    r2=2*r2;
end
while r2-r1>.001
    rr=(r1+r2)/2;
    if mVdot(rr,P,K)>0
        r2=rr;
    else
        r1=rr;
    end
end
r=r1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function x=mVdot(r,P,K)
% Computes the largest value of dot{V} on a circle of radius r
%P is matrix defining Lyapunov function V=x^TPx
%K is linear state feedback gain matrix
theta=linspace(0,2*pi);
x1=r*cos(theta);
x2=r*sin(theta);
alpha = 0.018;
x=-1e16;
for k=1:length(theta)
    u=-K*[x1(k);x2(k)];
    xdot=[x2(k);22.8*sin(x1(k))+(22.8/9.8)*cos(x1(k))*u-alpha*(x2(k)^2)*sign(x2(k))];
    Vdot=2*xdot'*P*[x1(k);x2(k)];
    if Vdot>x;x=Vdot;end
end
```

Fig. 4: Finding the maximum Radius R

6 Simulink Model

The complete model will consist of the non-linear energy controller that will swing up the pendulum to the inverted vertical position and the linear state-feedback regular that will stabilize the pendulum in that position. The model will consist of two main stages, the first stage will be the swing up stage. At this stage we will be using the energy equations in conjunction with the motor controller, the non-linear plant will swing the pendulum up to the inverted vertical position. The second stage will be the regulating the pendulum at the vertical upright position, meaning that we will use our linear regulator that we have designed to keep the pendulum stable in the vertical inverted position. Since there are two stage, a switching logic is required to transition the controller from the non-linear plant to the state-feedback regulator when the pendulum is near the inverted position. While the derivation of the linear regulator assume that the equilibrium point will be zero, meaning it will be at the inverted position, this is only the case when the pendulum swing from one direction. If the pendulum swings from the opposite direction we need the system to still stabilize, meaning the linear regulator needs to stabilize the system at both $x_1 = 0$ and $x_1 = 2\pi$. An angle compensation logic can be designed to account for the changes in the value of the angle as the pendulum swing up from either direction. Taking into consideration these additions the complete model of the control system is designed as shown in Fig. (5). When running the full simulink model a MATLAB script file shown in Fig. (6) need to be run prior to initiaze all the constant values for the model, motor control constants, linear regulator gain constants and the region of attraction calculations. Each of the sub-system in the complete model will be explained in detail below.

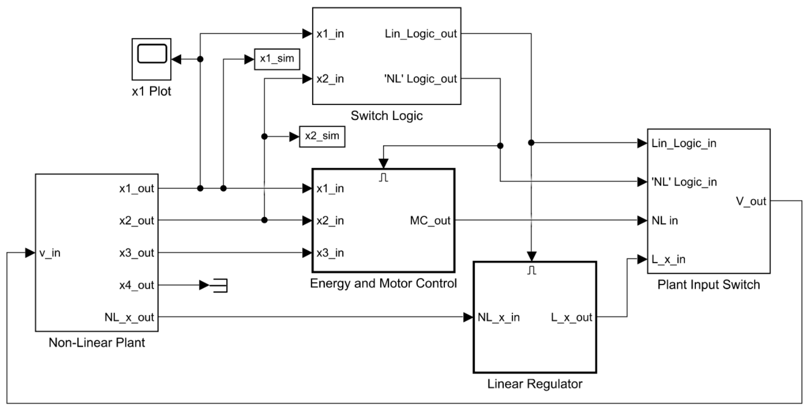

Fig. 5: Full Simulink Model

```
load sroots; %Load the roots to be used in spole calculations
% Initializing the constants that will be needed throughout the simulink
% model. The consants that are "hardware constants" are depend on the
% hardware the is being used for the physical system. These values should
% be double checked prior to running the program when integrated with the
% hardware system. The values used in this simulation is based on
% available resources from University of Rhode Island.
g=9.81; %Acceleration due gravity constant
A = 22.8; %Square of natural frequency of pendulum,will change for
          %different rod. (hardware constants)
n=495; % Transduction ratio to the lead screw (hardware constants)
C=25; % Motor Control Constants (hardware constants)
Cprime=28; % Motor Control Constants (hardware constants)
D=2350; % Motor Control Constants (hardware constants)
alpha=0.018; %Co-efficient of friction (User input)original=0.04; This
             %value must be estimated for each different rods used.

%The constants below are used in the "Energy and Motor Control" simulink
%model.
KM=((Cprime/2)^2)/D; %Motor Control Constants
C1= (2/Cprime)^2; % Motor Control Constants
C2= 4/Cprime; % Motor Control Constants
Ts=1; %Settling Time for Linear Regulator

% Calculations for determing the gain value (k) to be used for the linear
% regulator that will drive the pendulum system to the inverted position.
Amat = [0 1 0 0; A 0 0 ((A*C)/(n*g));0 0 0 1; 0 0 0 -C];
Bmat = [ 0; -((A*D)/(n*g)); 0 ;D];
plantpoles = eig(Amat);
%spoles = [s3/Ts plantpoles(4)];
spoles = [s2/Ts plantpoles(2) plantpoles(4)]; %Calculated based on the
                                              %plantpoles of the system.
k=place(Amat,Bmat,spoles); %The linear regulator gain value

% Calulations for the region of attraction model. This will determine
% when the system switch from the non-linear controller (initial
% swing-up) to the linear regulator to drive the system to the inverted
% position. Recall that our system is a fourth order system, but upon
% investigation it was found that the fourth order system produce a
% region of attration that is too small, and thus not useful. To work
% around this issue we take the second order version of our system and
% calculate its region of attraction
a_2 = [0 1; A 0];
b_2 = [0;A/g];
Ts_2 =1; %Settling Time for Linear Regulator
spoles_2 = s2/Ts_2;
k_2 = place(a_2,b_2,spoles_2);
P = lyap((a_2-b_2*k_2)',eye(2));
r = find_r(P,k_2); %Finding the largest value of r for for which Vdot is
                   %negative definite in a circle of radius r.
cc= r^2;
ce = (min(eig(P)))*cc; %This is the minimum eigenvalues of the P matrix
                       %multiplied by the square of the radius r.

%Plots for the r, where v_dot is negative definite and the regin of
%attraction is estimated to be the ellipse. The position and velocity of
%the simulink runs are plotted in overlay to show how the system respond
%as it enter the region of attraction.
plot(x1_sim,x2_sim);
hold on
plot_ellipse(eye(2),cc);
hold on
plot_ellipse(P,ce)
```

Fig. 6: Full Model Simulink Script

6.1 Energy and Motor Control

The energy and motor control sub-system, Fig.(7) is using the energy method to calculate the pivot acceleration for the pendulum-cart system. Using this acceleration the motor control system is translating it to the input voltage we need to drive the system. The architecture of this model follows closely to that of the phase-lead compensator with pre-filter shown in "A Control System Design". The "Energy and Motor Control" sub-system is responsible for swinging the pendulum up from its initial rest position, so when the simulation start this sub-system will swing the pendulum until the pendulum reaches very close to the inverted vertical position, after which this sub-system will stop running. To ensure the sub-system stops running we have an "Enable" block titled "Logic_in" which will execute the sub-system when the input for the "Enable" block is anything other than a zero value.

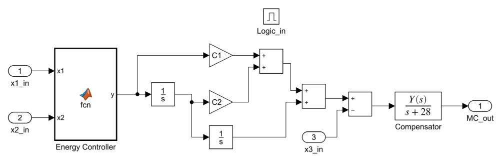

Fig. 7: Energy and Motor Control Simulink Sub-system

In the energy controller we are using our x_1 and x_2 inputs to calculate the total energy of the system, shown in Eq. (5) and our pivot acceleration shown in Eq. (9). In addition to the inputs the user must also define two key variables to ensure our controller works correctly. The first variable is the gain, k. This is not the same gain from the state feedback linear regulator. This gain value is used to ensure our energy controller is increasing the swing to approach our desired position. Via experimentation this value is chosen as $k = 0.006$. The second variable is the co-efficient of friction for the system. This is a value that is chosen to compensate for the frictional losses to the system. By default this value is chosen as $\alpha = 0.004$. When calculating the pivot acceleration we need to consider that as the angle, x_1, changes the value of $\cos(x_1)$ will approach to zero, and when $x_1 = \pi/2$, our control law Eq. (9) will go to infinity. To account for this we must "by-pass" this region by holding the value of $\cos(x_1)$to a specific value. This is shown in the MATLAB script for the Energy Controller block is shown in Fig. (8).

```
function y  = fcn(x1, x2)
A = 22.8;%Square of natural frequency of pendulum
g=9.81;%Gravity
n=495;% Transduction ratio to the lead screw
k = 0.02; %Gain for Energy System (User input) orignal = 0.016
alpha=0.018; %Co-efficient of friction (User input)original=0.04
E = 0.5*(x2^2)+A*(cos(x1)-1); %Enery of the pendulum system
if abs(cos(x1)) < 0.08 % This will account for when cos(X1)is close to 0
    w = 0.08*sign(cos(x1));
else
    w=cos(x1);
end
u1 = (1)*(k*E*x2*cos(x1))-((g/A)*alpha*(x2^2)*(sign(x2))/w); %Pivot acc.
u=u1*n; % We need to convert account for our transducer ratio
vdot = E*((alpha*(x2^2)*sign(x2))-((A/g)*x2*cos(x1)*u1));
y = u;
```

Fig. 8: Energy Controller MATLAB Script

The pivot acceleration output from the "Energy Controller" must be converted to our input signal, v. Recall that the pivot acceleration must account for the transduction ratio. From the transfer function shown in Eq. (13) and the values for C and C' we can design our pre-filer. The cart acceleration is passed through two integrator block and place the C_1 and C_2 constants between the appropriate integrator blocks to satisfy Eq. (13). This output is summed and passed through to the compensator transfer function.

6.2 Non-linear Plant & Linear Regulator

The non-linear plant is using our input v from the motor control and calculating our state variables, x_1, x_2, x_3 and x_4, Fig. (9).

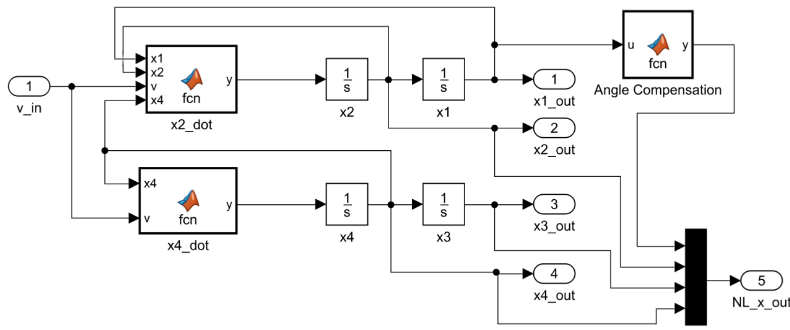

Fig. 9: Non-Linear Plant Simulink Sub-System

Using the fourth-order non-linear plant, Eq. (11), $\dot{x}_2$ and $\dot{x}_4$ are calculated using MATLAB script files. The output of the "x2_dot" script is passed through two integrator blocks to get the values for x_1 and x_2, recall that $\dot{x}_1 = x_2$. Similarly the output of the "x3_dot" script is passed through two integrator blocks to get the values for x_3 and x_4, recall that $\dot{x}_3 = x_4$, as shown Fig. (9). The non-linear plant is where we will initialize the system. The system is starting with pendulum hanging freely on the bottom vertical position, we set the initial condition for the $x_1$integrator to π. To start swinging our pendulum we need to give our system an initial velocity by setting the initial condition for the x_2 integrator to 0.17. This mimic lightly flicking the pendulum to jump start our control system and give it an initial velocity.

```
function y = fcn(x1, x2, v, x4)
A=41.5;
g=9.8;
D=2633;
n=495;
c=25;
alpha=0.04;
u= (A*sin(x1))-(((A/g)*cos(x1))*((-c*x4)+(D*v))/n)+(alpha*(x2^2)*sign(x2));
y = u;
```

```
function y = fcn(x4, v)
D=2350;
c=25;
u= (D*v)-(c*x4);
y = u;
```

Fig. 10: MATLAB Script for: "x2_dot" (left) & "x4_dot" (right)

Recall from the full simulink model, Fig.(5), that the "Non-Linear Plant" sub-system has an output "NL_x_out" that is feeding into the "Linear Regulator" sub-system "NL_x_in" input. This output signal is all four of our state variables , x_1, x_2, x_3 and x_4. But the value for $x_1$needs to account for the angle as we are looking to stabilize the system at the inverted vertical position, regardless of which direction the pendulum swings to get there. To address this, the MATLAB script, "Angle Compensation", takes the x_1 output form the system and substract it by 2π when x_1 is greater than π. The absolute value of this is then supplied to the multiplex vector signal junction ,"Mux", bus as a new value for x_1. Fig. (11) shows the MATLAB script used to compensate for the $x_1$value.

```
function y = fcn(u)
if u > pi
    y = (u - 2*pi);
else
y = u;
end
```

Fig. 11: Angle Compensation MATLAB Script

The state feedback linear regulator will take the input from the non-linear plant and utilize the feedback gain calculated in Eq. (18) to drive the system to stabilize in the inverted pendulum position. To model this in simulink utilize a gain block and assign the calculated k-value and multiply it by the input variable, see Fig. (12).

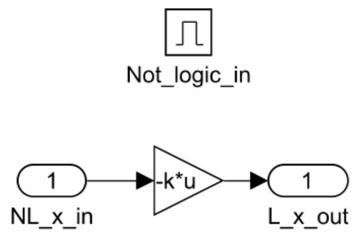

Fig. 12: Linear Regulator Simulink Sub-System

6.3 Switch

With the "Non-Linear Plant" coupled with the "Energy and Motor Control" we can swing our pendulum up and with our "Linear Regulator" we can catch our pendulum mid-swing and stabilize it in the inverted vertical position. To "catch" or more accurate switch between the energy control method and the linear regulator a "Switch Logic" sub-system is needed to properly turn on/off the appropriate sub-system depending on the position of the pendulum.

6.3.1 Switch Logic

The switch logic sub-system, Fig. (13), takes the input $x_1$and x_2 value from the "Non-Linear Plant" sub-system and output a Boolean signals, depending on the region of attraction calculation. The values "ce" and "P" are also imported from the workspace from Fig. (6).

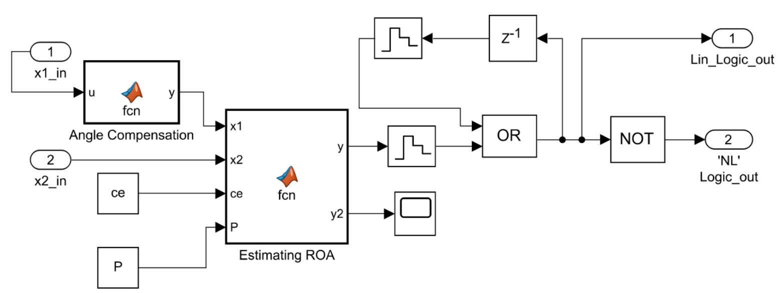

Fig. 13: Switch Logic Simulink

Similar to the nonlinear plant the $x_1$value requies an angle compensator to ensure that we are keeping track of the pendulum position regardless of the direction of swing. The two state variables $x_1$and x_2 are used to determine when the system should switch from the nonlinear controller to the linear regulator. This calculation is performed via the Matlab script "Estimating ROA" as shown in Fig.(14).

```
function [y,y2] = fcn(x1,x2,ce,P)
y2 = [x1; x2]'*P*[x1; x2];
if y2 < ce
   y=1;
else
    y = 0;
end
```

Fig. 14: Estimating ROA Script

This script will calculate the region of attraction and output a Boolean logic based on whether the system is within the region . From inspection we can see that when the system initially begin it will not be in the region of attractoin, thus the script will output a False. Since we want the nonlinear controller to be active in the beginning we want to negate the output. We also employ a rebound logic to ensure that once the system in within the region of attraction, i.e. the script output is True, we want the system to permanently switch to the linear regulator. To rebound logic works by sampling one sample prior to the current state of the simulation and then taking that signal and the output of the script through an OR gate. This means that once the script outputs a True value the resulting output henceforth will always be true.

6.3.2 Plant Input Switch

The last sub-system, "Plant Input Switch", also takes the output values from the "Switch Logic" sub-system and determines which input value will be routed back to the "Non-Linear Plant" subsystem. There are two input variables that are created, the first is from the "Energy and Motor Control" subsystem where we are using the energy controller to derive the state input, v, we need for the system. The second is the input that is created from the "Linear Regulator" sub-system. To ensure the correct input is being feed back into the "Non-Linear Plant" subsystem we take the Boolean logic outputs form the "Stitch Logic" subsystem and multiply by the state input variables that are created by the "Energy and Motor Control" and the "Linear Regulator" sub-system, see Fig. (15).

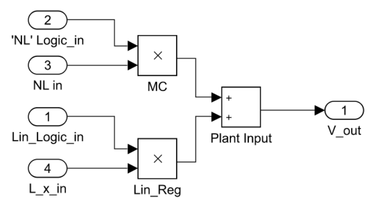

Fig. 15: Plant Input Switch Simulink Sub-System

7 Results & Conclusion

This report added a modification to the original energy method by considering friction losses. With our modified simulink model we can plot the angle of the pendulum, x_1, value as a function of time. Fig (16) shows that the pendulum start at $x_1 = \pi$ and swing up to the inverted vertical position and finally stabilizes at $x_1 = 0$. This shows that energy controller can be used to successfully swing up a pendulum to the inverted position. It is important to note that the model in this report are experimental and have not been proved on physical hardware. Further improvements can be made to the model by accounting for the length of the track, the current model does not account for the finite length of the track the cart is on.

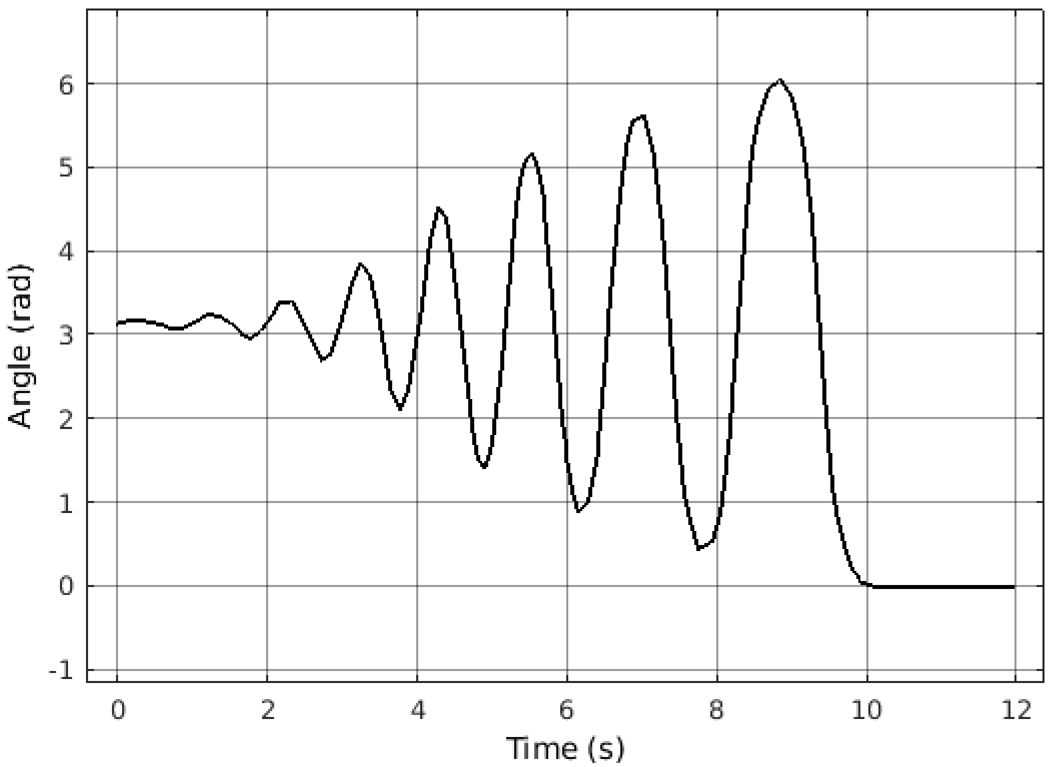

Fig. 16: Pendulum Swing (Angle) over Time

With the region of attraction model we expect that the system will drive the system to the equilibrium point when the state variables, $x_1$and x_2 are within the region of attraction. to show this we plot the region of attraction and overlay it with the two state variables. Fig. (7) showns below start with varing initial velocity, which emulate a light flick as $x_2 = 0.2$ and moderate flick as$x_2 = 2.5$ and a strong flick as $x_2 = 9.5$. The plots shows that with each different initial velocity the pendulum will swing until it is within the region of attraction and then the linear regulator takes over and drive the system to the local equilibrium point.

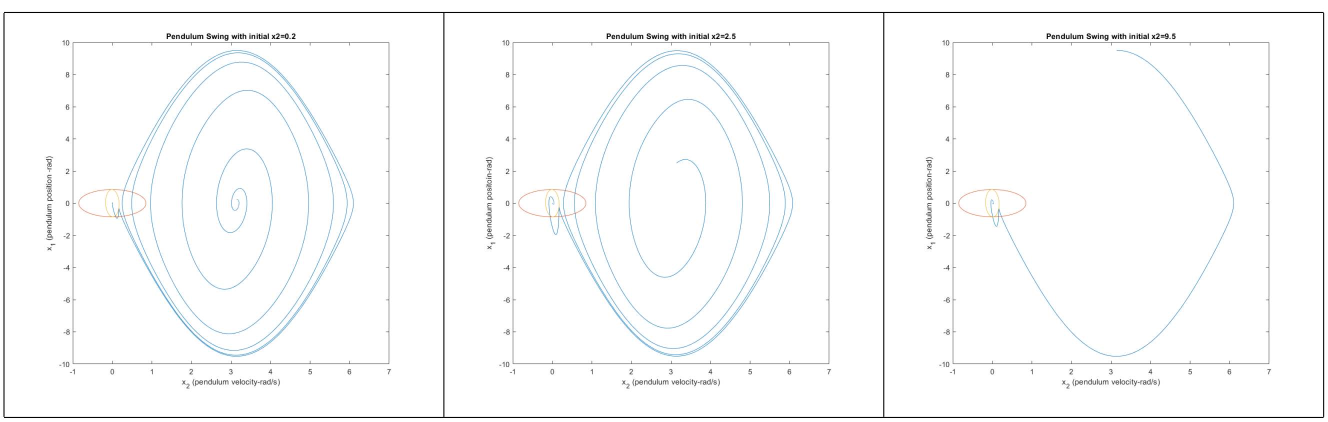

Fig. 17: Pendulum with varying initial velocity

References

[1.] Swinging up a pendulum by energy control" by K.J. Astrom, K.Furuta, Automatica 36 (2000) 287-295, dated May 1999.

[2.] A Control System Design A State-Space Approach by Richard Vaccaro, University of Rhode Island, dated September 2016.