Robust Control Algorithm for a Two Cart System and an Inverted Pendulum

Chris L. Wilson[1]
Mississippi State University, Mississippi State, MS, 39762

Pedro Capo-Lugo[2]
NASA Marshall Space Flight Center, Huntsville, AL, 35812

The Rectilinear Control System can be used to simulate a launch vehicle during liftoff. Several control schemes have been developed that can control different dynamic models of the rectilinear plant. A robust control algorithm was developed that can control a pendulum to maintain an inverted position. A fluid slosh tank will be attached to the pendulum in order to test robustness in the presence of unknown slosh characteristics. The rectilinear plant consists of a DC motor and three carts mounted in series. Each cart's weight can be adjusted with brass masses and the carts can be coupled with springs. The pendulum is mounted on the first cart and an adjustable air damper can be attached to the third cart if desired. Each cart and the pendulum have a quadrature encoder to determine position. Full state feedback was implemented in order to develop the control algorithm along with a state estimator to determine the velocity states of the system. A MATLAB program was used to convert the state space matrices from continuous time to discrete time. This program also used a desired phase margin and damping ratio to determine the feedback gain matrix that would be used in the LabVIEW program. This experiment will allow engineers to gain a better understanding of liquid propellant slosh dynamics, therefore enabling them to develop more robust control algorithms for launch vehicle systems.

Nomenclature

$\boldsymbol{m_1}$	= mass of cart one
$\boldsymbol{m_2}$	= mass of cart two
$\boldsymbol{m_p}$	= mass of pendulum
k	= spring constant
$\mathbf{c_1}$	= friction acting on cart one
$\mathbf{c_2}$	= friction acting on cart two
$\boldsymbol{F_f}$	= force of friction
L	= length of the pendulum arm
$\boldsymbol{g}$	= gravitational constant
$\boldsymbol{x}$	= state vector
$\vec{\boldsymbol{x}}$	= state estimate vector
$\boldsymbol{y}$	= output vector
$\vec{\boldsymbol{y}}$	= output estimate vector

I. Introduction

The Rectilinear Control System can be used to simulate a launch vehicle during liftoff by the use of the two cart system and the inverted pendulum. A control algorithm for the two cart system was developed in order to simulate the coupling of a multi-stage rocket. This algorithm controls the position of the first cart with the added disturbance of the second cart coupled with a spring.

[1] MSGR Intern, Control Systems Design and Analysis, NASA Marshal Space Flight Center.
[2] Aerospace Engineer, Control Systems Design and Analysis, 4600/EV41, NASA Marshall Space Flight Center.

Once a control algorithm is developed that will maintain the pendulum in an inverted position, a liquid slosh tank will be attached to the pendulum to test for robustness in the presence of unknown slosh characteristics. This experiment will provide engineers a cost effective simulation of a rocket, therefore enabling them to develop more robust control algorithms for launch vehicle systems.

II. Overview of the Model 210a Rectilinear Control System

The Model 210a Rectilinear Apparatus is designed to provide insight to control system principles through hands-on demonstration and experimentation [1]. The rectilinear plant consists of a DC motor and three carts mounted in series. The cart weight can be adjusted with brass masses and the carts can be coupled with different sized springs. The A-51 Pendulum Accessory is mounted on the first cart for the lower pendulum and the inverted pendulum systems. The pendulum length and location of the point mass can be adjusted. An optional disturbance motor can be attached to any cart and an adjustable damper can be attached to the third cart if desired. Each cart and the pendulum have a high resolution quadrature encoder to determine position. A fluid slosh tank has been constructed that can be attached to the pendulum in order to test for robustness in the presence of slosh dynamics. This attachment will enable engineers to simulate fuel slosh present in liquid propellant launch vehicles.

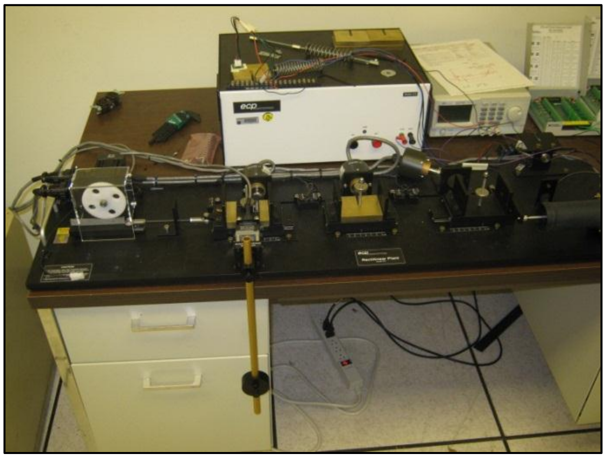


Figure 1. Rectilinear Control System

Originally, the Rectilinear Control System contained a data acquisition board which executed the control algorithm and collected data from the experiment. However, this board is no longer functioning and has been replaced with a LabVIEW user interface and Peripheral Component Interconnect (PCI) cards. LabVIEW is an icon based programming language which enables the user to make programming structures rather than lines of code. The PCI cards are located within the computer and allow the computer to access the data output from the system.

III. Procedure

In order to properly program the system dynamics into the algorithm, the equations of motion must be developed by using Newton’s Second Law. These equations are then converted to state space equations in order to input into the MATLAB program. This MATLAB program converts these state space equations from the continuous time domain to the discrete time domain. Using the pole placement method, this MATLAB program places the poles of the system at a desired location based on the phase margin and the damping ratio. Once the poles have been positioned, Ackermann’s formula is used to determine the feedback gains of the system at the specified pole locations. These feedback gains as well as the discrete state space equations are then programmed into the LabVIEW algorithm and tested.

A. Dynamic Systems

In order to get a better understanding of the Rectilinear Control System as well as the basics of controls theory, an algorithm was developed to control a one cart system. This program initiates a step function to make the cart move to a specified position or a sine function so the cart can oscillate at a certain amplitude and frequency.

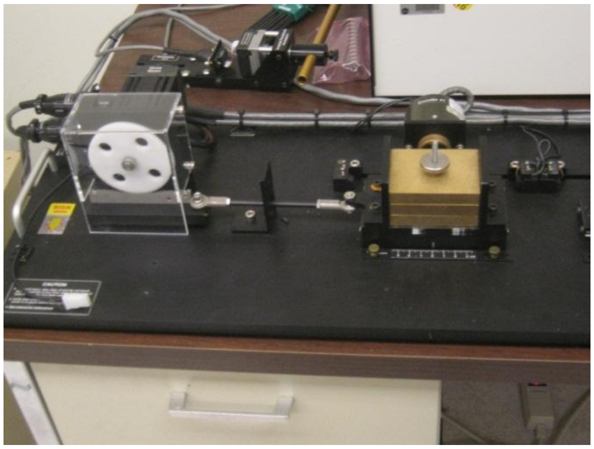


Figure 2. One Cart System

Once this program was functional, an algorithm to control a two cart system was created. This algorithm controls the position of cart 1 with the added disturbance of a second cart attached by a spring. Similar to the one cart system, this program can also implement a step function or a sine function.

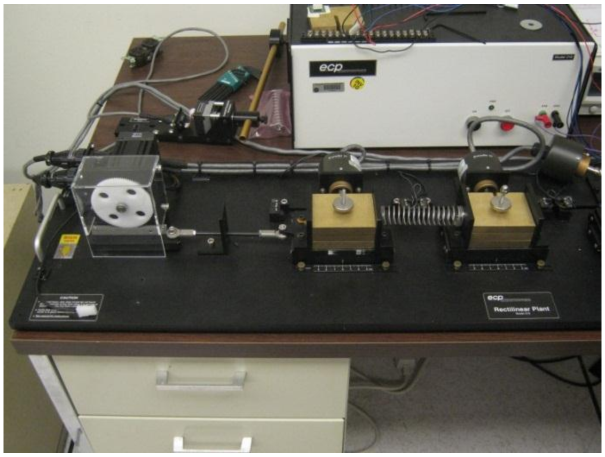


Figure 3. Two Cart System

The next control scheme that was developed was the lower pendulum system, which consists of the A-51 Pendulum Accessory attached to the first cart in a hanging position. This method can be seen in Figure 1. This algorithm is used to dampen out the swinging of the pendulum due to a minor disturbance. For instance, once the user gives the pendulum a tap, the cart will move in order to stop the swinging of the pendulum. The final control scheme that was used was the inverted pendulum system. This system uses the same algorithm as the lower pendulum with one sign change in order to change the direction the cart moves. This algorithm will maintain the pendulum in an inverted position by moving the cart and will correct for minor disturbances.

B. Development of Dynamic Equations

It was determined that the B state space matrix (force matrix) had to be multiplied by the hardware gain in order to convert the output from volts to newtons. The hardware gain was experimentally calculated to be $469\frac{N}{V}$. However, after the LabVIEW algorithm was tested with the feedback gains, it was discovered that the actual hardware gain was $\frac{469}{.005}\frac{N}{V}$. This difference might be due to the conversion of the state space matrices from continuous time to discrete time. Without dividing the hardware gain by .005, the feedback gains are too high to control the system.

The equations of motion for each system were obtained by using Newton's Second Law ($F = ma$) and summing the forces of the system in the horizontal direction. Because the plant has multiple degrees of freedom, Lagrange's equations were used to derive the state space representation of each dynamic system. If Q_i is called the *generalized force* in the direction of the i^{th} generalized coordinate, T is the kinetic energy and V is the potential energy, then Lagrange's equation is given by

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i$$

Expressing the Lagrangian, $L = T - V$, the equation can be written as:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i$$

The left side of this equation sums all energy that is conserved in the dynamic system and the right side represents the work done by external forces on the system. Solving these Lagrange equations will give the dynamic equations the form $\boldsymbol{M}\dot{x} = \boldsymbol{K}x + \boldsymbol{G}u$, where **M** is the mass matrix, **K** is the dissipation matrix, and **G** is the shaping matrix. Rearranging this equation to the form $\dot{x} = \boldsymbol{M}^{-1}\boldsymbol{K}x + \boldsymbol{M}^{-1}\boldsymbol{G}u$, where $\boldsymbol{A} = \boldsymbol{M}^{-1}\boldsymbol{K}$ and $\boldsymbol{B} = \boldsymbol{M}^{-1}\boldsymbol{G}$.

1. *One Cart System*

Equation of Motion:

$$m\ddot{x} + c\dot{x} = F(t)$$

State Space Equations:

$$\dot{x} = Ax + Bu \qquad x = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

$$y = Cx$$

$$u = -kx$$

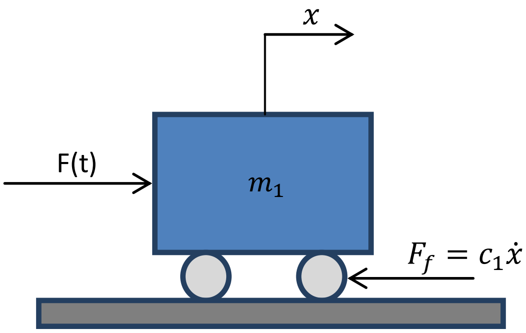


Figure 4. One Cart Block Diagram

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -c_1/m_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1/m_1 \end{bmatrix} u$$

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

2. *Two Cart System*

Equations of Motion:

$$m_1\ddot{x}_1 + c_1\dot{x}_1 + kx_1 - kx_2 = F(t)$$
$$m_2\ddot{x}_2 + c_2\dot{x}_2 + kx_2 - kx_1 = 0$$

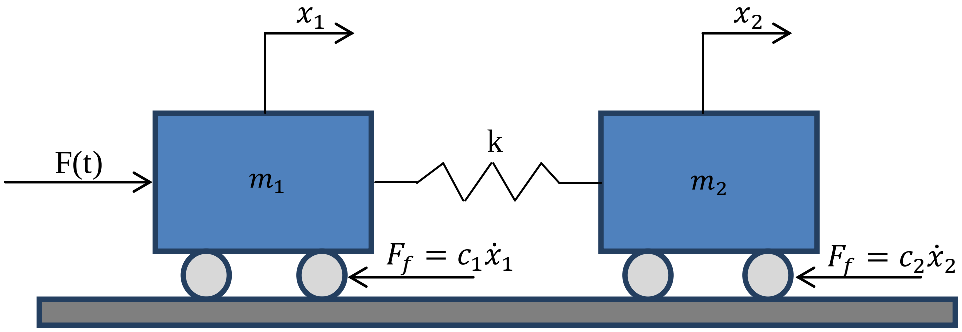


Figure 5. Two Cart Block Diagram

State Space Equations:

$$\dot{x} = Ax + Bu \quad x = \begin{bmatrix} x_1 & x_2 & \dot{x}_1 & \dot{x}_2 \end{bmatrix}^T$$

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\dfrac{k}{m_1} & \dfrac{k}{m_1} & -\dfrac{c_1}{m_1} & 0 \\ \dfrac{k}{m_2} & -\dfrac{k}{m_2} & 0 & -\dfrac{c_2}{m_2} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{bmatrix} u$$

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

3. *Lower Pendulum System*

Equations of Motion:

$$(m_1 + m_p)\ddot{x} - m_pL\ddot{\theta} = -F(t)$$
$$m_pL\ddot{\theta} - m_pL\ddot{x} - m_pgL\theta = 0$$

State Space Equations:

$$M\dot{x} = Ky + Gu \quad x = \begin{bmatrix} x & \theta & \dot{x} & \dot{\theta} \end{bmatrix}^T$$

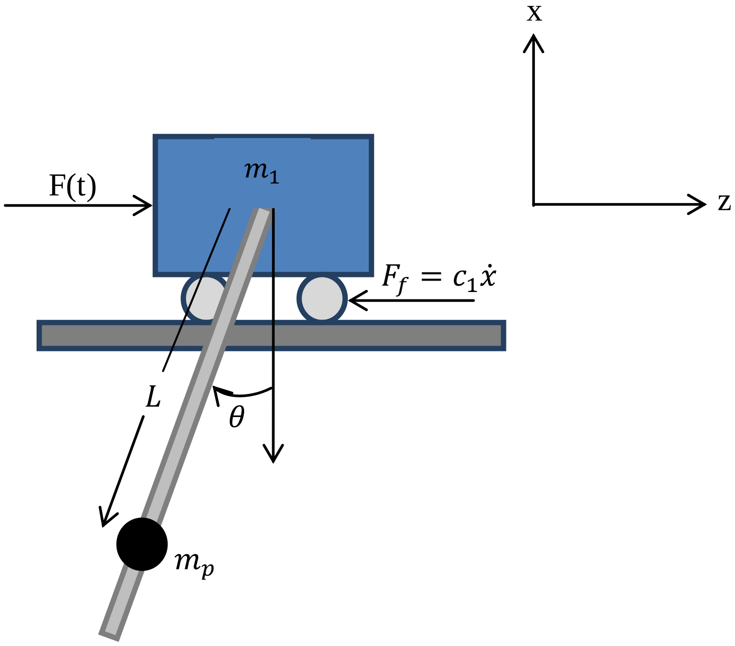


Figure 5. Lower Pendulum Block Diagram

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & m_1 + m_p & -m_p L \\ 0 & 0 & -m_p L & m_p L \end{bmatrix} \dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -c_1 \\ 0 & m_p g L & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} u$$

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

4. *Inverted Pendulum*

Equations of Motion:

$$(m_1 + m_p)\ddot{x} - m_p L\ddot{\theta} = F(t)$$
$$m_p L\ddot{\theta} - m_p L\ddot{x} + m_p g L\theta = 0$$

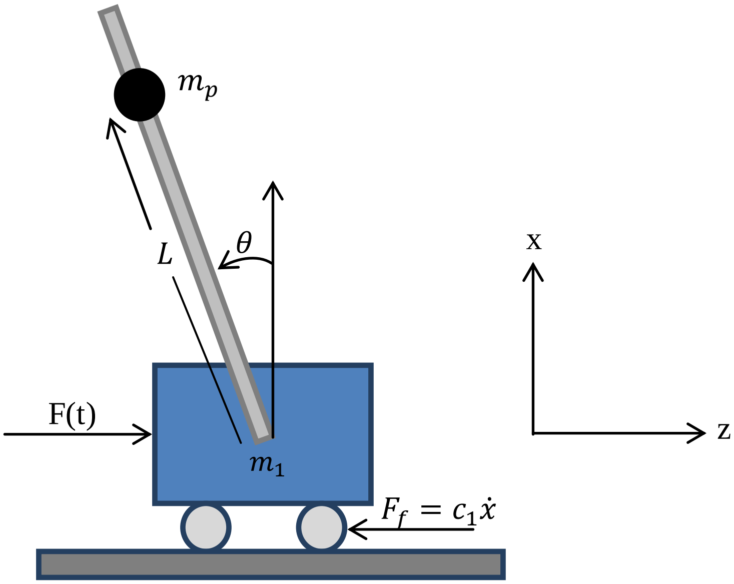


Figure 6. Inverted Pendulum Block Diagram

State Space Equations:

$$M\dot{x} = Ky + Gu \quad x = \begin{bmatrix} x & \theta & \dot{x} & \dot{\theta} \end{bmatrix}^T$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & m_1 + m_p & -m_p L \\ 0 & 0 & -m_p L & m_p L \end{bmatrix} \dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -c_1 \\ 0 & m_p g L & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} u$$

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

IV. Data and Analysis

A. Control System Design

A closed loop control system was used in the development of the LabVIEW algorithm. A closed loop control system uses a measurement from the sensor and compares it to the desired output in order to correct for error[2]. Figure eight displays a block diagram of a closed loop control system. Because the two cart system and the pendulum systems have multiple degrees of freedom, it was decided that full state feedback would be used to control the system. Full state feedback places the closed-loop poles of the system at a desired location. Ackermann's formula is then utilized to determine the feedback gain values. The command in MATLAB for Ackermann's formula is $K = acker(A, B, p)$. Where A and B are the state space matrices and p is the location of the poles. Figure seven shows the K matrix values found in each dynamic system.

Calculated Feedback Gains for Each System	
One Cart System	$\left[44.398 \quad 0.5228\right]$
Two Cart System (200 N/m spring)	$\left[1.1357 \quad 21.0286 \quad 0.0076 \quad 0.764\right]$
Two Cart System (390 N/m spring)	$\left[1.3475 \quad 24.1461 \quad 0.0077 \quad 0.6118\right]$
Two Cart System (830 N/m spring)	$\left[1.1485 \quad 4.8572 \quad 0.0076 \quad 0.1974\right]$
Lower Pendulum System	$\left[128.19 \quad -140.725 \quad 20.982 \quad -21.3168\right]$
Inverted Pendulum System	$\left[44.28514 \quad -44.1144 \quad -0.13861 \quad 0.138383\right]$

Figure 7. Feedback Gains Using Ackermann's Formula

In order to use full state feedback, all states of the system must be known. However, the encoders on the Rectilinear Control System only have the capability to measure position. Therefore, a method called state estimation must be used in order to determine the velocity states of the system. State estimation is determined using the following equation: $\dot{\vec{x}} = A\vec{x} + Bu + L(y - \vec{y}) = (A - LC - BK)\vec{x} + Ly$. Where A is the system dynamics matrix, B is the force matrix , K is the feedback gain matrix, and L is the observer gain matrix.

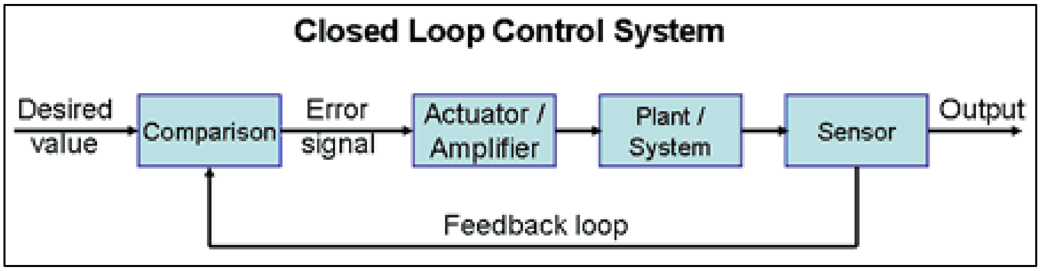


Figure 8. Diagram of closed loop control system

B. Data Analysis

Figure nine shows the step function input of the one cart system. The first graph is the cart's position, the second graph is the cart's velocity, and the last graph is the voltage input to the motor. In this experiment, the cart was programed to move one centimeter. As you can see the cart overshoots one centimeter and the closed loop control system corrects the error and moves the cart back to the correct position. Also, the cart had a relatively quick rising time and its steady state error was approximately zero. The amount of overshoot can be controlled by selecting a lower set of feedback gains. This requires lowering the phase margin and the damping ratio within the MATLAB program.

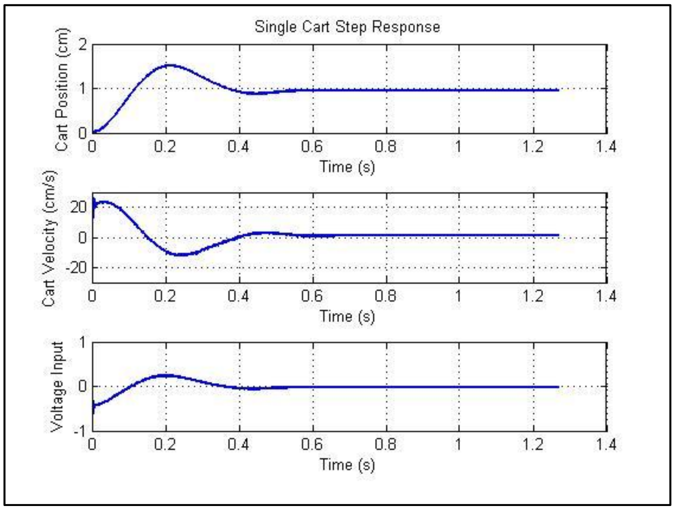


Figure 9. Single Cart Step Response

Similarly, figure ten shows the one centimeter step response of the two cart system using the three different sized springs: 200N/m, 390 N/m, and 830 N/m. The first graph is the position of cart one, the second graph is the position of cart two, and the final graph is the voltage input to the motor. This response shows how the added disturbance of the second cart attached with a spring disrupts the system. The two cart system takes almost seven milliseconds to go into steady state, which is two milliseconds longer than the singe cart system. By taking a look at the individual spring constants and how the system responds, we can learn a lot about control systems and how to develop more robust control algorithms. For instance, as the spring constant increases, the cart becomes easier to control and the overshoot of the cart decreases. Also, due to the increased spring constant the rising time of the system greatly increases and the system goes into steady state much quicker.

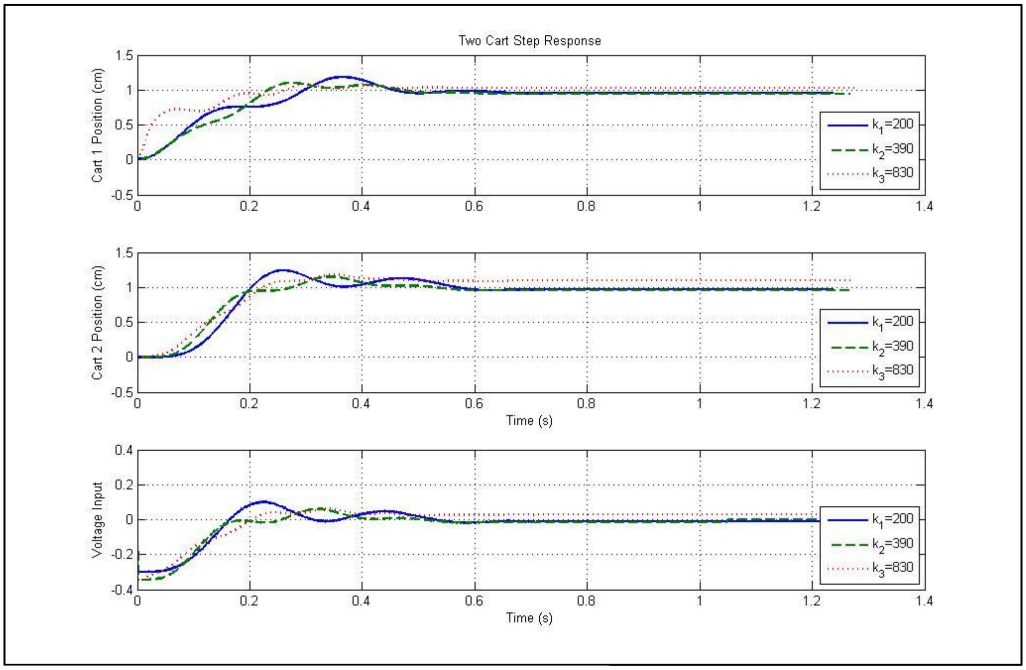


Figure 10. Step Response of the Two Cart System

Figure eleven shows the response of the lower pendulum system. The first graph is the cart position, the second graph is the pendulum angle, and the last graph is the voltage input to the motor. The pendulum was given an initial tap of seven degrees. As a result the cart moved almost two centimeters in order to dampen out the motion. The pendulum swing lost three degrees on just the first swing and two more on the next swing. The gains that were used in this experiment could not correct the pendulum for anything less than two degrees. This system was used to determine how the cart would respond to the unbalance of the pendulum and was used to develop the inverted pendulum algorithm. The only difference between the lower pendulum algorithm and the inverted pendulum algorithm is the cart moves in the opposite direction to balance the system.

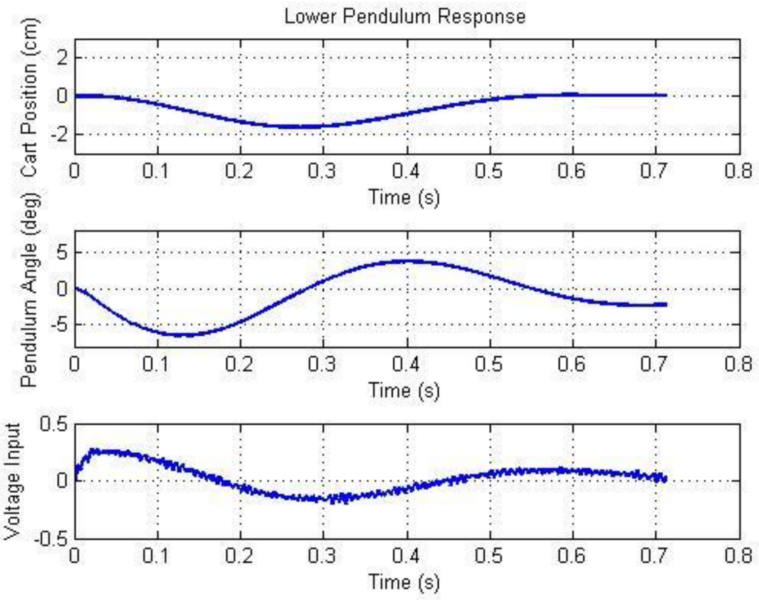


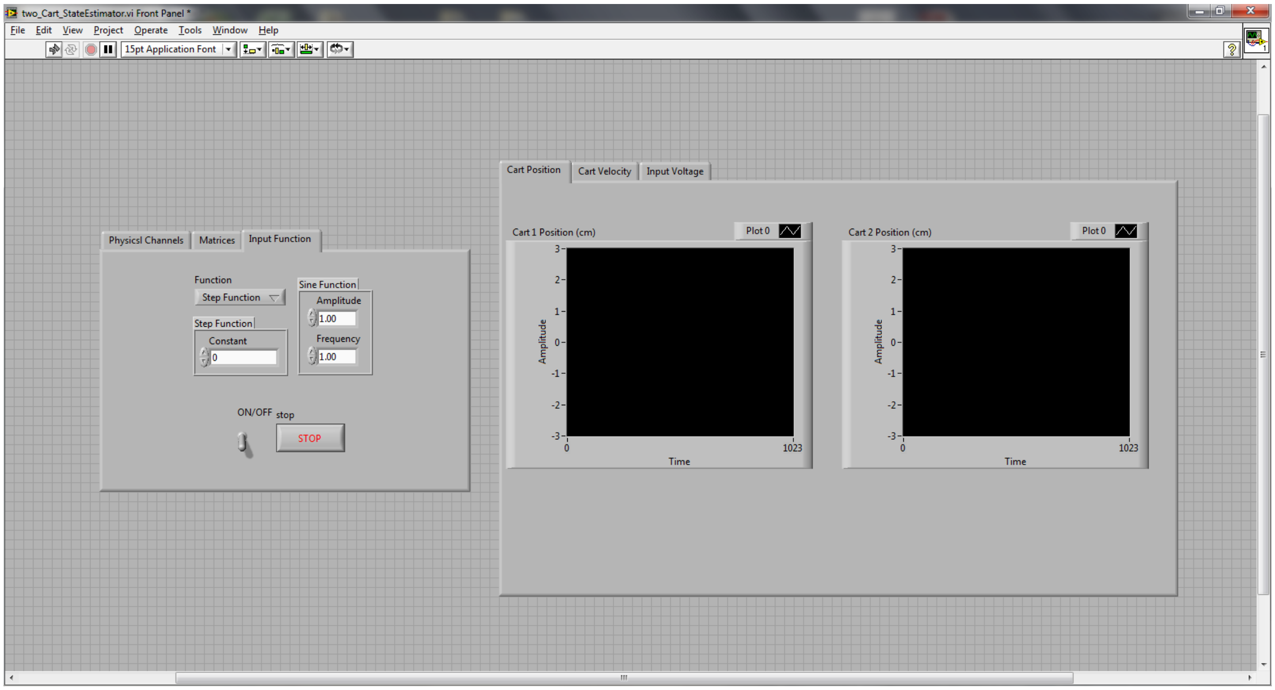
Figure 11. Lower Pendulum Response

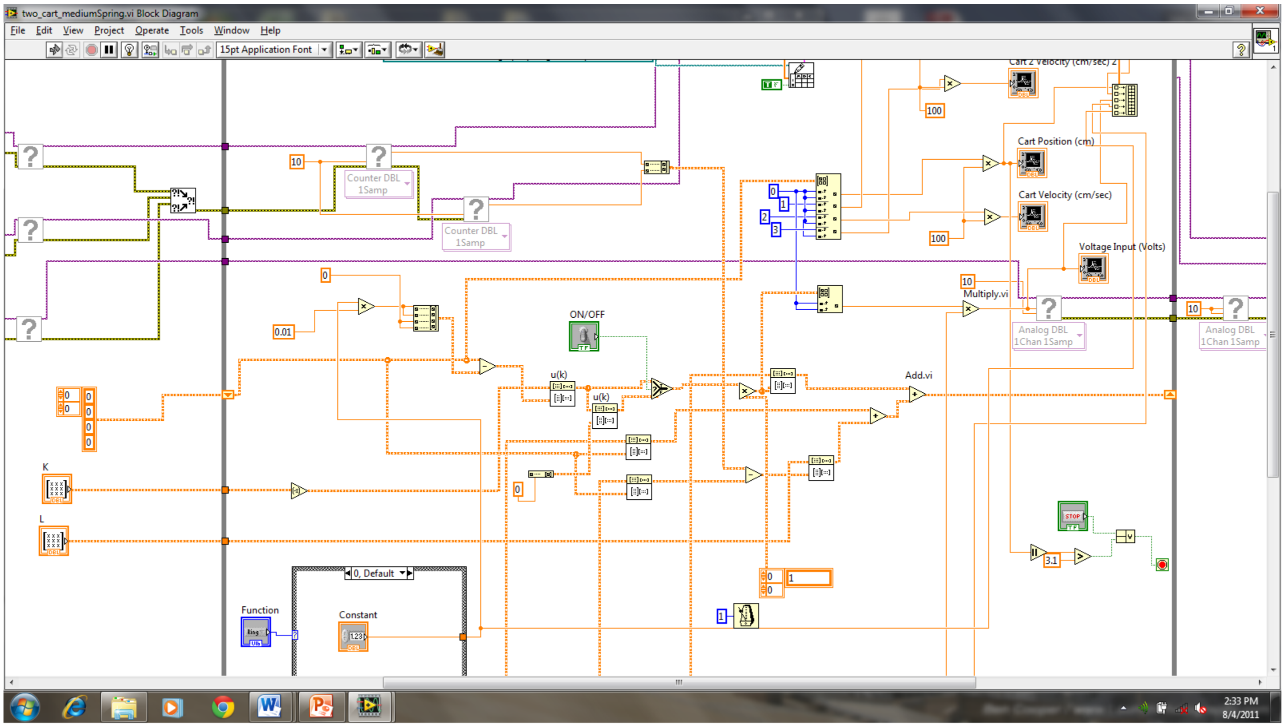
The inverted pendulum algorithm currently only maintains the pendulum in an inverted state for approximately ten seconds. A systems check is currently underway in order to determine the error in the hardware gain. Once the correct hardware gain of the system is determined, better feedback gain values will enable the algorithm to maintain the pendulum in the inverted position indefinitely. Therefore, no data has been reported for this system.

V. Conclusions

In conclusion, by the use of the Rectilinear Control System engineers have access to a low cost demonstration test bed that has the ability to simulate a rocket on liftoff. This will enable them to develop more robust and efficient control algorithms for launch vehicle systems. Also, this test bed will allow engineers to study slosh dynamics that will be experienced in liquid propellant rockets. By observing the two cart system we can conclude that the coupling between rocket stages is easier to control and will react faster if it has a greater stiffness constant. The necessity of a closed loop control system can be observed in the step responses of each system. The closed loop system enables the algorithm to make corrections for error in the motor output.

Appendix A: LabVIEW Program





Appendix B: MATLAB Program

```
%% Pole Placement for 2-cart system
%  2 cart continuous to discrete
%  Chris Wilson (06/13/2011)

clc,clear

c1=0.014;   % Friction on cart 1 (N*s/m)
c2=0.014;   % Friction on cart 2 (N*s/m)
m1=2;       % Mass of cart 1 (kg)
m2=2;       % Mass of cart 2 (kg)
k=200;      % Spring between carts (N/m)

Khw=469/.005;   %Hardware gain

%Continuous Model
F=[0 0 1 0;0 0 0 1;-k/m1 k/m1 -c1/m1 0;k/m2 -k/m2 0 -c2/m2];    %A matrix
G=[0;0;Khw/m1;0];   %B matrix
C=[1 0 0 0;0 1 0 0];        %output matrix
T=0.001;  %sampling rate

%Discrete Model
[A,B]=c2d(F,G,T);   %converts to discrete form

PM=.1;     %phase margin
damp=.9;  %damping ratio
p=damp*[cos(PM*pi/180)+sin(PM*pi/180)*i,cos(PM*pi/180)-sin(PM*pi/180)*i, ...
        (0.01+cos(PM*pi/180))+sin(PM*pi/180)*i,(0.01+cos(PM*pi/180))-
sin(PM*pi/180)*i]   %pole locations


K=acker(A,B,p); %Developes the K matrix based on the desired pole locations

%figure(get_fig('2cart place'))
pzmap(tf(1,poly(p)))

%%

PM=1;
damp=0.9;
p=damp*[cos(PM*pi/180)+sin(PM*pi/180)*i,cos(PM*pi/180)-sin(PM*pi/180)*i, ...
        (0.01+cos(PM*pi/180))+sin(PM*pi/180)*i,(0.01+cos(PM*pi/180))-
sin(PM*pi/180)*i];
L=place(A',C',p)';

%figure(get_fig('2cart acker'))
%pzmap(tf(1,poly(p)))

u=0;
q=[0;0;0;0];
tact=0;
```

```
x=[0;0;0;0];
xd=[1;1;0;0]*1e-2;
Nf=100;

for k=1:Nf          %state estimation equations
    y(:,k)=C*x(:,k);
    u(k)=-K*(q(:,k)-xd);
    q(:,k+1)=A*q(:,k)+L*(y(:,k)-C*q(:,k))+B*u(k);
    x(:,k+1)=A*x(:,k)+B*u(k);
    tact(k+1)=(k+1)*T;
end

figure
subplot(2,1,1)
plot(tact(1:end-1),x(1,1:end-1)*100,tact(1:end-1),q(1,1:end-1)*100,'--
','linewidth',2)
subplot(2,1,2)
plot(tact(1:end-1),x(2,1:end-1)*100,tact(1:end-1),q(2,1:end-1)*100,'--
','linewidth',2)
figure
plot(tact(1:end-1),u)
```

Acknowledgments

I would like to thank my mentor Pedro Capo-Lugo for providing guidance and support throughout the course of my project. Also, I would like to thank Brent Hipp for taking the time to explain various controls theory concepts as well as helping determining the source of any programming errors in LabVIEW. Without the help of Pedro and Brent, the success of my project would not have been possible.

References

[1] Educational Control Products, *Manual For Model 210/210a Rectilinear Control System*, Manual Version 1.4, ECP, Bell Canyon, CA, 2003.

[2] Bishop, R., Dorf, R., *Modern Control Systems*, 12th ed., Pearson Education, Inc., New Jersey, 2011.