Standup and stabilization of the inverted pendulum /

2 authors, including:

Kelechukwu Andrew
University of Nigeria
2 PUBLICATIONS **22** CITATIONS

SEE PROFILE

Standup and Stabilization of the Inverted Pendulum

by

Andrew K. Stimac

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science

at the

Massachusetts Institute of Technology

June 1999

Signature of Author: .....
Department of Mechanical Engineering
May 5, 1999

Certified by: .....
David L. Trumper
Associate Professor of Mechanical Engineering
Thesis Supervisor

Accepted by: .....
Ernest G. Cravalho
Chairman, Undergraduate Thesis Committee
Department of Mechanical Engineering

Standup and Stabilization of the Inverted Pendulum

by

Andrew K. Stimac

Submitted to the Department of Mechanical Engineering on
May 5, 1999, in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Mechanical Engineering.

Abstract

The inverted pendulum is a common, interesting control problem that involves many basic elements of control theory. This thesis investigates the standup routine and stabilization at the inverted position of a pendulum-cart system. The standup routine uses strategic cart movements to add energy to the system. Stabilization at the inverted position is accomplished through linear state feedback. Methods are implemented using the Matlab Simulink environment, with a dSPACE DSP controller board for interaction with the physical system. Algorithms described in this report were successful and consistently produced the desired results. This report serves as a guide to the current working system and as background information on the inverted pendulum.

Thesis Supervisor: David L. Trumper
Title: Associate Professor of Mechanical Engineering

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Purpose

This report explores the standup and stabilization of an inverted pendulum. It contains a theoretical analysis of the system dynamics and control methods, as well as a summary of the apparatus and implementation. There are two purposes of this report: to summarize and explain the methods of controlling this specific pendulum-cart system; and to explore and illustrate in general the inverted pendulum problem.

A major focus of this project was the use of Simulink software for control implementation. Simulink provides a block diagram representation of signal processing methods. Allowing for easy visualization, Simulink facilitates the design of complex control algorithms. While code scripts have been used previously for this project, Simulink provides a more intuitive and organized solution.

1.2 Project Overview

This thesis covers several topics. Primarily, there are two control problems: the standup routine and the stabilization in the inverted position. Also important, however, is the implementation of these control algorithms in the Simulink environment.

The standup routine uses strategic cart movements to gradually add energy to the pendulum. This involves first placing the cart under position control. Then a routine is developed to prescribe the cart's movement. This movement is such that the cart does work on the pendulum, in a consistent and efficient manner. It is also important to gradually reduce cart movement amplitude so that the standup routine delivers the pendulum to the inverted pendulum position with small angular velocity.

Inverted pendulum stabilization can be accomplished through several methods. This analysis uses state feedback to provide the desired response. Using LQR optimal design tools as a design hangle, the controlled system poles are placed to provide a fast, stable response.

Simulink implementation requires the exploration of specific Simulink techniques. While state feedback control is well suited to the Simulink environment, the standup routine includes some logic that would be more easily represented in a program script. Adaptation to Simulink requires the generation of several information variables that describe the current system state. These variables are the processed with logical operators in order to determine current actions. The Simulink adaptation is slightly complex. However, the Simulink implementation is superior to a script in many areas because routines have little or no preset order, depending primarily on the present system state.

1.3 Organization

The sections of this report cover both the theoretical and experimental aspects of this project. Section 2 contains a theoretical discussion of the system dynamics. This analysis is necessary for system modeling, controller design, and for general background understanding of the pendulum-cart system. Section 3 describes the details of the apparatus, including modeling assumptions and specific parameter values. Section 4 explores the controller design, for both standup and stabilization. Stabilization is covered in Section 4.1, and involves linear state feedback methods. Standup is covered in Section 4.2, which explores the strategic cart motion that delivers energy to the pendulum. Section 5 documents the specific Simulink implementation. This section seves as a user's guide to the current functioning system, as well as a tutorial on Simulink methods. Finally, Section 6 gives Recommendations for further improvements. Appendix A includes the Matlab files used and Appendix B contains simple instructions on how to operate the system.

1.4 Acknowledgments

I would like to voice my appreciation to everyone who helped me on this project. Special thanks to Professor Trumper for suggesting this thesis and for his guidance along the way, to Mike Liebman for his assistance and support in the Mechatronics Lab, and to Steve Ludwick for sharing his understanding of the current project setup. Finally, many thanks to those who made contributions on this project before me, particularly Steve Ludwick, Ming-chih Weng, and Pradeep Subrahmanyan, and Professor Will Durfee and his students who originally built the pendulum-cart experiment hardware at MIT.

2. Pendulum Cart Modeling and Dynamics

For the purpose of this report, it is necessary to understand the dynamics of the pendulum cart system. In the follow pages, a theoretical analysis is conducted, using the Lagrangian approach to derive the state equations. The result is linearized around two points of operation, pendulum up and down, and transfer functions are derived and analyzed. This approach provides an in depth theoretical explanation for the dynamics of this system. Throughout this analysis, vector and matrix quantities are distinguished using bold text.

2.1 Lagrangian Dynamic Analysis

A complete theoretical model of the pendulum cart system can be done using Lagrangian Dynamics. First we choose generalized coordinates, and then derive expressions for the generalized forces, energy functions, and Lagrangian. Finally, we can use Lagrange's Equation to derive the equations of motion [3].

A schematic of the system is shown in Figure 2.1. This model and coordinate system will be used in the analysis. For completeness, this derivation involves inertia about the pendulum center of mass, I, though in the case of a point mass this term becomes zero.

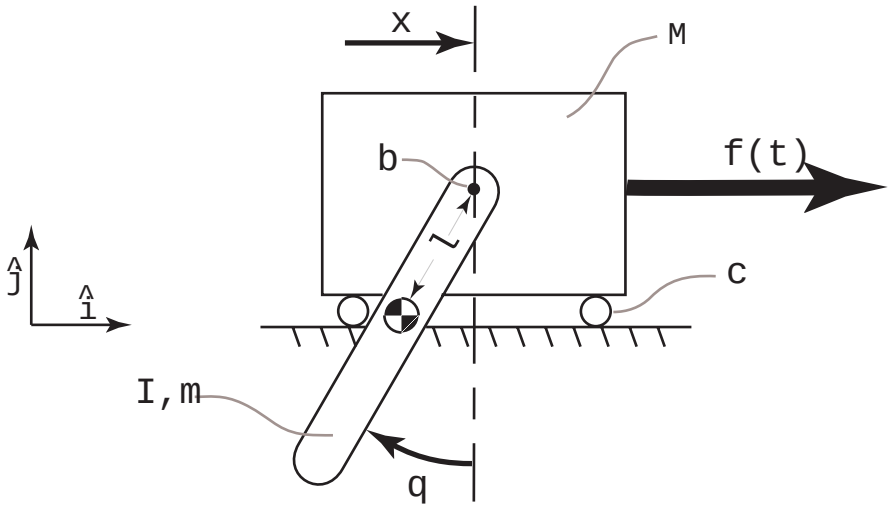

Figure 2.1: Schematic of Pendulum Cart. Model includes cart mass M, pendulum of mass m and inertia I about its center of mass, which is a distance l from the pendulum pivot. Also included is rotational and translational viscous damping coefficients b and c.

Generalized Coordinate System

The cart pendulum system has two degrees of freedom and can therefore be fully represented using two generalized coordinates. For this analysis, the generalized

coordinates are chosen as the horizontal displacement of the cart, x, and the rotational displacement of the pendulum, θ:

$$\xi_j : x, \theta \tag{2.1}$$

The positive direction of x is to the right and the positive direction of θ is clockwise, measured from the downward position. Positive θ was chosen in the clockwise direction so that x and θ are both measured to the right when the pendulum is in its inverted position. A complete and indepentent set of the associated admissible variations is given by

$$\xi_j : \delta x, \delta\theta \tag{2.2}$$

so the system is holonomic.

Generalized Forces

An expression for the generalized forces, $\Xi_{j,}$ can be derived from the nonconservative work, given by

$$\delta W^{nc} = \sum_{i=1}^{N} \mathbf{f^{nc}} \bullet \delta \mathbf{R_i} = \sum_{j=1}^{n} \Xi_j \delta \xi_j \tag{2.3}$$

where $\mathbf{R_i}$ is the position vector where the i^{th} nonconservative force acts. The nonconservative forces in this case result from the input force and the system damping,

$$\delta W^{nc} = f(t)\delta x - c\dot{x}\delta x - b\dot{\theta}\delta\theta \tag{2.4}$$

Comparing Equations 2.3 and 2.4 yields expressions for the generalized forces:

$$\begin{aligned} \Xi_x &= f(t) - c\dot{x} \\ \Xi_\theta &= -b\dot{\theta} \end{aligned} \tag{2.5}$$

Kinetic and Potential Energy Functions

The kinetic coenergy function for the cart mass is simply

$$T_M^* = \frac{1}{2} M \dot{x}^2 \tag{2.6}$$

For the pendulum, the coenergy function can be derived from

$$T_m^* = \frac{1}{2} m \mathbf{v}_c \bullet \mathbf{v}_c + \frac{1}{2} I \omega^2 \tag{2.7}$$

where I is the moment of inertial around the pendulum's center of mass and $\mathbf{v}_c$ is the velocity of the pendulum's center of mass. This velocity can be related to the position of the pendulum's center of mass,

$$\mathbf{r}_c = (x - l\sin\theta)\hat{\mathbf{i}} - l\cos\theta\hat{\mathbf{j}} \tag{2.8}$$

by

$$\mathbf{v}_c = \frac{d\mathbf{r}_c}{dt} = (\dot{x} - l\cos\theta\dot{\theta})\hat{\mathbf{i}} + l\sin\theta\dot{\theta}\hat{\mathbf{j}} \tag{2.9}$$

The angular velocity of the pendulum is simply

$$\omega = \dot{\theta} \tag{2.10}$$

Substituting Equations 2.9 and 2.10 into Equation 2.7 yields

$$T_m^* = \frac{1}{2} m(\dot{x}^2 - 2\dot{x}l\cos\theta\dot{\theta} + l^2\cos^2\theta\dot{\theta}^2 + l^2\sin^2\theta\dot{\theta}^2) + \frac{1}{2} I\dot{\theta}^2 \tag{2.11}$$

which, upon simplifying, becomes

$$T_m^* = \frac{1}{2} m(\dot{x}^2 - 2\dot{x}l\cos\theta\dot{\theta} + l^2\dot{\theta}^2) + \frac{1}{2} I\dot{\theta}^2 \tag{2.12}$$

The total kinetic coenergy is

$$T^* = T_M^* + T_m^* = \frac{1}{2} M\dot{x}^2 + \frac{1}{2} m(\dot{x}^2 - 2\dot{x}l\cos\theta\dot{\theta} + l^2\dot{\theta}^2) + \frac{1}{2} I\dot{\theta}^2 \tag{2.13}$$

Since the cart moves only in the horizontal direction, the potential energy of the system is determined entirely by the angle of the pendulum, given by

$$V = -mgl\cos\theta \tag{2.14}$$

Lagrangian

From the kinetic coenergy and potential energy functions, the Lagrangian is given by

$$L = T^* - V \tag{2.15}$$

Using Equations 2.13 and 2.14, the Lagrangian can be written as

$$L = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}^2 - 2\dot{x}l\cos\theta\dot{\theta} + l^2\dot{\theta}^2) + \frac{1}{2}I\dot{\theta}^2 + mgl\cos\theta \tag{2.16}$$

Lagrange's Equation

State equations can be generated using Lagrange's Equation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\xi}_j}\right) - \frac{\partial L}{\partial \xi_j} = \Xi_j \tag{2.17}$$

The equation for x is

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = \Xi_x \tag{2.18}$$

Using Equation 2.16 and evaluating the partial derivatives yields

$$\frac{d}{dt}(M\dot{x} + m\dot{x} - ml\cos\theta\dot{\theta}) - 0 = f(t) - c\dot{x} \tag{2.19}$$

which reduces to

$$(M+m)\ddot{x} - ml\cos\theta\ddot{\theta} + ml\sin\theta\dot{\theta}^2 = f(t) - c\dot{x} \tag{2.20}$$

The equation for θ is

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \Xi_\theta \tag{2.21}$$

Using Equation 2.16 and evaluating the partial derivatives yields

$$\frac{d}{dt}(-ml\dot{x}\cos\theta + ml^2\dot{\theta} + I\dot{\theta}) - (ml\dot{x}\sin\theta\dot{\theta} - mgl\sin\theta) = -b\dot{\theta} \tag{2.22}$$

which reduces to

$$(ml^2 + I)\ddot{\theta} - ml\ddot{x}\cos\theta + ml\dot{x}\sin\theta\dot{\theta} - ml\dot{x}\sin\theta\dot{\theta} - mgl\sin\theta = -b\dot{\theta} \tag{2.23}$$

Simplifying and rearranging, the system state equations are

$$\begin{cases} (M+m)\ddot{x} + c\dot{x} - ml\cos\theta\ddot{\theta} + ml\sin\theta\dot{\theta}^2 = f(t) \\ -ml\ddot{x}\cos\theta + (ml^2 + I)\ddot{\theta} + b\dot{\theta} + mgl\sin\theta = 0 \end{cases} \tag{2.24}$$

2.2 Linearization and Transfer Function Generation

Before proceeding with further analysis, we must linearize the state equations. There are two equilibrium points: $\theta=0$ (pendulum down, stable) and $\theta=\pi$ (pendulum up, unstable). Focusing on small variations of θ about the equilibrium point θ_o:

$$\begin{aligned} \theta &= \theta_0 + \varepsilon \\ \dot{\theta} &= \dot{\varepsilon} \end{aligned} \tag{2.25}$$

From a Taylor Series expansion, a first order approximation of any function of θ is

$$f(\theta) \approx f(\theta_0) + \varepsilon \left.\frac{df}{d\theta}\right|_{\theta_0} \tag{2.26}$$

Also, because higher order terms are neglected,

$$\dot{\varepsilon}^2 \approx 0 \tag{2.27}$$

2.2.1 Pendulum Down ($\theta=0$)

For $\theta=0$, Equation 2.26 yields to first order

$$\begin{aligned} \cos\theta &\approx \cos(0) + \theta[-\sin(0)] = 1 \\ \sin\theta &\approx \sin(0) + \theta[\cos(0)] = \theta \end{aligned} \tag{2.28}$$

Substituting these linearizations into the system state equations (Equation 2.24) and neglecting the higher order term yields

$$\begin{cases} (M+m)\ddot{x} + c\dot{x} - ml\ddot{\theta} = f(t) \\ -ml\ddot{x} + (ml^2 + I)\ddot{\theta} + b\dot{\theta} + mgl\theta = 0 \end{cases} \tag{2.29}$$

Taking the Laplace Transform

$$\begin{cases}(M+m)s^2X(s)+csX(s)-mls^2\Theta(s)=F(s)\\-mls^2X(s)+(ml^2+I)s^2\Theta(s)+bs\Theta(s)+mgl\Theta(s)=0\end{cases} \tag{2.30}$$

Using substitution to eliminate either X(s) or Θ(s) yields two transfer functions:

$$\begin{aligned}G_1(s)&=\frac{X(s)}{F(s)}=\frac{a_2s^2+a_1s+a_0}{b_4s^4+b_3s^3+b_2s^2+b_1s}\\G_2(s)&=\frac{\Theta(s)}{F(s)}=\frac{c_2s}{b_4s^3+b_3s^2+b_2s+b_1}\end{aligned} \tag{2.31}$$

where

$$\begin{aligned}a_2&=ml^2+I\approx ml^2\\a_1&=b\\a_2&=mgl\\b_4&=(M+m)(ml^2+I)-m^2l^2\approx Mml^2\\b_3&=(M+m)b+(ml^2+I)c\approx(M+m)b+ml^2c\\b_2&=(M+m)mgl+bc\\b_1&=mglc\\c_2&=ml\end{aligned}$$

The approximations shown in the coefficients above are for I=0, which is the ideal case where the pendulum is constructed from a point mass and massless rod, so that it has no moment of inertia about its center of mass. Using this approximation, the transfer functions reduce to:

$$\begin{aligned}G_1(s)&=\frac{X(s)}{F(s)}=\frac{ml^2s^2+bs+mgl}{Mml^2s^4+[(M+m)b+ml^2c]s^3+[(M+m)mgl+bc]s^2+mglcs}\\G_2(s)&=\frac{\Theta(s)}{F(s)}=\frac{mls}{Mml^2s^3+[(M+m)b+ml^2c]s^2+[(M+m)mgl+bc]s+mglc}\end{aligned} \tag{2.32}$$

The pole-zero plots and bode diagrams for these transfer functions are shown in Figure 2.2.

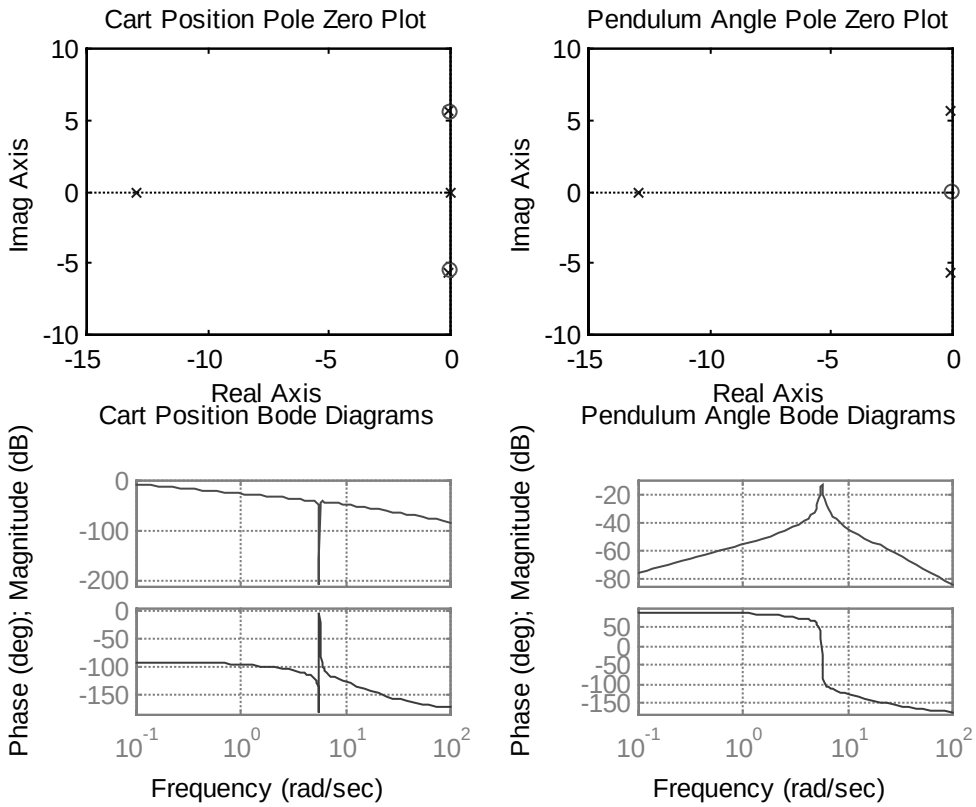

Figure 2.2: Pole-zero plots and Bode diagrams with the system linearized around θ=0 (Pendulum down)

2.2.2 Pendulum Up (θ=π)

For θ=π, Equation 2.26 yields to first order

$$\begin{aligned} \cos\theta &\approx \cos\pi + (\pi - \theta)(-\sin\pi) = -1 \\ \sin\theta &\approx \sin(\pi) + (\pi - \theta)\cos(\pi) = \pi - \theta \end{aligned} \tag{2.33}$$

Analysis is simplified by defining a new coordinate [1]

$$\theta' = \theta - \pi \tag{2.34}$$

This is nothing more than θ measured clockwise from the up pendulum position. Now

$$\sin\theta \approx \pi - (\theta' + \pi) = -\theta' \tag{2.35}$$

From here on in the analysis, θ' will be written as θ, but it should be noted that θ now measures from a new reference. Note also that

$$\dot{\theta}' = \dot{\theta} \quad \text{and} \quad \ddot{\theta}' = \ddot{\theta} \tag{2.36}$$

so this substitution does not have any other effects on the state equations.

Substituting these linearizations into the system state equations and neglecting the higher order term:

$$\begin{cases} (M+m)\ddot{x} + c\dot{x} + ml\ddot{\theta} = f(t) \\ ml\ddot{x} + (ml^2 + I)\ddot{\theta} + b\dot{\theta} - mgl\theta = 0 \end{cases} \tag{2.37}$$

Taking the Laplace Transform

$$\begin{cases} (M+m)s^2X(s) + csX(s) + mls^2\Theta(s) = F(s) \\ mls^2X(s) + (ml^2 + I)s^2\Theta(s) + bs\Theta(s) - mgl\Theta(s) = 0 \end{cases} \tag{2.38}$$

Using substitution to eliminate either X(s) or Θ(s) yields two transfer functions:

$$\begin{aligned} G_1(s) &= \frac{X(s)}{F(s)} = \frac{a_2s^2 + a_1s + a_0}{b_4s^4 + b_3s^3 + b_2s^2 + b_1s} \\ G_2(s) &= \frac{\Theta(s)}{F(s)} = \frac{c_2s}{b_4s^3 + b_3s^2 + b_2s + b_1} \end{aligned} \tag{2.39}$$

where

$$\begin{aligned} a_2 &= ml^2 + I \approx ml^2 \\ a_1 &= b \\ a_2 &= -mgl \\ b_4 &= (M+m)(ml^2 + I) - m^2l^2 \approx Mml^2 \\ b_3 &= (M+m)b + (ml^2 + I)c \approx (M+m)b + ml^2c \\ b_2 &= -(M+m)mgl + bc \\ b_1 &= -mglc \\ c_2 &= -ml \end{aligned}$$

The approximations shown above are for I=0, which is the ideal case where the pendulum is constructed from a point mass and massless rod, so that it has no moment of inertia

about its center of mass. Using this approximation, the transfer functions reduce to:

$$G_1(s) = \frac{X(s)}{F(s)} = \frac{ml^2 s^2 + bs - mgl}{(Mml^2 s^4 + [(M+m)b + ml^2 c]s^3 + [-(M+m)mgl + bc]s^2 - mglcs}$$
$$G_2(s) = \frac{\Theta(s)}{F(s)} = \frac{-mls}{Mml^2 s^3 + [(M+m)b + ml^2 c]s^2 + [-(M+m)mgl + bc]s - mglc} \quad (2.40)$$

The pole-zero plots and bode diagrams for these transfer functions are shown in Figure 2.3.

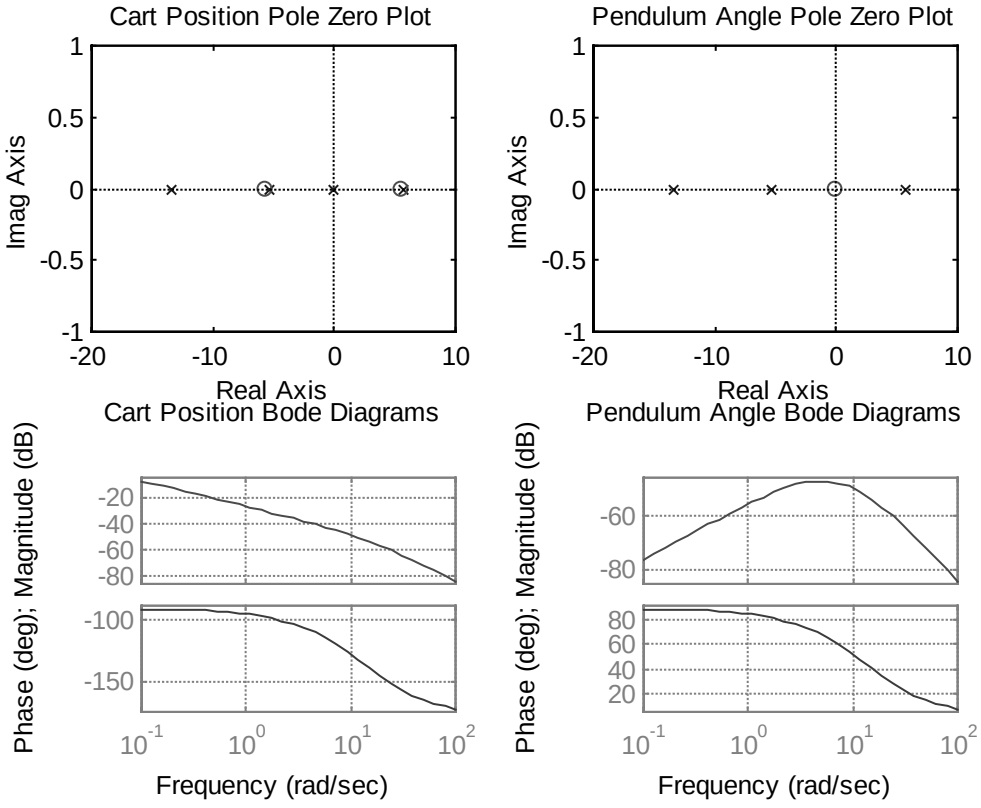

Figure 2.3: Pole-zero plots and Bode diagrams with the system linearized around θ=π (Pendulum up).

2.3 State Space Representation

For linear state control of the inverted pendulum, it is necessary to convert the state equations to state space representation in the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2.41}$$

For the state vector **x** a change in variable notation is needed, defined by

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{pmatrix} \tag{2.42}$$

Referring back to the linearized state equations for Pendulum Up (Equation 2.37), and neglecting I, this change of variables leads to

$$\begin{cases} (M+m)\dot{x}_2 + cx_2 + ml\dot{x}_4 = f(t) \\ ml\dot{x}_2 + ml^2\dot{x}_4 + bx_4 - mglx_3 = 0 \end{cases} \tag{2.43}$$

From the variable definitions:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_3 &= x_4 \end{aligned} \tag{2.44}$$

Expressions for $\dot{x}_2$ and $\dot{x}_4$ can be found by using substitution to eliminate either $\dot{x}_4$ or $\dot{x}_2$ from Equation 2.43. The result, in matrix form, is given by

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -c/M & -mg/M & b/Ml \\ 0 & 0 & 0 & 1 \\ 0 & c/Ml & (M+m)g/Ml & -(M+m)b/Mml^2 \end{bmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} + \begin{bmatrix} 0 \\ 1/M \\ 0 \\ -1/Ml \end{bmatrix} f(t) \tag{2.45}$$

3. Apparatus

A real pendulum-cart system was used for this thesis project. This system includes the necessary equipment to constrain motion, apply force, measure states, and implement control schemes.

A block diagram of the system is shown in Figure 3.1, with a full schematic in Figure 3.2. The components involved are a current amplifier, DC motor with gearing, a cart with pendulum, an array of feedback instruments, and a computer with Simulink software for signal processing.

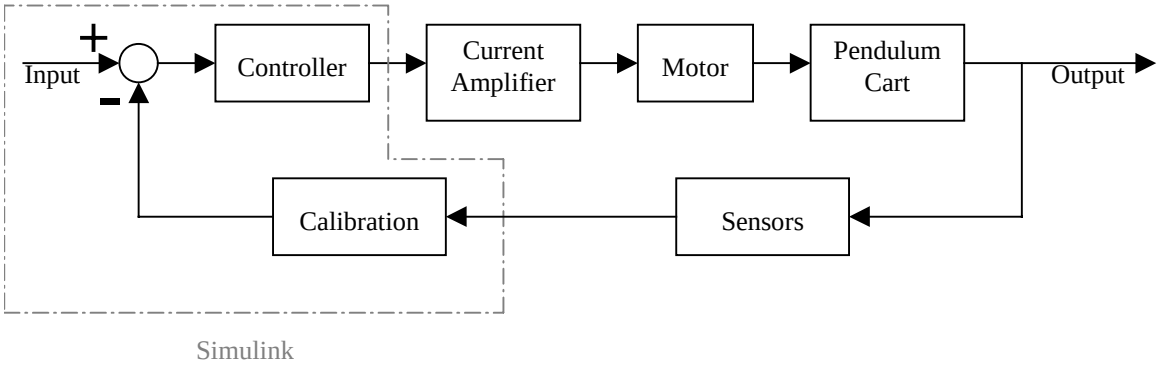

Figure 3.1: Pendulum Cart System Block Diagram.

Current is amplified using an Aerotech 4020 DC Servo Controller in current mode. This amplifier has a gain of 2 A/V, with a bandwidth of 2 kHz. Maximum ratings at 25° C are 40 V, 5 A continuous, and 20 A peak (2 seconds). The amplifier input power required is 100-125 V at 50-60 Hz, which is taken directly from an electrical outlet. For general operation, amplifier dynamics are negligible and the amplifier is modeled as a single gain K_a=2 A/V.

The driving motor is an Aerotech Permanent Magnet DC Servo Motor, Model 1135 Standard. The motor has a torque constant K_t=0.17 N-m/A, inertia J=3.6×10^{-4} kg-m^2, and viscous damping b=0.007 N-m/krpm. Because this motor is driven by a current amplifier, the back emf, K_b=18.2 V/krpm, need not be considered. Electrical dynamics can also because of the current drive. In any case, the electrical time constant is τ_e=2.2 ms, which is small compared to the mechanical time constant of τ_m=16 ms. Armature resistance, 1.4 Ω, is not important because a current drive is used. The motor torque is connected directly to the cart using a chain drive with a radius of 4 cm. Including this gearing, the total motor gain can be expressed as K_m=4.25 N/A. Motor inertia and damping is lumped with parameters of the pendulum cart. This motor is also equipped with a tachometer with constant K_{tach}=3 V/krpm, used for cart velocity feedback.

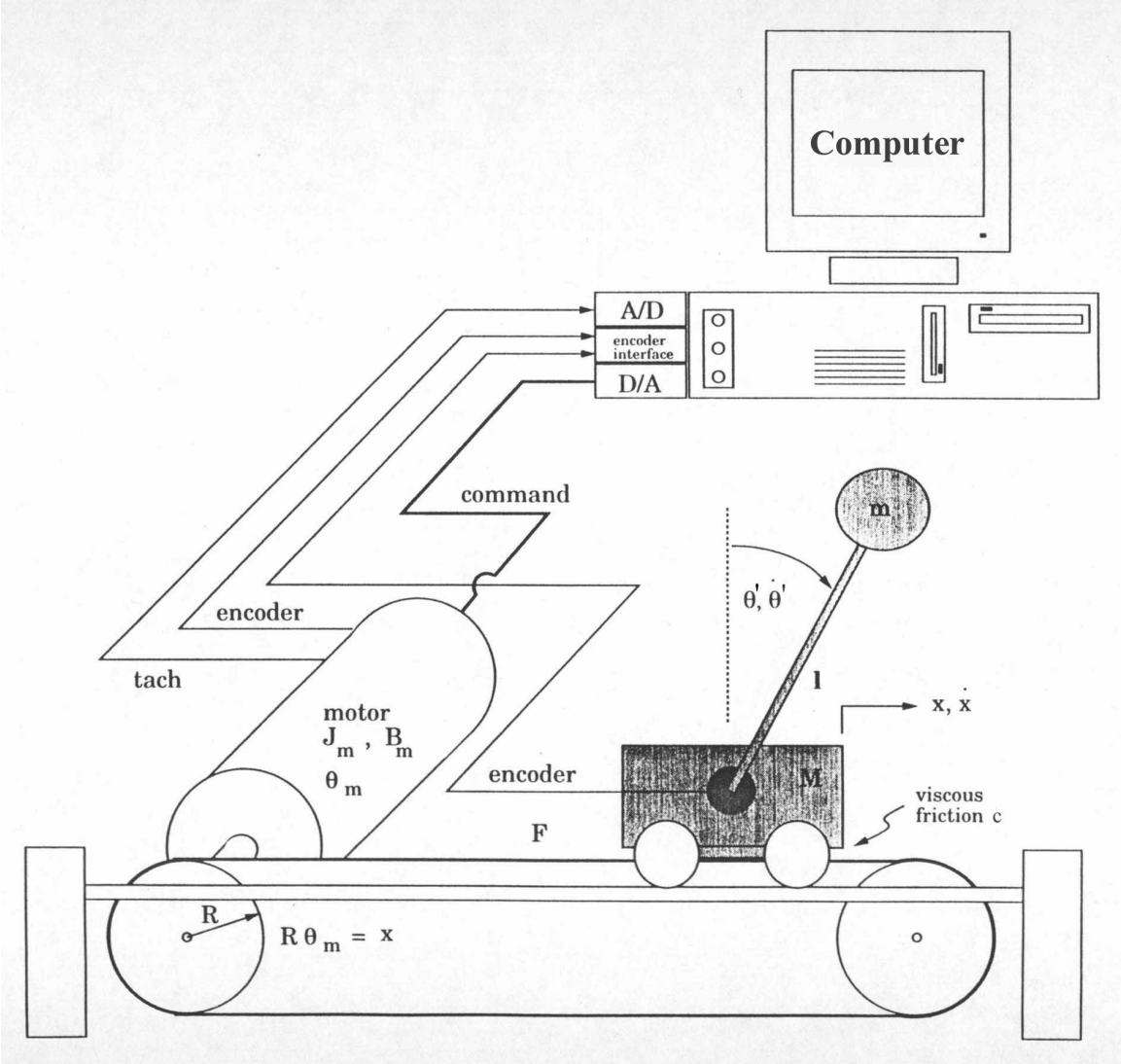

Figure 3.2: Schematic of Pendulum-Cart System.

A photograph of the pendulum-cart system is shown in Figure 3.3. Rod mass and inertia about the pendulum center of mass are small and can be neglected. The pendulum is thus approximated as a point mass m=0.15 kg on a massless rod of length l=0.314 m. The pendulum is attached to the cart with a pivot joint that allows full rotational motion with very low viscous friction. The cart is mounted on a horizontal track, which constrains cart motion to one dimension. The track imposes some sliding friction (modeled approximately as viscous damping) and has end stops that limit travel to 0.323 m. Force can be applied to the cart through the chain drive.

For low frequency operation, including the frequencies of interest herein, the cart mass and motor inertia are not independent because the chain and gearing that connects them are relatively rigid. Motor inertia can thus be expressed as an equivalent mass. Using the relations between torque T, force F, angular

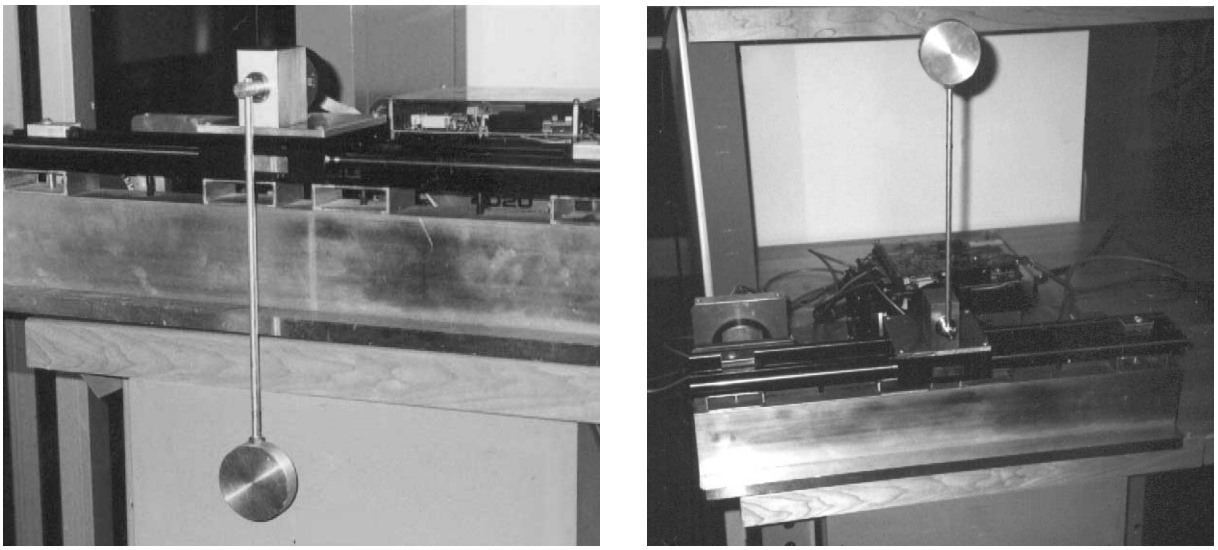

Figure 3.3: Photographs of the Pendulum Cart with pendulum down (left) and up (right).

velocity ω, and velocity v for a wheel of radius R:

$$T = FR = J\dot{\omega} = J\frac{\dot{v}}{R} \tag{3.1}$$

$$F = \frac{J}{R^2}\dot{v} \tag{3.2}$$

$$M_{eq} = \frac{J}{R^2} \tag{3.3}$$

For the actual values, this equivalent mass is M_{eq}=0.225 kg, which is added to the mass of the cart (1.3 kg) for a total effective mass M=1.525 kg. Higher frequency operation would require more detailed theoretical analysis, but this is well above system crossover and therefore does not warrant consideration.

Cart position and velocity and pendulum angular position are measured directly for feedback. Pendulum angular position (θ) and cart position (x) are measured using two mechanical encoders mounted at the pendulum joint and motor shaft. Both encoders have 4000 counts per revolution. Considering the setup and gearing, this gives a resolution of 63 μm in cart position x and 1.57×10^{-3} rad or 0.09° in pendulum angle θ.

Cart velocity is measured using the tachometer on the motor shaft. This signal is processed using a voltage follower and a unity gain low pass filter with a bandwidth 2 kHz.

Simulink software, with a dSPACE controller board, is used for signal processing and controller implementation. Signals to and from the hardware are connected using a DS1102 interface card and processed digitally. Simulink is used to manage and calibrate

the array of feedback signals, and to implement both the standup routine and linear controller. The sampling rate as implemented was 2000 Hz, which is high enough to produce negligible phase loss at crossover.

4. Controller Design

Control algorithms for this project had two functions: 1) to gradually swing the pendulum to the inverted position and then 2) to balance the pendulum at this unstable equilibrium point. The first requires a routine of precision cart movement that gradually adds energy to the system. The second is solved using state feedback. This section focuses on both the theoretical design and experimental results, as these topics are frequently inseparable in controller design. The specific Simulink implementation is covered later in section 5.

4.1 Linear Control (Pendulum Inverted)

In the inverted position, the pendulum is unstable without control. The transfer function between cart position x and control input contains both a pole and a zero in the right half plane. Because of the proximity of this pole zero pair, it is difficult to obtain a stable response through classical feedback methods.

Instead, stabilization of the pendulum is conducted through full state feedback. Using this method, the system poles can be placed arbitrarily.

For a Linear Time Invariant System

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.1}$$

the system poles are given by the eigenvalues of **A**, defined as the solutions λ to

$$\left|\lambda\mathbf{I} - \mathbf{A}\right| = 0 \tag{4.2}$$

where **I** is the identity matrix of proper dimension and the symbol '| |' refers to the matrix determinant. An array of negative feedback gains is used, so that the input **u** is proportional to the given states:

$$\mathbf{u} = -(\mathbf{k}_1\mathbf{x}_1 + \mathbf{k}_2\mathbf{x}_2 + \cdots + \mathbf{k}_n\mathbf{x}_n) = -\mathbf{K}\mathbf{x} \tag{4.3}$$

where

$$\mathbf{K} = [\mathbf{k}_1 \quad \mathbf{k}_2 \quad \cdots \quad \mathbf{k}_n] \tag{4.4}$$

For systems with single inputs, **u** and $\mathbf{k}_1$ through $\mathbf{k}_n$ reduce to scalar quantities, and **K** is a row vector. Substituting Equation 4.3 into Equation 4.1,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}\mathbf{x} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} \tag{4.5}$$

Analysis is simplified by defining the controlled state matrix

$$\mathbf{A_c} = \mathbf{A} - \mathbf{BK} \tag{4.6}$$

yielding

$$\dot{\mathbf{x}} = \mathbf{A_c}\mathbf{x} \tag{4.7}$$

Now the system poles are given by the eigenvalues of $\mathbf{A_c}$, defined by

$$\left|\lambda\mathbf{I} - \mathbf{A_C}\right| = 0 \tag{4.8}$$

It can be shown for most systems that by choosing the proper gain matrix, the eigenvalues of $\mathbf{A_c}$, which are the controlled system poles, can in theory be placed anywhere. Using this method, a fast, stable response can be accomplished.

Actually performance is limited by the physical hardware, which includes terms such as friction, finite track length, and amplifier saturation. Best results are achieved by optimizing between response speed and control effort. This was accomplished numerically using a linear-quadratic regulator (LQR) method. The Matlab function 'lqr' performs this operation, and allows for weighting of both the state errors and the control effort. Figure 4.1 shows the original and controlled pole placement. Figure 4.2 shows a simulated response to an initial error in theta under this control scheme.

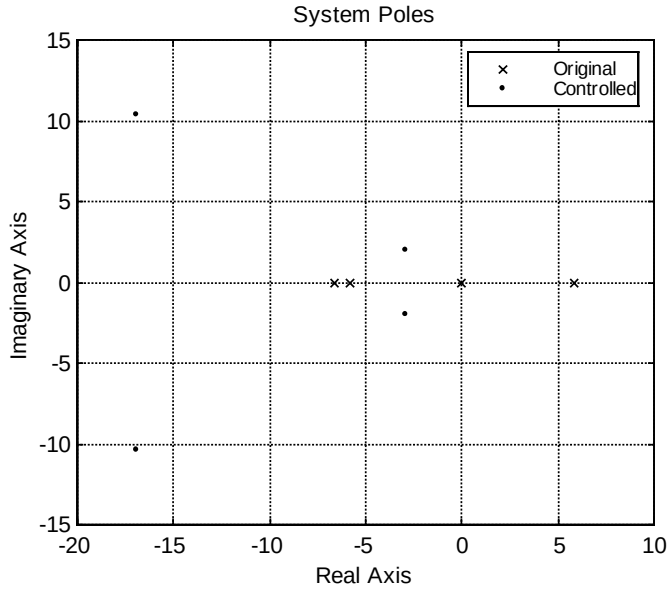

Figure 4.1: Resulting LQR Pole Placement.

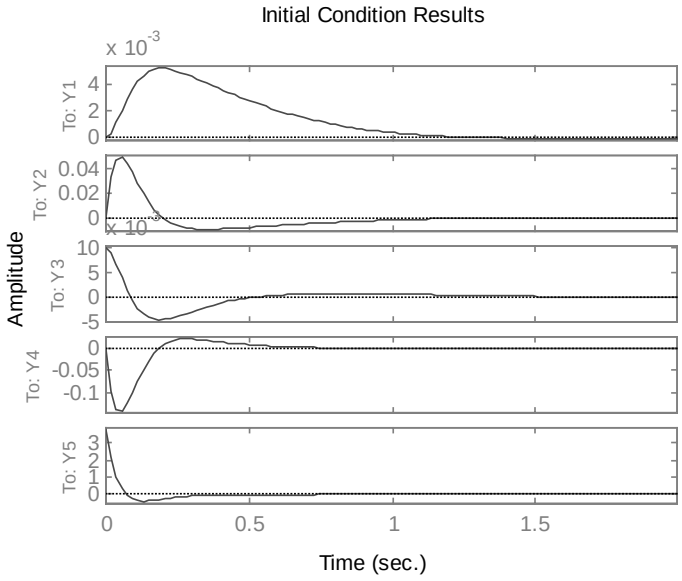

Figure 4.2: Simulated system response for an initial error in θ. From top to bottom, output shown is $x, \dot{x}, \theta, \dot{\theta}$, and control effort.

4.2 Standup Routine

The pendulum begins at rest, hanging downward in a stable equilibrium. The standup routine raises the pendulum to the inverted position, where the linear controller can stabilize it. It is important that the standup routine delivers the pendulum to the inverted position in a controlled, predictable fashion and at small angular velocity.

The basic strategy is to move the cart in such a motion that energy is gradually added to the pendulum. This first requires putting the cart under position control. Then a routine is needed to drive the cart's position.

It is critical that this cart motion is synchronized with the pendulum swing. Precalculated movements and pauses will not suffice, being prone to system disturbances and uncertainty. Instead, a control method is needed which reacts to the current system state, and prescribes cart position accordingly.

4.2.1 Cart Position Control

The control of the cart position is straight forward, and will only be covered briefly. There exist many position control schemes alternate to what is presented here. However, it is required only that this controller act fast compared to pendulum movement and reject the disturbance forces caused by the pendulum swing. Significant overshoot is undesirable because it causes unpredictability.

For the design of this controller, pendulum motion is ignored. This results in a simple cart model two pole system with mass and damping. A bode plot is shown in Figure 4.3. It can be seen that even at low frequencies, both poles contribute a significant phase shift.

Higher crossovers can be achieved using a lead compensator with proportional gain. Bode plots of the suggested controller and the resulting transfer function are shown in Figure 4.4. The controller shown provides an additional 42° phase margin for a closed loop system crossover at 40 rad/s with 55° total phase margin.

Experimental results differed significantly from the theoretical modeling. An experimental bode plot (Figure 4.5) shows reduced amplitude and further phase degradation. Significant care was taken to ensure that this discrepancy is not a calculation error. Direct measurements of closed loop force vs. displacement show that all gains are accounted for correctly. Instead, the difference must lie in modeling assumptions and parameter values. In practice, it was necessary to significantly increase the proportional gain in order to produce the desired crossover.

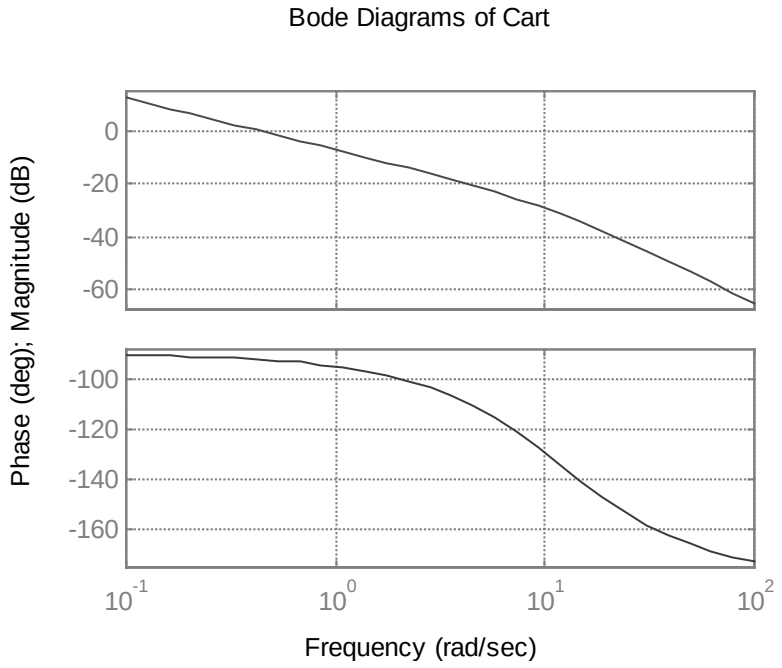

Figure 4.3: Bode Diagrams of Cart System.

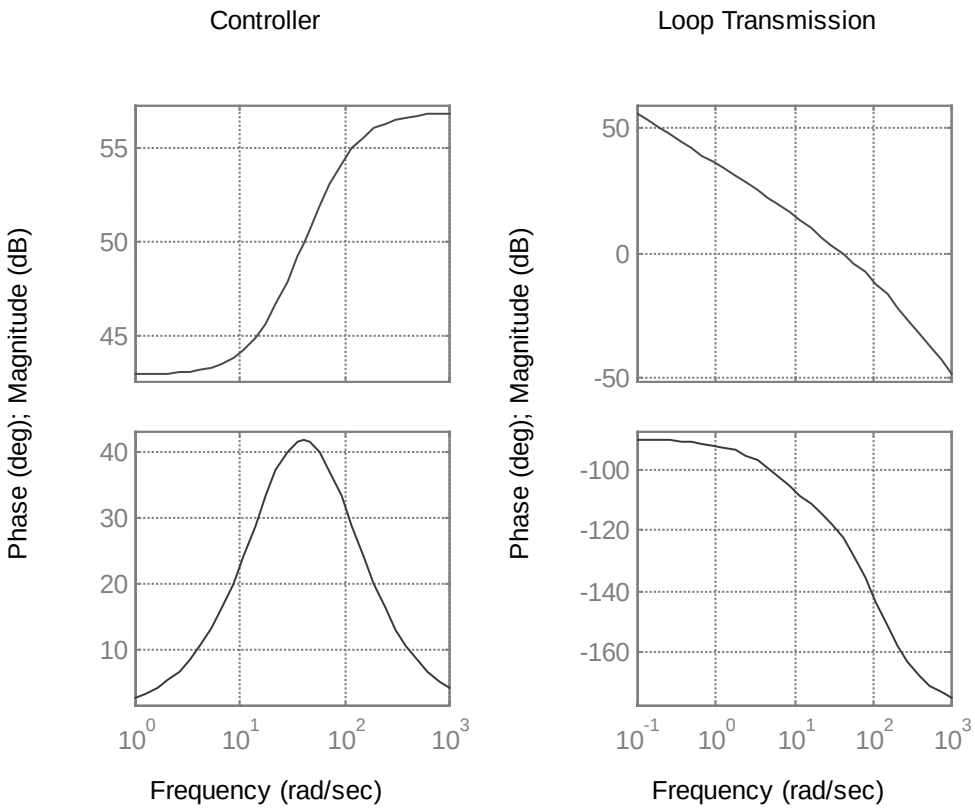

Figure 4.4: Lead Controller Designed for Crossover at 40 rad/sec.

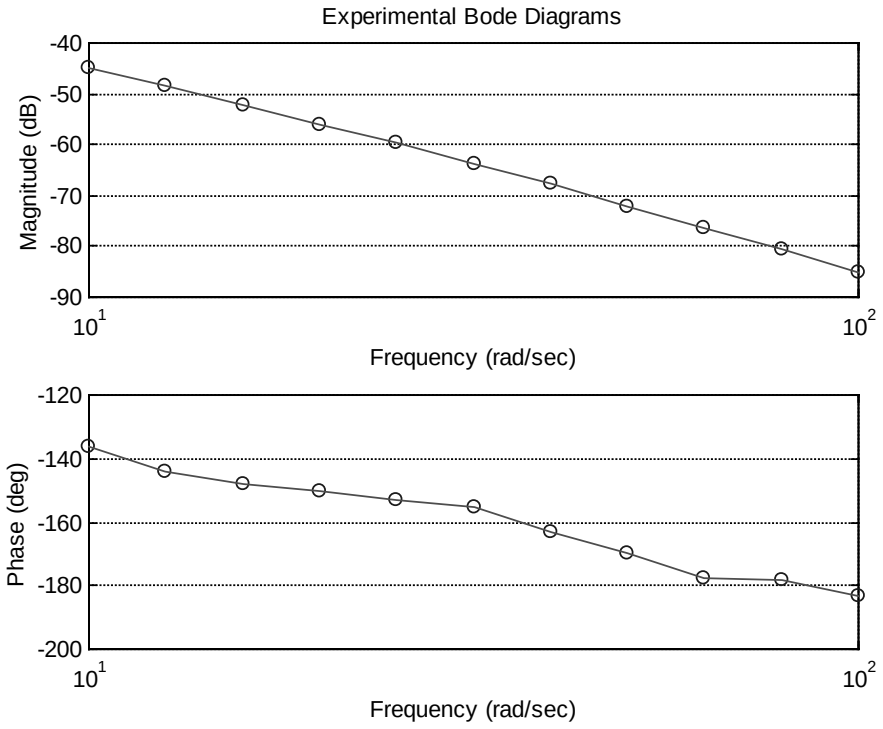

Figure 4.5: Experimental Bode Plot of the Cart System.

4.2.2 Motion Strategy: A Work Analysis

With the ability to prescribe cart motion, we should now explore how to best add energy to the pendulum. We desire a cart trajectory that raises the pendulum efficiently, consistently, and predictably.

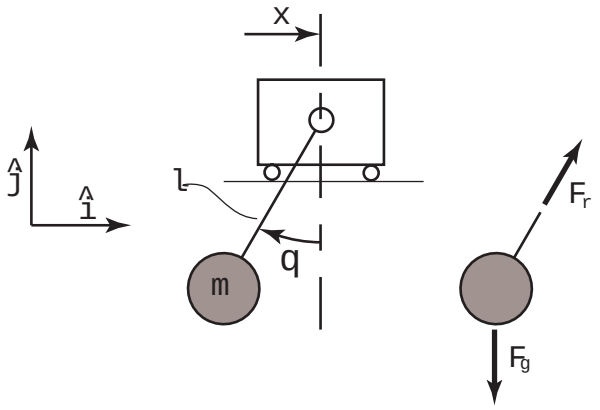

Figure 4.6: Coordinate System and Free Body Diagram of Pendulum Cart System.

Using a Newtonian approach, expressions can be derived relating pendulum and cart motion. With the cart under position control, it is natural to think of cart position, velocity, and acceleration all as input parameters which we can prescribe. Figure 4.6 shows again a diagram of the pendulum, with the same coordinates used earlier, and a free body diagram. Using Cartesian coordinates, the motion and forces on the pendulum can be described. Throughout this analysis, bold characters are used to denote vector quantities, while non-bold characters represent scalar quantities. Velocity and acceleration of the pendulum can be found by differentiating position:

$$\mathbf{r} = (x - l\sin\theta)\hat{\mathbf{i}} - l\cos\theta\hat{\mathbf{j}} \tag{4.9}$$

$$\mathbf{v} = \frac{d\mathbf{r}}{dt} = (\dot{x} - l\cos\theta\dot{\theta})\hat{\mathbf{i}} + l\sin\theta\dot{\theta}\hat{\mathbf{j}} \tag{4.10}$$

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = (\ddot{x} + l\sin\theta\dot{\theta}^2 - l\cos\theta\ddot{\theta})\hat{\mathbf{i}} + (l\cos\theta\dot{\theta}^2 + l\sin\theta\ddot{\theta}\hat{\mathbf{j}} \tag{4.11}$$

Two forces act on the pendulum mass: the force of the rod,

$$\mathbf{F_r} = F_r(\sin\theta\hat{\mathbf{i}} + \cos\theta\hat{\mathbf{j}}) \tag{4.12}$$

and the force of gravity,

$$\mathbf{F_g} = -mg\hat{\mathbf{j}} \tag{4.13}$$

Using Newton's Law,

$$\sum \mathbf{F} = m\mathbf{a} \tag{4.14}$$

$$F_r\sin\theta\hat{\mathbf{i}} + (F_r\cos\theta - mg)\hat{\mathbf{j}} = m(\ddot{x} + l\sin\theta\dot{\theta}^2 - l\cos\theta\ddot{\theta})\hat{\mathbf{i}} + m(l\cos\theta\dot{\theta}^2 + l\sin\theta\ddot{\theta})\hat{\mathbf{j}} \tag{4.15}$$

This can be rewritten as two equations, one for each direction:

$$\begin{aligned} F_r\sin\theta &= m(\ddot{x} + l\sin\theta\dot{\theta}^2 - l\cos\theta\ddot{\theta}) \\ F_r\cos\theta - mg &= m(l\cos\theta\dot{\theta}^2 + l\sin\theta\ddot{\theta}) \end{aligned} \tag{4.16}$$

This can produce several results. Using substitution to eliminating F_r and rearranging, an expression is found that relates the relative movements of the cart and pendulum:

$$\ddot{x} - \frac{l\ddot{\theta}}{\cos\theta} - g\tan\theta = 0 \tag{4.17}$$

Examining Equation 4.17, several points can be made. Without cart acceleration, this system is a simple pendulum, with gravity causing pendulum rotational acceleration at a magnitude increasing with swing angle. Cart acceleration also causes pendulum rotation,

in addition to the effects of gravity. It can be seen that the direction of pendulum rotational acceleration depends on the direction of cart acceleration and whether the pendulum angle is above or below 90°. The effect of a cart acceleration on pendulum rotation is greatest when θ=0° (pendulum down) and at θ=180° (pendulum up).

Alternatively, Equation 4.16 can be solved for F_r, by eliminating $\ddot{\theta}$:

$$F_r = m(\sin\theta\ddot{x} + l\dot{\theta}^2 + g\cos\theta) \tag{4.18}$$

It is desirable to know how much work is done on the system when the cart is moved a small distance. This is simply the component of the force that is in the direction of movement, times the small displacement,

$$\delta W = \mathbf{F_r} \bullet \partial\mathbf{x} = F_r \sin\theta\partial x = m(\sin^2\theta\ddot{x} + l\sin\theta\dot{\theta}^2 + g\sin\theta\cos\theta)\delta x \tag{4.19}$$

Equation 4.19 shows three terms contributing to the force in the rod, and therefore contributing to the work done: an inertial force, a centripetal force, and a gravitational force. These components have their maximum work contributions at ±90°, ±90°, and ±45°, respectively.

The first term in Equation 4.19 is of great importance, as it describes the work effect of a given acceleration at any angle. Figure 4.6 shows a plot of this term. For θ near ±90°, a given acceleration and displacement will do a lot of work; for θ near 0°, hardly any work is done. If the acceleration and displacement are in the same direction, the work done is positive and energy is added to the system. If the acceleration and displacement are in opposite directions (deceleration), the work done is negative and energy is taken out of the pendulum.

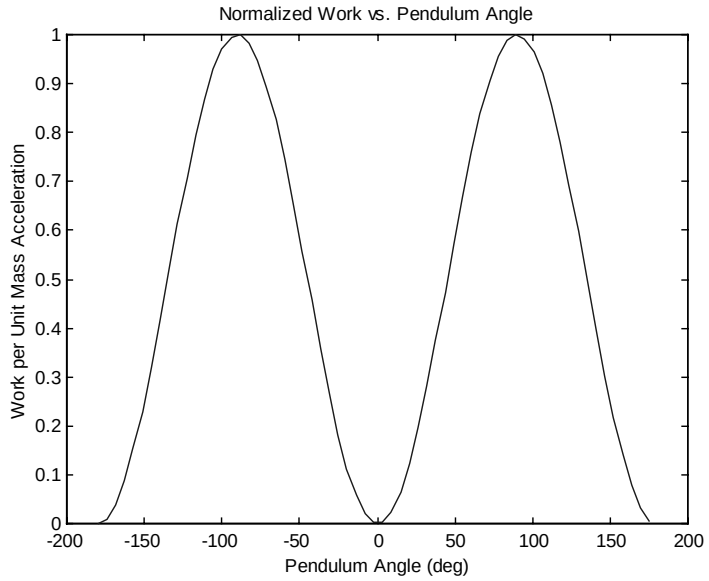

Figure 4.7: Ratio of Work to Acceleration as a Function of Pendulum Angle.

The strategy is to prescribe a cart trajectory that is driven by θ. Because the cart track is of finite length, the trajectory must contain both acceleration and deceleration. To maximize the positive work done on the pendulum, the trajectory should be such that the cart accelerates when there is high work transfer (θ near ±90°) and decelerates when there is low work transfer (θ near 0°). Figure 4.8 depicts a strategy that provides acceleration and deceleration in the proper regions.

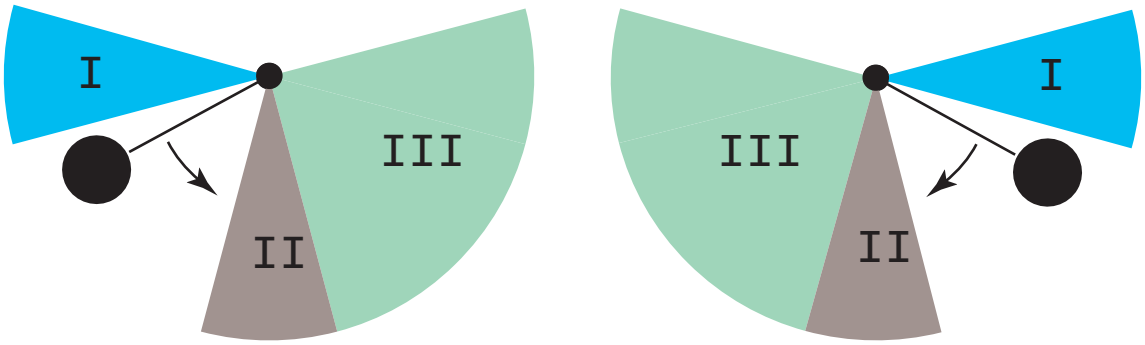

Figure 4.8: Cart Motion Strategy: Accelerate the cart when the pendulum is in region I, decelerate when the pendulum reaches region II, and then wait while the pendulum goes through region III until it changes direction. Deceleration in region III is undesirable because it will do negative work, taking energy out of the pendulum.

Because track length is limited, a cart position is specified instead of acceleration. The position trajectory must be chosen to have appropriate acceleration characteristics, accelerating and decelerating at the correct sections of the pendulum swing. One suitable trajectory is to use half a period of a cosine wave, adjusted to fit a section of the pendulum,

$$x_d = A\cos\left(\frac{\theta}{\theta_{max}}\pi\right) \quad \text{for} \quad 0 < \theta < \theta_{max} \quad \text{and} \quad \dot{\theta} < 0 \tag{4.20}$$

where A is the amplitude and θ_{max} is the angle at which motion starts and should be an angle with high positive work transfer. If the pendulum is not yet swinging past 90°, θ_{max} should be the highest point in the swing. Once the pendulum swings out of this region, to $\theta<0$, the cart should remain motionless and wait for the pendulum to begin swinging downward again. Then a similar movement in the opposite direction should be prescribed. Figure 4.9 shows a plot of this desired position function.

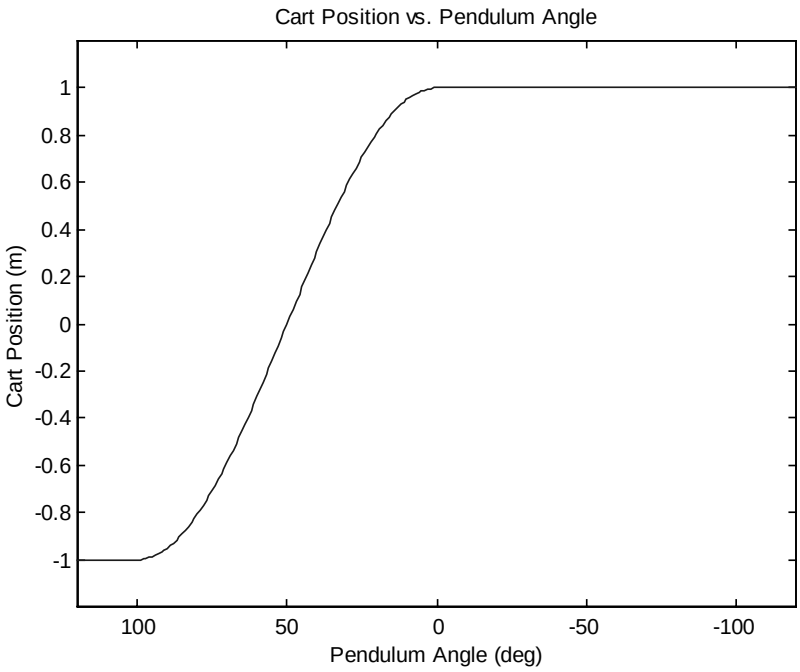

Figure 4.9: Cart Position versus Angle for the Pendulum Swinging to the Right.

It should be noted that because the trajectory given by Equation 4.20 is a function of θ, cart velocity and acceleration depend on the derivatives of θ. Specifically,

$$\dot{x}_d = \frac{\partial x_d}{\partial \theta}\dot{\theta} = -\frac{A\pi}{\theta_{\max}}\sin\left(\frac{\theta}{\theta_{\max}}\pi\right)\dot{\theta} \tag{4.21}$$

and

$$\ddot{x}_d = \frac{\partial \dot{x}_d}{\partial \theta}\dot{\theta} + \frac{\partial x}{\partial \dot{\theta}}\ddot{\theta} = -\frac{A\pi^2}{\theta_{\max}^2}\cos\left(\frac{\theta}{\theta_{\max}}\pi\right)\dot{\theta}^2 - \frac{A\pi}{\theta_{\max}}\sin\left(\frac{\theta}{\theta_{\max}}\pi\right)\ddot{\theta} \tag{4.22}$$

So the cart acceleration is not actually a cosine function, but involves the pendulum motion. When pendulum motion is considered, it can be shown that the cart acceleration and deceleration are in fact in the desired swing regions discussed earlier, although their magnitudes are quite dependent on swing dynamics.

For this analysis, the remaining terms in Equation 4.19 have been greatly ignored. This is mainly because unlike acceleration, $\dot{\theta}$ and gravity are beyond our control. Also, it is hoped that a fast controller will produce much greater effects through the first term. If anything, these later two terms suggest times when energy is added simply by moving the cart. This brings advantages to a trajectory that moves large distances while the pendulum is at θ of ±90° and ±45.

In choosing this trajectory, it was critical that cart motion would allow the pendulum to move through the proper swing areas. Because both acceleration and deceleration is necessary for the finite length track, the pendulum must move from regions of high to low work transfer. For the trajectory described by equation 4.20, it is necessary that the pendulum indeed swings from $\pm\theta_{max}$ to 0°. Certain rapid cart movements will in fact cause the pendulum to accelerate upwards (see Equation 4.17), especially when the pendulum begins at a low height. In all considered trajectories, pendulum dynamics must be analyzed.

The pendulum motion can be analyzed using a Matlab numerical solver. Figure 4.10 shows the pendulum swing and cart movement with the pendulum beginning motionless at initial angles of 40°, 100°, and 160°. For small swings, the pendulum begins moving slowly, both because gravity has a smaller accelerating effect and because cart movement tends to push the pendulum in the opposite direction. Great acceleration at low starting angles would cause the pendulum to swing upwards. Matlab scripts are included in Appendix A.

While the trajectory described so far can efficiently add energy to the system, the standup routine requires additional algorithms at the start and finish. This trajectory is ineffective when the pendulum is at rest, because there is no work transfer when θ=0° and because of the pendulum dynamics discussed earlier. In this region, a quick change in cart position can be used to start the pendulum swinging. Several, well-timed jumps improve performance.

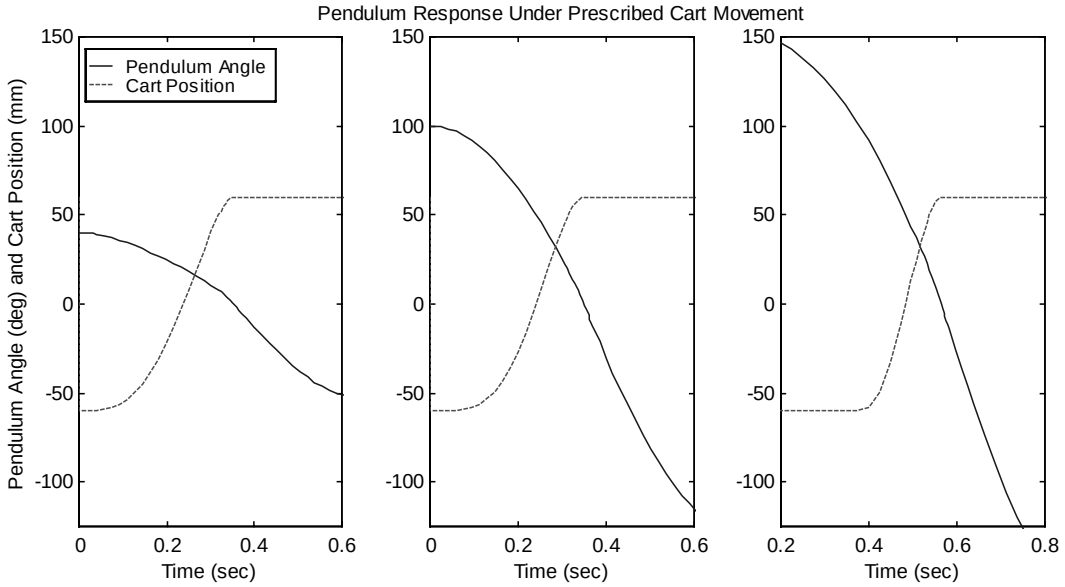

Figure 4.10: Pendulum and Cart Motion at Start Angles of 40°, 100°, and 160°. The cart moves only in a specific region of the swing in a trajectory that does work on the pendulum. For smaller start angles, the pendulum motion begins very slowly.

As the pendulum approaches the inverted position, caution must be used that the pendulum reaches vertical at a low velocity. The trajectory described in equation 4.20 is still effective, but the amplitude must be reduced.

In summary, the standup routine should consist of the following subroutines, in this order:

1. Begin with a series of quick cart movements that begin the pendulum swinging.
2. Move the cart in a fashion that efficiently adds energy to the pendulum.
3. As the pendulum swings higher, gradually reduce cart movement amplitude so that the pendulum approaches the inverted position in small increments and ultimately reaches vertical with small angular velocity.

Trajectory amplitude for subroutine 3 can be calculated as some function of the maximum height of the previous swing. This amplitude function can be tuned to produce the desired result of a gradual approach to vertical. More elaborate calculations could calculate the current energy of the pendulum, and prescribe a trajectory that delivered a certain amount of work. However, this calculation is computationally intensive, as it involves a work integral with continually changing parameters.

This algorithm is incapable of recovering if the pendulum is thrown over vertical with large velocity. It is therefore necessary to approach vertical carefully and gradually. Developing an algorithm that can recover opens the door to new strategies that are less cautious or perhaps intentionally throw the pendulum over the vertical, but this is beyond the scope of the present work.

5. Simulink Implementation

The routines and controllers described earlier, along with all general signal handling, are implemented using a single Simulink model. A Matlab script is used so that many system parameters are set from a single source. This section focuses on the details of the Simulink model, and assumes that the reader is somewhat familiar with general Simulink usage. First, we look at some useful Simulink methods. Then we give a complete explanation of the actual Simulink model. As the model contains numerous levels constructed with Simulink Subsystem blocks, analysis proceeds from the top down, beginning with the overall model before examining the Subsystems. In this fashion, the reader should learn the basic workings of the entire system before complicating matters with the internal details.

5.1 General Simulink Techniques

A few block combinations appear so frequently in the model that it is worth explaining them separately, before looking at the actual model. This will ease the explanation of later systems.

Methods of Switching Outputs

It is often desired to use different signals under different conditions. This is comparable to "if then else" statements and other logical operations in program scripts. Two methods of choosing outputs were found useful: enabled Subsystems and Multiport Switches.

A model with enabled Subsystems is shown in Figure 5.1. The Subsystems contain the Enable block, and run only when an enabling signal greater than zero is passed to them. Because an disabled system outputs zero, it is possible to sum the Subsystem outputs as long as only one is enabled at a time.

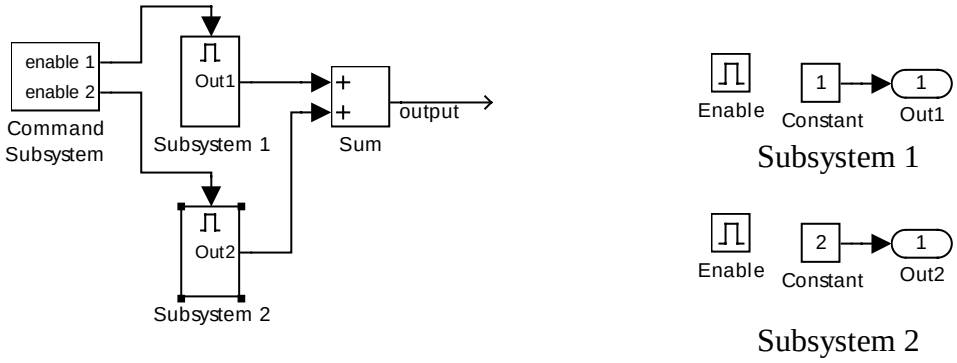

Figure 5.1: Simulink model Using Enabled Subsystems to Choose an Output. The Command Subsystem is used to generate two enabling signals, and should only enable one Subsystem at a time. Depending which system is enabled, the output is either 1 or 2.

An alternative is the Multiport Switch, shown in Figure 5.2. This block chooses an output according to the command signal. The command signal is rounded to the nearest positive integer, and the corresponding input is passed through.

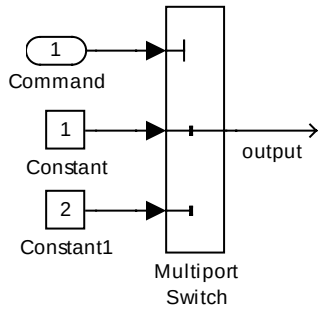

Figure 5.2: Multiport Switch Used to Choose an Output.

Both methods can be expanded to handle choices between more than two inputs. The enabled Subsystem method requires an enable signal for each input, and a Sum block that sums the signals from all of the Subsystems. The Multiport Switch can take more inputs, and requires only a single command signal.

Both methods were employed, as they each have benefits in certain situations. For switching of computationally intensive inputs, the enabled Subsystem method is superior, because only one Subsystem is run at a time. For switching simple signals, where Subsystems are unnecessary and computational efficiency is less critical, the Multiport Switch is convenient.

Signal Holding

Simulink provides an array of logical and relational operators that can be used to process information about the current system state. However, it is often important to know if an event has happened. For this it is desirable to generate a variable (usually Boolean) that is held at its maximum or minimum value over time.

Such a holding function can be created using a combination of a MinMax and a Unit Delay block, as shown in Figure 5.3. The MinMax block, usually used to compare two signals, is here used to compare a signal with is previous value. The Unit Delay block is necessary to prevent algebraic loops, and its delay period should be set to match the simulation or controller sample period. Such a block combination outputs the maximum or minimum value of the input. It should be noted that the initial output value depends on the initial output of the Unit Delay block (which can be specified) and of the input signal. As many signals in Simulink are initialized to zero, it is likely that the initial output will be zero. The Simulink Memory block can alternately be substituted for the Unit Delay block.

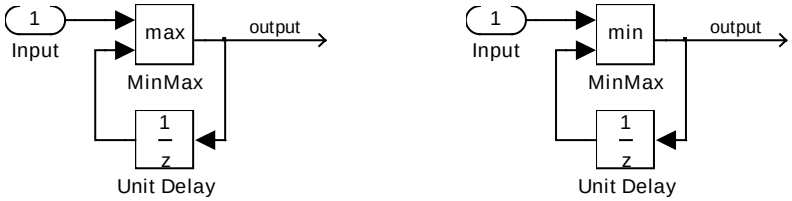

Figure 5.3: Simulink constructions that hold the maximum (left) and minimum (right) of the input values. Special attention must be given to the initialization values of the Input and Unit Delay block.

A slight variation of this system allows for the output to be reset to zero (Figure 5.4). The system then holds the maximum (minimum) value of the input since the last reset, where input below (above) zero produces a zero output.

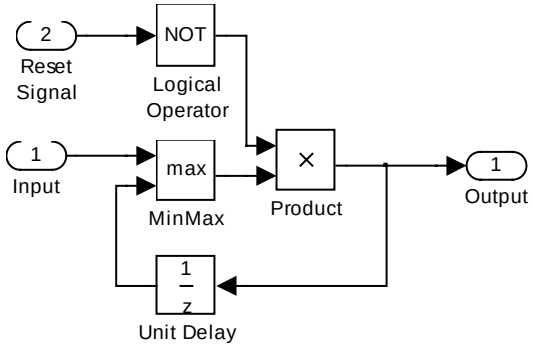

Figure 5.4: Hold Maximum with Reset. A reset signal forces the output to zero. The output is the maximum input value since the last reset. The output cannot go below zero.

Holds and hold-resets are used throughout the final Simulink model. Particularly, they are useful in generating command signals for switching, counting, and recording the height of the latest pendulum swing.

5.2 Overview of the Simulink Model

In this section, the entire Simulink model is outlined and explained. The model contains five levels of Subsystems. Analysis proceeds from the top level down. At each level, the function of all blocks and connections are described in moderate detail. Then the individual Subsystems are examined in the same manner.

Signals, blocks, and subsystems have been named to describe their function in the system. While this simplifies the model, it abstracts the original Simulink blocks. Specific Simulink blocks are best recognized in this report by their shape and appearance. Using the actual Simulink model file, blocks can be identified by their dialog boxes.

The system layout is shown in tree form in Figure 5.5, with branches indicating lower Subsystems. A quick reference guide to the Subsystem hierarchy and their functions is shown in Table 5.1.

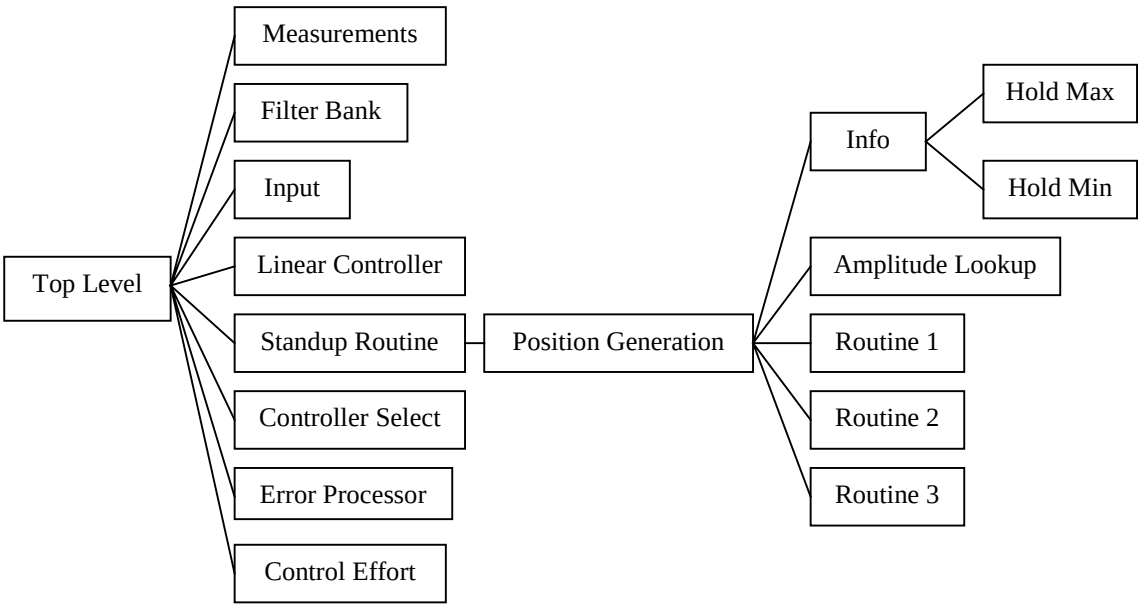

Figure 5.5: Tree Diagram of Subsystem Hierarchy.

Subsystem	Subsystems Contained	Function	Section	Page
Top Level	Measurements, Filter Bank, Input, Linear Controller, Standup Routine, Controller Select, Error Processor, Control Effort	Process top level signals from feedback to control effort.	5.2.1	38
Measurements	None	Read and calibrate feedback measurements from A/D.	5.2.2	40
Filter Bank	None	Filter measurements for noise.	5.2.3	41
Input	None	Set desired state for linear control.	5.2.4	42
Linear Controller	None	Implement matrix gain state feedback.	5.2.5	43
Standup Routine	Position Generation	Implement cart movement for standup routine.	5.2.6	43
Position Generation	Info, Amplitude lookup, Routine 1, Routine 2, Routine 3	Create the desired position waveform for the standup routine.	5.2.6	44
Info	Hold Max, Hold Min	Create information and switching command variables.	5.2.6	46
Hold Max	None	Remember the highest point of the last swing to the left.	5.2.6	46
Hold Min	None	Remember the highest point of the last swing to the right.	5.2.6	46
Amplitude Lookup	None	Determine movement amplitude for standup routine.	5.2.6	48
Routine 1	None	Start pendulum swinging.	5.2.6	49
Routine 2	None	Efficiently add energy to pendulum ($\|\theta\|<100°$).	5.2.6	50
Routine 3	None	Continue adding energy to pendulum ($\|\theta\|>100°$).	5.2.6	52
Controller Select	None	Choose between the standup routine and linear control.	5.2.7	53
Error Processor	None	Disable output according to certain error conditions.	5.2.8	54
Control Effort	None	Calibrate and output to DAC	5.2.9	55

Table 5.1: Subsystem Hierarchy and Function

5.2.1 Top Level

The top level system is shown in Figure 5.6. Signals flow from left to right, with feedback signals entering in the Measurements block, traveling through various routines and controllers, and exiting to the hardware through the Control Effort block.

Feedback signals are interpreted in the Measurements Subsystem. This Subsystem includes the dSpace A/D conversion blocks and all necessary measurement calibrations. The resulting output from this block represents the physical quantities in SI units. Cart displacement x is represented in meters, cart velocity xdot is represented in meters per second, and pendulum angle theta is represented in radians. The directions of these measurements are consistent with that described in the theoretical derivation and apparatus summary earlier in this report.

These measurements are processed in the Filter Bank Subsystem. An array of low-pass filters is used for noise reduction. The angular velocity of the pendulum, thetadot, and a repeating angular position variable, theta_mod, are created in this block. All signals are labeled with the '_f' extension to indicate that they have been filtered.

From here, the signals split. For linear control, the Input and Linear Controller Subsystems are used. For the standup routine, the Standup Routine Subsystem is used. Both generate an output signal representing the force, in Newtons, that should be applied to the cart. Signal switching is conducted using the method of enabled Subsystems, described in Section 5.1. The enable signal is generated by the Controller Select Subsystem, and the outputs are combined at the Sum block.

The Notch Filter is a Butterworth band stop filter for frequencies between 90 Hz and 120 Hz. This is necessary to eliminate a resonant mode involving deflection in the pendulum rod.

The Limiter block is used to protect the hardware. In this block, the requested force output is limited to 20 Newtons, limiting amplifier current to 4.6 amps.

The Error Processor Subsystem and Error Cutoff block form a safety system. The Error Cutoff block is simply a Multiport Switch, with zero as the second input. Under normal conditions, the Error Processor's Subsystem outputs a value of 1, thereby selecting the Limiter block output, which is the force command signal. Under certain error conditions, the Error Processor will send a command cutoff signal of 2, which will select the zero input and thereby disable the output and stop the cart.

The Control Effort Subsystem calibrates the Force signal, and writes to the DAC. The analog signal from the DAC is used to drive the current amplifier.

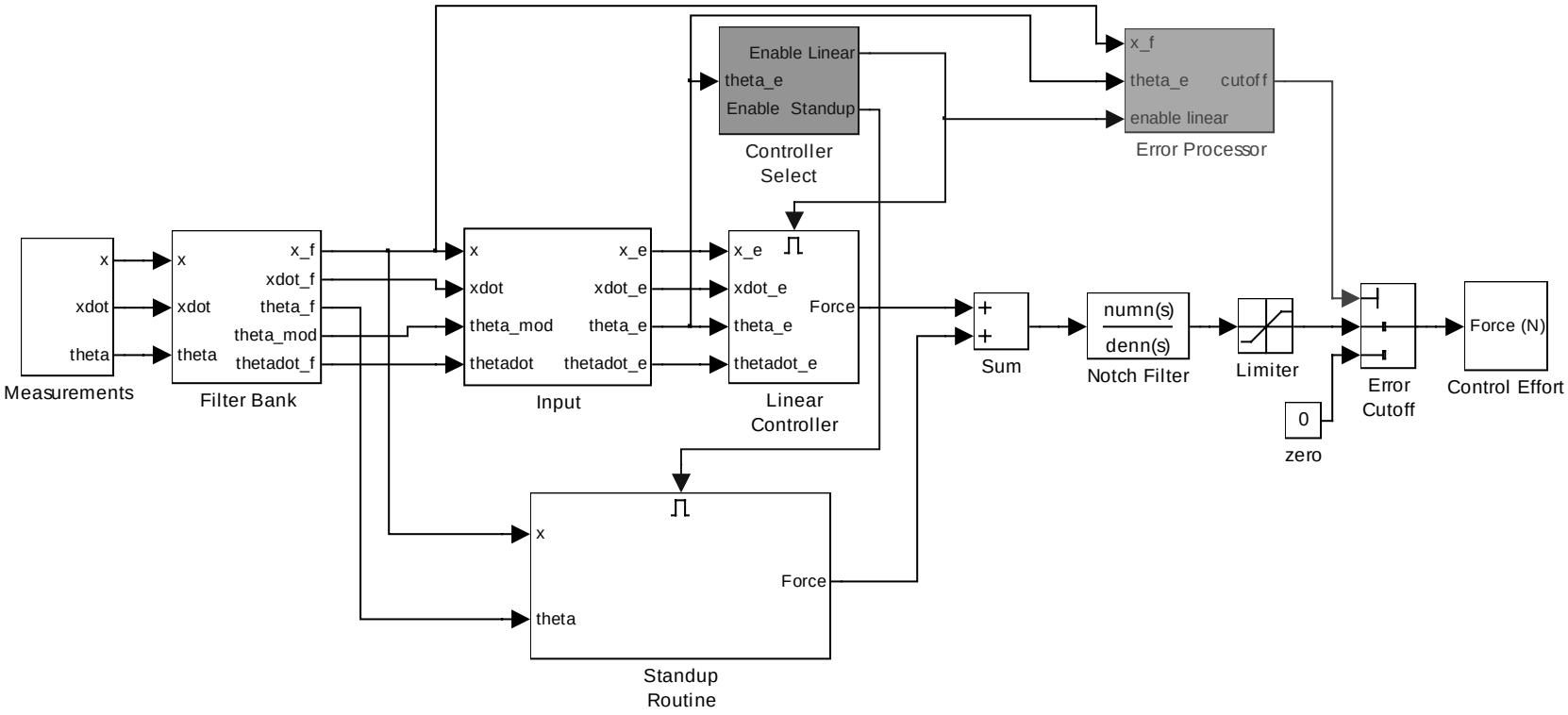

Figure 5.6: Top Level Simulink Model.

5.2.2 Measurements

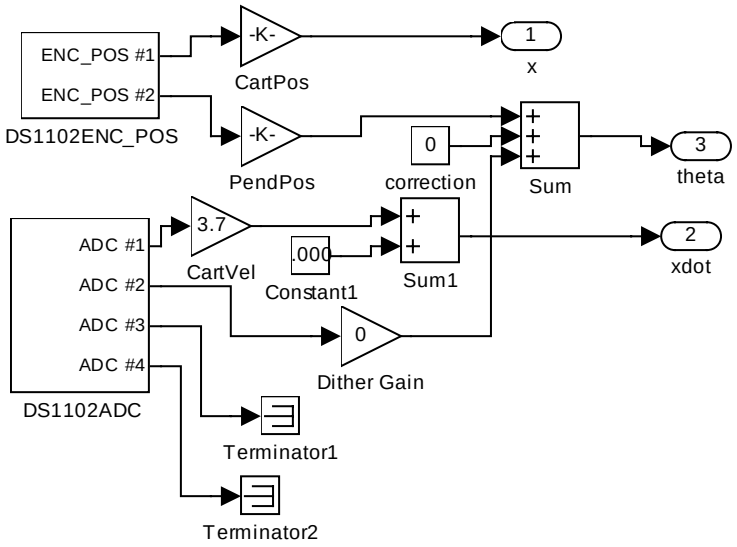

Figure 5.7: Measurements Subsystem.

The Measurements Subsystem (Figure 5.7) reads signals from the DS1102 interface card and performs the necessary calibrations. Output variables are in standard SI units: meters, radians, and meters per second.

Cart position x (via motor angle) and pendulum angle θ are measured using rotary quadrature, which are input using the DS1102ENC_POS block. Matlab calculates counts directly from the quadrature, but scales the full counter range to between –1 and 1. Cart Velocity is input using an A/D converter in the DS1102ADC block, which scales the input by 1/10.

Scaling gains (labeled CartPos, PendPos, and CartVel) must account for several effects. They must consider the calibration of the sensor, any gearing or circuitry gain involved, and the scaling of the DS1102, so that the output is in real units. The gain expressions, which convert the input block to a number representing the physical quantity in SI units, are given by

$$\frac{\text{x}}{\text{ENC\_POS\#1}} = \underbrace{\left(\frac{2^{23}\text{counts}}{1}\right)}_{\text{dSpace}}\underbrace{\left(\frac{1\text{rev}}{4000\text{counts}}\right)}_{\text{encoder}}\left(\frac{2\pi\text{rad}}{\text{rev}}\right)\underbrace{\left(\frac{0.04\text{meters}}{\text{rad}}\right)}_{\text{gearing}} \tag{5.1}$$

$$\frac{\text{theta}}{\text{ENC\_POS\#2}} = \underbrace{\left(\frac{2^{23}\text{counts}}{1}\right)}_{\text{dSpace}}\underbrace{\left(\frac{1\text{rev}}{4000\text{counts}}\right)}_{\text{encoder}}\left(\frac{2\pi\text{rad}}{\text{rev}}\right) \tag{5.2}$$

$$\frac{\text{xdot}}{\text{ADC\#1}} = \underbrace{\left(\frac{10\text{volts}}{1}\right)}_{\text{dSpace}}\underbrace{\left(\frac{1\text{volt}}{1\text{volt}}\right)}_{\text{circuit}}\underbrace{\left(\frac{1000\text{rpm}}{3\text{volts}}\right)}_{\text{encoder}}\left(\frac{1\text{min}}{60s}\right)\left(\frac{2\pi\text{rad}}{\text{rev}}\right)\underbrace{\left(\frac{0.04\text{m/s}}{\text{rad/s}}\right)}_{\text{gearing}} \tag{5.3}$$

A Dither Gain block was used in an earlier attempt to smooth the theta signal quantization. Ultimately, this gain was set to zero and the dither was not used. However, if desired, a dither signal could be brought in from ADC #2 on the DS1102.

All input values from the DS1102 are initialized to zero when the model is compiled. The state of the system at the time of compiling will therefore determine the zero value of all measurements. As designed, this model should be compiled with the cart stationary near the center of the track with the pendulum down and motionless.

5.2.3 Filter Bank

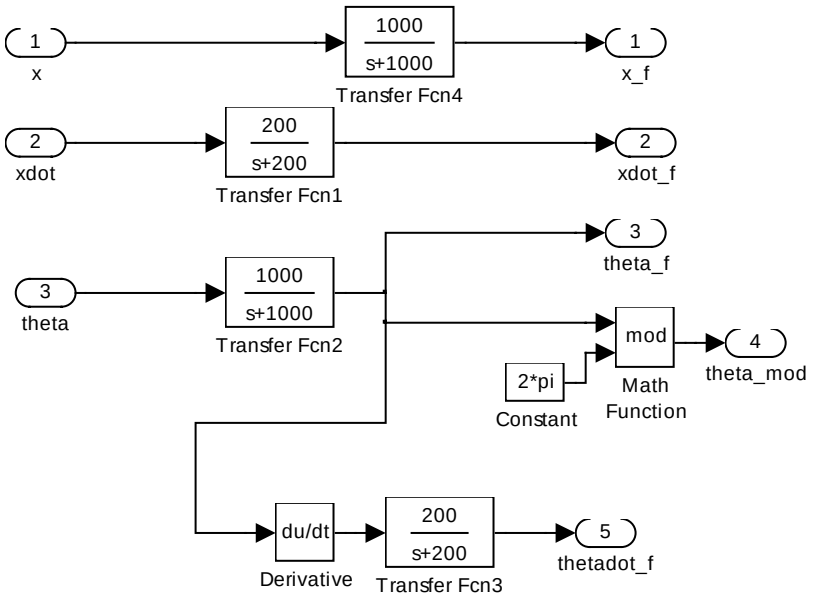

Figure 5.8: Filter Bank Subsystem.

Feedback signals are processed in the Filter Bank Subsystem (Figure 5.8). An array of low-pass filters is implemented using Transfer Fcn blocks. These filters are tuned to reduce noise in the feedback signals. All of these filters have unity DC gain, and have break frequencies significantly above system crossover, so they do not affect stability.

Pendulum angular velocity is calculated using a Derivative block. This is filtered a second time to further reduce noise in the numerical derivative.

A repeating theta variable theta_mod is created using the mod function in a Math Function block. While the encoder measuring theta will measure multiple revolutions, the mod function subtracts these out, producing an output that is always between 0 and 2π. This variable is used for the linear controller, so that extra rotations do not produce an error. This is important because the pendulum can reach vertical in either the clockwise or counterclockwise direction.

5.2.3 Input

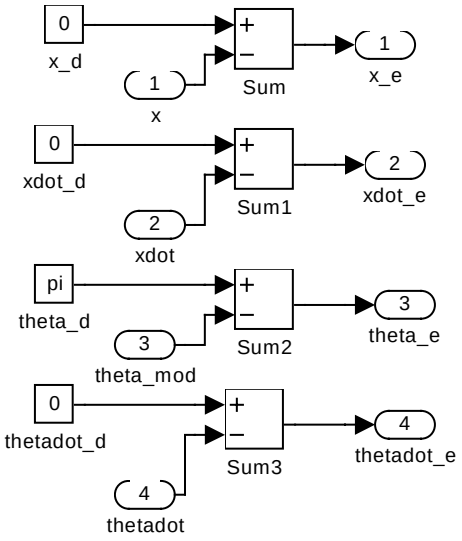

Figure 5.9: Input Subsystem.

The Input Subsystem (Figure 5.9) explicitly shows the desired values of the state variables that the linear controller will stabilize about. These inputs x_d, xdot_d, theta_d, and thetadot_d are compared with the feedback measurements, and the error is determined. The meaning and units of these setpoints is determined by the feedback signal calibration.

The inputs theta_d and thetadot_d must be set to π and 0, respectively, so that the system linearization is valid. For general operation, the desired value is zero for x_d and xdot_d. Input waveforms can be used instead of constants to prescribe movements, within the limitations of the system response.

5.2.4 Linear Controller

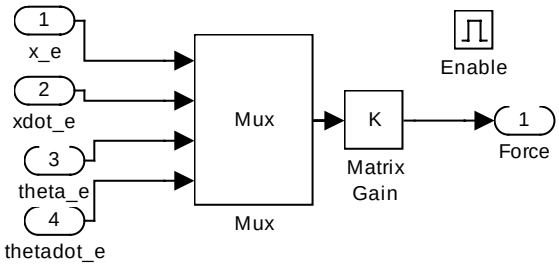

Figure 5.10: Linear Controller Subsystem.

The Linear Controller Subsystem (Figure 5.10) is used to implement the control algorithm described in Section 4.1. State variable errors are combined into a vector using the Mux block. This vector is then multiplied by K, a gain row vector determined using lqr in a Matlab script, which is included in Appendix A. The result is a Force signal in Newtons that drives the cart when under linear control. This system is enabled by the Controller Select Subsystem, so it does not run during the standup routine.

5.2.5 Standup Routine

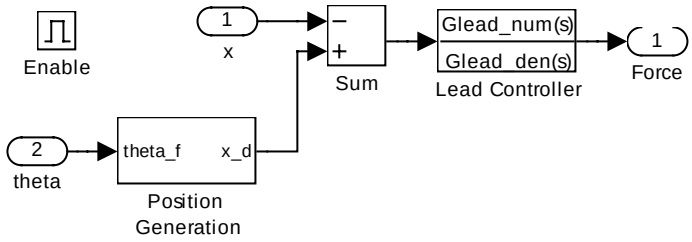

Figure 5.11: Standup Routine Subsystem.

The Standup Routine Subsystem (Figure 5.11) implements the controls that bring the pendulum to an upward vertical position. This Subsystem is enabled by a command from the Controller Select Subsystem, so that it is only executed during the standup routine.

This level contains basic cart position control. The measured cart position x is compared with a desired position x_d. The Lead Controller, which is the Simulink implementation of the controller described in Section 4.2.1, creates a Force signal based on this error. The Force signal is in Newtons, and is the force that should be applied to the cart.

The desired position x_d is determined by the Position Generation Subsystem. This Subsystem looks at the current pendulum state, and determines an appropriate cart trajectory. Many algorithms would require both theta and theta_dot as inputs, but this one does not, for reasons to be described later.

Position Generation

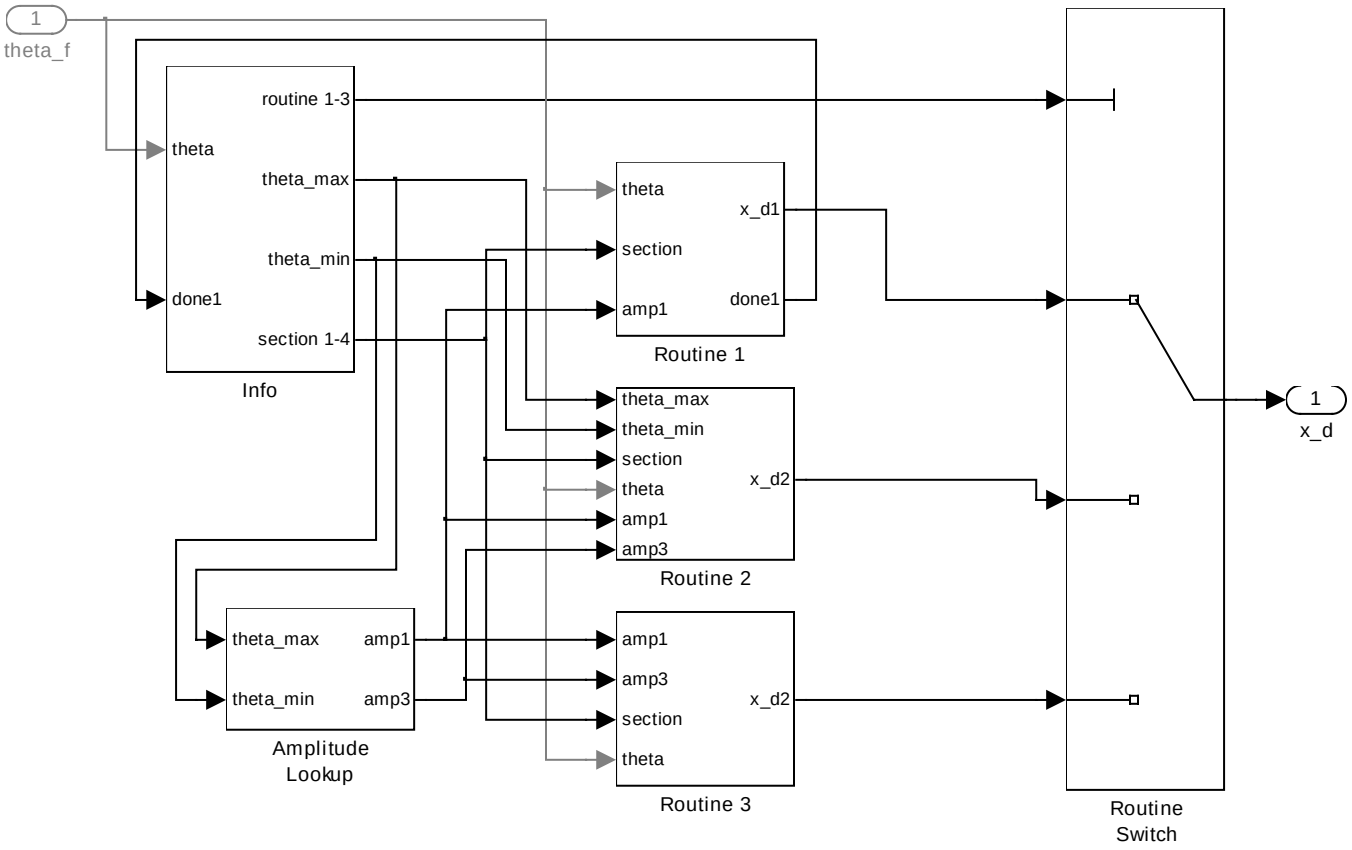

Figure 5.12: Position Generation Subsystem.

The Position Generation Subsystem is the primary component of the standup routine (Figure 5.12). Given the pendulum angle, this Subsystem determines a variety of information about the pendulum motion and calculates a movement trajectory.

The Info Subsystem is responsible from creating the command signals and information variables necessary to conduct the cart trajectory calculations. Four variables are generated: routine 1-3, theta_max, theta_min, and section 1-4. The routine variable is a command signal for the Routine Switch for output selection. The variables theta_max and theta_min are the highest points to the left and right, respectively, of the last pendulum swing. Because of the coordinate system used, theta_max is a positive angle measured in

radians and theta_min is a negative angle measured in radians. The section 1-4 variable represents the section of the swing that the pendulum is in, determined by both position and direction of motion, as shown in Figure 5.13. Sections 2 and 4 are active regions, where the cart should be moved to increase the energy of the system. Sections 1 and 3 are inactive, where the cart should remain at a fixed position.

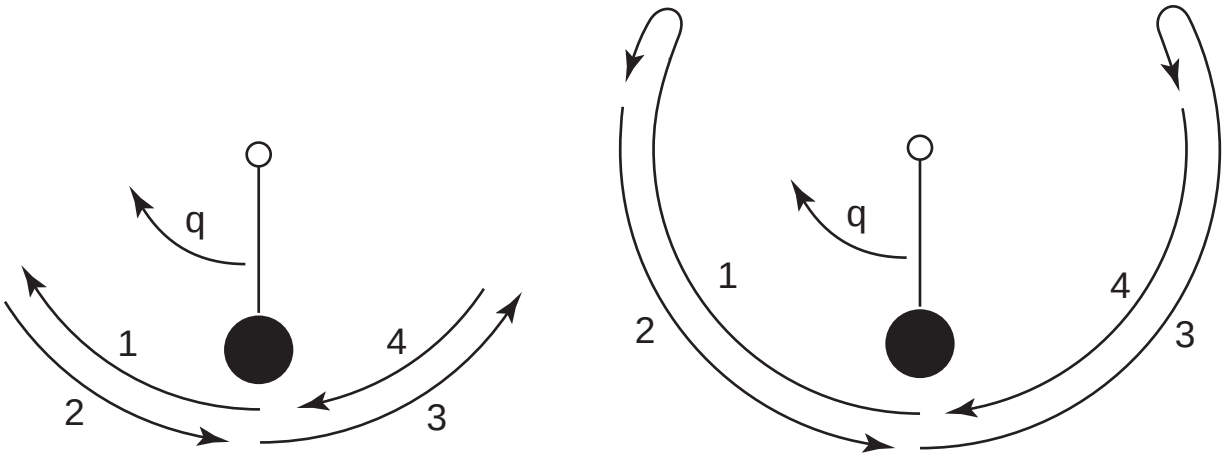

Figure 5.13: Swing section for low amplitude (left) and high amplitude swings (right), as assigned by the section 1-4 variable from the Info Subsystem.

The Amplitude Lookup Subsystem is used to determine the magnitude of the cart movement, in meters. As the pendulum swings higher, amplitude is reduced so that the pendulum approaches vertical gradually and without significant velocity. The theta_max and theta_min variables are used alternately for this calculation, depending on the direction that the pendulum is swinging.

Three routines, Routine 1, Routine 2, and Routine 3, determine the cart trajectory. They correspond roughly to the three steps described in Section 4.2.2 for the standup routine. Routine 1 consists of two abrupt changes in cart position, and is designed to get the pendulum started. Routine 2 uses smooth movement to add energy efficiently for pendulum swings below 100°. Routine 3 also uses smooth movement and is designed for higher pendulum swings. All of these routines take information from the Amplitude Lookup Subsystem in order to calculate trajectory magnitude. In addition, Routine 2 uses the theta_max and theta_min variables from the Info Subsystem in order to match the cart movement to the current pendulum height. Output from these routines is selected using the Routine Switch and the routine 1-3 command signal from the Info Subsystem.

Info

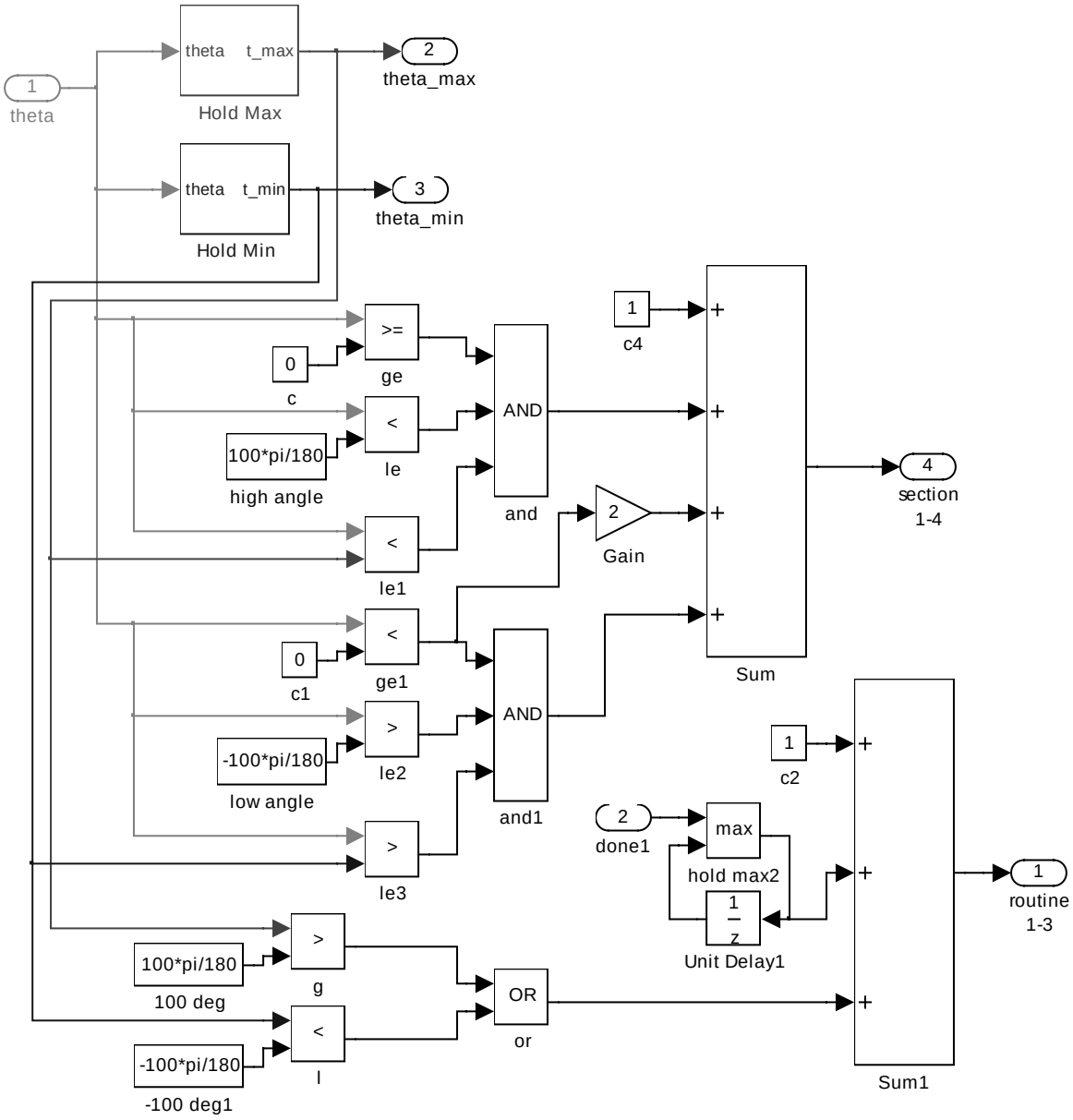

Figure 5.14: Info Subsystem.

The Info Subsystem (Figure 5.14) generates four variables, which are used to calculate and select output trajectories.

The variables theta_max and theta_min are used to remember the highest point of the last swing. These calculations are done in the Hold Max and Hold Min Subsystems (Figure 5.15), which involve the hold reset mechanism that was discussed in Section 5.1.

Hold Max holds the greatest positive value of theta, representing a swing to the left, while Hold Min holds the greatest negative value of theta, representing a swing to the right. Both measurements are reset to zero in between swings, at a time when they are not being used for computation. As Hold Max represents the highest point left, it is reset when the pendulum crosses zero moving left, so each time a new maximum is recorded. Likewise, Hold Min is reset when the pendulum crosses zero moving right. The reset point is specified by the conditions in the Hit block, which uses a ODE solver to detect crossing of a specific value in a specific direction of the input variable. When output of the Hit block goes high, output is forced to zero through the Not and Product blocks.

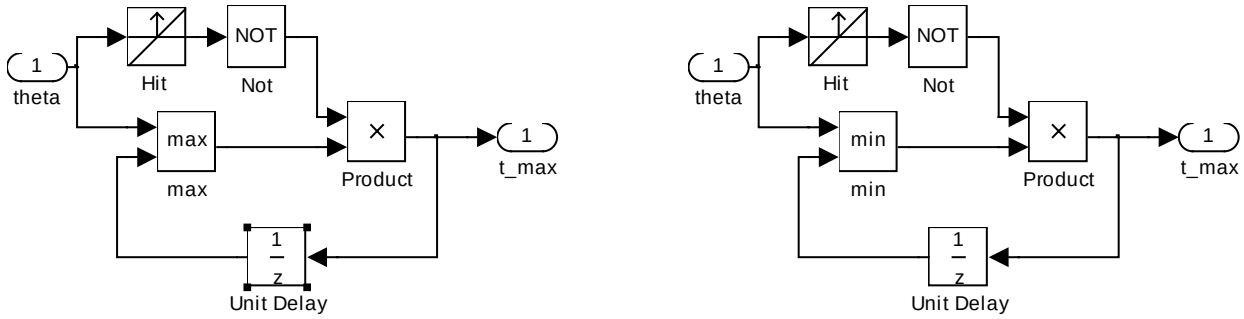

Figure 5.15: Hold Max (left) and Hold Min (right) Subsystems

The section 1-4 variable is an integer from 1 to 4 representing the present swing section of the pendulum, as shown earlier in Figure 5.13. Output can be understood from the 4-input Sum block. This block adds integers that check a series of logical conditions. The first Sum input insures that Sum output is 1 when none of the conditions are met. The third sum input adds 2 if the pendulum is on the right side ($\theta<0$). The second and fourth Sum inputs add 1 if the pendulum is in a specific part of the swing, and cannot be satisfied simultaneously.

The second Sum input is true (true has a value of 1 in Simulink) if the pendulum is in swing section 2. This checks that $0°<\theta<100°$ and that the pendulum is swinging downwards. Swinging downwards is determined by checking if theta is less than theta_max. Because of the reset point in the theta_max variable, theta equals theta_max while the pendulum swings upwards in this region.

The fourth Sum input is similar to the second, but is true if the pendulum is in swing section 4. This involves checking that $-100°<\theta<0°$ and that the pendulum is again swinging downwards. In this case, swing downward is determined by checking that theta is greater (more positive) than theta_min.

An alternative method for determining that the pendulum is swinging downward would have been to evaluate the sign of pendulum angular velocity theta_dot. In many aspects, both methods would produce equivalent results. A difference occurs when cart acceleration causes pendulum velocity to reverse in mid swing. This can cause jumps in

the section 1-4 variable, leading to discontinuities in the desired position command. The method chosen is more tolerant, but still affected. Moreover, it is important to use cart trajectories that do not reverse the pendulum movement.

The routine 1-3 variable is also determined by summing logical signals. The first input to Sum1 begins output at 1. The second input holds the maximum value of done1, a variable generated inside the Routine 1 Subsystem that indicates that Routine 1 is finished. The third input checks that the pendulum is swing above 100°. But checking both theta_max and theta_min, the periodic resets of these variables do not cause routine 1-3 to drop back to a value of 2. However, the routine 1-3 variable can drop from 3 to 2 if the swing amplitude is for some reason reduced below 100°, perhaps through a disturbance.

Amplitude Lookup

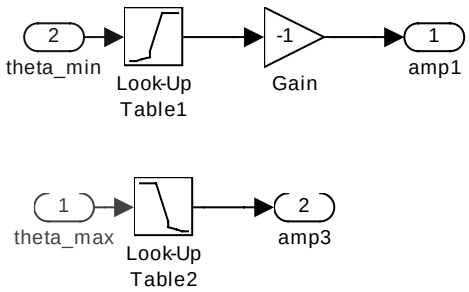

Figure 5.16: Amplitude Lookup Subsystem.

The Amplitude Lookup Subsystem (Figure 5.16) calculates the amplitude of cart movement, based on the current maximum height that the pendulum has reached. The variable amp1 represents the position where the cart should be in swing section 1, at the far left. The variable amp3 represents the position where the cart should be at in swing section 3, at the far right of the track. (Recall that the cart should be stationary in swing sections 1 and 3.) The theta_max and theta_min variables are used alternately, so that this amplitude is based on the height of the last swing.

Each amplitude is calculated using a Look-Up Table block. This block allows for a tabular relation between input and output, and performs linear interpolation between entries. The tables are symmetric, with sign differences caused by the sign of the pendulum angle and cart position. Table 5.2 show the corresponding input and output for these blocks. Also, the Look-Up Table block itself depicts a general plot of this relation.

The Look-Up Table values were tuned by hand until the desired standup routine performance was achieved. Test runs were conducted, beginning with amplitude values that were too low throughout the table, so that the pendulum would not reach the inverted

position. The maximum angle that the pendulum reached was noted, and the corresponding table entry was increased. Then the test was rerun, the new maximum height observed, and this corresponding table entry was increased. Adjustment gradually moved towards the right of Table 5.2. Values were adjusted until each cart movement added a small increment in pendulum height. If the pendulum was ever thrown to vertical too quickly from a particular angle, the corresponding table entry for that angle was reduced. Repeated observations, sometimes with disturbances created, increased the robustness of the final tuning.

theta_min (°)	0	-60	-110	-111	-145	-165	-180	-190
amp1 (mm)	-80	-80	-20	-10	-4.7	-3.4	-2.8	-2.8

theta_max (°)	0	60	110	111	145	165	180	190
amp3 (mm)	80	80	20	10	4.7	3.4	2.8	2.8

Table 5.2: Corresponding values of input and output using the Lookup Tables in the Amplitude Lookup Subsystem.

Routine 1

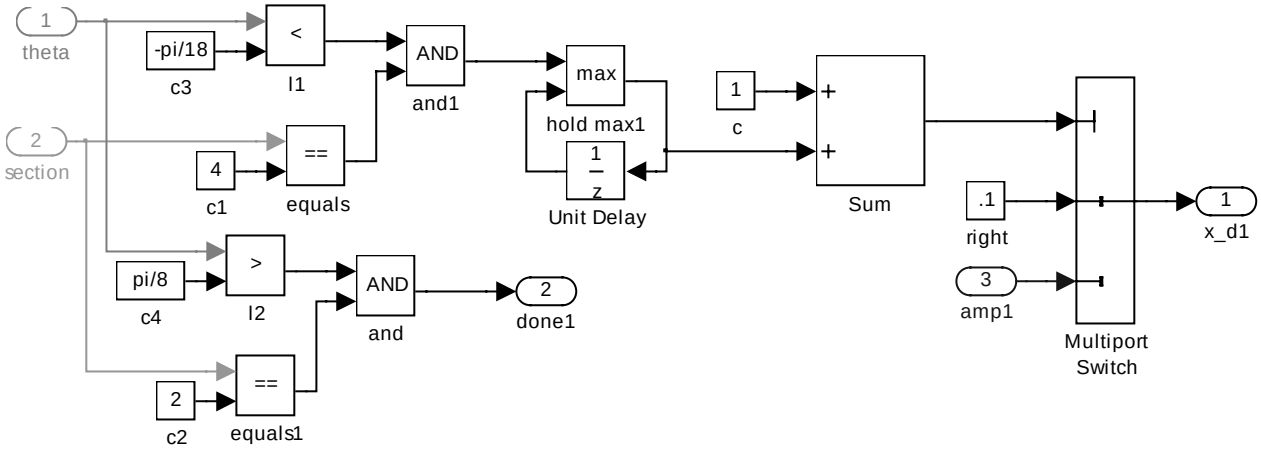

Figure 5.17: Routine 1 Subsystem.

The Routine 1 Subsystem (Figure 5.17) outputs cart position commands that start the pendulum swinging. This involves jumping to the right, then left, and then passing control on to the next routine.

Output begins with the Sum output equaling 1 and the Multiport Switch choosing 0.1 (right) as the desired position x_d. The desired position switches to a left position, determined by the Amplitude Lookup Subsystem, when a condition is true: The

pendulum must be in swing section 4 (moving downwards) at an angle of $\theta<-\pi/18$. The initial cart jump to x=0.1 happens immediately after compiling or as soon as the hardware is powered up. Checking that the pendulum in swing section 4 causes the cart to wait until the pendulum is swinging downward again before moving left. Checking that $\theta<-\pi/18$ ensures that this condition is not met prematurely as wind or table motion cause the pendulum to rock slightly before the algorithm starts.

Further logic determines the done1 variable, which is used in the Info Subsystem to calculate the routine 1-3 variable. This algorithm waits for the pendulum to have swung back over to the left ($\theta>\pi/8$) and begin moving downward (swing section 2). Recall that the blocks in the Info Subsystem hold the maximum value of done1, so control is never passed back to this routine.

Routine 2

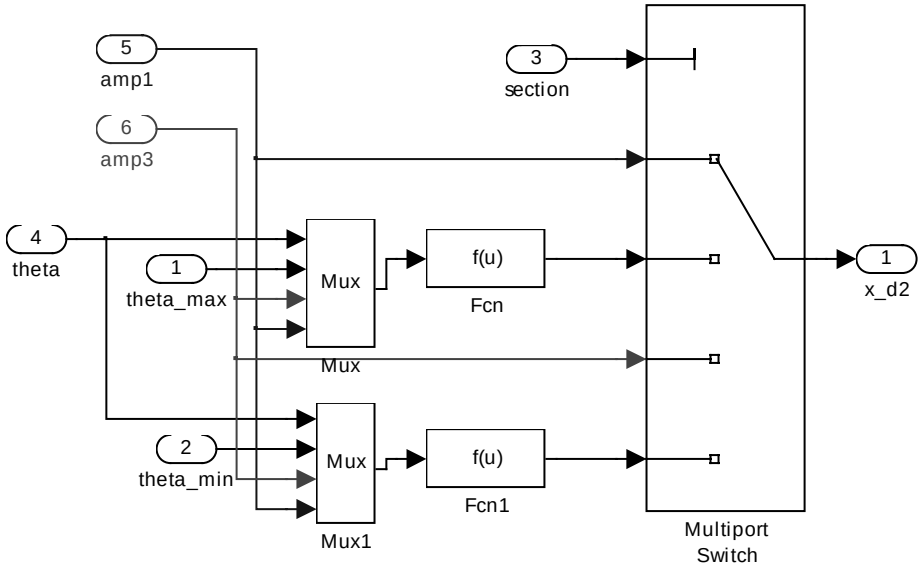

Figure 5.18: Routine 2 Subsystem.

The Routine 2 Subsystem (Figure 5.18) is used following Routine 1 and raises the pendulum up to 100°. This routine uses a smooth trajectory between amp1 and amp3, synchronized to match pendulum motion between the high point of the last swing and 0°.

Output is selected with a Multiport Switch. The four switch inputs correspond to inputs for the four different swing sections. For the inactive sections 1 and 3, the cart should remain stationary at a position of amp1 and amp3, respectively. For the active sections, a suitable motion should be prescribed, as was discussed in Section 4.2.2. Recall the trajectory described by Equation 4.20:

$$x_d = A\cos\left(\frac{\theta}{\theta_{max}}\pi\right) \quad \text{for} \quad 0 < \theta < \theta_{max} \text{ and } \dot{\theta} < 0 \tag{5.4}$$

This can be adapted using amp1, amp3, theta_max, and theta_min. A discontinuity is the prescribed cart position function is caused between certain swing sections if amp1 and amp3 have different values. This will be the case because amp1 and amp3 are reset at different times and are based on the pendulum height at different sides of the swing. A solution is to fade linearly between amp1 and amp3 as a function of theta. The result, with Simulink variable names substituted for the theoretical variables in Equation 5.4 is shown below. The terms in brackets are the expressions for amplitude A. The trajectories x_d_2 and x_d_4 for swing sections 2 and 4, respectively, are given by

$$x\_d_2 = \left\{-\text{amp1}\left(\frac{\text{theta}}{\text{theta\_max}}\right) + \text{amp3}\left(1 - \frac{\text{theta}}{\text{theta\_max}}\right)\right\}\cos\left(\frac{\text{theta}}{\text{theta\_max}}\pi\right) \tag{5.5}$$

$$x\_d_4 = \left\{-\text{amp1}\left(\frac{\text{theta}}{\text{theta\_max}}\right) + \text{amp3}\left(1 - \frac{\text{theta}}{\text{theta\_max}}\right)\right\}\cos\left(\frac{\text{theta}}{\text{theta\_min}}\pi\right) \tag{5.6}$$

The negative signs in front of amp1 are necessary because the value is calculated as a negative number by the Amplitude Lookup Subsystem. Referring back to Figure 5.18, the trajectories given by Equations 5.5 and 5.6 are implemented by first combining variables into a vector and then using a Fcn block to express a vector function.

Routine 3

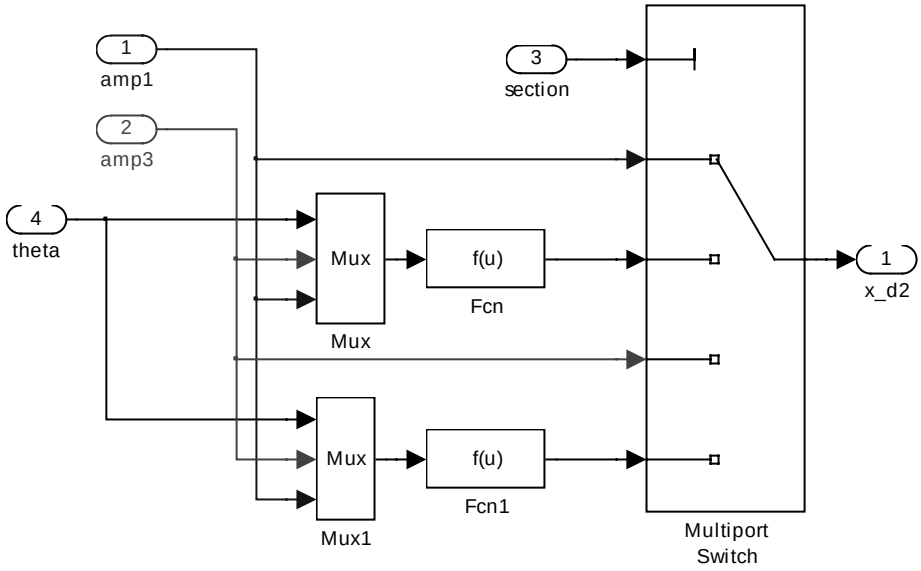

Figure 5.19: Routine 3 Subsystem.

The Routine 3 Subsystem (Figure 5.19) performs a similar function as Routine 2, but is designed for high pendulum swings. For large swings, it is no longer necessary, in fact inefficient, to distribute cart movement between the pendulum angles of theta_max and 0°. Instead, movement should be distributed between some fixed high angle (100° was used) and 0°. This involves a simpler calculation with a constant θ_{max} in the denominator of Equation 5.4. Amplitude blending and Simulink implementation is very similar to that of Routine 2.

Both Routines 2 and 3 exhibit an undesirable property: an effective delay in amplitude adjustment. Because amplitude is blended between amp1 and amp3, it is calculated from the last two swing high points (left and right) instead of only the most recent. The problem stems from the fact that all cart movements are centered around x=0, so the total displacement depends on both the new and previous desired positions. If the Amplitude Lookup Subsystem calculates an amplitude that would be appropriate for the total cart displacement, then this effect creates an error.

This error decreases the robustness of the standup routine, and adds difficulty to the tuning of the amplitude Look-Up Tables. The sign of the error depends on whether the maximum pendulum swing height is increasing or decreasing on successive swings. If the pendulum is gaining height on each swing, amplitude is gradually being reduced, and the delay error creates extra cart movement. If the pendulum begins losing height, amplitude is being increased, and the delay error creates a reduction in amplitude.

The delay error was easily overcome. If the pendulum is raised gradually, this error is insignificant. However, different schemes could be implemented where cart movements were not centered around x=0. This would require either a memory of the last position, or prescribing velocity and integrating. The difficulty with these methods is that the cart must remain between the end stops of the track.

5.2.6 Controller Select

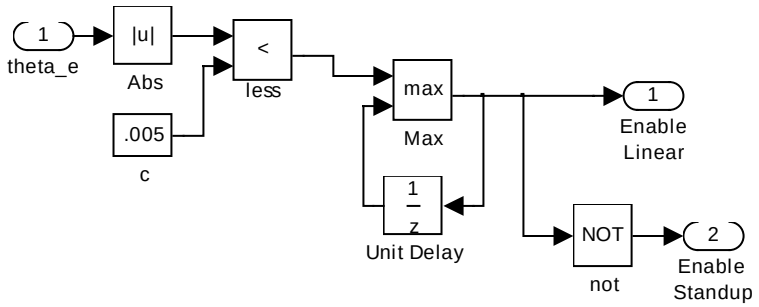

Figure 5.20: Controller Select Subsystem.

The Controller Select Subsystem (Figure 5.20) chooses between the Standup Routine and Linear Controller Subsystems by creating two enabling signals. This system checks that $|\theta_{error}|<0.005$, meaning that the pendulum is very close to vertical. At start, the pendulum is down, so this condition is not met, giving Enable Linear a value of zero and Enable Standup a value of one. The standup routine is thus run, until it brings the pendulum to vertical and the condition is met. Once this condition is met, it will be held using the Max and Unit Delay configuration, enabling the Linear Controller and disabling the Standup Routine.

The condition for switching to linear control could be improved using more information. In addition to pendulum position, calculations could involve pendulum velocity or energy. Smarter algorithms could consider the current position of the cart, so that the track could best be utilized for stabilization. These ideas, however, are unnecessary for operation.

It should be noted that for theta_e an absolute value less-than block combination was used instead of an equality. This is because with digital processing there is no guarantee that the pendulum position error is actually equal to zero in any given sample period, even if the error crosses zero. As alternative would be to use a Hit Crossing block, which uses an ODE solver to detect a crossing. Unfortunately, a Hit Crossing in this Subsystem consistently registered a hit at compile time. This was probably because theta_e, like most variables, was initialized to zero. The extra hit in this configuration caused the selection to jump directly to the Linear Controller without any Standup Routine.

5.2.8 Error Processor

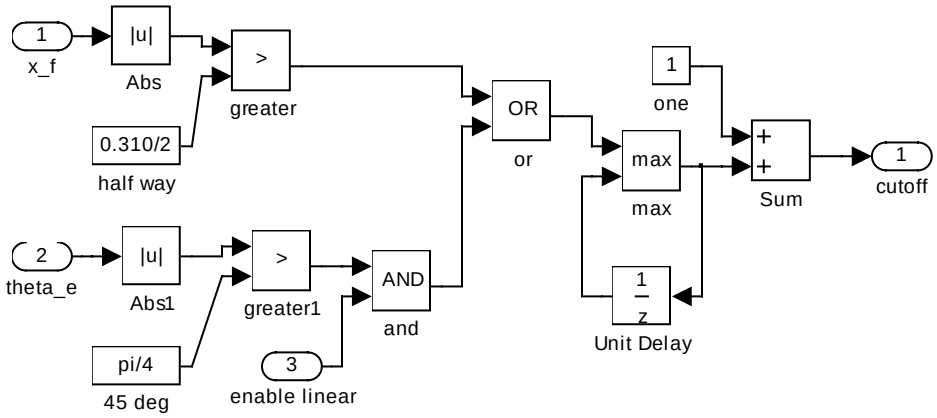

Figure 5.21: Error Processor Subsystem.

The Error Processor Subsystem (Figure 5.21) is not directly involved in the balancing of the pendulum, but is important for safety and hardware protection. Program malfunctions often produce unstable response, which drives the cart violently against the end stops. The function of this subsystem is to evaluate several error conditions, and to cutoff output if these condition are met.

Review of the top level model (Figure 5.6) shows that the cutoff signal from this block is the command signal to the Error Cutoff switch. A command signal of 1 allows the Control Effort signal from either controller to be passed through to the output. A command signal of 2 disconnects the controller signal and instead passes a value of 0 to the output.

The error conditions in the Subsystem can be expanded. Here, two primary conditions have been included. The first condition checks to see if the cart is near the end of the track. The second condition determines if the pendulum is under linear control and is at a large angle from upward vertical. If any error condition is met (this can be expanded to handle more error condition by using an Or block with more inputs), the error signal is held high, cutoff equals 2, and system output is cut off.

Resetting errors in this model requires recompiling. This is undesirable, and can be corrected using a hold-reset configuration instead of the simple hold. The reset signal could be driven externally, using either a physical reset switch connected to a digital I/O pin on the dSpace board or a virtual switch from Cockpit or Trace software to interrupt the simulation. For implementation, certain other variables, such as routine 1-3 and done1, would have to be reset to an appropriate starting point.

5.2.9 Control Effort

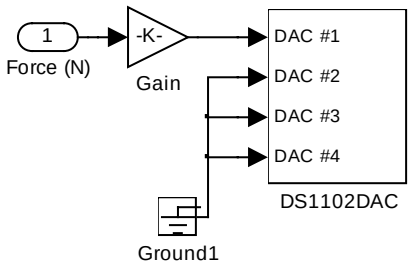

Figure 5.22: Control Effort Subsystem.

The Control Effort Subsystem calibrates and outputs the Control Effort to the DS1102. The input Force refers to the force, in Newtons, that should be delivered to the cart. For calibration, a gain is necessary which counteracts gains in the dSpace DAC write, current amplifier, and motor.

$$K = \underbrace{\left(\frac{1}{10\text{volts}}\right)}_{\text{dSpace}}\underbrace{\left(\frac{1\text{volt}}{2\text{amps}}\right)}_{\text{amplifier}}\underbrace{\left(\frac{1}{4.25\text{N/amp}}\right)}_{\text{motor}} \tag{5.7}$$

6. Recommendations

Although the current system is fully function, several improvements can be made. These are outlined here for the benefit of anyone continuing this project, and to increase the general understanding of the subject. Many of these recommendations are also covered in specific sections of this report.

When cart position control was designed for the standup routine (Section 4.2.1), a significant discrepancy between theoretical predictions and experimental results was discovered. This was attributed to unmodeled effects, and was overcome by adjusting gain to match the experimental response. However, design of the linear controller is based completely on theoretical behavior, and involves many of the parameters found to be in error for cart position control. While the linear controller is functional, it is suspected that improvements could be made if this discrepancy were resolved. It is therefore recommended that additional behavior, including dry friction, belt compliance, and motor and amplifier dynamics, be considered. Alternatively, some of these effects could be reduced or eliminated.

The physical hardware is unsuccessful at constraining the pendulum to planar motion. In fact, a mode of vibration exists in which the pendulum deflects in a direction parallel to its axis of rotation. This unfortunately couples with cart motion, despite their orthogonality. This causes a resonance at a frequency of about 9.5 Hz, producing significant jitter in linear control. This mode should be eliminated by replacing the pendulum joint and rod with a structure that is rigid in this direction.

The Simulink model could be improved in several ways, which were discussed at appropriate points in Section 5. A significant improvement would be to allow for resets of the model without recompile. The current model runs the routine only once per compile, and will shut off when errors occur. It would be desirable to have an external reset switch that would restart the routine.

A second modification of the Simulink model involves the cart motions in the standup routine. Currently, all motion is centered about x=0. This is undesirable because the total magnitude of each movement depends on the starting position. The solution is to prescribe cart movement magnitudes, instead of prescribing the next desired location. This can be accomplished either be prescribing velocity and integrating, or by calculating the new desired position relative to the previous one. Implementing such a scheme would improve standup routine reliability and allow for faster standups.

A final recommendation is to use a reduced-order observer for the determination of pendulum angular velocity $\dot{\theta}$. Such an implementation would eliminate the numerical derivative that is used in the current model. It must be noted that this observer can only be used for linear control, where the system can be modeled using a LTI system. Also, the observer must be initialized correctly at an appropriate time.

Appendix A: Matlab Scripts and Function Files

File: setup.m
Description: This file contains model parameters and generates controllers for linear state control and for cart position control. Also included is the notch filter and several general system parameters. The calculated variables are used in the simulations and in the real-time Simulink-based controllers.

```
% setup.m
clear all; close all;

%System Parameters
m = 0.15;
l = 0.314;
M = 1.3 + 3.6e-4/(0.04^2);%includes motor inertia
I = 7.38e-5;%ignored
g = 9.80665;
km = 4.4825;  % N/A
kdac = 20;    % scaled DAC write, includes Amp gain
b = 20;
Mt=M+m;

%State Space Representation
A = [0 1 0 0; 0 -b/M -m*l/M 0; 0 0 0 1; 0 b/(M*l) Mt*g/(M*l) 0];
B = [0; 1/M; 0; -1/(M*l)];
C = [1 0 0 0; 0 1 0 0; 0 0 1 0];
D = [0; 0; 0];

sysp = ss(A,B,C,D);

Q = zeros(4,4);
Q(1,1) = 1;
Q(2,2) = 1e-4;
Q(3,3) = .5;
Q(4,4) = 1e-4;
R = 1e-5;

K = lqr(A,B,Q,R);
%K = [-316.2278 -186.3904 -479.6232  -72.6099];

%Generate Plots
c_poles=eig(A-B*K)
poles=eig(A);

figure(1);
sys_cl = ss(A-B*K,zeros(4,1), [eye(4); -K], zeros(5,1));
x0 = [0; 0; 0.01; 0];
initial(sys_cl,x0);

figure(2);
plot(real(poles),imag(poles),'kx',real(c_poles),imag(c_poles),'k.');
legend('Original', 'Controlled');
grid;
```

```
title('System Poles');
xlabel('Real Axis');
ylabel('Imaginary Axis');

% create a notch filter at the resonant frequency
[numn,denn] = butter(3,2*pi*[90 120],'stop','s');

%Controller (lead), label 2, Values for lag are included, but not used.
Ka=2;
Gsys=tf([0 0 1],[M+m b 0]);
wc=40;
wctr=wc;
Kc=10*274.5909;
a1=5;

t1=1/(wctr*sqrt(a1));
t3=1/(wc/50);%lag
a3=10;
Glead_num=Kc*[a1*t1 1];
Glead_den=[t1 1];
Glag_num=a3*[t3 1];
Glag_den=[a3*t3 1];
Gc_num=conv(Glead_num,Glag_num);
Gc_den=conv(Glead_den,Glag_den);
Glead=tf(Glead_num,Glead_den);
Glag=tf(Glag_num,Glag_den);
Gc2=Glead*Glag;
L2=Glead*Glag*Ka*km*Gsys;

%For controller design
[m p]=bode(L2,wc);
1/m

%standup params (many were ultimately not used)
c=0.07;
theta_high=100*pi/180;
sample_T=0.0005;
sat=40/Kc;%not used
```

File: swing.m
Descirption: Numerical simulation of cart and pendulum movements. Cart movement is defined by Equation 4.20 in this report. This script calls the function 'ydot', included, which defines the system differential equations.

```
%swing_r
global A t_max;

t_max=100*pi/180;%Angle at which cart motion begins
%motion ends at theta=0
A=.06;%cart motion amplitude
y0=[160*pi/180 0];%initial angle and velocity of pendulum

[t y]=ode45('ydot',[0 1],y0);%use ODE solver for theta and theta_dot
theta=y(:,1);
```

```
theta_deg=theta*180/pi;

%calculate cart position x from theta
n=size(t);
x_d=zeros(n);
for i=1:n
   if and(theta(i)<t_max, theta(i)>0)
      x_d(i)=A*cos(theta(i)*pi/t_max);
   else if theta(i)>t_max
         x_d(i)=-A;
      else
         x_d(i)=A;
      end
   end
end

%plot x and theta vs. time
plot(t,theta_deg,'-',t,1000*x_d,'--');
title('Pendulum Response Under Prescribed Cart Movement');
xlabel('Time (sec)');
ylabel('Pendulum Angle (deg) and Cart Position (mm)');
legend('Pendulum Angle', 'Cart Position')
```

File: ydot.m
Description: Matlab function defining the system differential equation General dynamics were derived in Equation 4.17, where $\ddot{x}$ is defined by 4.22. Because *x* is a function of θ, differential analysis involves only θ and $\dot{\theta}$.

```
function dydt=ydot(t,y);

global A t_max;%use values set in the swing.m file

theta=y(1);
theta_dot=y(2);

g=9.8;
l=.314;

ddt_theta=theta_dot;

if and(theta<t_max, theta>0)%Cart moving, ddt_theta_dot involves cart
acceleration
   ddt_theta_dot=(-(A*pi^2/t_max^2)*cos(theta*pi/t_max)*theta_dot^2-
g*tan(theta) )/( l/cos(theta)+(A*pi/t_max)*sin(theta*pi/t_max) );
else%cart stationary, x_double_dot=0
   ddt_theta_dot=-g*tan(theta)*cos(theta)/l;
end

dydt(1)=ddt_theta;
dydt(2)=ddt_theta_dot;
dydt=dydt';
```

Appendix B: Operating the Pendulum-Cart System

1. Check the Hardware

The system has several components, currently wired together but not physically inseparable. You should have (1) a large platform with pendulum-cart and motor, (2) a current amplifier, and (3) a small aluminum box housing a circuit and a DS1102 interface card. There are two switches; set them to **OFF** and **DISABLE**. Check that the amplifier is plugged in, and that the computer is connencted to the interface card. **Be sure that the pendulum motion is unobstructed for all cart and pendulum positions.**

2. Software

To operate the pendulum system, you need two files: the Simulink model **pendem.mdl** and a Matlab script **setup.m**. For simplicity, copy both files to a common directory, and run Matlab in that directory.

At the Matlab prompt:

Type 'setup' to run setup.m.
Type 'pendem' to open Simulink with the pendem.mdl Simulink model.

3. Building the Simulink model

Before you build, the hardware must be set accordingly:

1) Make sure the 'Enable/Disable' switch is set to **DISABLE**.
2) Set amplifier switch to **ON** (this affects velocity measurement initialization).
3) **Center** the Pendulum on the track, and stabilize with **pendulum down**.

Now build the model, using **RTW Build** in the Simulink Tools menu.

Wait until the model is built, then flip the 'Enable/Disable' switch to **ENABLE** to begin routine. **Cart and pendulum motion will begin immediately.**

4. Operation Notes:

If the system malfunctions, flip the 'Enable/Disable' switch to **DISABLE**. Make sure this switch is safely away from pendulum motion (before you compile).

The model also disables under certain error conditions, sending zero to output. If it disables, it is necessary to recompile the model. You can deliberately disable the model by using these error conditions: move the cart to the end of the track or cause a pendulum angle of more than 45° from vertical while the system is under linear control.

If the 'Enable/Disable' switch is set to **ENABLE** while compiling, Pendulum motion will begin when compiling is complete. However, while compiling, output is commanded by

the previous DSP application. This is fine **only** if this model was the previous application **and** an error condition was met. Otherwise, make sure the 'Enable/Disable' switch is set to **DISABLE**.

5. Cockpit Interface

A Cockpit file, called **pendem.css**, has been included. This file contains displays of the state variables and other basic model parameters. To use, run cockpit, then open **pendem** from the Cockpit File menu. After compiling the Simulink model, load the trace file **pendem**, using the Cockpit File menu. Click on the **Start** button to begin.

References

1. Franklin, Gene F., Powell, David J., and Emami-Maeini, Abbas, *Feedback Control of Dynamic Systems*, Addison-Wesley, New York, 1994.

2. Dorf, Richard C., *Modern Control Systems*, Addison-Wesley, New York, 1992.

3. Williams, James H. Jr., *Fundamentals of Applied Dynamics*, John Wiley & Sons, New York, 1996.

4. Friedland, Bernard, *Control System Design*, McGraw-Hill, Boston, 1986.

5. Rowell, Derek and Wormley, David N., *System Dynamics: An Introduction*, Prentice-Hall, Upper Saddle River, New Jersey, 1997.

6. Gould, L. A., Markey, W. R., Roberge, J. K., and Trumper, D. L., *Control Systems Theory*, 1997.

7. Spong, Mark, W., *The Swing Up Control Problem for the Acrobot*, IEEE Control Ststems, pp. 49-55, February 1995.

8. *Using Simulink*, The Mathworks, 1998.