

2006

Comparison of linear, non-linear and adaptive control techniques on a reaction wheel pendulum

Sangeetha Sangameswaran
University of Dayton

Follow this and additional works at: https://ecommons.udayton.edu/graduate_theses

Recommended Citation

Sangameswaran, Sangeetha, "Comparison of linear, non-linear and adaptive control techniques on a reaction wheel pendulum" (2006). *Graduate Theses and Dissertations*. 5361.
https://ecommons.udayton.edu/graduate_theses/5361

This Thesis is brought to you for free and open access by the Theses and Dissertations at eCommons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of eCommons. For more information, please contact mschlangen1@udayton.edu, ecommons@udayton.edu.

Comparison of Linear, Non-linear and Adaptive Control
Techniques on a Reaction Wheel Pendulum

A Thesis

Submitted to

The School of Engineering of the
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for
The Degree

Master of Science in Electrical Engineering

by

Sangeetha Sangameswaran

UNIVERSITY OF DAYTON

Dayton, Ohio

December, 2006

Comparison of Linear, Non-linear and Adaptive Control Techniques
on a Reaction Wheel Pendulum

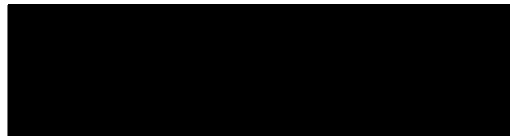
APPROVED BY:



Raúl Ordóñez, Ph.D.
Advisory Committee Chairman
Associate Professor
Department of Electrical and
Computer Engineering



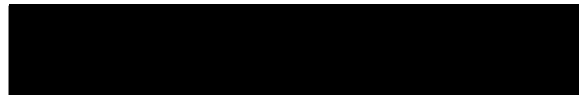
John S. Loomis, Ph.D.
Committee Member
Associate Professor
Department of Electrical and
Computer Engineering



Russell C. Hardie, Ph.D
Committee Member
Professor
Department of Electrical and Computer Engineering



Donald L. Moon, Ph.D.
Associate Dean
Graduate Engineering Program &
Research, School of Engineering



Joseph E. Saliba, Ph.D., P.E.
Dean, School of Engineering

To my parents and sister.

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my advisor, Professor Raúl Ordóñez, whose expertise, understanding, and patience, added considerably to my graduate experience. I can never thank him enough for his advice throughout the working of my thesis. I would also like to thank Dr. John Loomis and Dr. Russel Hardie for serving in my thesis committee and helping me to improve this thesis.

I would like to thank Jaideep Prasad for the work he had done on the pendulum. Having the main structure designed and built has given me a very handy head-start. It has also provided direction for the rest of the project.

Moreover, I want to thank my research colleagues Shreecharan Kanchanavally, Kayode Ajayi-Majebi, Jingyi Yao and Sangameshwar Sonth, for their valuable discussions and suggestions. I also wish to thank my friends Chaitanya Gogineni, Ashwin Karnam and Suman Jaltar who helped me get through two years of graduate school.

I would also like to thank my family for the support they provided me through my entire life and without whose love and encouragement I would not have finished this thesis.

ABSTRACT

The reaction wheel pendulum is a simple pendulum with a symmetric wheel attached to one end which is free to spin about an axis parallel to the axis of rotation of the pendulum. The coupling torque generated by the angular acceleration of the disc can be used to control the system. The balancing of the pendulum involves the use of three different controllers, namely the balancing controller, the swing-up controller and the switching controller. The balancing controller is used to balance the pendulum in the upright position. The swing-up controller raises the pendulum to the inverted position in a controlled manner, where the balancing controller can stabilize it. The switching controller is used to switch between the two as required. A comparative analysis is performed by using linear, non-linear and adaptive controllers as balancing controllers. The results of this thesis can be used to illustrate how adaptive techniques can be used practically with increased degree of robustness. Both simulation and laboratory results are shown to illustrate the conclusions.

TABLE OF CONTENTS

	Page
Abstract	v
List of Figures	ix
List of Tables	xi
Chapters:	
1. Introduction	1
1.1 Motivation	1
1.2 The Reaction Wheel Pendulum	5
1.3 Significance of the thesis	6
1.4 Research Questions	6
1.5 Organization of the thesis	7
2. The Reaction Wheel Pendulum	8
2.1 Description of the Plant	8
2.2 Input and Output Signals	8
2.3 Modeling	9
2.3.1 Parameters of the System	9
2.3.2 Mathematical Model	10
2.3.3 State Space Representation	11
2.4 Torque to voltage converter	12
2.5 Angle convention	12
3. Implementation	15
3.1 dSpace Technology	15
3.1.1 RTI and Control Desk	15
3.1.2 dSpace Hardware	16
3.1.3 Real-time Implementation	17
3.1.4 Algorithm for the control process	18

3.2	Motor Control	19
3.2.1	Pulse Width Modulation	19
3.2.2	H-Bridge LMD18200	20
3.3	Velocity generation	21
3.4	Swing-up Controller	21
3.5	Switching controller	22
4.	Linear Controller	24
4.1	Linear model of the pendulum	24
4.2	Analysis of the system	24
4.2.1	Investigation of poles	24
4.2.2	Controllability and Observability	25
4.3	Pole Placement	26
4.3.1	Linear Quadratic Regulator	26
4.4	Simulation results	27
4.5	Experimental results	27
5.	Feedback linearizing controller	31
5.1	I/O feedback linearizable system	31
5.2	Diffeomorphism	32
5.3	Controller	34
5.4	Simulation Results	34
5.5	Experimental Results	35
6.	Adaptive Controller	37
6.1	Direct Adaptive Controller	38
6.1.1	Derivation	38
6.1.2	Lyapunov Analysis	38
6.2	Fuzzy Systems and Neural Networks as Approximators	40
6.2.1	Fuzzy Systems	41
6.2.2	Neural Networks	41
6.2.3	Universal Approximation Property	42
6.3	Simulation results	42
6.4	Experimental results	43
6.5	Case studies	45
6.5.1	Case 1 : Mass Added to the Motor	46
6.5.2	Case 2 : Mass Added to the Rod	48
6.5.3	Case 3 : Mass Added to the wheel	51
7.	Conclusions	53

Appendix A	56
Appendix B	58
Appendix C	62
Bibliography	71

LIST OF FIGURES

Figure	Page
2.1 Reaction Wheel Pendulum	8
2.2 Parameters of Reaction Wheel Pendulum	10
2.3 Angle convention for a: Swing-up controller b: Balancing controller .	13
3.1 Simulink model.	18
3.2 Swing up.	21
3.3 Switching Controller.	23
4.1 Pole-zero plot.	25
4.2 Simulation results using LQR.	28
4.3 Experimental results using LQR.	29
5.1 Simulation result with feedback linearizing controller.	35
5.2 Experimental result with feedback linearizing controller.	36
6.1 Simulation result with adaptive controller.	43
6.2 Experimental result: DAC with $u_k = 0$	44
6.3 Experimental result: DAC with $u_k = u_{fbl}$	45
6.4 Experimental result: DAC with $u_k = u_{lqr}$	46
6.5 Mass added to motor.	47
6.6 LQR Controller with mass added to motor.	48

6.7	FBL Controller with mass added to motor.	48
6.8	Adaptive Controller with mass added to motor.	49
6.9	Mass added to rod	49
6.10	LQR Controller with mass added to the rod.	50
6.11	FBL Controller with mass added to the rod.	50
6.12	Adaptive Controller with mass added to the rod.	51
6.13	Mass added to wheel.	52
6.14	Adaptive Controller with mass added to the wheel.	52
B.1	Functional block diagram of the LMD18200 (taken from [1])	58
B.2	Pin diagram of the LMD18200 (taken from [1])	59
B.3	Circuit diagram (taken from [1])	60
B.4	Sign/Magnitude PWM Control	61
C.1	Simulink model.	67
C.2	Motor angle and velocity.	67
C.3	Pendulum angle and velocity.	68
C.4	Controllers.	68
C.5	Switch.	69
C.6	Torque to voltage converter.	69
C.7	Direction and PWM generator.	70

LIST OF TABLES

Table	Page
2.1 Plant parameters	10
2.2 Calculated parameters	11
2.3 Constants of the DC motor	12
3.1 Swing-up controller.	22
6.1 Results of the Case Studies	52
7.1 Balancing Metrics	54
7.2 Control Energy	54
7.3 Conclusions	55

CHAPTER 1

INTRODUCTION

1.1 Motivation

For the past 50 years, aerospace and robotics applications remained as some of the most influential sources of motivation for rigorous analysis and control of both mechanical systems and nonlinear systems. All along the way, advances achieved by researchers in mechanics and nonlinear control theory mutually affected and enhanced each other.

Control of mechanical systems is currently among one of the most active fields of research due to the diverse applications of mechanical systems in real-life [2]. During the past century, a series of scientific, industrial, and military applications motivated rigorous analysis and control design for mechanical systems. On the other hand, the theoretically challenging nature of analysis of the behavior of non-linear dynamical systems attracted many mathematicians to study control systems. As a result, the efforts of engineers and scientists together led to creation of linear control and non-linear control theories [3]. More recently, robust control theory was born and added to the above picture because of an inevitable need to deal with the presence of uncertainties in real-life control systems.

Control system theory has evolved from linear systems to nonlinear systems. Questions regarding controllability, observability, stabilization, and tracking for a linear system using state or output feedback have been quite well-understood for a long

time [4]. However, adding constraints or further specifications to the description of the system might make the control design rather complicated. None of these problems can be addressed as easily as the control problems for linear systems. One can observe that minor deviations from the standard problem of stabilization of a linear time invariant control system and additional constraints make the system rather complex. An example of such a situation can be found in [5] where the system has a forward path that is linear and time-invariant, and a feedback path that contains a memory-less, possibly time-varying, non-linearity.

Thus, although linear control theory has evolved a variety of powerful methods and has had a long history of successful applications, it has been found to be inadequate in many applications, for many reasons such as increasingly stringent performance requirements and large operating range, which invalidate the use of linearized models. Many physical systems have so-called “hard nonlinearities,” such as Coulomb friction, saturation, dead zones, backlash, and hysteresis. These nonlinearities are non-smooth or discontinuous, and do not allow linear approximations. They often cause undesirable behavior in the control system, such as instability and limit cycles if not properly handled [6]. It may be necessary to apply nonlinear control to obtain acceptable performance.

The main focus in this thesis is on nonlinear control of under actuated mechanical systems. This is motivated by broad applications of under actuated systems and theoretically challenging problems that they have to offer.

During the last decades, nonlinear control became a field of growing interest. The reason is twofold. Firstly, new achievements of nonlinear control theory combining Lyapunov and geometric methods strengthened its power. The book [6] describes the Lyapunov analysis in detail. Secondly, new demands for high performance control arose in science and engineering. Nonlinearity plays an especially strong role

in control of mechanical systems such as cars, robots, vibrational units, helicopters, ships, etc., as described in [7]. Many of those systems are characterized by a high level of complexity: high dimension of the state space, multiple inputs and outputs, parametric uncertainty and unmodeled dynamics. With the advances of low-cost microprocessors, it is neither difficult nor costly to implement nonlinear controllers.

Earlier results in nonlinear control required exact knowledge of the system dynamics. With adaptive control coming into the picture, this requirement is no longer necessary. The controller is approximated online to achieve best results.

Control of uncertain nonlinear dynamics is, thus, essential for successful applications. In fact, during the past twenty years, the control of uncertain dynamics has been very popular. Numerous algorithms have been proposed. Function approximators have been used to represent the unmodeled dynamics of the plant in the controller. One method for training approximators is using genetic algorithms [8]. Another approach is using fuzzy systems or neural networks as function approximators which will be used in this thesis. The paper [9] provides excellent examples of implementation of adaptive controllers using fuzzy systems and neural networks.

Fuzzy systems capture the decision-making capabilities of humans and neural networks capture the parallel processing and learning capabilities of biological nervous systems. It has been shown that both these systems possess the universal approximation property. A detailed description of the universal approximation property of fuzzy systems can be seen in [10]. The papers [11] and [12] describe in detail the theory of approximation and training using neural networks for non-linear systems. Thus they can be used to approximate the unknown dynamics of the system to provide robust control.

Under-actuated mechanical systems are systems that have fewer control inputs than configuration variables. Under-actuated systems appear in a broad range of

applications including robotics, aerospace systems, marine systems, flexible systems, mobile systems, and locomotive systems [13]. The “under actuation” property of under-actuated systems is due to the following four reasons:

- dynamics of the system (e.g., aircraft, spacecraft, helicopters, underwater vehicles, locomotive systems with-out wheels);
- design for reduction of the cost or some practical purposes (e.g., satellites with two thrusters and flexible-link robots);
- by actuator failure (e.g., in a surface vessel or aircraft);
- imposed artificially to create complex low-order nonlinear systems for the purpose of gaining insight in control of high-order under actuated systems (e.g., the beam-and-ball system, the rotating pendulum)

The book [14] deals with the application of modern control theory to some important under-actuated mechanical systems. It presents modeling and control of systems like the inverted pendulum, the convey-crane system, the pendubot system, the Furuta pendulum, the inertia wheel pendulum, the planar flexible-joint robot and the ball and beam system. The paper [15] deals with under-actuated mechanical systems with two degrees of freedom and symmetry.

The pendulum is one of the most important examples in dynamics and control and has been studied perhaps more than any other mechanical system. There have been several devices to illustrate the dynamics of the pendulum and to allow control design and implementation. The oldest of these ideas is the cart pole system [16] where the pivot point of the pendulum is moved linearly in order to control the pendulum motion. Later innovations of this idea were the pendubot [17] and the rotational pendulum [18] and [19]. The paper [20] discusses some simple strategies

for swinging up the pendulum based on energy control. The thesis [21] has the simulation and experimental results of a linear controller and fuzzy controller for an inverted pendulum while [22] deals with control of the inverted pendulum using neural networks. [23] and [24] are papers that deal with robust control of a double inverted pendulum. The paper [25] describes the stabilization of a triple inverted pendulum. The reaction wheel pendulum is the newest and the simplest of the various pendulum experiments due to the symmetry of the wheel attached to the end of the pendulum.

The reaction wheel pendulum was first introduced by Spong et al. [7] where the problems of swing-up and balance were based on feedback linearization and passivity based control. Since then, it has been widely used by many researchers for various experiments. The thesis [26] contains the design and development of the laboratory apparatus of the reaction wheel pendulum. The paper [27] provides the global asymptotic stabilization of the reaction wheel pendulum around its up-right position.

Though non-linear control and adaptive control have been active topics of research for the past few years, we find that they have not gained popularity in practice. In this thesis, we try to investigate the advantages and drawbacks of practically using an adaptive controller.

1.2 The Reaction Wheel Pendulum

The reaction wheel pendulum, being an under-actuated mechanical system and inherently open loop unstable with highly non-linear dynamics, is the perfect test-bed for the design of a wide range nonlinear control strategies. Its applications range widely from robotics to rocket science. Originally, systems like this were used to illustrate ideas in linear control theory. Their inherent non-linear nature helped them to maintain their usefulness along the years and they are now used to illustrate several emerging ideas in the field of modern non-linear control. It consists of a pendulum

with a rotating uniform inertia-wheel at its end. The pendulum is under actuated and the system has to be controlled via the rotating wheel. The control task is to carry the pendulum towards the upright position (swing-up) and then balancing it at this inverted position. In this thesis, three controllers that solve the problem are proposed. The different control strategies are analyzed, both by simulation and using an actual laboratory pendulum.

1.3 Significance of the thesis

The control of an inverted pendulum is a well known and challenging problem that serves as a popular bench mark in modern control system studies. Theoretically, the linear controller used to stabilize the pendulum in the inverted position has many disadvantages because in linearizing the system we lose some information about the actual system. Similarly the nonlinear controller designed using feedback linearization techniques has the disadvantage that there is no guarantee for boundedness. This experiment is used to illustrate how well the adaptive control techniques using fuzzy systems or neural networks as function approximators are able to perform, especially relative to linear and feedback linearization techniques. The problem of balancing an unstable system occurs in practice in the areas of missile stabilization, control of a space rocket during take off and in the control of robots.

1.4 Research Questions

The main question here is which of the three control techniques, linear control, feedback linearization control or adaptive control is more robust. The main tasks involved in this project are

1. Design implementation of linear and feedback linearization controllers.

2. Design and implementation of adaptive controllers using fuzzy systems as function approximators.
3. Comparative study of the performance of all the three controllers.

1.5 Organization of the thesis

This thesis consists of 6 chapters. Chapter 2 provides a description of the reaction wheel pendulum. A detailed derivation of its mathematical model is presented. Chapter 3 describes the experimental setup and the hardware used in the experiment. Chapter 4 is devoted to the linear controller and both simulation and experimental results are presented. Chapter 5 introduces the feedback linearization controller. A derivation of the controller equation is presented along with graphical results. In chapter 6, the adaptive controller is developed and three case studies are carried out to compare the performance of the three controllers. The results are analyzed and a conclusion is provided in chapter 7.

CHAPTER 2

THE REACTION WHEEL PENDULUM

2.1 Description of the Plant

The plant is an inverted pendulum freely spinning around its axis. A DC-motor is attached at its outer end. This motor runs a heavy wheel which is freely spinning in the air. The only way to apply a torque to the pendulum is to accelerate this wheel. A schematic representation of the reaction wheel pendulum is shown in figure 2.1.

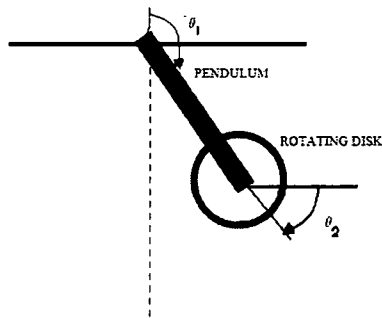


Figure 2.1: Reaction Wheel Pendulum

2.2 Input and Output Signals

There are two output signals from the plant. The angle of the pendulum θ_1 (measured with respect to the positive vertical axis) and the angle of the wheel θ_2 (measured with

respect to the positive horizontal axis). The signals we are interested in for our controller are the angle of the pendulum θ_1 , the angular velocity of the pendulum $\dot{\theta}_1$ and the angular velocity of the wheel $\dot{\theta}_2$. The angles of the wheel θ_2 and the pendulum θ_1 can be measured from two encoders:

- DC Servo Motor - Reference Voltage: 24 V - Encoder Resolution: 1000 CPR
- Optical Shaft Encoder - Encoder Resolution: 1000 CPR

The encoder on the motor can measure the angle of the wheel θ_2 , and the optical shaft encoder of the pendulum measures its angle θ_1 . The angular velocities are derived by differentiating the angles. Euler's backwards difference method is used for this. The angular velocities need to be filtered to reduce noise. The output from the controller is a voltage signal. This is converted to a Pulse Width Modulated (PWM) signal. It means that the proportion of time when the signal is high is manipulated. This signal is sent to a motor driver chip (LMD 18200) which transforms it to an analog voltage and controls the speed and direction of the motor attached to the wheel.

2.3 Modeling

The first step in any control system design is to derive a mathematical model of the system to be controlled. In this section we will develop a mathematical model for the reaction wheel pendulum. A non-linear model will be derived using the Lagrangian approach. This model will later be linearised and a linear controller designed for the system. Later we will return to the non-linear model and investigate the application of the more advanced feedback linearization and adaptive control strategies.

2.3.1 Parameters of the System

Figure 2.2 shows a schematic diagram of the reaction wheel pendulum with its parameters. The various parameters used in the system are shown in Table 2.1.

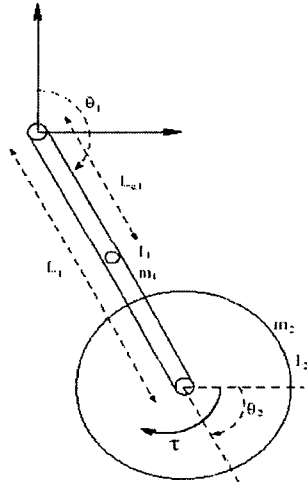


Figure 2.2: Parameters of Reaction Wheel Pendulum

Furthermore, we calculate the quantities shown in Table 2.2.

Table 2.1: Plant parameters

Variable	Description of the variable	Value
m_1	Mass of the pendulum	0.060 kg
m_2	Mass of the reaction wheel	0.0834 kg
m_{motor}	Mass of the DC motor	0.116 kg
l_1	Length of the pendulum	0.133 m
r_2	Radius of the reaction wheel	0.04 m
r_{motor}	Radius of the DC motor	0.013 m
I_{rotor}	Moment of inertia of the motor's rotor	1.06e-6 kg m^2

2.3.2 Mathematical Model

Using Lagrangian method, the mathematical model of the system can be obtained.

For detailed derivation see Appendix A. We obtain the differential equations that

Table 2.2: Calculated parameters

Variable	Description of the variable	Formula
l_{c1}	Distance to the center of mass of the pendulum	$\frac{m_2 l_1 + m_1 (l_1/2)}{m_1 + m_2}$
I_1	Moment of inertia of the pendulum	$\frac{m_1 (l_1^2 + b_1^2)}{l_1}$
I_2	Moment of inertia of the reaction wheel	$\frac{m_2 r_1^2}{2}$
I_{motor}	Moment of inertia of the motor	$\frac{m_{motor} r_{motor}^2}{2}$
m	Total mass of the system	$m_1 l_{c1} + (m_2 + m_{motor}) l_1$

describe the dynamics of the reaction wheel pendulum as,

$$\begin{aligned} d_{11}\ddot{\theta}_1 + d_{12}\ddot{\theta}_2 - mg \sin \theta_1 &= 0 \\ d_{21}\ddot{\theta}_1 + d_{22}\ddot{\theta}_2 &= \tau \end{aligned} \quad (2.1)$$

where,

$$d_{11} = m_1 l_{c1}^2 + m_2 (l_1 - l_d)^2 + m_{motor} (l_1 - l_d)^2 + I_1 + I_2 + I_{motor}$$

$$d_{12} = I_2 + I_{rotor} = d_{21} = d_{22}$$

2.3.3 State Space Representation

Starting from the mathematical model of the reaction wheel pendulum given in equation (2.1), a state space model can be developed. The state variables have been chosen as

$$x_1 = \theta_1, x_2 = \dot{\theta}_1, x_3 = \theta_2, x_4 = \dot{\theta}_2$$

Rewriting equation 2.1 we get

$$\begin{aligned} \ddot{\theta}_1 &= \frac{d_{22}}{|D|} mg \sin x_1 - \frac{d_{12}}{|D|} \tau \\ \ddot{\theta}_2 &= -\frac{d_{21}}{|D|} mg \sin x_1 + \frac{d_{11}}{|D|} \tau \end{aligned}$$

where

$$D = d_{11}d_{22} - d_{21}d_{12}$$

The state space representation is therefore

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{d_{22}}{|D|}mg \sin x_1 - \frac{d_{12}}{|D|}\tau \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -\frac{d_{21}}{|D|}mg \sin x_1 + \frac{d_{11}}{|D|}\tau
\end{aligned} \tag{2.2}$$

2.4 Torque to voltage converter

The controller that we design gives us the torque necessary to stabilize the system. We need to convert this torque to velocity so that it can directly control the DC motor. This can be done by using the formula

$$V = \frac{R_a k_a}{\tau} + k_e \dot{\theta}_2 \tag{2.3}$$

where R_a is the armature resistance of the motor, k_a is the torque constant of the motor and k_e is the error constant. Referring to the datasheets of the motor we get the value of the constants as

Table 2.3: Constants of the DC motor

Parameter	value
R_a	30 Ω
k_a	0.0283 Nm/A
k_e	0.0283 Nm/A

2.5 Angle convention

Both the optical encoders give pulses which must be converted to radians. The formula used for this is

$$\text{Angle} = 2\pi * (\text{CPR of the encoder}) \quad (2.4)$$

The angle convention used for both the swing-up and balancing controller of the pendulum are shown in Fig 2.3. In order to get the angles as shown in the figure, the

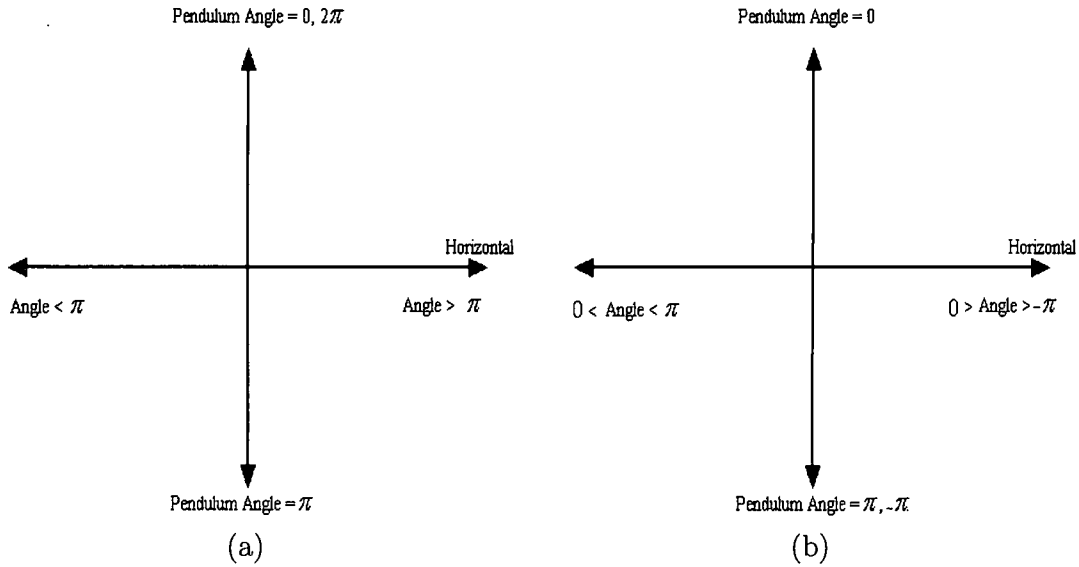


Figure 2.3: Angle convention for a: Swing-up controller b: Balancing controller

following algorithm was used.

For swing-up control:

- *while* (*angle* < $-\pi$) *angle*+ = 2π ;
- *while* (*angle* > π) *angle*- = 2π ;

For balancing control:

- *while* (*angle* < -2π) *angle*+ = 2π ;
- *while* (*angle* > 2π) *angle*- = 2π ;

- $angle = abs (angle);$

where abs denotes absolute value.

CHAPTER 3

IMPLEMENTATION

The implementation of the controller has been realized in dSpace. dSpace system provides real-time hardware for rapid prototyping. This software is used to compile a C-language program and then load it on a DS1104 R and D Controller Board.

3.1 dSpace Technology

One of the best features of the dSpace package is the ease of building real-time applications. Basically a real-time application can be created by means of two methods:

- using Simulink for building the model and then automatically generating the C code and downloading it into the DSP memory;
- hand-coding in C and compiling the model into DSP code.

The fastest way is developing the model in Simulink. Once one has completed a Simulink model, dSpace software creates the real-time DSP code, downloads it on the DS1104 board (a dSpace controller board used in the laboratory), and then automatically starts the hardware execution in real-time.

3.1.1 RTI and Control Desk

The dSpace comes with its own software for integration into Simulink called Real-Time Interface (RTI), and a separate program called Control Desk. RTI is the link

between dSPACE's real-time hardware and the MATLAB/Simulink development software from MathWorks. It extends the C code generator Real-Time Workshop so your Simulink models can be implemented very easily on dSPACE real-time hardware. RTI provides added functionality of Input/Output (I/O) ports for Simulink block diagrams and also integrates the C code generated from RTW into the dSpace system for hardware-in-the-loop (HIL) simulation. It is also possible to provide a handwritten C-code model. Control Desk software is similar to an oscilloscope allowing access to all the variables in the dSpace system and displays it on the monitor. It allows the designer to observe the behavior of the real-time application and to change application-specific parameters.

3.1.2 dSpace Hardware

dSpace hardware mainly consists of DS1104 R and D Controller board which can be plugged into the PCI slot of a PC. It is a complete real-time control system based on a 603 Power PC floating point processor running at 250 Mhz. For advanced I/O purposes the board includes a slave-DSP subsystem based on the TMS320F240 DSP microcontroller. dSpace has a connector panel CP1104 which provides easy-to-use connections between DS1104 and devices to be connected to it. Devices can be easily connected, disconnected and interchanged without soldering. The CP1104 Connector panel is equipped with the following connectors

- BNC Connectors
- Digital I/O connector
- Slave I/O PWM connector
- Incremental encoder interface
- UART RS232 Connector

For this experiment will only be using three of the above connectors.

1. Encoder interface to read the pendulum and wheel angle.
2. Slave I/O PWM connector for controlling the voltage to motor through H-bridge.
3. Digital I/O connector for controlling the direction of the motor through H-bridge.

3.1.3 Real-time Implementation

For real-time implementation of the system, the model is built in Simulink but is compiled into C codes which are then loaded on the DS1104 board. After that, the codes run on the master PPC of the DS1104 instead of the host PC. The DS1104 board receives a signal from the encoders, digitally processes the input signal by its master PPC, and then presents the output signal to motor. The simulink model used for the experiment is shown in Fig 3.1

- The 'Motor Angle and Velocity' block receives the pulses from the motor encoder and calculates the motor angle and velocity.
- The 'Pendulum Angle and Velocity' block receives the pulses from the pendulum encoder and calculates the pendulum angle and velocity.
- The 'Controllers' block consists of both the swing-up and balancing controllers.
- The 'switch' block switches between the two controllers based on the position of the pendulum.
- The 'Safety' block stops the experiment whenever the wheel velocity exceeds a particular value.

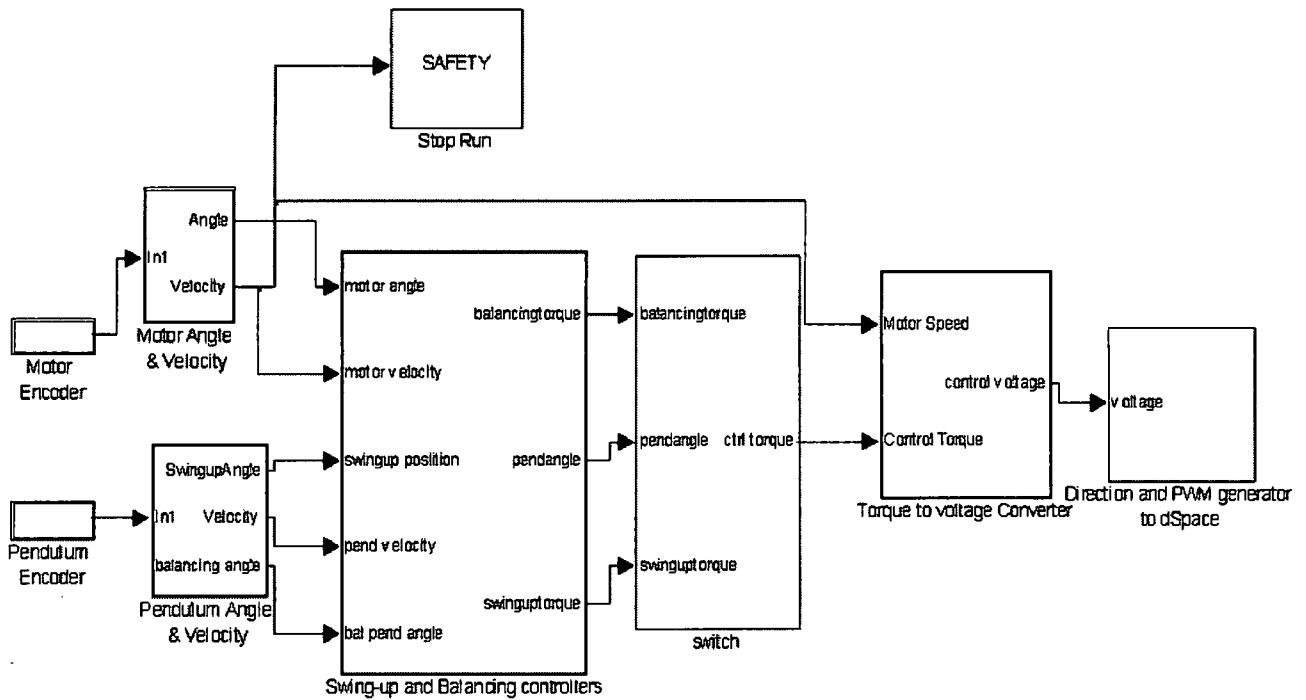


Figure 3.1: Simulink model.

- The control torque is then converted to voltage and then to PWM and sent to the H-bridge.

3.1.4 Algorithm for the control process

All the control algorithms are performed in a periodic function. This periodic function runs every millisecond and has the following tasks:

- Get θ_1 and θ_2 from encoders.
- Calculate angular velocities and filter them.
- Decide whether swing up or stabilizing controller should be used.
- Calculate the control torque.
- Convert the torque to voltage.

- Calculate duty cycle of the PWM and the direction signal.
- Send to the H Bridge.
- H-bridge controls direction and speed of motor.

3.2 Motor Control

The motor used in this experiment is an A-max 26 from Maxon Motors. This is a 24 V DC motor with ironless core and no load speed of 7910 rpm. The main signal type used to drive the motor is a pulse width modulated rectangular wave. This is just like connecting a switch to the motor. If the motor should go fast, the switch is repetitively turned on for a long time and then turned off for a short time so that the time the motor is powered is much longer than the time it is not powered. To make the motor go slowly, the time the switch is on is very short compared to the time the switch is off. The H-Bridge is the primary means for driving a small DC motor in the forward and reverse directions.

3.2.1 Pulse Width Modulation

The major reason for using pulse width modulation in DC motor control is to avoid the excessive heat dissipation. The heat dissipation problem often results in large heat sinks and sometimes forced cooling. PWM greatly reduces this problem. Moreover, the input signal to the PWM driver may be directly derived from any digital system without the need for any D/A converters. But PWM has its own disadvantages. The desired signal is not translated to a voltage amplitude but rather the time duration (or duty cycle) of a pulse. This is obviously not a linear operation. But with a few assumptions, which are usually valid in motor control, PWM may be approximated as being linear (i.e., a pure gain). The linear model of the PWM is based on the average voltage being equal to the integral of the voltage waveform.

Thus

$$V_s * T_{on} = V_{eq} * T_{sw} \quad (3.1)$$

where

- V_s is the supply voltage (+24 volts);
- T_{on} is the pulse duration;
- V_{eq} is the average or equivalent voltage seen by the motor;
- T_{sw} is the switching period.

The control variable is the duty cycle which is T_{on}/T_{sw} . The duty cycle must be recalculated at each sampling time. The voltage that the motor sees is thus V_{eq} , which is equal to the duty cycle times the supply voltage.

3.2.2 H-Bridge LMD18200

The dSpace DS1104 board, which provides our pulse width modulated motor driving signals, simply does not have the power to drive the motor. In addition, the dSpace board cannot control the direction of the motor, because the pulse width modulated signals operate on 5V TTL logic. This means that the motor driving pulses can be either 0V or 5V, and as such can only turn the motor in one direction. In order to solve these problems, a very commonly used H-Bridge circuit was used. LMD18200 is a 3A, 55 V H-Bridge designed for motor-control applications. The input requirements for this IC are a PWM signal for proportional speed control and a logic high (+5V) or low (0V) for direction control which are provided by the DS1104. The functional block diagram and pin configuration of this IC are given in Appendix B.

3.3 Velocity generation

The angular velocities can be calculated from the angles by differentiation. This is done in Simulink using the differential block which uses the backwards difference method. The velocity then has to be filtered to get rid of the high frequency noise that is present. For this a first order filter with transfer function $H(s) = \frac{15}{s+15}$ is used.

3.4 Swing-up Controller

The aim of this controller is to make the pendulum go up as fast as possible. The motor and the wheel are too weak to swing the pendulum to its upright position in one swing. They need to add energy to the system with many swings to get it upright. One way to swing-up the pendulum is to give maximum torque to the pendulum to increase its speed in the direction it is going. But when should the direction of the torque be reversed? Since we need to get the most out of the torque, this is best done when $\dot{\theta}_1$ is almost zero, i. e., when the pendulum is about to change direction. This algorithm was first reported in [26]. Fig 3.2 is an illustration of the swing-up controller. After some trial and error, the following controller was designed.

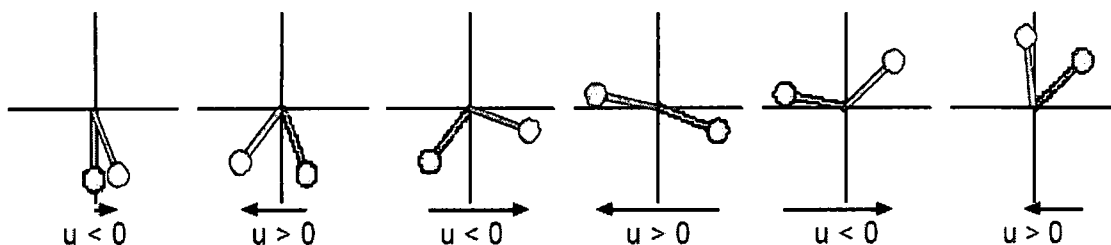


Figure 3.2: Swing up.

Table 3.1: Swing-up controller.

θ_1	Control Torque
$\geq \Pi$	-0.0072
$< \Pi$	0.0072

3.5 Switching controller

To get a complete controller for the pendulum the swing up and the stabilizing controller need to be combined into one. Or in other words, one needs to know when to switch between the two controllers. To switch at the right time a switching controller is developed as shown in Fig 3.3 (taken from [28]). The basic idea is to switch to the balancing controller as soon as the pendulum reaches the upright position and at this time also terminate the swing up controller. To give the stabilizing controller some space to work within, the switching controller is constructed so that it switches to the balancing controller when the pendulum has swung up and is within some small angle, δ , of the upright position. The value of δ is taken as 0.2 radians in both simulation and experiment.

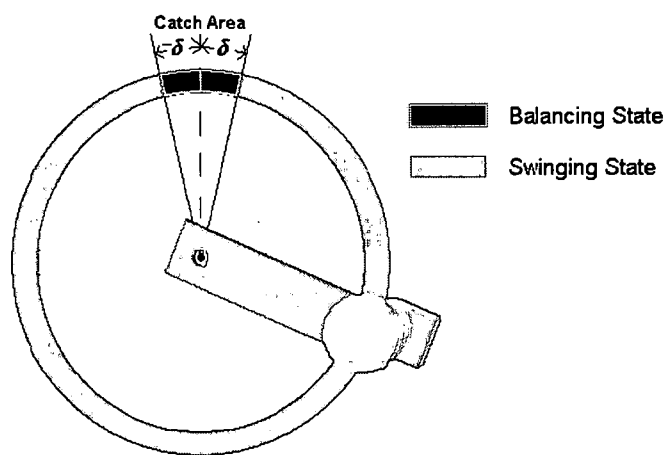


Figure 3.3: Switching Controller.

CHAPTER 4

LINEAR CONTROLLER

4.1 Linear model of the pendulum

After being moved by the swinging process to its upright position, the pendulum is supposed to be in the linearized region around $\theta = 0$. Thus in this region we can use the linear model of the plant as shown below

$$\dot{x} = Ax + Bu \quad (4.1)$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{d_{22}mg}{|D|} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-d_{21}mg}{|D|} & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{-d_{12}}{|D|} \\ 0 \\ \frac{d_{11}}{|D|} \end{bmatrix} \quad (4.2)$$

Substituting the values for the various parameters, we get

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 77.0057 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -77.0057 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ -257 \\ 0 \\ 15430 \end{bmatrix} \quad (4.3)$$

4.2 Analysis of the system

4.2.1 Investigation of poles

The system has four poles. Fig 4.1 shows the location of the poles for the linearization around the unstable equilibrium point. The poles in the right half of the s plane show that the system is unstable.

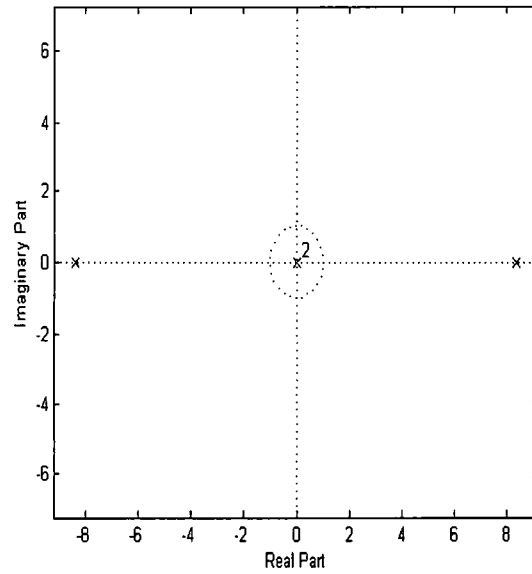


Figure 4.1: Pole-zero plot.

4.2.2 Controllability and Observability

To be able to fully control the system it needs to be controllable. Therefore, the controllability matrix is calculated.

$$C = \begin{bmatrix} 0 & -257 & 0 & -19819 \\ -257 & 0 & -19819 & 0 \\ 0 & 15430 & 0 & 19819 \\ 15430 & 19819 & 0 & 0 \end{bmatrix}$$

Rank of C is 4 which is equal to number of states of the system. Thus all the states can be controlled by the input signal u .

The states x_1 and x_3 can be measured using encoders. The other two states, x_2 and x_4 are derived by differentiation and filtering of the sensor signals. Thus obviously, the plant is observable as well.

4.3 Pole Placement

Given a controllable system in state-space form, we can always stabilize the system using full-state feedback. One method we have to find these feedback gains is pole placement. The power of pole placement is that we can place the system poles anywhere. The designer is likewise free to shape the system's response any way he/she desires. The strength of pole placement is also its weakness. Since the system's poles can be placed anywhere, the designer is not given any help in choosing where the poles should be placed. This is a problem since, when practical considerations are taken into account, such as noise effects, limited controller size, etc., some pole locations are better than others.

4.3.1 Linear Quadratic Regulator

A method to design the coefficients for optimal performance is to use a Linear Quadratic (LQ) controller. Such a control law u is derived by minimizing a cost function depending on the weights Q , R and N :

$$J(v) = \int_0^{\infty} (x^T Q x + u^T R u + 2x^T N u) dt$$

where x and u are linked to the state space equation $\dot{x} = Ax + Bu$.

The choice of Q lies on the fact that more emphasis should be given to x_1 . Q is chosen in the following way.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix}$$

R is a scaling factor and is set to 900 in order to avoid having a too high control voltage. Finally, N is equal to 0 because the cost due to the correlation between x and u has been neglected.

4.4 Simulation results

For the chosen values of Q , R and N , we obtain the value of gain as

$$k = [-0.7844, -0.0896, -0.0001, -0.0002]^T \quad (4.4)$$

This places the poles of the system at -8.7824, -8.7741, -1.2631+0.3482i and -1.2631-0.3482i. The simulation results with the LQR controller as balancing controller are shown in Fig 4.2. We see that the controller drives all the states to zero and balances the pendulum.

4.5 Experimental results

The experimental results using LQR as the balancing controller are shown in Fig 4.3. The response time in the experiment is more when compared to the simulation. More number of swings were necessary for the pendulum to reach the inverted position. The swing-up controller had to be tuned in the following way so that the angular velocity of the pendulum was not too high when it arrived at the inverted position.

- *if* ($angle < \pi$)
 if ($0 < angle < 0.7850$)
 torque = 0.0052;
 else
 torque = 0.008;
- *if* ($angle > \pi$)
 if ($5.4962 < angle < 2\pi$)
 torque = -0.0052;
 else
 torque = -0.008;

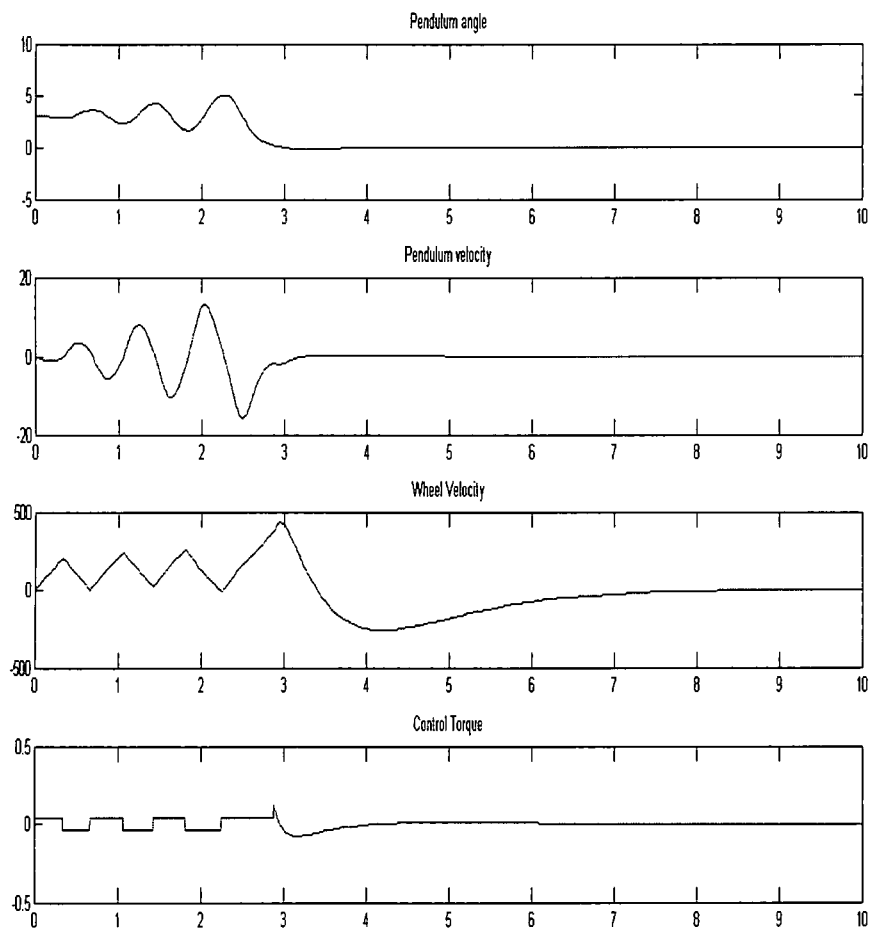


Figure 4.2: Simulation results using LQR.

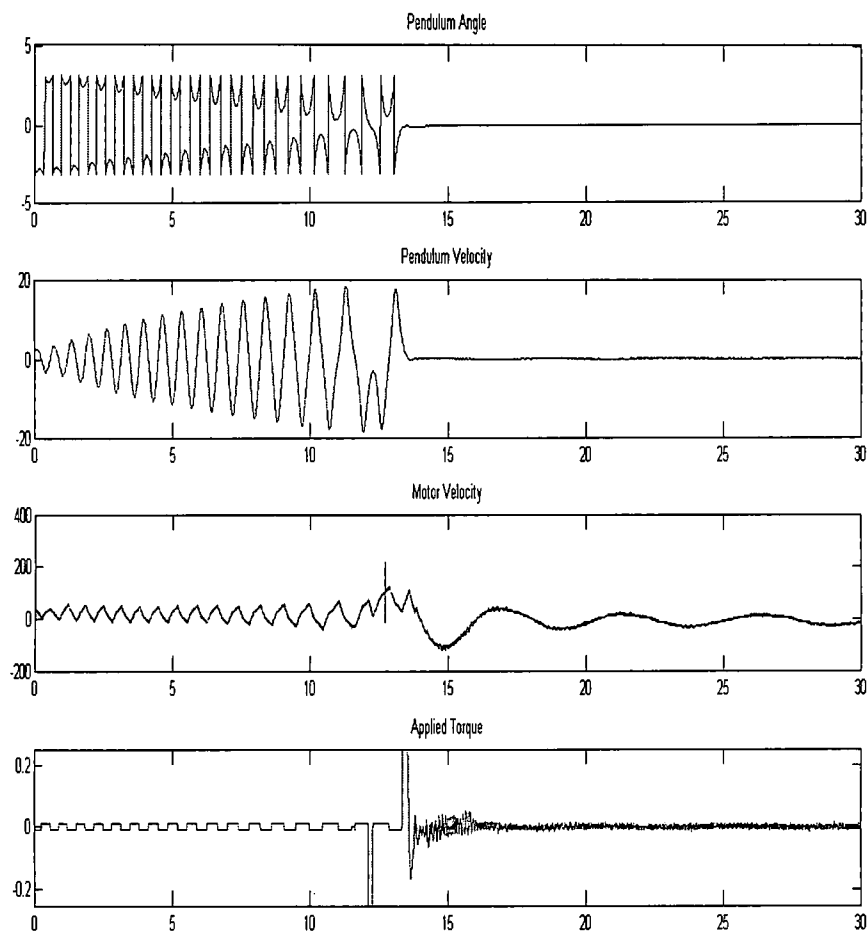


Figure 4.3: Experimental results using LQR.

nl

CHAPTER 5

FEEDBACK LINEARIZING CONTROLLER

A possible way to approach the stabilization problem for a non-linear system is to make use of the powerful results available in the linear case via feedback linearization. Some nonlinear systems can be transformed to linear systems via feedback and possibly a non-linear change of variables. After this transformation, a stabilizing linear state feedback controller is designed. In this section, we design a feedback linearizing controller for the balancing of the reaction wheel pendulum. There are two types of feedback linearization - state feedback linearization and I/O feedback linearization. Here we will consider I/O feedback linearization.

5.1 I/O feedback linearizable system

The output equation of the system is given by the first component of the generalized momentum as shown below.

$$y = d_{11}x_1 + d_{12}x_3 \quad (5.1)$$

Proceeding from (2.2) and (5.1) and taking the successive differentiation on (5.1), we get

$$\begin{aligned}
y &= d_{11}x_1 + d_{12}x_3 \\
\dot{y} &= d_{11}\dot{x}_1 + d_{12}\dot{x}_3 \\
&= d_{11}x_2 + d_{12}x_4 \\
y^{(2)} &= d_{11}\dot{x}_2 + d_{12}\dot{x}_4 \\
&= mg \sin x_1 \\
y^{(3)} &= mgx_2 \cos x_1 \\
y^{(4)} &= -mg \sin x_1 x_2^2 + m^2 g^2 \cos x_1 \sin x_1 \frac{d_{22}}{|D|} - \frac{d_{12}}{|D|} mg \cos x_1 \tau
\end{aligned} \tag{5.2}$$

Clearly, the relative degree $n = 4$, i. e., we need to differentiate the output 4 times until the input u appears for the first time and it doesn't vanish $\forall x \in D \subset \mathbb{R}^{n+d}$ where d refers to the residual dynamics of the system. Here $d = 0$, so there are no zero dynamics. Thus the system is input-output feedback linearizable. But it is not globally I/O feedback linearizable. When $\cos x_1 = 0$ we lose controllability, i. e., the feedback linearization is only valid in the region $-\frac{\pi}{2} < x_1 < \frac{\pi}{2}$. This condition is easily satisfied in our experiment because the feedback linearizing controller is only active in the region $-0.2 < x_1 < 0.2$.

5.2 Diffeomorphism

Since the system is I/O linearizable, we can therefore define new state variables z_1, z_2, z_3 and z_4 and choose a diffeomorphism (that is, a smooth, invertible change of

coordinates) which puts the system in canonical form as follows

$$\begin{aligned}
z &= T(x) \\
z_1 &= y \\
z_2 &= \dot{y} \\
z_3 &= \ddot{y} \\
z_4 &= y^3
\end{aligned} \tag{5.3}$$

From 5.2, we get

$$\begin{aligned}
z_1 &= d_{11}x_1 + d_{12}x_3 \\
z_2 &= d_{11}x_2 + d_{12}x_4 \\
z_3 &= mg \sin x_1 \\
z_4 &= mgx_2 \cos x_1
\end{aligned} \tag{5.4}$$

This transformation $z = T(x)$ is a diffeomorphism since it is continuously differentiable and T^{-1} exists. We can now define the new state equations

$$\begin{aligned}
\dot{z}_1 &= z_2 \\
\dot{z}_2 &= z_3 \\
\dot{z}_3 &= z_4 \\
\dot{z}_4 &= f + gu
\end{aligned} \tag{5.5}$$

where

$$\begin{aligned}
f &= -mg \sin x_1 x_2^2 + m^2 g^2 \cos x_1 \sin x_1 \frac{d_{22}}{\det D} \\
g &= \frac{-d_{12}}{\det D} mg \cos x_1
\end{aligned}$$

Using the diffeomorphism, we have transformed the system into a canonical form.

5.3 Controller

For balancing the pendulum at its inverted position the output needs to be regulated to zero. For the system in the form (5.5), the feedback linearizing controller is given by

$$u = \frac{1}{g}(-f + v) \quad (5.6)$$

so that the system becomes the linear chain of integrators

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ \dot{z}_3 &= z_4 \\ \dot{z}_4 &= v \end{aligned} \quad (5.7)$$

The state transformation and hence the feedback linearization control strategy is valid as long as $|x_1| < \frac{\pi}{2}$. The new control variable can be taken as

$$v = -k_1 z_1 - k_2 z_2 - k_3 z_3 - k_4 z_4 \quad (5.8)$$

and k_1 , k_2 , k_3 and k_4 are chosen such that $s^4 + k_4 s^3 + k_3 s^2 + k_2 s + k_1$ is a Hurwitz polynomial. This ensures that the poles are in the left half of the s -plane, thus making the system stable.

5.4 Simulation Results

The system was simulated using Runge-Kutta method with a step size of 0.001. For simulation of the controller we took $k_1 = 350$, $k_2 = 550$, $k_3 = 220$ and $k_4 = 50$ which places the poles of the system at -47.9181, $-0.6816 \pm 2.6066i$, -0.7187. The simulation results are shown in Fig 5.1.

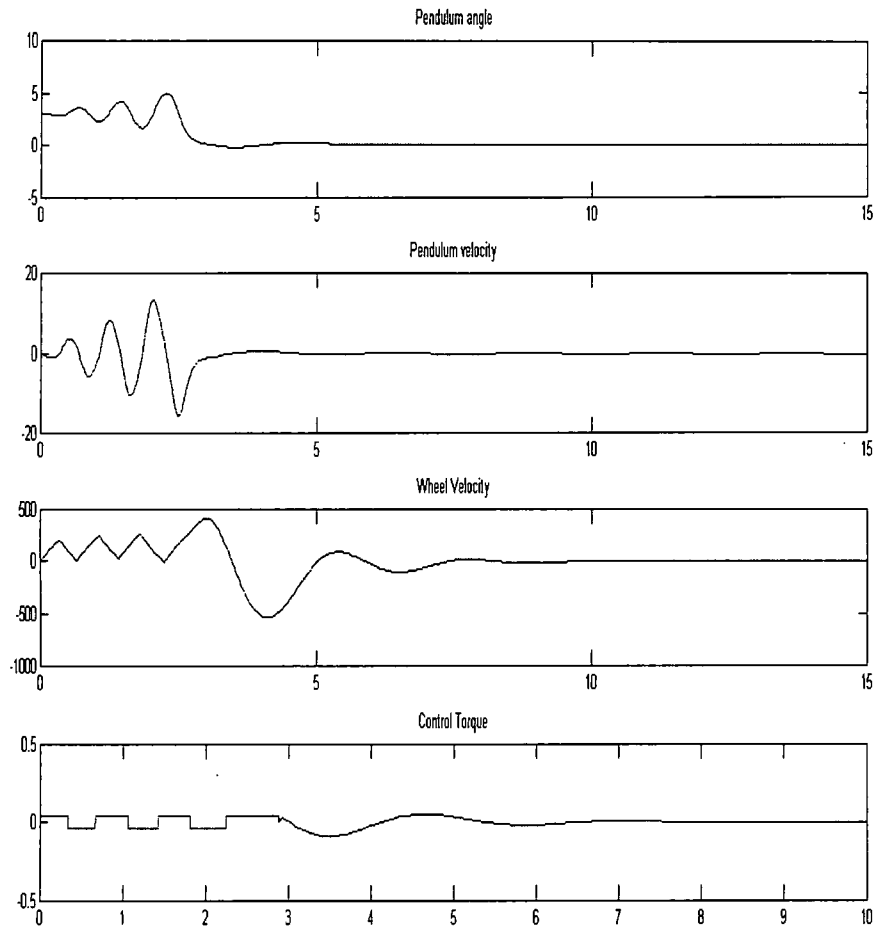


Figure 5.1: Simulation result with feedback linearizing controller.

5.5 Experimental Results

The same values of the gain k were used for the experiment as well and we found that the pendulum was able to balance itself at the inverted position. The results are shown in Fig 5.2

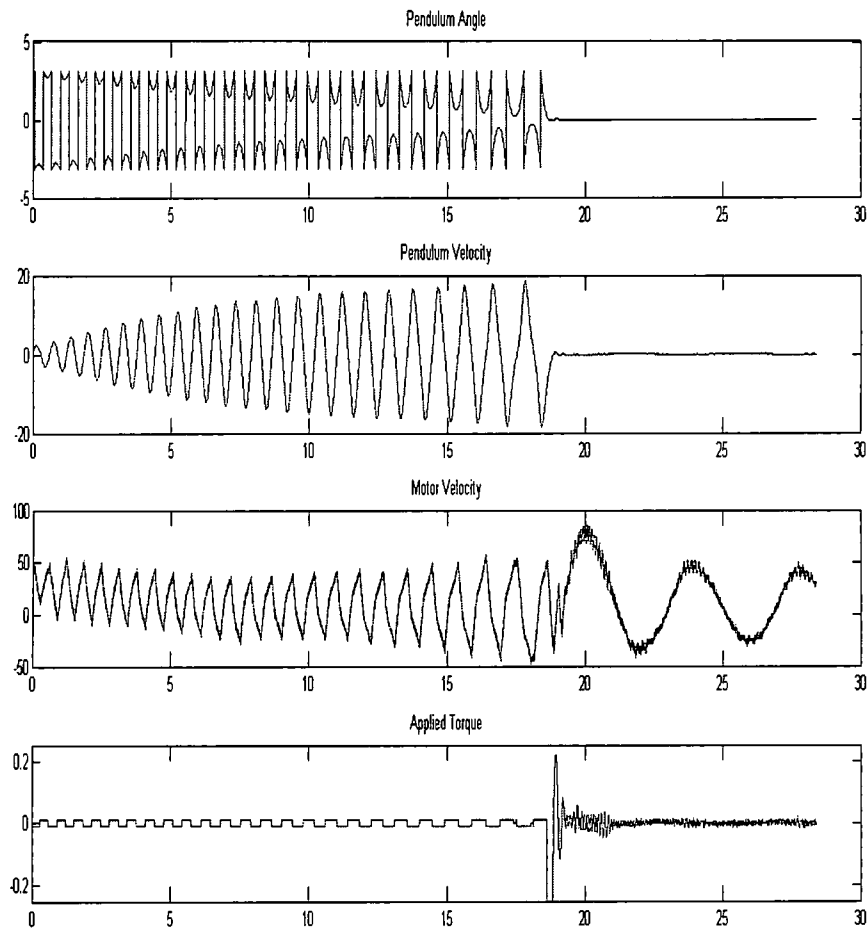


Figure 5.2: Experimental result with feedback linearizing controller.

CHAPTER 6

ADAPTIVE CONTROLLER

Both the feedback linearizing controller and the LQR fail when the system parameters are altered. In such a situation an adaptive controller provides stable control. Adaptive controllers are suitable for systems whose nominal mathematical model contains an uncertain parameter ϕ . Adaptive designs involve the construction of a time varying parameter $\hat{\phi}$ whose value is controlled by an adaptation law. The adaptive control mechanisms attempts to 'identify' or 'estimate' the uncertain parameter ϕ of the plant by a 'parameter estimator.' In general, there are two types of adaptive control.

- Indirect Adaptive Control
- Direct Adaptive Control

In indirect adaptive control, we approximate the unknown plant dynamics and use the approximations to construct the control law. In direct adaptive control, we directly approximate the stabilizing controller. For this particular system, the dynamics are complex and its easier to directly approximate the controller rather than approximating the dynamics and then designing the controller. Also this method allows us to incorporate the linear and feedback linearizing controller. Hence we will only consider direct adaptive control.

6.1 Direct Adaptive Controller

6.1.1 Derivation

In order to design a direct adaptive controller, we make the following assumptions.

- $g(z)$ is bounded by two constants g_0 and g_1 such that $g_1 \leq g(z) \leq g_0 < 0$.
- $|g(z)| \leq B < \infty \forall z \in S \subset \mathbb{R}^{n+d}$.

In this particular experiment we have chosen the values for these constants as $g_0 = -76$, $g_1 = -100$ and $B = 200$. From (5.6), the ideal feedback linearizing controller is obtained as

$$u_{fl}^* = \frac{1}{g(z)}(-f(z) + v) \quad (6.1)$$

This can be represented as

$$u_{fl}^* = \mathcal{F}(z, \bar{r}, \phi_u^*) + w_u(z, \bar{r})$$

where $\phi_u^* \in \mathbb{R}^P$ is an unknown vector of ideal parameters, \bar{r} is the reference vector and $|w_u(z, \bar{r})| \leq W_u$ is the largest approximation error between the approximated controller and u_{fl}^* . Here ϕ_u^* is unknown while W_u can be estimated by trial and error and is chosen as 0.1. Let $\hat{\phi}_u$ be the parameter estimate of ϕ_u^* which will be updated online. We define our controller u as

$$u = \mathcal{F}(z, \bar{r}, \hat{\phi}_u) + u_s \quad (6.2)$$

where \mathcal{F} is a universal approximator such as a fuzzy system and u_s is the stabilizing term.

6.1.2 Lyapunov Analysis

Consider the reference vector $r(t)$ as

$$\bar{r} = [r, \dot{r}, r^{(2)}, r^{(3)}]$$

where r, \dot{r}, \dots are the reference, first derivative of the reference and so on. The error is defined as

$$e = k_1(z_1 - r) + k_2(z_2 - \dot{r}) + k_3(z_3 - r^{(2)}) + z_4 - r^{(3)}$$

The error dynamics are given by

$$\dot{e} = k_1(z_2 - \dot{r}) + k_2(z_3 - \ddot{r}) + k_3(z_4 - r^{(3)}) + f(z) + g(z)u - r^{(4)}$$

Lets define

$$\bar{\chi} = k_1(z_2 - \dot{r}) + k_2(z_3 - \ddot{r}) + k_3(z_4 - r^{(3)}) - r^{(4)} \quad (6.3)$$

so that

$$\dot{e} = \bar{\chi} + f(z) + g(z)u \quad (6.4)$$

From (6.2)

$$\dot{e} = \bar{\chi} + f(z) + g(z)(\mathcal{F}(z, \bar{r}, \hat{\phi}_u) + u_s)$$

Adding and subtracting $\mathcal{F}(z, \bar{r}, \phi_u^*)$, we get

$$\dot{e} = \bar{\chi} + f(z) + g(z)(\mathcal{F}(z, \bar{r}, \hat{\phi}_u) + u_s) + \mathcal{F}(z, \bar{r}, \phi_u^*) - \mathcal{F}(z, \bar{r}, \phi_u^*)$$

The parameter error is given by $\tilde{\phi}_u = \hat{\phi}_u - \phi_u^*$. Thus

$$\dot{e} = \bar{\chi} + f(z) + g(z)(\mathcal{F}(z, \bar{r}, \tilde{\phi}_u) + u_s + u_{fl}^* - w_u)$$

since $u_{fl}^* = \mathcal{F}(z, \bar{r}, \phi_u^*) + w_u$. But we know that $\mathcal{F}(z, \bar{r}, \tilde{\phi}_u) = \tilde{\phi}_u^T \zeta(z, \bar{r})$ and substituting u_{fl}^* in the above equation, it becomes

$$\dot{e} = -ke + g(\tilde{\phi}_u^T \zeta(z, \bar{r}) - w_u + u_s)$$

Defining the Lyapunov candidate as

$$V = -\frac{1}{2g}e^2 + \frac{1}{2\gamma}\tilde{\phi}_u^T \tilde{\phi}_u \geq 0$$

where the adaptation gain $\gamma > 0$, we obtain

$$\begin{aligned}\dot{V} &= -\frac{1}{g}e\dot{e} + \frac{\dot{g}}{2g^2}e^2 + \frac{1}{\gamma}\tilde{\phi}_u^T\dot{\hat{\phi}}_u \\ &= \frac{k}{g}e^2 - \tilde{\phi}_u^T\zeta_ue - (u_s - w_u)e + \frac{\dot{g}}{2g^2}e^2 + \frac{1}{\gamma}\tilde{\phi}_u^T\dot{\hat{\phi}}_u\end{aligned}$$

From the above equation, the adaptation law can be given as

$$\dot{\hat{\phi}} = \gamma\zeta_ue \quad (6.5)$$

Using the above adaptation law, \dot{V} becomes

$$\dot{V} = \frac{k}{g}e^2 - (u_s - w_u)e + \frac{\dot{g}}{2g^2}e^2$$

Finding the upper bound for \dot{V} ,

$$\begin{aligned}\dot{V} &\leq \frac{k}{g}e^2 + \left|\frac{\dot{g}}{2g^2}e^2\right| + |W_ue| - u_se \\ &\leq \frac{k}{g}e^2 + \left(\frac{B}{2g_0^2}|e| + W_u\right)|e| - u_se\end{aligned}$$

From the above equation, the stabilizing term can be obtained as,

$$u_s = \text{sgn}(e)\left(W_u + \frac{B}{2g_0^2}|e|\right) \quad (6.6)$$

such that

$$\begin{aligned}\dot{V} &\leq \frac{k}{g}e^2 \\ &\leq \frac{k}{g_1}e^2 \leq 0\end{aligned}$$

where g_1 is the upper bound of $g(z)$ and is negative. By LaSalle's theorem we can prove that $e(t)$ converges to 0 asymptotically, and the parameter error $\tilde{\phi}(t)$ is bounded.

6.2 Fuzzy Systems and Neural Networks as Approximators

Fuzzy systems and neural networks are universal approximators and can be used to approximate any non-linear function. Here they are used to approximate the controller.

6.2.1 Fuzzy Systems

Fuzzy systems are a mechanism to encode heuristic knowledge about how human do certain tasks. They are non-linear functions that are parameterized by membership function parameters. They are based on a set of “if-then” rules that is implemented with the help of membership functions. These membership functions describe the extent to which a linguistic statement is true. The most commonly used membership functions are Gaussian membership functions which are represented as

$$\mu_i(x) = e^{-\left(\frac{\|x-c_i\|^2}{\sigma_i^2}\right)}$$

where c_i are the centers of the Gaussian functions and σ_i are the spreads for the Gaussian functions.

Fuzzification is the process of obtaining the inputs x_i and computing $\mu(x_i)$ for each membership function. The inference mechanism decides which rules must be used and applies them. Finally we compute the output of the fuzzy system using “center average defuzzification” given by

$$y = \mathcal{F}(x) = \frac{\sum_{j=1}^p b_j \mu_j(x)}{\sum_{j=1}^p \mu_j(x)}$$

where b_j is the center of the output membership function for the j th rule.

6.2.2 Neural Networks

Artificial neural networks are computer algorithms of a connected set of neurons that form biological neural networks. The multilayer perception (MLP) should be viewed as a nonlinear network whose nonlinearity can be tuned by changing the weights, biases, and parameters of the activation functions. An MLP has three layers. The input layer transforms the input based on the ‘weights’, which can be tunable. The activation function then produces the output. A few of the commonly used activation functions are linear, hyperbolic tangent and radial basis functions. Neural

networks using Radial Basis Functions (RBFs) as activation functions have been used in a number of applications because of their mathematical properties. An RBF is given by

$$\psi(s) = \exp\left(\frac{-s^2}{\nu^2}\right) \quad (6.7)$$

where $s, \nu \in \mathbb{R}$. The relation between input and output in an RBF (with inner product output mapping) is given by

$$y = \mathcal{F}(x, \phi) = \sum_{j=1}^p W_j \exp\left(\frac{\|X - C_j\|^2}{\nu_j^2}\right) \quad (6.8)$$

6.2.3 Universal Approximation Property

It has been shown by many authors that fuzzy systems and neural networks possess the universal approximation property. That is, if $f : D \rightarrow \mathbb{R}$ is continuous and bounded in the compact region $D \subset \mathbb{R}^n$, then, given a desired approximation error bound ϵ , a large enough p exists so that f can be uniformly approximated by a fuzzy system or neural network in D .

Here p refers to size of the approximator, i. e., the number of parameters. For the direct adaptive control of the reaction wheel pendulum, the fuzzy system approximator is designed with $p = 3$ rules per membership function. The membership functions are Gaussian functions.

6.3 Simulation results

When the designed controller was simulated in Matlab, the results obtained are shown in Fig 6.1. We observe that the overshoot at the time of switching is higher in the case of adaptive controller. The control value goes to zero once the pendulum is stabilized.

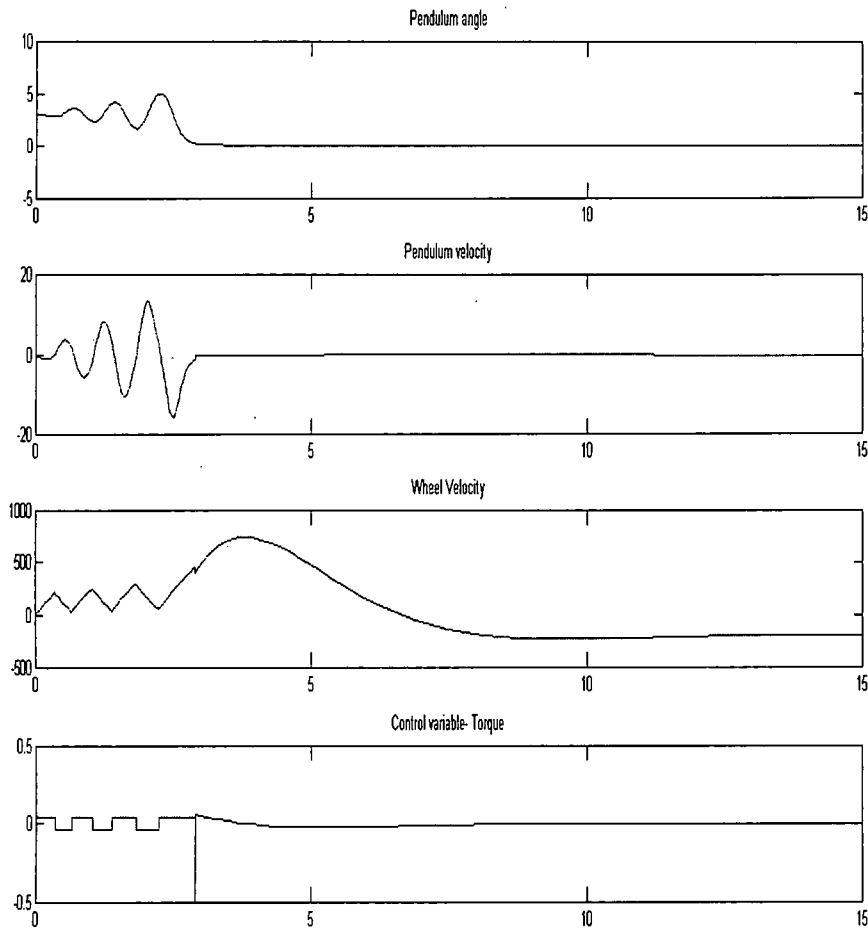


Figure 6.1: Simulation result with adaptive controller.

6.4 Experimental results

When the direct adaptive controller was implemented on the actual apparatus, we found that for a sampling rate of 0.001, the system indicated a run-time error. So the sampling rate had to be changed to 0.0016. The experimental results of DAC with the known part of the controller $u_k = 0$ are shown in Figure 6.2.

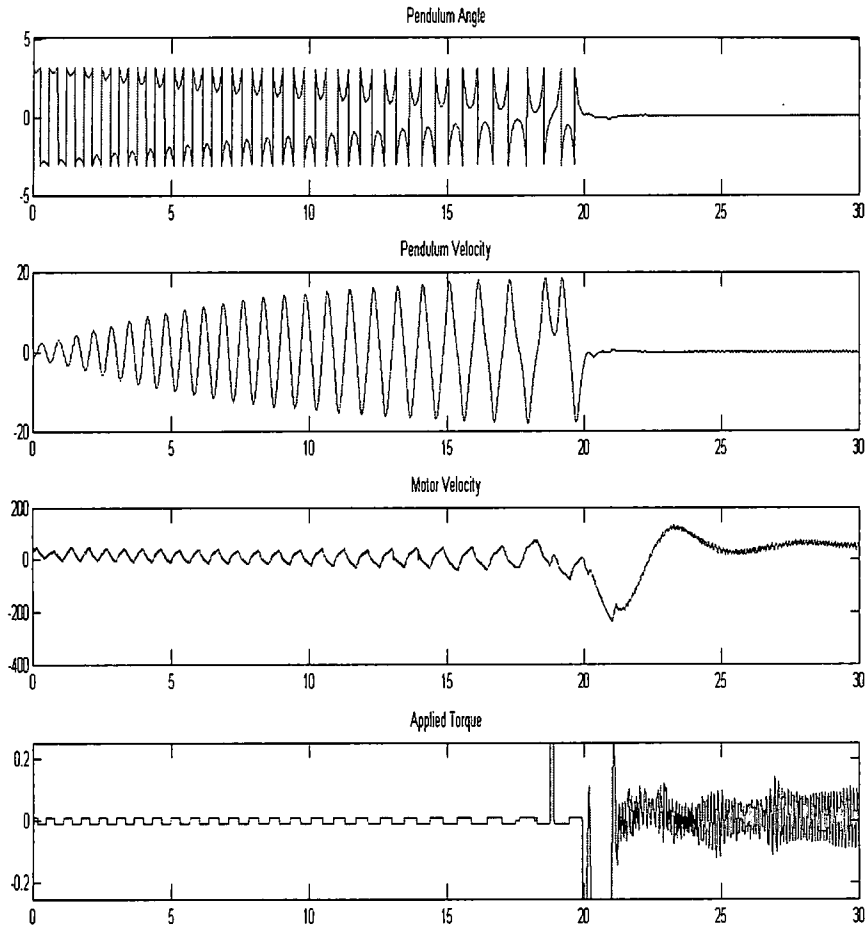


Figure 6.2: Experimental result: DAC with $u_k = 0$.

Since we already know that both the LQR and feedback linearizing controller can be used for the system, we will use this information and let the approximator approximate only the unknown part of the controller. The results of using a known controller are shown in Figures 6.3 and 6.4.

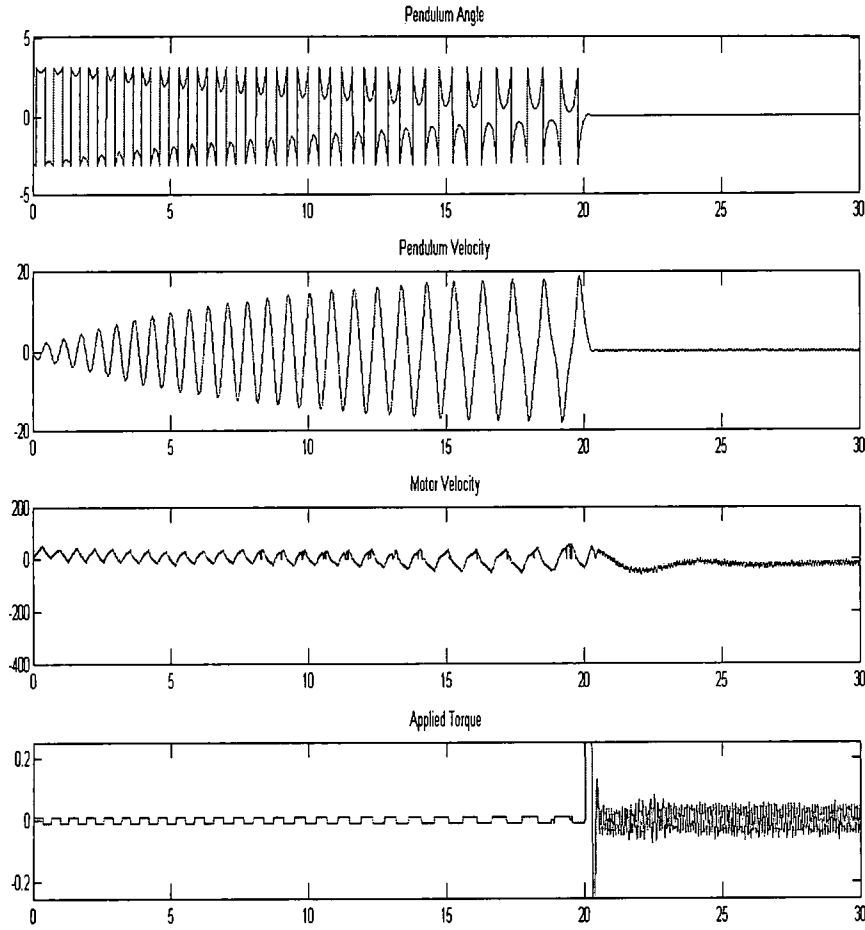


Figure 6.3: Experimental result: DAC with $u_k = u_{fb}$.

6.5 Case studies

So far we have found that all the three controllers that were designed and implemented were able to maintain the pendulum balanced in the inverted position. In order to test the robustness of these controllers, the original system was modified a bit and the following case studies were carried out.

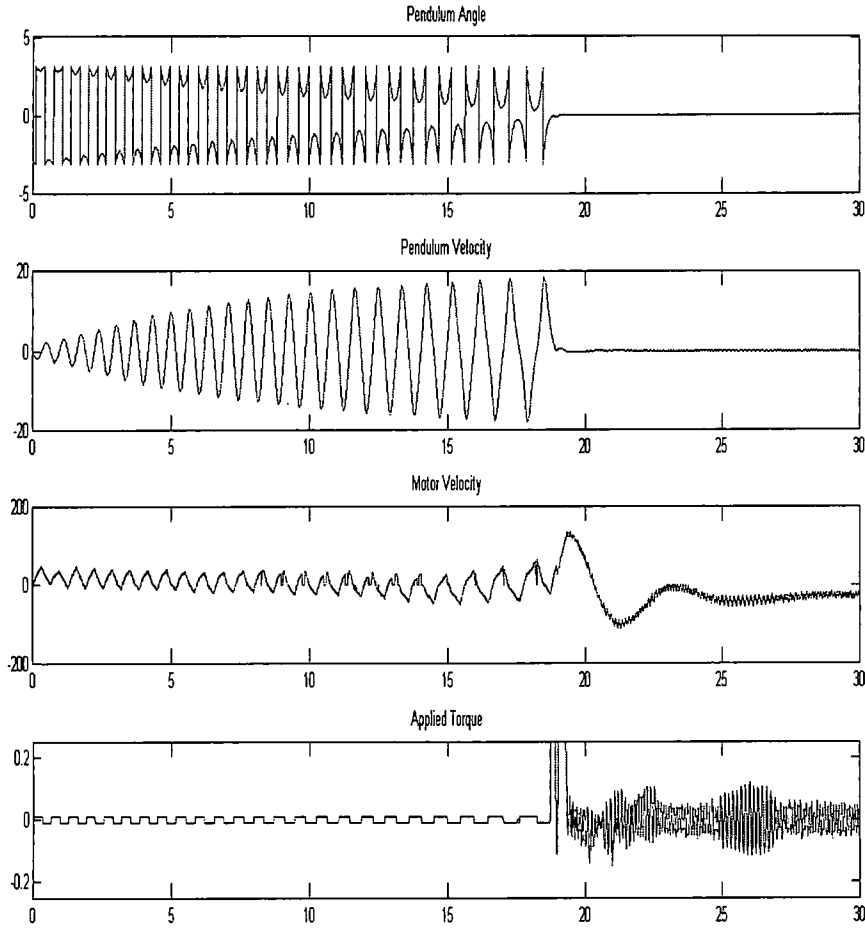


Figure 6.4: Experimental result: DAC with $u_k = u_{lqr}$.

6.5.1 Case 1 : Mass Added to the Motor

Two magnets were attached to the motor and to those magnets we attached two AA batteries as shown in Fig 6.5. This alters the mass of the motor, which in turn affects the mass of the pendulum and also its center of mass. The swing-up controller had to be re-tuned in order to make the pendulum reach the inverted position. It was tuned in the following manner.

- *if* ($angle < \pi$)
 - if* ($0 < angle < 0.5850$)
 - $torque = 0.0045;$
 - else*
 - $torque = 0.0093;$
- *if* ($angle > \pi$)
 - if* ($5.2962 < angle < 2\pi$)
 - $torque = -0.0045;$
 - else*
 - $torque = -0.0093;$

We found that all the controllers performed reasonably well in this case. Figs

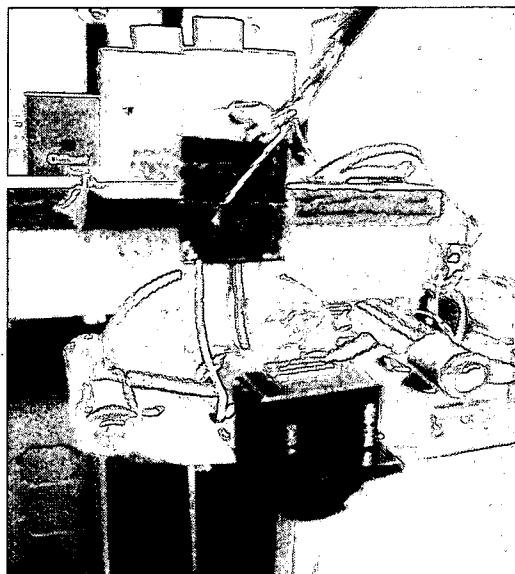


Figure 6.5: Mass added to motor.

6.6, 6.7 and 6.8 show the experimental results of the three controllers.

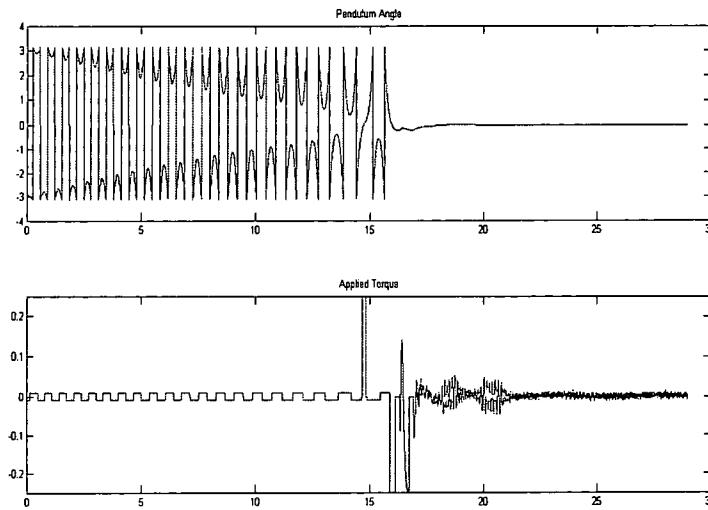


Figure 6.6: LQR Controller with mass added to motor.

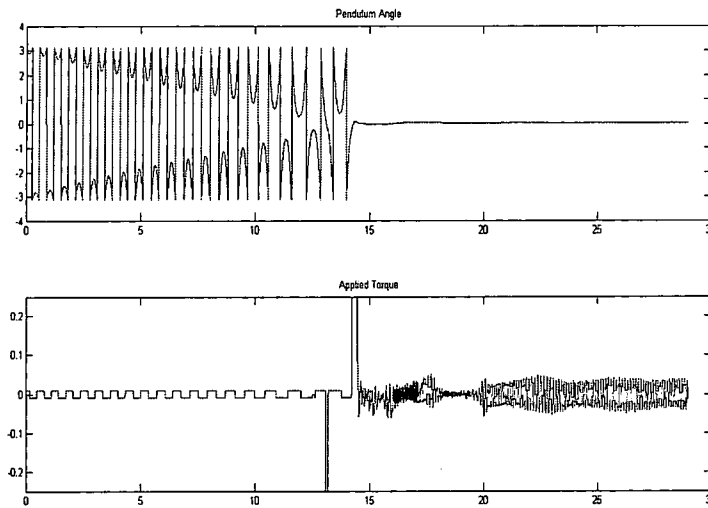


Figure 6.7: FBL Controller with mass added to motor.

6.5.2 Case 2 : Mass Added to the Rod

A block of modeling clay was attached to the rod as shown in the Fig 6.9. This alters the mass of the rod, thus affecting the mass of the system. In this case,

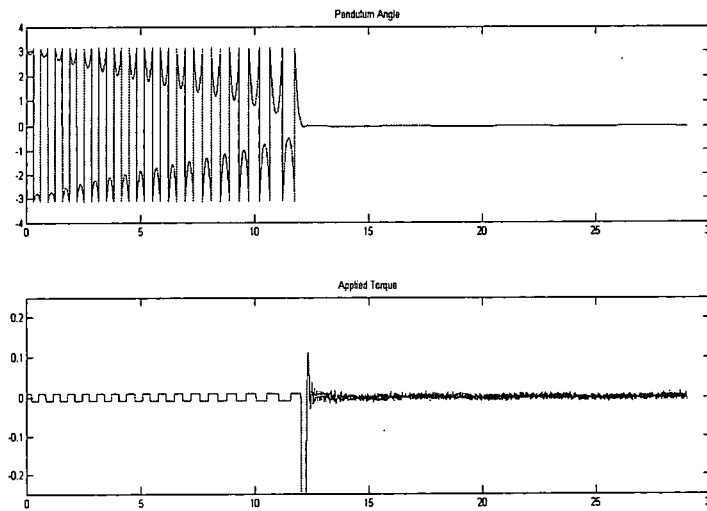


Figure 6.8: Adaptive Controller with mass added to motor.

the linear and the adaptive controllers were able to balance the pendulum but the feedback linearizing controller failed. Fig 6.10, 6.11 and 6.12 show the experimental



Figure 6.9: Mass added to rod

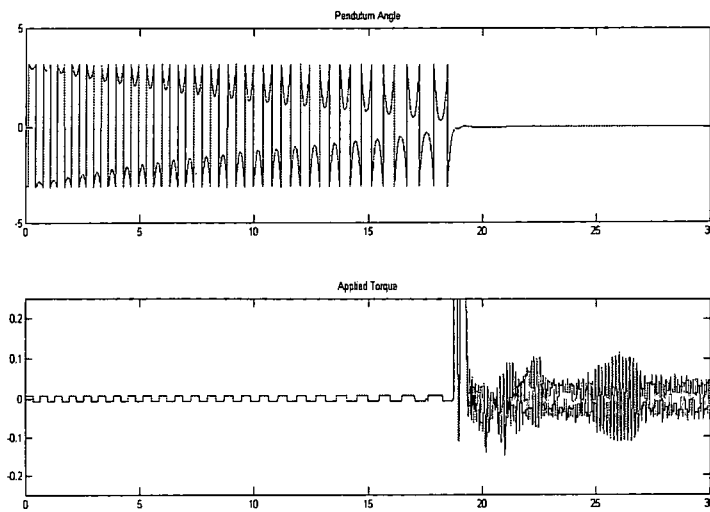


Figure 6.10: LQR Controller with mass added to the rod.

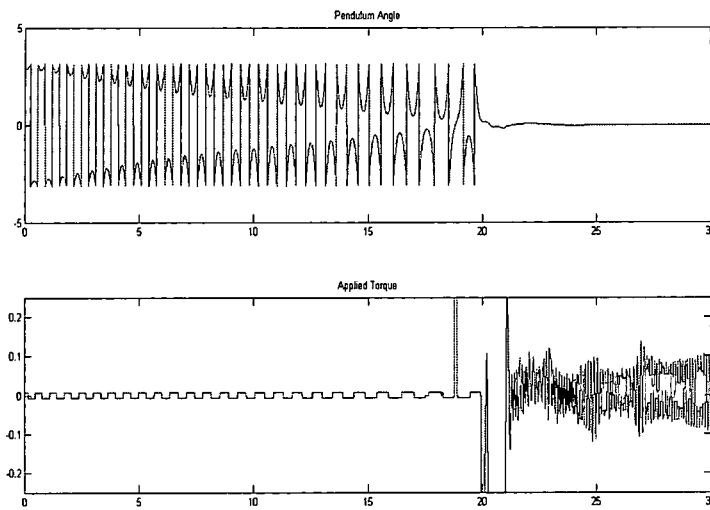


Figure 6.11: FBL Controller with mass added to the rod.

results of the three controllers.

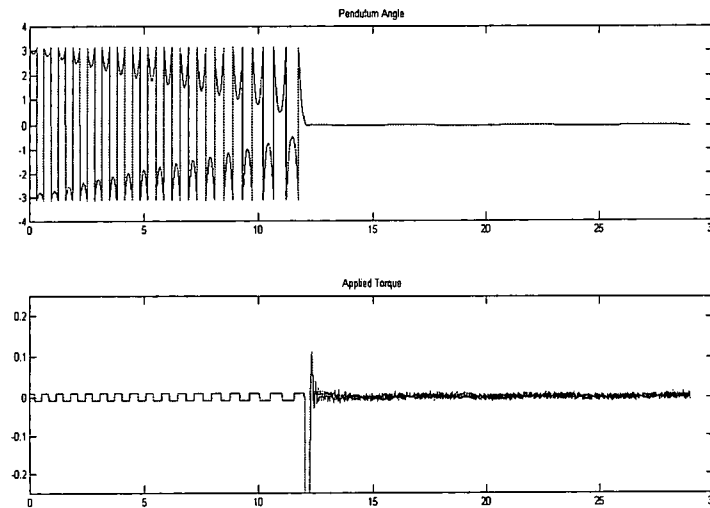


Figure 6.12: Adaptive Controller with mass added to the rod.

6.5.3 Case 3 : Mass Added to the wheel

One AA battery was attached to the wheel as shown in Fig 6.13. Now the mass of the wheel is changed. Also the system now loses symmetry. Here, both the linear and feedback linearizing controllers failed. Only the adaptive controller was able to balance the pendulum. In order to represent changing dynamics, some small magnets were attached to the battery after it reached the inverted position. The adaptive controller was able to tune itself and maintain the pendulum balanced even in this case. Fig 6.14 shows the experimental results of the adaptive controller in case 3. At the 40th second a disturbance was created by attaching magnets to the system and this is seen in the plot by sudden increase in torque. The results are summarized in Table 6.1.

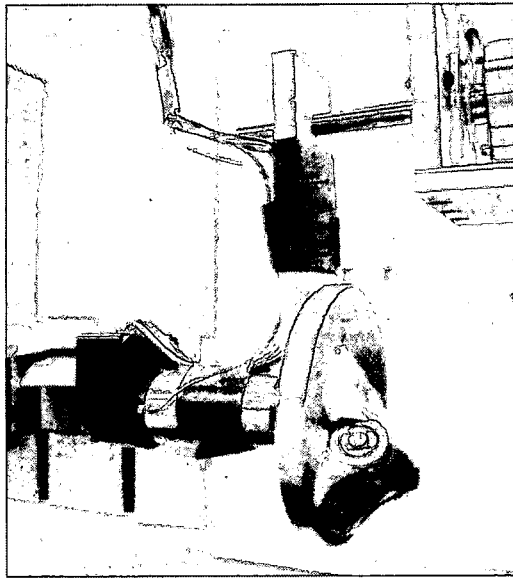


Figure 6.13: Mass added to wheel.

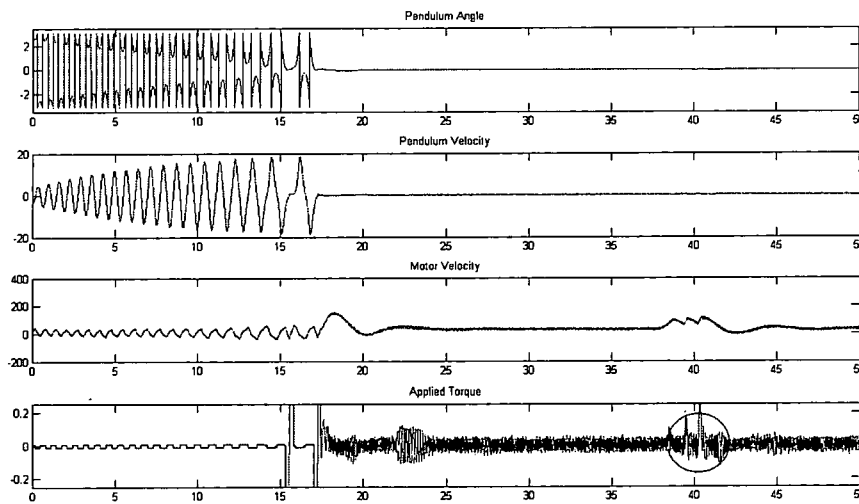


Figure 6.14: Adaptive Controller with mass added to the wheel.

Table 6.1: Results of the Case Studies

	Linear	Non-linear	Adaptive
Mass attached to motor	✓	✓	✓
Mass attached to Rod	✓	X	✓
Mass attached to Wheel	X	X	✓

CHAPTER 7

CONCLUSIONS

Three different controllers were designed and successfully implemented on the reaction wheel pendulum. The swing-up controller swings the pendulum up rapidly and makes it arrive within the catch area with low velocity. Then the balancing controller takes over and stabilizes the pendulum.

The swing-up controller implementation required a lot of trial and error and had to be tuned so that the pendulum arrived at the inverted position with less velocity. Also, it was observed that there is a significant difference between simulation and experimental results. It was realised that a big problem is to minimize the CPU utilization while keeping a good control performance. This is due to the trade off between these two requirements. The biggest issue was in handling the hardware. The H-Bridge, in spite of the heat sink, used to get burned and had to be replaced a few times. Also the wires from the motor and the encoder created a lot of disturbance for the pendulum.

The mean, variance and maximum value of the pendulum angle using the three different balancing controllers was calculated and is shown in Table 7.1. It was noted that the mean and variance were lesser for the adaptive controller while the maximum value was greater.

While all the three balancing controller were able to balance the pendulum, each of these techniques has advantages and drawbacks. The performance of the three

Table 7.1: Balancing Metrics

	Linear Controller	Non-linear Controller	Adaptive Controller
Mean	0.0013	0.0027	0.00165
Maximum	0.0487	0.0359	0.0623
Variance	0.0021	0.0023	0.00193

controllers was different. In terms of CPU utilization, the linear and non-linear controllers showed better performance. This was evident from the fact that while we could use 1000 samples/sec for these two controllers, for the adaptive controller it had to be reduced to 600 samples/second to avoid run-time errors.

In terms of control energy, the adaptive controller required much more energy than the linear and non-linear controllers. This was obvious from the graphs obtained and also by noting to the excessive heating of the motor. The control energies for the various controllers, calculated using least squares method, are shown in Table 7.2.

Table 7.2: Control Energy

	Linear Controller	Non-linear Controller	Adaptive Controller
Control Energy (in Nm)	0.0001	0.0008	0.0024

Even though the adaptive controller had the above mentioned two disadvantages, in terms of robustness, its performance surpassed that of the other two controllers. In spite of changes in the system as demonstrated by the three case studies, it was able to stabilize the system where the linear and non-linear controllers failed.

Thus, it can be concluded that the type of controller for a particular application depends on the requirements. If control energy and CPU utilization are of primary importance, linear and non-linear controllers are the best choices. But if robustness is

the issue, the adaptive controller is the best solution. The conclusions are summarised in table 7.3.

Table 7.3: Conclusions

	Linear Controller	Non-linear Controller	Adaptive Controller
Control Energy	low	low	high
CPU Utilization	low	low	high
Robustness	low	low	high

APPENDIX A

A convenient way to derive the equations of motion of mechatronic systems is the Lagrangian method. The Lagrangian method is very simple because it allows us to deal with scalar energy functions.

The Lagrangian method begins by defining a set of generalized coordinates which are typically position coordinates. Since the reaction wheel pendulum has two degrees of freedom, we take the generalized coordinates as angle θ_1 of the pendulum and angle θ_2 of the wheel.

In terms of these generalized coordinates we need to compute the kinetic and potential energies. In general, potential energy is only a function of the generalized coordinates while the kinetic energy is a function of the generalized coordinates and their derivatives. Kinetic energy for the system is the sum of the pendulum kinetic energy and rotor kinetic energy and can be written as

$$KE = \frac{1}{2}(m_1 l_{c1}^2 + I_1) \dot{\theta}_1^2 + \frac{1}{2}((m_2 + m_{motor})(l_1 - l_d)^2 + I_{motor}) \dot{\theta}_1^2 + \frac{1}{2}(I_2 + I_{rotor})(\dot{\theta}_1 + \dot{\theta}_2)^2$$

Potential energy for the system is calculated as

$$PE = mg(\cos \theta_1 - 1)$$

The Lagrangian L may now be defined as the difference between the kinetic and potential energies. L is therefore a function of the generalized coordinates and their derivatives.

$$L = KE - PE \tag{A-1}$$

$$L = \frac{1}{2}(m_1 l_{c1}^2 + m_2(l_1 - l_d)^2 + m_{motor}(l_1 - l_d)^2 + I_1 + I_2 + I_{motor})\dot{\theta}_1^2 + (I_2 + I_{rotor})\dot{\theta}_1\dot{\theta}_2 + \frac{1}{2}(I_2 + I_{rotor})\dot{\theta}_2^2 - mg(\cos \theta_1 - 1) \quad (A-2)$$

The equations of motion can now be found from

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_k} \right) - \frac{\partial L}{\partial \theta_k} = \tau_k, k = 1, \dots, n \quad (A-3)$$

where τ_k represents the generalized torque/force in the θ_k direction and n is the number of degrees of freedom of the system.

For the reaction wheel pendulum, the equations of motion are given by

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} &= 0 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} &= \tau \end{aligned} \quad (A-4)$$

where τ is the control input, the torque generated by a motor. Following equations (A-2) and (A-4), we get

$$\begin{aligned} (m_1 l_{c1}^2 + m_2(l_1 - l_d)^2 + m_{motor}(l_1 - l_d)^2 + I_1 + I_2 + I_{motor})\ddot{\theta}_1 + (I_2 + I_{rotor})\ddot{\theta}_2 - mg \sin \theta_1 &= 0 \\ (I_2 + I_{rotor})\ddot{\theta}_1 + (I_2 + I_{rotor})\ddot{\theta}_2 &= \tau \end{aligned}$$

Thus the mathematical model of the reaction wheel pendulum can be written as,

$$\begin{aligned} d_{11}\ddot{\theta}_1 + d_{12}\ddot{\theta}_2 - mg \sin \theta_1 &= 0 \\ d_{21}\ddot{\theta}_1 + d_{22}\ddot{\theta}_2 &= \tau \end{aligned} \quad (A-5)$$

where

$$\begin{aligned} d_{11} &= m_1 l_{c1}^2 + m_2(l_1 - l_d)^2 + m_{motor}(l_1 - l_d)^2 + I_1 + I_2 + I_{motor} \\ d_{12} &= I_2 + I_{rotor} = d_{21} = d_{22} \end{aligned}$$

APPENDIX B

The LMD18201 is a 3A H-Bridge designed for motion control applications. The device is built using a multi-technology process which combines bipolar and CMOS control circuitry with DMOS power devices on the same monolithic structure. The H-Bridge configuration is ideal for driving DC and stepper motors. The LMD18201 accommodates peak output currents up to 6A. Current sensing can be achieved via a small sense resistor connected in series with the power ground lead.

Figure B.1 shows the functional block diagram. Fig B.2 shows the pin diagram for the LMD18200.

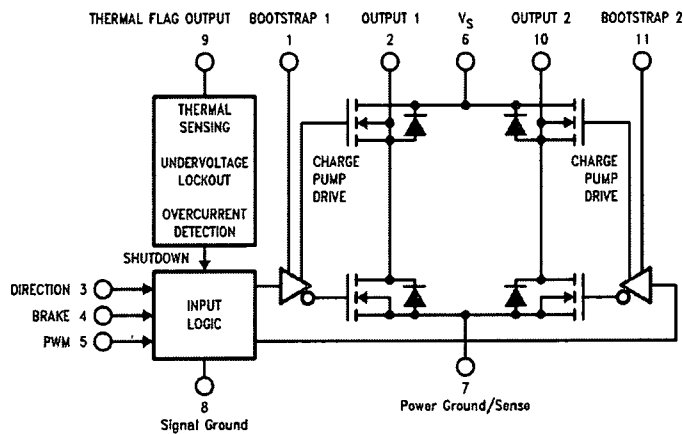


Figure B.1: Functional block diagram of the LMD18200 (taken from [1])

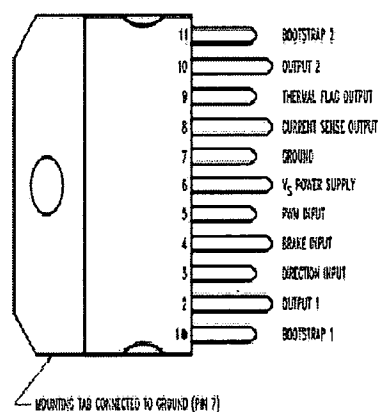


Figure B.2: Pin diagram of the LMD18200 (taken from [1])

Absolute Maximum Ratings

Total Supply Voltage (VS, Pin 6)	60V
Voltage at Pins 3, 4, 5 and 9	12V
Voltage at Bootstrap Pins (Pins 1 and 11)	$V_{OUT}+16V$
Peak Output Current (200 ms)	6A
Power Dissipation ($T_A = 25^{\circ}C$, Free Air)	3W
Junction Temperature, $T_J(\max)$	150°C

The connections to the IC are as follows and are also depicted in Fig B.3

Pin 1, BOOTSTRAP 1 Input: Bootstrap capacitor pin for half H-Bridge number

1. The recommended capacitor (10 nF) is connected between pins 1 and 2.

Pin 2, OUTPUT 1: Half H-Bridge number 1 output.

Pin 3, DIRECTION Input: This input controls the direction of current flow between OUTPUT 1 and OUTPUT 2 (pins 2 and 10) and, therefore, the direction of rotation of a motor load.

Pin 4, BRAKE Input: This input is used to brake a motor by effectively shorting its terminals.

Pin 5, PWM Input: See Table 1. How this input (and DIRECTION input, Pin 3) is used is determined by the format of the PWM Signal.

Pin 6, VS Power Supply

Pin 7, POWER GROUND/SENSE Connection: This pin is the ground return for the power DMOS transistors of the H-Bridge. The current through the H-Bridge can be sensed by adding a small, 0.1W, sense resistor from this pin to the power supply ground.

Pin 8, SIGNAL GROUND: This is the ground return for the internal logic circuitry used to control the PWM switching of the H-Bridge.

Pin 9, THERMAL FLAG Output: This pin provides the thermal warning flag output signal. Pin 9 becomes active-low at 145C (junction temperature). However the chip will not shut itself down until 170C is reached at the junction.

Pin 10, OUTPUT 2: Half H-Bridge number 2 output.

Pin 11, BOOTSTRAP 2 Input: Bootstrap capacitor pin for half H-Bridge number 2. The recommended capacitor (10 nF) is connected between pins 10 and 11.

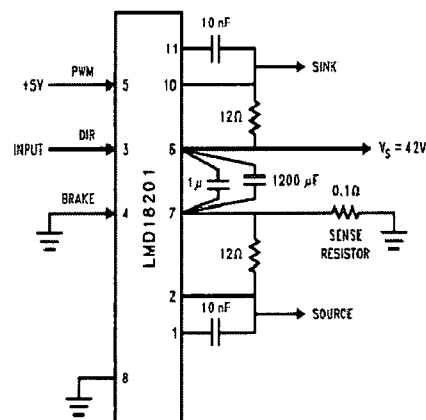


Figure B.3: Circuit diagram (taken from [1])

TYPE OF PWM SIGNAL USED

The type of PWM used here is sign/magnitude PWM which consists of separate direction (sign) and amplitude (magnitude) signals (see Figure B.4). The (absolute) magnitude signal is duty-cycle modulated, and the absence of a pulse signal (a continuous logic low level) represents zero drive. Current delivered to the load is proportional to pulse width. For the LMD18201, the DIRECTION input (pin 3) is driven by the sign signal and the PWM input (pin 5) is driven by the magnitude signal.

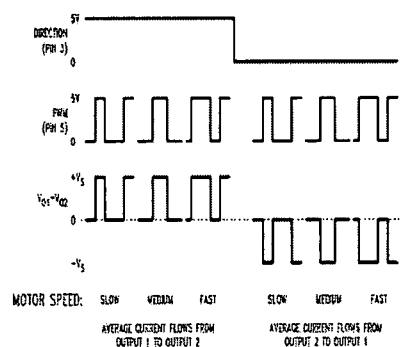


Figure B.4: Sign/Magnitude PWM Control

APPENDIX C

MATLAB CODE:

The various parameters required by the controller are calculated using the file parameters.m which is given below.

```
g=9.81;      % Gravity
```

% ROD DETAILS

```
l1=0.133;      % length of the rod
b=0.03;        % Breadth of the rod
thickness=0.00635; % Thickness of rod
rod=0.060;     % Mass of the rod
```

% MOTOR DETAILS

```
motor=0.116;    % mass of motor in KG by measuring
radius=0.013;   % radius of the motor in m
ka=0.0283;     % torque constant
Ra=17.9;       % Armature resistance
```

%DISC DETAILS

```
rdisc=.04;      % radius of the disc
mdisc=0.0834;   % actual measured weight
```

% CALCULATING THE CENTER OF MASS AND TOTAL MAS OF THE SYSTEM

m1=rod; %Mass of the pendulum (in kg)(including the masss of rod)

m2=mdisc+motor;

lc1=(m2*l1+m1*(l1/2))/(m1+m2); % Calculating the center of mass

m=m1*(l1/2)+m2*(l1); %Combined mass

% CALCULATING THE MOMENT OF INERTIAS

I1=m1*(l1^2+b^2)/12;

I2=mdisc*rdisc*rdisc/2;

Irotor=0.00106/1000;

Imotor=(motor*radius*radius)/2;

% Calculating the D matrix

d11=I1+I2+m1*((l1/2)^2)+m2*((l1)^2);

d12=I2;

d21=I2;

d22=I2;

detd=d11*d22-d12*d21;

Code for Swing-up Controller:

if (0<x(2) & x(2)<0.5)

flag1=1;

counter=1;

else

```

        flag1=0;

end

if (flag1) % if flag1=1 then change direction else do not change direction
    if (x(1)-pi>=0)
        u=swing;direction=1;
    else
        u=-swing;direction=0;
    end
else
    if (direction)
        u=swing;
        direction=1;
    else
        u=-swing;
        direction=0;
    end
end
end

```

Code for LQR Controller:

```

A=[0 1 0 0;d22*m*g/detd 0 0 0;0 0 0 1;-d21*m*g/detd 0 0 0];
B=[0;-d12/detd;0;d11/detd];
C=[d11 0 d12 0];
D=0;
G=lqr(A,B,diag([1,0.1,0.01,0.001]),800);
u=-G*y;

```

Code for Feedback Linearizing Controller:

```
k1=250;k2=400;k3=108;k4=50;

v=k1*(d11*y(1)+d12*y(3))+k2*(d11*y(2)+d12*y(4))+k3*(m*g*sin(y(1)))+k4*(m*g*(cos(y(1)))*y
num=-v-((-y(2)^2*m*g*sin(y(1)))+(m*g)^2*d22*cos(y(1))*sin(y(1))/detd));
den=-d12*m*g*cos(y(1))/detd;

u=num/den;
```

Code for Direct Adaptive Controller:

```
x1=x(1);%pend position
x2=x(2);%pend velocity
x3=x(3);%disc position
x4=x(4);%disc velocity

% reference model:
xr(1)=0; xr(2)=0; xr(3)=0; xr(4)=0;

% controller parameters
g0 = -80;      % lower limit for g
B = 200;      %
W = 0.1;      % a guess
gamma=.01;

k=8; k1=5; k2=500; k3=1000; k4=800;

%transformed states
z(1)=d11*x(1)+d12*x(3); z(2)=d11*x(2)+d12*x(4); z(3)=m*g*sin(x(1));
```

```
z(4)=m*g*(cos(x(1)))*x(2);
```

```
% error manifold
```

```
e = k1*(z(1)-xr(1)) + k2*(z(2)-xr(2)) +k3*(z(3)-xr(3))
```

```
+k4*(z(4)-xr(4)); chi = k1*z(2) +k2*z(3)+k4*z(4);
```

```
e_bar= -chi - k*e;      % let this be an input to improve the approximation
```

```
% function approximator
```

```
alpha=[0.1*ones(1,6),1]; [Fu,zeta] =
```

```
Fu_app([z(1);z(2);z(3);z(4);e_bar], theta_hat,alpha);
```

```
% continuous stabilizing term
```

```
eps=0.1; u_s = -(W - B*abs(e)/(2*g0^2))*sat(e/eps);
```

```
% know controller
```

```
u_k = 0;
```

```
u_adaptive = Fu + u_s + u_k; sigma=1;
```

```
% adaptation laws
```

```
theta_hat_dot = gamma*zeta*e-sigma*theta_hat;
```

```
u=u_adaptive;
```

Simulink model:

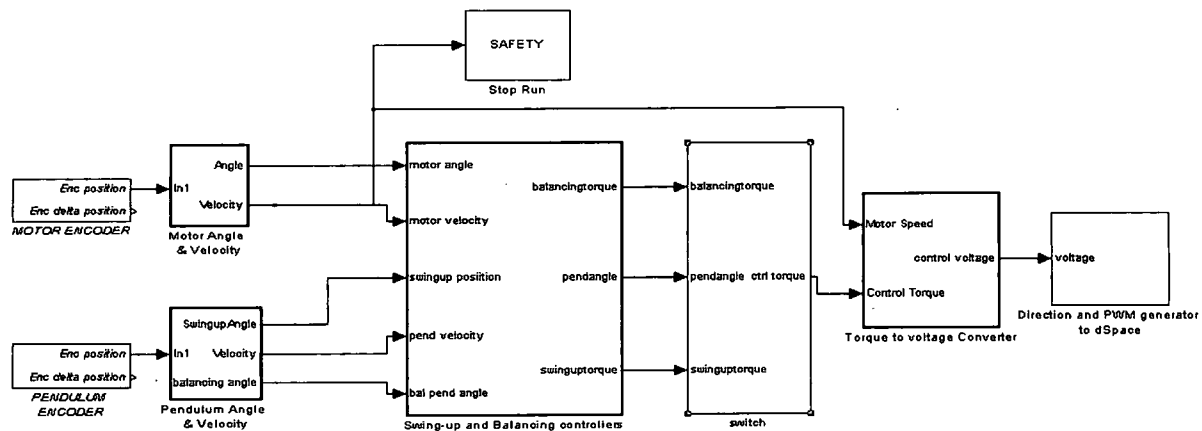


Figure C.1: Simulink model.

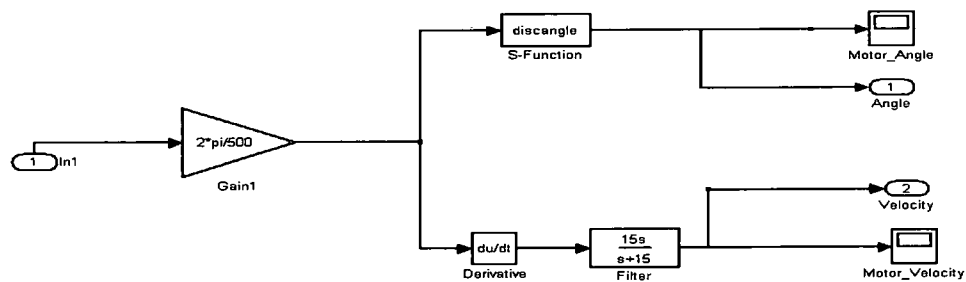


Figure C.2: Motor angle and velocity.

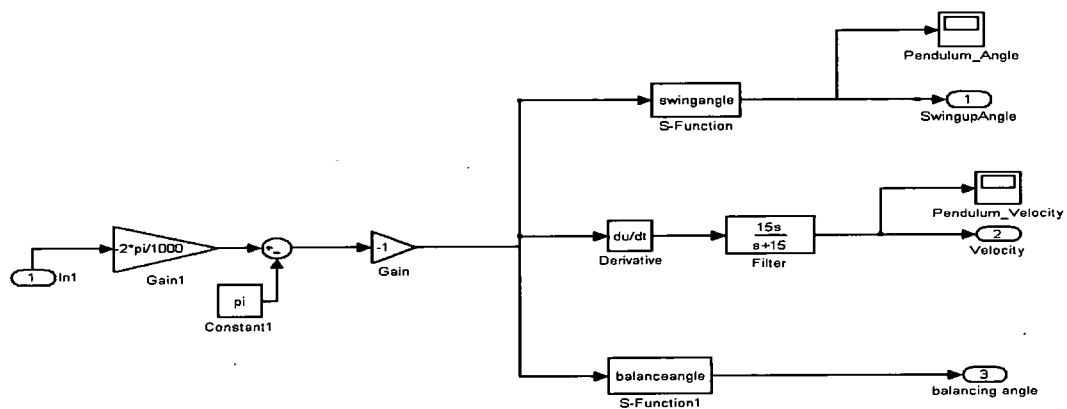


Figure C.3: Pendulum angle and velocity.

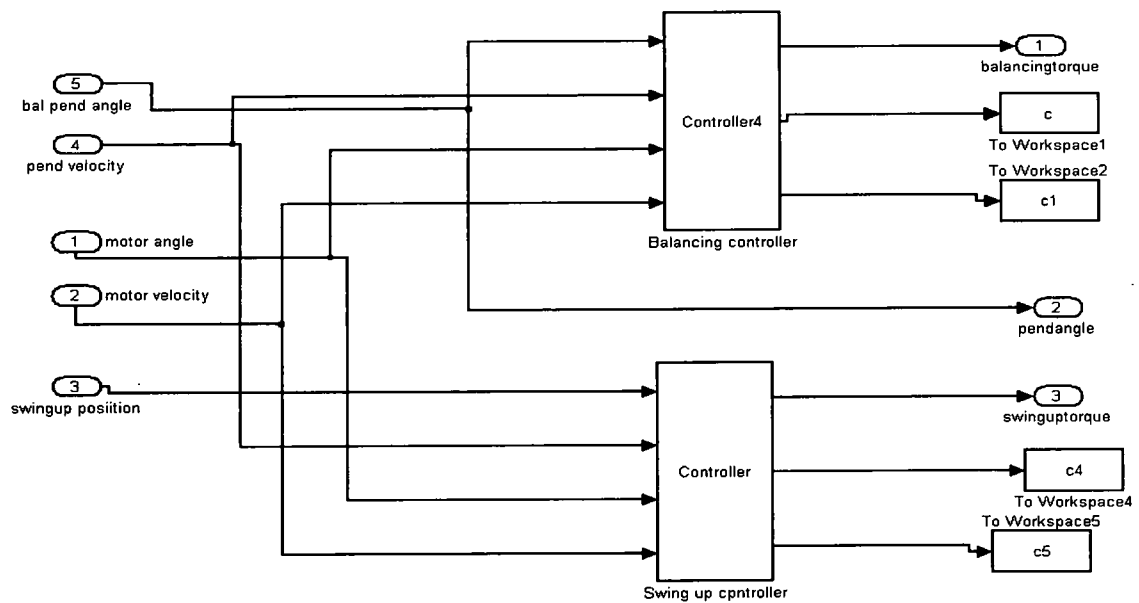


Figure C.4: Controllers.

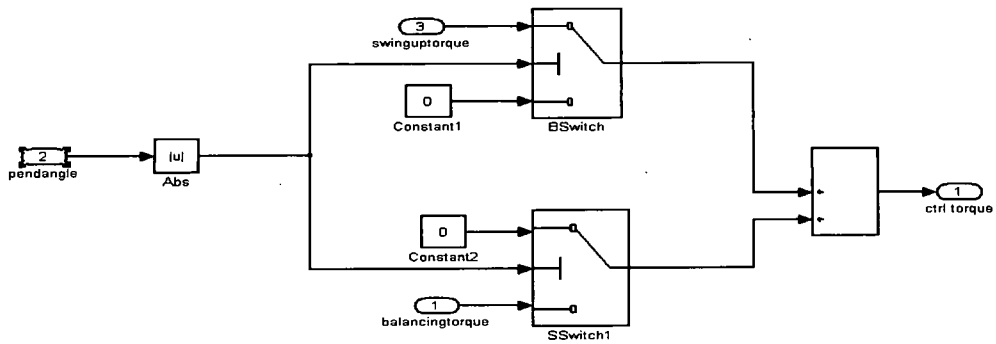


Figure C.5: Switch.

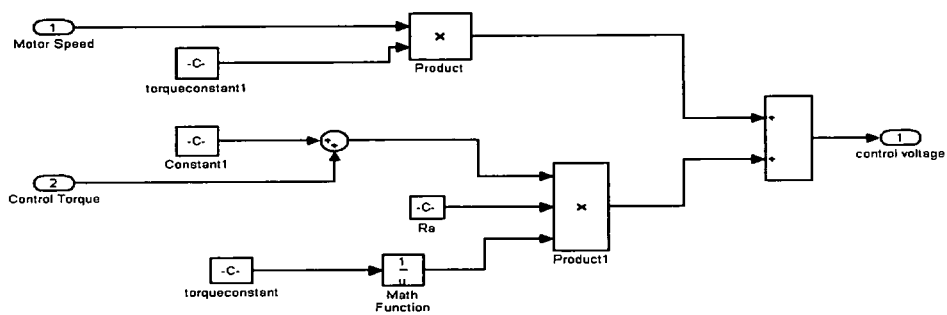


Figure C.6: Torque to voltage converter.

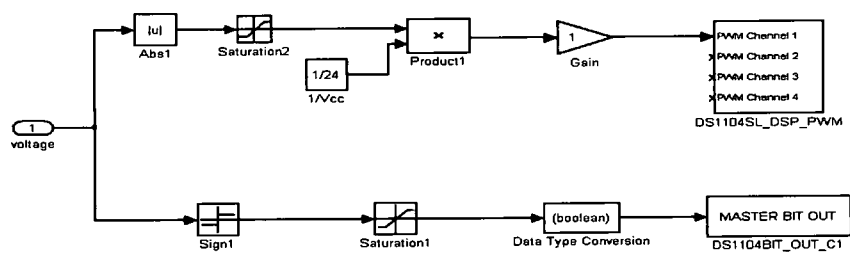


Figure C.7: Direction and PWM generator.

BIBLIOGRAPHY

- [1] National Semiconductor, *LMD 18200 Data Sheet*.
- [2] J. J. Craig, *Introduction to robotics: mechanics and control*. Addison-Wesley, 1989.
- [3] M. S. de Queiroz, D. M. Dawson, S. P. Nagarkatti, and F. Zhang", "*Lyapunov-Based Control of Mechanical Systems*". Springer, 2000.
- [4] C.-T. Chen, *Linear System Theory and Design*. New York: Oxford University Press, 1999.
- [5] A. Lur'e and V. Postnikov, "On the theory of stability of control systems,,"
- [6] H. K. Khalil, ed., *Nonlinear Systems- 3rd ed*. Upper Saddle River, New Jersey: Prentice Hall, 1996.
- [7] M. W. Spong, P. Corke, and R. Lozano, "Nonlinear control of the inertia wheel pendulum," *Automatica*, pp. 1845–1871, 2001.
- [8] J. W. Hauser, *Approximation of Nonlinear Functions for Fixed-Point and Asic Applications Using A Genetic Algorithm*. PhD thesis, University of Cincinnati, 2001.
- [9] R. Ordonez, J. Zumberge, J. Spooner, and K. M. Passino, "Adaptive fuzzy control: Experiments and comparative analyses," *IEEE Trans. on Fuzzy Systems*, vol. 5, pp. 167–188, Aug. 1997.
- [10] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Trans. Computers*, vol. 3, no. 2, pp. 1329–1333, 2000.
- [11] T. Poggio and F. Girosi, "A theory of networks for approximation and learning," Tech. Rep. AIM-1140, 1989.
- [12] R. J. Schilling, J. J. C. Jr., and A. F. Al-Ajlouni, "Approximation of nonlinear systems with radial basis function neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 1, pp. 1–15, 2001.
- [13] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: Wiley, 1989.

- [14] I. Fantoni and R. Lozano, eds., *Non-linear Control for Underactuated Mechanical Systems*. Great Britain: Springer, 2002.
- [15] R. Olfati-Saber, "Control of underactuated mechanical systems with two degrees of freedom and symmetry," *IEEE Trans. Computers*, vol. 6, pp. 4092–4096, 2000.
- [16] H. Kimura and S. Kobayashi, "Efficient non-linear control by combining q-learning with local linear controllers," in *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, (San Francisco, CA, USA), pp. 210–219, Morgan Kaufmann Publishers Inc., 1999.
- [17] H. Thomas and E. Sebastian, "Energy based control of the pendubot," *IEEE Trans. Automatic Control*, vol. 45, pp. 725–729, 2000.
- [18] M. Widjaja and S. Yurkovich, "Intelligent control for swing up and balancing of an inverted pendulum system," *Proc. of IEEE Conference on Control Applications*, Albany, NY, Sept. 1995.
- [19] K. J. Åström and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley Publishing Company, 1989.
- [20] K. J. Astrom and K. Furuta, "Swinging up a pendulum by energy control," *Automatica*, vol. 36, pp. 2193–2198, 2000.
- [21] V. Arjuva, "Nonlinear control systems laboratory," Master's thesis, University of Dayton, 2003.
- [22] C. Anderson, "Learning to control an inverted pendulum using neural networks," *IEEE Control Systems Magazine*, vol. 9, no. 3, pp. 31–37, 1989.
- [23] M. Yamakita, M. Iwashiro, Y. Sugahara, and K. Furuta, "Robust swing up control of double pendulum," in *Proceedings of the American Control Conf.*, vol. 1, pp. 290–295, 1995.
- [24] S. A. Bortoff, "Robust swing-up control for a rotational double pendulum," in *Proc. IFAC World Congress (Vol. F)*, (San Francisco), pp. 413–418, July 1996.
- [25] H. M. zu Farwig and H. Unbehauen, "Discrete computer control of a triple-inverted pendulum," *Optim. Control Appl. Methods*, vol. 11, no. 2, pp. 157–169, 1990.
- [26] J. Prasad, "Design and development of a reaction wheel pendulum," Master's thesis, University of Dayton, 2004.
- [27] R. Olfati-Saber, "Global stabilization of a flat underactuated system: the inertia wheel pendulum," in *Proceedings of the 40th IEEE Conference on Decision and Control, 2001*, vol. 4, pp. 3764 – 3765, 2001.
- [28] G. Chauveau, D. Chazal, D. Nakayama, E. Olsen, and S. Palm, "Controlling the reaction wheel pendulum," 2005.

R002592863