Design, Modeling and Control of an Inverted Pendulum on a Cart

by

Vignesh Namasivayam

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2020 by the
Graduate Supervisory Committee:

Konstantinos Tsakalis, Chair
Armando Rodriguez
Jennie Si
Md. Ashfaque Bin Shafique

ARIZONA STATE UNIVERSITY

May 2021

ABSTRACT

The Inverted Pendulum on a Cart is a classical control theory problem that helps understand the importance of feedback control systems for a coupled plant. In this study, a custom built pendulum system is coupled with a linearly actuated cart and a control system is designed to show the stability of the pendulum. The three major objectives of this control system are to swing up the pendulum, balance the pendulum in the inverted position (i.e. $180°$), and maintain the position of the cart. The input to this system is the translational force applied to the cart using the rotation of the tires.

The main objective of this thesis is to design a control system that will help in balancing the pendulum while maintaining the position of the cart and implement it in a robot. The pendulum is made free rotating with the help of ball bearings and the angle of the pendulum is measured using an Inertial Measurement Unit (IMU) sensor. The cart is actuated by two Direct Current (DC) motors and the position of the cart is measured using encoders that generate pulse signals based on the wheel rotation. The control is implemented in a cascade format where an inner loop controller is used to stabilize and balance the pendulum in the inverted position and an outer loop controller is used to control the position of the cart. Both the inner loop and outer loop controllers follow the Proportional-Integral-Derivative (PID) control scheme with some modifications for the inner loop.

The system is first mathematically modeled using the Newton-Euler first principles method and based on this model, a controller is designed for specific closed-loop parameters. All of this is implemented on hardware with the help of an Arduino Due microcontroller which serves as the main processing unit for the system.

To my Family, Friends and Teacher's for their constant support and encouragement towards all my endeavors.

ACKNOWLEDGMENTS

First and foremost, I would like to thank Dr. Konstantinos Tsakalis for guiding me throughout this research with dedication and patience. He always had the time to answer my questions and helped me in building my knowledge in control theory and its practical applications.

Secondly, I would like to thank Dr. Md. Ashfaque Bin Shafique, who found the potential in me and motivated me to take up this research. He was always ready to answer my questions regardless of the time of day or night. I am thankful to both my professors for all the long conversations and for taking their time to help me understand and move forward with my research work.

I owe my sincere gratitude to my friend Vishwam Aggarwal for helping me understand major concepts in modeling and control theory. I also want to thank him for his constant support, encouragement, and motivation. I also would like to thank my friend Sai Sravan who helped with hardware components to complete my research.

I would also like to thank my defense committee members, Dr. Armando Rodriquez for explaining to me the key concepts of feedback systems and Dr. Jennie Si for introducing me to Artificial Neural Networks and Machine Learning.

Finally, none of this would have been made possible without the constant love and support of all my family members and dear friends. They were always there for me in times of need and motivated me through all my tough times.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Overview

The Inverted Pendulum on a Cart system has been a great platform for engineers to research and develop different types of applications like self-balancing scooters, segways etc. It is a classical problem used for learning the ideas of a coupled system. For a controls engineer, this system gives a very good insight into developing different feedback control mechanism.

The Inverted Pendulum on a Cart system has a pendulum subsystem and a cart subsystem which are coupled together by mechanical design. Due to the coupled nature of this system, an effect on one subsystem also affects the other subsystem. In this thesis, the pendulum system on its own is considered as a rigid body that has a weight attached to one end while the other end is fixed to a pivot so that the pendulum can swing freely about the rotational axis of the pivot. The pendulum system always remains vertical when at rest due to the effect of gravitational force. This resting point of the pendulum is called the stable equilibrium of the system (i.e the system will always converge to this point in any given point of time). The pendulum will also have the same effect of the gravitational field when it is displaced $180°$ from its resting point, this point is called the unstable equilibrium of the system because even a very small disturbance in the inverted position will make the system unstable. The main objective of this thesis is to swing up and stabilize the pendulum around its unstable equilibrium point which is similar to balancing a long rod with our hand where our hand plays an important role in keeping the rod steady at all times.

Similarly, a cart subsystem is designed and coupled with the pendulum subsystem so that it can balance the pendulum when it is in the inverted position. The cart subsystem is constructed with the help of two high powered DC motors which are used for actuation of the system. To balance the pendulum, the cart has to accelerate in one direction which provides a torque to swing the pendulum. Therefore, a controller designed for this type of system must have an inverse response due to the effect of right half plane zeros present in the system (i.e For the cart to go right, it must first move left and unbalance the pendulum).

This system can be either modeled using the Newton-Euler method or the Euler-Lagrange method. In this thesis, the Newton-Euler approach of first principles modeling is used to derive the equations of motion. Based on the equations of motion Force is the only input given into the system. But, Force can only be used in a simulation environment as it cannot be measured accurately in a real-world system. Hence, the equations of motion should replace the Force input with the dynamics of the two DC motors used so that a better approximate model of the system is derived.

Several research papers show different methods to build this system and the motivation for this thesis is the design, model, and construction of a robot that balances a free rotating pendulum with the help of digital sensors like accelerometer and gyroscope which are used to estimate the angle of the pendulum and a motor encoder sensor to estimate the position of the robot. Using these sensors to balance an inverted pendulum while maintaining the position of the cart makes this problem more interesting and difficult to control.

1.2 Literature Review

The Inverted Pendulum on a Cart problem has been in the control theory world since the early 1950's enabling engineers with abundant research area. The idea of this problem is to understand the effects of linear controllers on unstable systems like the pendulum model [26, 19]. Following the history shown in [15], many developments give a clear insight into this control problem.

Many sources give the idea of modeling this type of nonlinear system either by following the Newton-Euler modeling or Euler- Lagrange modeling approach [5, 24, 25]. Most of the research for these type of systems is done on a cart which is attached to a rail like track actuated by a DC motor and a belt [22]. The swinging up of the pendulum using energy control was proposed in [3] and there are several developments added to that research [8, 34, 20, 22, 14]. Later the restrictions of cart track length were considered into design [7, 33].

Controllers for this system are designed based on the Linearized model following the root locus approach, PID control approach [24]. There are developments with the use of Fuzzy-logic controllers over PID controller design [21], and other nonlinear approaches to swing and balance the pendulum[8]. In this thesis, the focus of the linear controller is based on Frequency loop shaping approaches which help in better stabilization of the system [11, 12, 13, 29, 28, 30]. [23] discusses the idea of swinging the pendulum and controlling using the frequency loop shaping adaptive control design. [2] gives an idea about Discrete time PID control tuning using Frequency loop shaping. There are other modern techniques to design controllers for this system using the ideas of LQR, LQG control design which can be extended as future work of this thesis.

The purpose of this thesis is to design a control system to swing up and balance the inverted pendulum on a cart actuated by two DC motors as shown in Figure 4.1. The DC motor model [24, 27] is added to the nonlinear system of equations to develop a better approximate model. A new idea of energy control is proposed in Section 5.3 based on the ideas from [3]. The controller is designed and implemented on an Arduino Due microcontroller [10] based on the first principles model and the performance on hardware is shown in Chapter 5.

1.3 Report Organization

This thesis report is organized into 6 chapters.

- Chapter 1 is the Introduction chapter for this thesis.
- Chapter 2 explains the Modeling of the Inverted Pendulum on a Cart system using Newton-Euler's first principles method.
- Chapter 3 describes the design and hardware components used to construct the robot.
- Chapter 4 briefly shows the derivation of parameters for the inverted pendulum on a cart system.
- Chapter 5 explains in detail the types of controllers used in this thesis along with the results based on the first principles model
- Chapter 6 concludes the thesis with details and discusses the future work that can be done on this robot.

Chapter 2

MODELING OF INVERTED PENDULUM ON A CART

2.1 Introduction

The Inverted Pendulum on a Cart shown in Figure 2.1 is modeled based on the Newton-Euler first principles method. This system has one input which is translational force F which displaces the cart, and the pendulum connected to it. The two outputs of concern in this system are the position of the cart and the angle of the pendulum.

Consider a rigid pendulum of length l, with mass m attached to one end of the pendulum while the other end is fixed to a rotating pivot attached to a cart with mass M, which displaces when a force F is applied to it. x is the displacement of the cart and pendulum, and y is the displacement of the cart. The angle of the pendulum is θ. Both the cart and the pendulum experiences frictional forces which are represented as c_c and c_p respectively. The acceleration due to gravity in the system is represented as g.

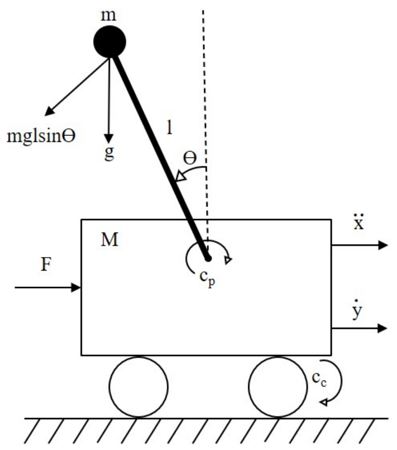

Figure 2.1: Inverted Pendulum on a Cart

2.2 Newton Euler Equations

When a force F is applied to the cart and pendulum system, it accelerates along the direction of the force applied based on Newton's second law of motion. The cart also experiences a retarding force due to the friction between wheels and ground. So, we can write the equations of motion

$$(m+M)\ \ddot{x} = F - c_c\ \dot{y} \tag{2.1}$$

From Equation 2.1, we can derive the cart's position

$$y = x - \frac{lmsin\theta}{m+M} \tag{2.2}$$

From Equations 2.1 and 2.2 we can obtain the equation for the cart's motion

$$(m+M)\ \ddot{y} = F - c_c\ \dot{y} - ml\ddot{\theta}cos\theta + ml\dot{\theta}^2 sin\theta \tag{2.3}$$

The pendulum rotates about a pivot axis and the acceleration of the pendulum is reduced due to the effect of gravity and frictional force. Summing up these tangential forces, we get

$$ml^2\ddot{\theta} = -c_p\ \dot{\theta} - mglsin\theta - ml\ddot{y}cos\theta \tag{2.4}$$

Equations 2.3 and 2.4 describes the nonlinear model of an Inverted Pendulum on a Cart system. Here, both the equations are dependent on each other and hence realized as a Single Input Multi Output System (SIMO). The cart system has two states y and $\dot{y}$ i.e cart position in meters (m) and cart velocity in meters / second (m/sec) respectively. The pendulum also has two states θ and $\dot{\theta}$ i.e pendulum angle in radians (rad) and pendulum angular rate in radians / second (rad/sec). Both the states of the cart system are 0 at equilibrium. The pendulum has two equilibrium points $\theta = 0$ which is the stable equilibrium and $\theta = \pi$ which is the unstable equilibrium.

In this thesis we deal with the pendulum in inverted position, therefore $\theta = \pi$ and $\dot{\theta} = 0$ at equilibrium. In order to get the linear model, Equations 2.3 and 2.4 are linearized about the equilibrium points.

Before linearization, Equation 2.3 is substituted in Equation 2.4 and vice-versa to get two equations with all the four states. The output of the system is cart position y and pendulum angle θ. The only input to this coupled system is the force F which can be used in a simulation environment, but in real-world systems, the force cannot be provided as an input. The cart is actuated by two wheels that are connected to two DC motors. Hence, the dynamics of the DC motor should be modeled in the system equations to replace the force.

2.3 DC Motor Model

The DC motor produces angular velocity ω, and torque τ for an applied input voltage. The DC motor circuit is shown in Figure 2.2. The input voltage to the motor is Vin, Resistance of the armature is R, Inductance of the armature is L, the current flowing through the armature is i and the back emf produced is Ve.

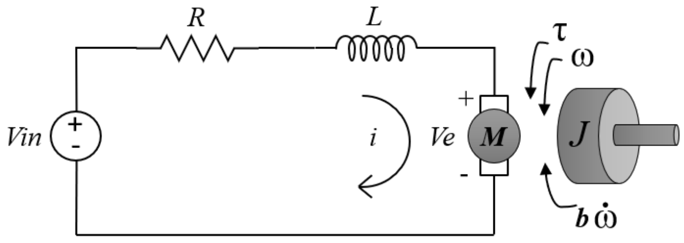

Figure 2.2: DC Motor Circuit

In the DC motor, the motor torque τ is proportional to the armature current i by a motor torque constant K_τ.

$$\tau = K_\tau i \tag{2.5}$$

The back emf e is proportional to the angular velocity of the shaft by a back emf constant K_e.

$$e = K_e \dot{\omega} \tag{2.6}$$

Both K_τ and K_e are given by the motor manufacturer. Ideally, for a DC motor, mechanical power output is equal to the electrical power. Therefore, theoretically K_τ and K_e are equal and represented as K [24, 18].

$$K_\tau = K_e = K \tag{2.7}$$

The DC motor model can be derived using Krichoff's Voltage Law and Newton's second law.

$$V(t) = Ri(t) + L\frac{di}{dt} + K\dot{\omega}(t) \tag{2.8}$$

$$J\ddot{\omega}(t) = Ki(t) - b\dot{\omega}(t) \tag{2.9}$$

Here J is the Moment of inertia of the rotor, and b is the motor friction constant. Now writing Equation 2.8 and Equation 2.9 in the form of a state space representation

$$\begin{bmatrix} \dot{x1} \\ \dot{x2} \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} u \tag{2.10}$$

where the states are $x1 = \dot{\omega}(t)$, $x2 = \dot{i}(t)$ and $u = V(t)$. This can be represented as shown in Figure 2.3

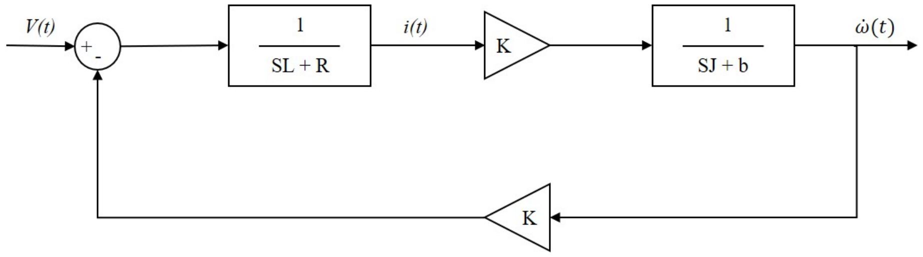

Figure 2.3: DC Motor Block Diagram Representation

In the DC motor circuit, there will be no change in current after a short period which results in no effect of inductance L. Therefore, when current reaches steady state $L \approx 0$ and Equations 2.8 and 2.9 becomes,

$$V(t) = Ri(t) + K\dot{\omega}(t) \tag{2.11}$$

$$J\ddot{\omega}(t) = Ki(t) - b\,\dot{\omega}(t) \tag{2.12}$$

We can write the equation for current from Equation 2.11,

$$i(t) = \frac{V(t)}{R} + \frac{K}{R}\dot{\omega}(t) \tag{2.13}$$

Substituting the above Equation in 2.11 we get our second Equation based on the angular acceleration of the rotor

$$\ddot{\omega}(t) = \frac{K}{JR}V(t) - \frac{1}{J}\left[\frac{K^2}{R} + b\right]\dot{\omega}(t) \tag{2.14}$$

Using the above equations, we can derive the transfer function from input voltage V to angular velocity $\dot{\omega}$ shown in Equation 2.15

$$\frac{\dot{\omega}(S)}{V(S)} = \frac{K/JR}{S + \left(\frac{1}{J}\left[\frac{K^2}{R} + b\right]\right)} \tag{2.15}$$

and Torque τ can be calculated as

$$\tau = J\dot{\omega}(t) \tag{2.16}$$

Therefore, the transfer function from input voltage V to torque τ is given in Equation 2.17

$$\frac{\tau(S)}{V(S)} = \frac{(K/R)S}{S + \left(\frac{1}{J}\left[\frac{K^2}{R} + b\right]\right)} \tag{2.17}$$

2.4 Modified Cart and Pendulum Equations

The linear acceleration of the cart and pendulum system can be given by

$$\ddot{x} = r\dot{\omega} \tag{2.18}$$

Where, r is the radius of the wheels, and ω is the angular velocity of the wheels. The states of the system should be transformed in terms of angular position ψ of the wheel. Thus, assuming new states $\dot{\omega} = \ddot{\psi}$ and $\omega = \dot{\psi}$ the Equation 2.18 can be rewritten as

$$\ddot{x} = r\ddot{\psi} \tag{2.19}$$

and therefore, the new states of the cart and pendulum system is represented by the state vector X,

$$X = \begin{bmatrix} \psi \\ \dot{\psi} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{2.20}$$

The input force F to the system can be replaced by the torque τ produced by the two DC motors as per Equation 2.21

$$F(t) = \frac{2\tau}{r} - \frac{2Iw}{r}\ddot{\psi} \tag{2.21}$$

where Iw is the moment of inertia of one wheel.

Based on Equation 2.5 the above equation can be written as,

$$F(t) = \frac{2Ki(t)}{r} - \frac{2Iw}{r}\ddot{\psi} \tag{2.22}$$

Later substituting Equations 2.13 and 2.19 in Equation 2.22 we can replace the force in Equation 2.3. The resulting equations for the cart and pendulum system combined with the DC motor dynamics are given as

$$(m+M)r\ddot{\psi} = \frac{2K}{Rr}V - \frac{2K^2}{Rr}\dot{\psi} - \frac{2Iw}{r}\ddot{\psi} - c_c r\dot{\psi} + ml\dot{\theta}^2 sin(\theta) - mlcos(\theta)\ddot{\theta} \qquad (2.23)$$

$$ml^2\ddot{\theta} = -c_p\dot{\theta} - mglsin(\theta) - mlr\ddot{\psi}cos(\theta) \qquad (2.24)$$

Equations 2.23 and 2.24 define the actual cart and pendulum system where the input to the system is Voltage V and the outputs are angular position of the wheel ψ and pendulum angle θ. The linear position of the cart y can be obtained by multiplying the radius of the wheel r with the angular position of the wheel ψ.

2.5 Linearization

The Inverted Pendulum on a Cart is a nonlinear system. This thesis deals with designing a feedback control system for a linear system. To get a linear approximation, the system is linearized about equilibrium points $\psi = 0$ and $\dot{\psi} = 0$ for the cart, and $\theta = \pi$ and $\dot{\theta} = 0$ for the pendulum since it is controlled in the inverted position.

To perform linearization, Equations 2.23 and 2.24 are transformed such that, they contain all the four states. This can be done by substituting Equations 2.23 in 2.24 and vice-versa. The state space for the system can be computed using Taylor's expansion theorem which yields a (1x2) system with one input and two outputs as shown in Equation 2.25.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -a_1 & a_2 & -a_3 \\ 0 & 0 & 0 & 1 \\ 0 & -a_4 & a_5 & -a_6 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ b_1 \\ 0 \\ b_2 \end{bmatrix} C = \begin{bmatrix} c_1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 \end{bmatrix} \tag{2.25}$$

where,

$$\begin{aligned}
a_1 &= \frac{\frac{2K^2+c_cRr^2}{rR}}{Mr+\frac{2I_w}{r}}, \quad a_2 = \frac{mg}{Mr+\frac{2I_w}{r}}, \quad a_3 = \frac{\frac{c_p}{l}}{Mr+\frac{2I_w}{r}}, \\
a_4 &= \frac{\frac{2K^2+c_cRr^2}{Rl}}{Mr+\frac{2I_w}{r}}, \quad a_5 = \frac{\frac{g}{l}((m+M)r+\frac{2I_w}{r})}{Mr+\frac{2I_w}{r}}, \quad a_6 = \frac{\frac{c_p}{ml^2}((m+M)r+\frac{2I_w}{r})}{Mr+\frac{2I_w}{r}}, \\
b_1 &= \frac{\frac{2K}{Rr}}{Mr+\frac{2I_w}{r}}, \quad b_2 = \frac{\frac{2K}{Rl}}{Mr+\frac{2I_w}{r}}, \quad c_1 = r
\end{aligned}$$

From the state space equations in Equation 2.25 we can compute two transfer functions for the system. The cart's transfer function from input voltage V to linear position y is given in Equation 2.26 and the pendulum transfer function from input voltage V to pendulum angle θ is given in Equation 2.27.

$$\frac{y(s)}{V(s)} = \frac{c_1[b_1s^2 - (a_3b_2 - a_6b_1)s + a_2b_2 - a_5b_1]}{s[s^3 + (a_1+a_6)s^2 + (a_1a_6 - a_3a_4 - a_5)s - a_1a_5 + a_2a_4]} \tag{2.26}$$

$$\frac{\theta(s)}{V(s)} = \frac{-b_2s + a_1b_2 - a_4b_1}{s^3 + (a_1+a_6)s^2 + (a_1a_6 - a_3a_4 - a_5)s - a_1a_5 + a_2a_4} \tag{2.27}$$

2.6 Conclusion

In this Chapter, the Inverted Pendulum on a Cart system was modeled using Newton-Euler Equations (Equation 2.3 and Equation 2.4), which resulted in a nonlinear system. The force input F was transformed to Voltage V after the DC motor dynamics were combined with the actual model in Equations 2.23 and 2.24.

To work with the nonlinear model we linearized the transformed system equations about the equilibrium point and derived the state space for the linear model in Equation 2.25. Linearization resulted in a 1x2 system with one input i.e Voltage V and two outputs i.e Cart's position y and Pendulum angle θ. Finally, we computed the transfer functions for the coupled system which are shown in Equations 2.26 and 2.27.

Chapter 3

HARDWARE AND SYSTEM INTEGRATION

3.1 Introduction

In this chapter, the hardware components, interconnections, and the system integration for the design and control of the Inverted Pendulum on a Cart system are described.

3.2 Hardware

This section describes the list of all the hardware components used for the design and construction of the inverted pendulum on a cart system.

Cart Body Frame

The base for the cart system was designed using Fusion360 CAD design software. The base is a 200mm diameter circular disk which is designed to mount the motors, battery, and a power distribution breakout board. The base supports the pendulum structure on top of it using four standoffs. The design of the Cart body frame is shown in Figure 3.1.

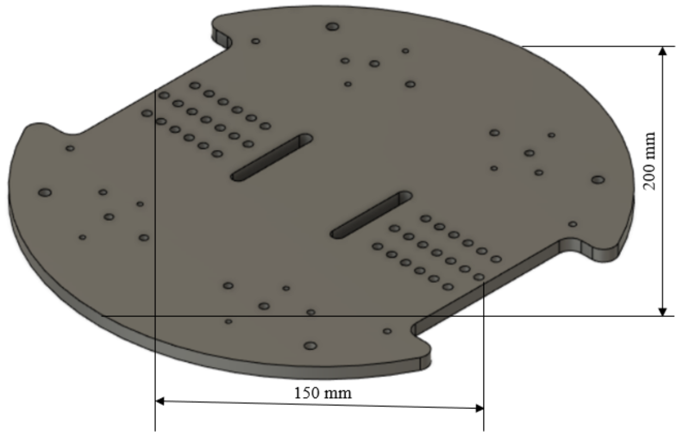

Figure 3.1: Cart Base

Pendulum Structure

The pendulum body frame is designed such that it holds the pendulum on top of the cart's base. The pendulum structure has a base which mounts the microcontroller and a sensor to measure heading. The pendulum is fixed to a T-shaped pendulum arm which is suspended between two poles with the help of a ball bearing on each side to enable free rotation of the pendulum. The pendulum structure is designed for a height of 200 mm and the pendulum suspended between the structure is of length 115 mm. The pendulum structure also holds a sensor bed which is attached to one end of the pendulum arm, this sensor bed is used to mount a sensor that measures the angle of the pendulum. The sensor is powered by using a slip ring commutator which is mounted on one of the poles which support the pendulum. Figure 3.2 shows the design of the pendulum structure.

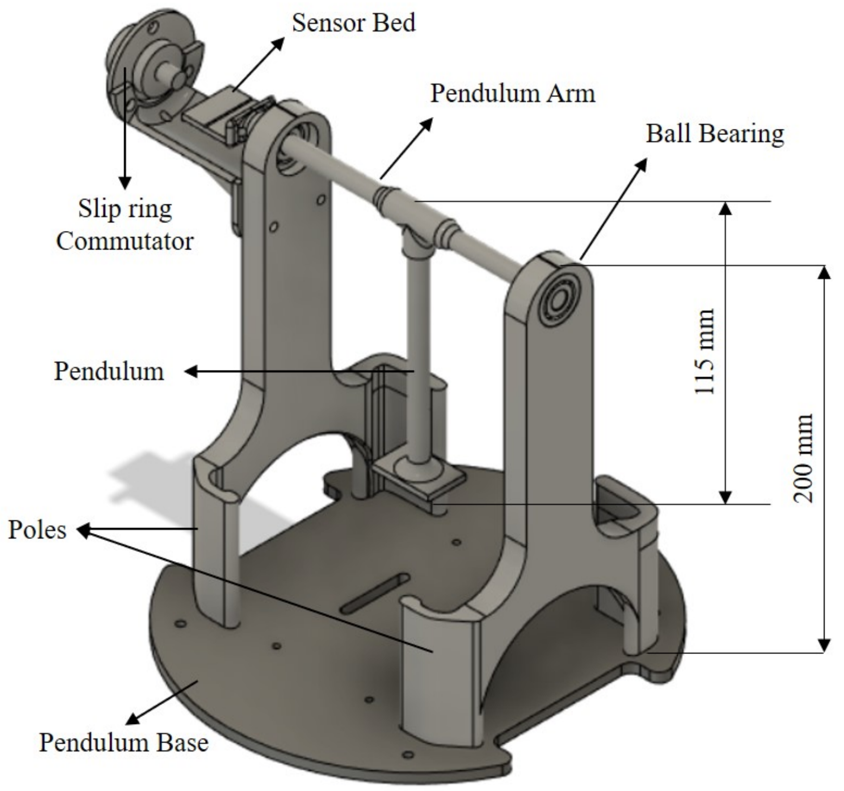

Figure 3.2: Pendulum Structure

DC Motors

As mentioned in Section 2.2, the cart and pendulum system is actuated by two DC motors. The DC motor generates torque τ for an input voltage V. For this particular robot, two 12 V high power brushed DC motor (25D x 63L mm) combined with a 9.7:1 metal spur gearbox from Pololu is being used. The motor has a no-load speed of 1030 RPM consuming 300 mA and can handle a stall torque of 3.16 kg-cm at 5600 mA. The DC motors are mounted on the cart's base using a 25D mm Pololu metal Gearmotor mount as shown in Figure 3.3

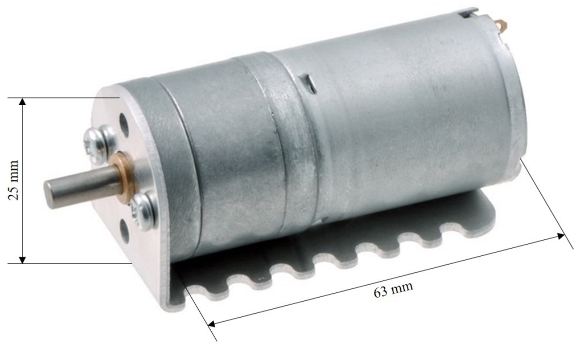

Figure 3.3: Pololu 12V High Power DC Motor Connected to 9.7:1 Metal Gearbox with Mounting Bracket

Motor Driver

The Motor Driver is used to drive the motors as per the given input signal. For the Motor Driver, a Pololu Dual VNH5019 Shield for Arduino was used. It is equipped with two VNH5019 motor drivers with H-bridge which operates between 5.5 V and 24 V while delivering up to 12 A continuous current. The shield is designed for Arduino boards and is compatible with the Arduino Due. The motor driver also can operate the PWM frequency at 20 Khz. This motor driver shield has an Arduino library that is used for programming.

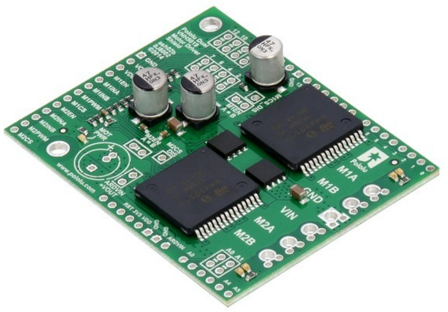

Figure 3.4: Pololu Dual VNH5019 Shield for Arduino

Encoders

An Encoder is an electro-mechanical device that can measure the angular position of a shaft. The encoder is equipped with two channel hall effect sensors which sense the rotation of a magnetic disk. The magnetic disk is attached to one end of the motor shaft. As the motor rotates, the magnetic disk rotates the hall effect sensor detects the change in magnetic poles and generates two pulse signals. For a quadrature encoder, the two pulse signal generated will be found to be 90° from each other which is used to determine the direction of rotation. For this thesis, a quadrature encoder from Pololu shown in Figure 3.5 was used which provided a resolution of 48 counts per revolution of the motor shaft and 464.64 counts per revolution with the gearbox. This sensor is mainly used to measure the position and angular velocity of the cart.

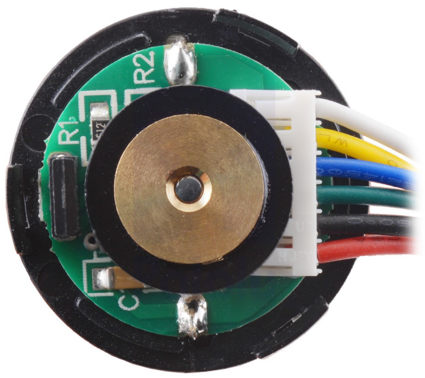

Figure 3.5: Pololu Motor Encoder

Wheels

For the wheels two INJORA 70x30 mm Plastic Wheel Rim and Rally tire were used as shown in Figure 3.6. The wheels are connected to the DC motor shafts using Pololu 12 mm Hex wheel adapters shown in Figure 3.7.

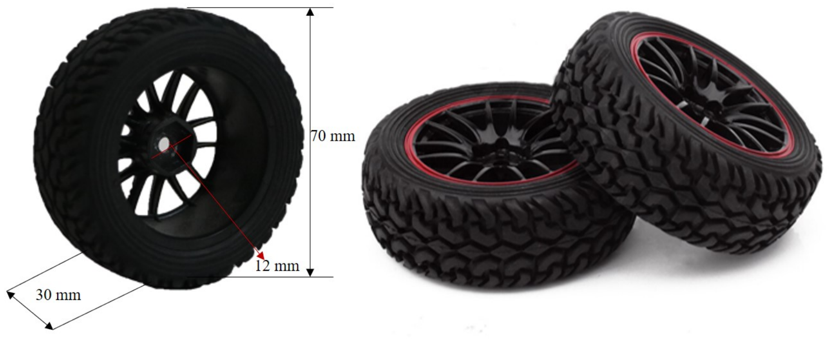

Figure 3.6: INJORE 70x30 mm Wheels

Figure 3.7: Pololu 12 mm Hex Wheel Adapters

Inertial Measurement Unit

To measure the angle of the pendulum an Inertial Measurement Unit (IMU) is used. The GY-86 is a 10-DOF breakout IMU which houses an MPU6050 sensor which has 3-Axis Accelerometer and 3-Axis Gyroscope, an MS5611 Barometric Pressure sensor, and an HMC5883L 3-Axis Magnetometer. There is another IMU sensor breakout board named GY-521 that is placed on the cart body which has an MPU6050 sensor with a 6-DOF which is used for measuring the yaw and yaw rate of the cart.

For the measurement of the pendulum angle, the data from the accelerometer, gyroscope and magnetometer of the GY-86 were fused to estimate Euler angles using a Sensor Fusion algorithm described in [17, 16]. The problem of using Euler angles is the effect of the Gimbal lock explained in [6]. Both the sensors communicate with the Arduino Due microcontroller by using the standard I2C communication protocol. The two breakout boards are shown in Figure 3.8..

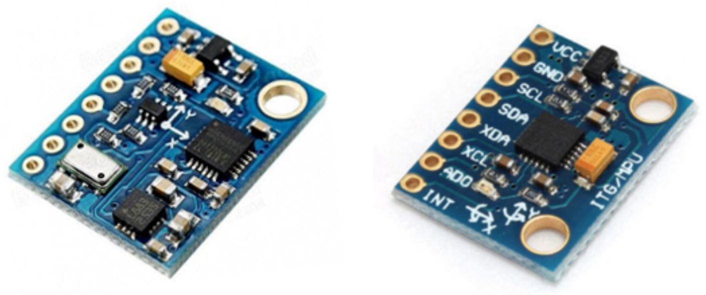

Figure 3.8: GY-86 IMU and GY-521 IMU Breakout Boards

Data Logger

A micro SD data logger is used to log the data from different sensors on the pendulum cart system. The data logger receives data from the Arduino Due microcontroller and creates a text file and stores it on a micro SD card. The data transfer is done using the UART Serial Communication protocol at 115200 bps. This module helps to store accurate data with very minimum data loss. The data logger module used in this thesis is shown in Figure 3.9

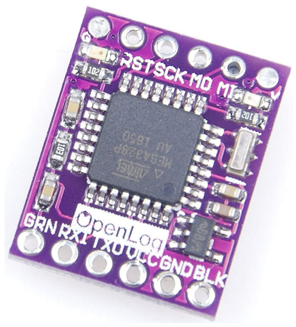

Figure 3.9: Micro SD Card Data Logger Module

Microcontroller

The mathematical computations and operation for the pendulum cart system is done using an Arduino Due microcontroller [10] shown in Figure 3.10. The Due board has Atmel SAM3X8E ARM Cortex -M3 32-bit core running an 84 Mhz clock. The Arduino Due is packed with 54 digital I/O pins, 12 analog input pins and 2 DAC ports. It operates at 3.3 V and has 12 pins that are capable of providing PWM outputs. The Serial communication for the Due is done using the 4-UART bus, 2-I2C bus and 1-SPI bus. The Due can be powered using a micro USB cable, which is also used for programming the microcontroller using Arduino IDE. The Arduino Due is sampled at a 100 Hz frequency to process all the required operations and to generate control inputs for the Motor driver to actuate the pendulum cart system.

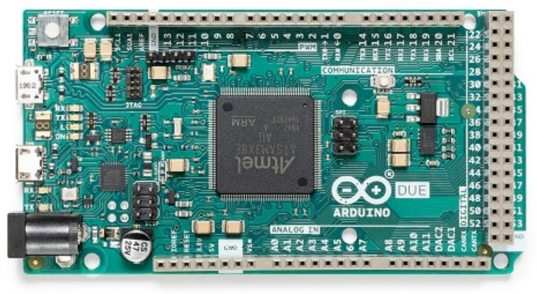

Figure 3.10: Arduino Due Microcontroller

Battery

A Floureon 3S 11.1V 3000mAh 30C LiPo battery pack shown in Figure 3.11 is used to power all the components of the pendulum cart system. It has three 3.7V LiPo cells connected in series which can provide a peak current of 90 A when operated at 11.1 V.

Figure 3.11: Floureon 3S 11.1V 3000mAh 30C LiPo Battery

3.3 System Integration

All of the hardware mentioned in 3.2 are integrated to build the final working model of the Inverted Pendulum on a Cart System. The Arduino Due microcontroller acts as the brain of the system where all the mathematical computations are done. The LiPo battery provides power to all the components of the system, the sensors send measured sensor data to the microcontroller. The IMU sensor uses Madgwick's algorithm [16, 17] to estimate Euler angles which determine the angle of the pendulum. The Encoder sensors provide the position estimate of the cart. Based on these two sensor data the microcontroller sends in desired PWM control signals to the motor driver which controls the motors and actuates the system.

The System Integration and signal flow for the system is represented as a block diagram shown in Figure 3.12. Finally, the Arduino Due microcontroller is programmed using the Arduino IDE software which supports the use of C and C++ languages with some special set of rules for running the code instructions.

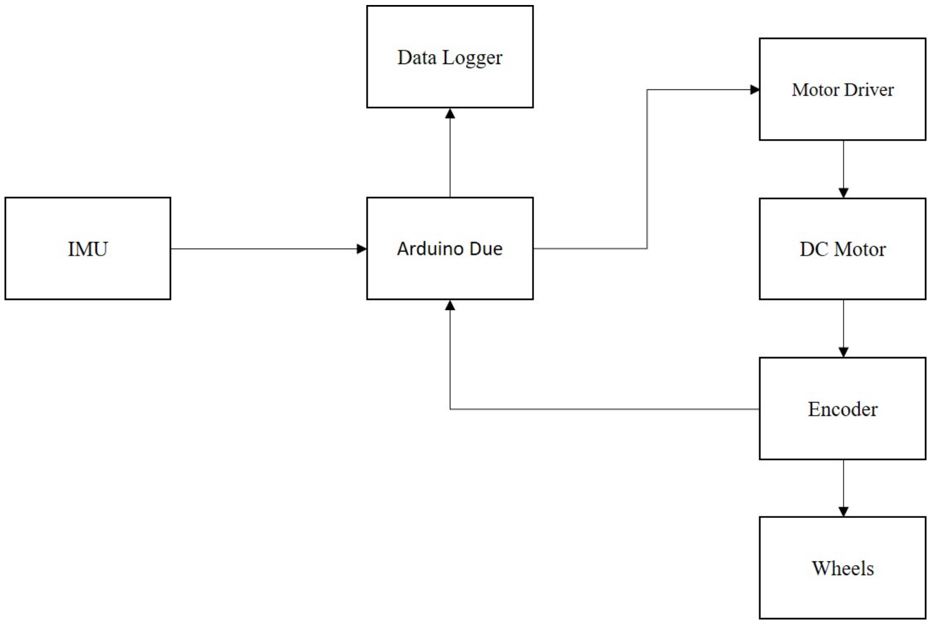

Figure 3.12: System Integration and Signal Flow for Inverted Pendulum on a Cart

3.4 Conclusion

In this Chapter, the hardware components used to design and build the pendulum cart system were briefly described and the basic operations were discussed. All of these components put together forms the complete Inverted Pendulum on a cart model as shown in Figure 4.1. The Microcontroller acts as the brain of this system and commands the DC motors to perform operations to balance a pendulum using control laws and techniques based on the data received from the IMU and encoder sensors.

Chapter 4

PARAMETRIZATION OF FIRST PRINCIPLES MODEL

4.1 Introduction

In this Chapter, we will parameterize the model such that we get the complete parameterized Linear model for designing controllers. Some of the parameters are based of the construction of the robot and some of the parameters are calculated and estimated for the design of the system. For example, the mass of the cart M, the mass of the pendulum m are based on the construction of 3D print parts, the moment of inertia for the pendulum I_p, the moment of inertia of the wheels I_w are calculated and the coefficient of friction of cart cc, coefficient of friction of pendulum cp are estimated by a few experiments. All of these parameters were calculated or estimated for the Inverted Pendulum on a Cart robot as shown below in Figure 4.1.

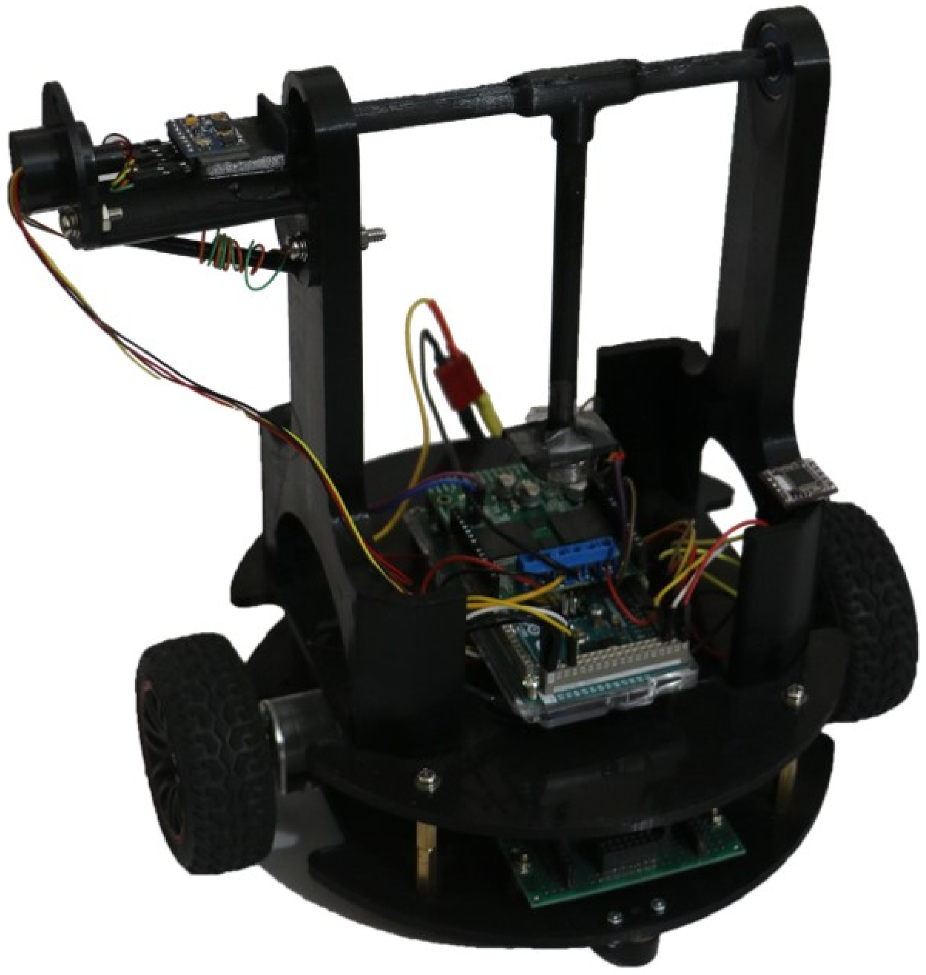

Figure 4.1: The Complete Inverted Pendulum on a Cart System

4.2 Parameter Estimation

This subsection will describe the details of the parameters derived for the Inverted Pendulum on a Cart robot system. Firstly, the parameters based on construction will be discussed, followed by the parameters that were calculated and finally the parameters that were estimated using experimentation. The parameters for the Inverted pendulum on a cart system shown in Figure 4.1 are listed as -

Mass of the Cart, M

The Mass of the Cart M is the total mass of the cart system which is the combined mass of all the components of the robot. The Mass of the cart is measured by using a weight scale with units set to the standard SI units kg. Hence,

$$M = 1.21\ kg \tag{4.1}$$

Mass of the Pendulum, m

The Mass of the Pendulum is the combined mass of the pendulum rod and the bob attached at the end of the pendulum rod. In this project, we assume a weightless pendulum rod hence the mass of the bob dominates the total weight of the pendulum.

$$m = 0.020\ kg \tag{4.2}$$

Acceleration Due to Gravity, g

Acceleration due to gravity is a constant calculated based on Newton's Second Law of Motion and Newton's Law of Universal Gravitation.

$$g = 9.81\ m/s^2 \tag{4.3}$$

Radius of the Wheel, r

The radius of the wheel is given by the wheel manufacturer as shown in Subsection 3.2 and the mass of one wheel is 0.04 kg

$$r = 0.0375\ m \tag{4.4}$$

Length of the Pendulum, l

As stated in Subsection 3.2, the length of the pendulum is 0.115 m. The length of the pendulum is calculated as the end to end measurement of the pendulum from the pivot to the bob.

$$l = 0.115\ m \tag{4.5}$$

Moment of Inertia of the Wheel, I_w

The Moment of Inertia of the Wheel I_w is calculated as the sum of the individual moments of inertia of wheels and the connector shafts using the Parallel Axis Theorem. Both the wheel and the connector shaft is assumed to be a solid cylinder rotating in the y-axis about its center as shown in Figure 4.2. The formula to calculate I_w is given below,

$$\begin{aligned} I_w &= \frac{1}{2} M_{wheel}\, r_{wheel}^2 + \frac{1}{2} M_{shaft}\, r_{shaft}^2 \ \ kg\, m^2 \\ I_w &= 2.8395 \times 10^{-5} \ \ kg\, m^2 \end{aligned} \tag{4.6}$$

where, M_{wheel} is the mass of the wheel, r_{wheel} is the radius of the wheel, M_{shaft} is the mass of the connector shaft and the r_{shaft} is the radius of the connector shaft.

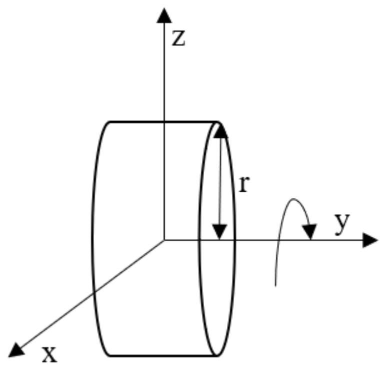

Figure 4.2: Moment of Inertia for a Solid Cylinder Rotating about y Axis

Moment of Inertia of the Pendulum, I_p

The pendulum is considered as a solid rod with Length l described in Subsection 4.2 with a Pendulum bob of Mass m described in Subsection 4.2. The Solid rod is rotating about one end along the y-axis as shown in Figure 4.3. The Pendulum bob is considered as a point mass rotating along the y-axis. Hence the total Moment of Inertia is the sum of the individual moment of inertia of the solid rod and the point mass. The formula to calculate the Moment of Inertia for the Pendulum I_p is given below using the Parallel Axis Theorem,

$$\begin{aligned} I_p &= \frac{1}{3} m_{rod}\, l^2 + m_{bob}\, l^2 \;\; kg\, m^2 \\ I_p &= 26.45 \times 10^{-5} \;\; kg\, m^2 \end{aligned} \tag{4.7}$$

where, m_{rod} is the mass of the rod, m_{bob} is the mass of the bob and l is the length of the pendulum shown in Equation 4.5.

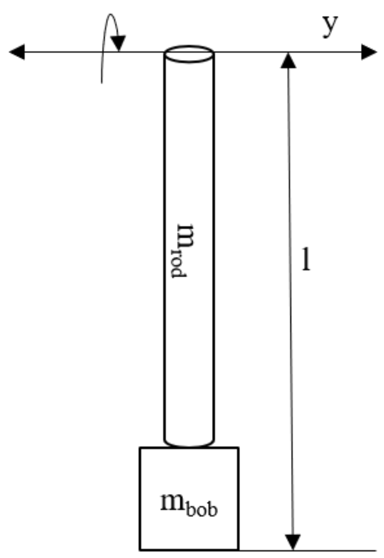

Figure 4.3: Moment of Inertia for a Solid Rod Rotating about Pivot Axis

Motor Constant, K

From Section 2.3 we know that the Motor Constant K is provided by the motor manufacturer. However, the motor manufacturer will not give the motor constant directly. Hence we derive a formula to calculate the Motor Torque Constant K_τ

$$K_\tau = \frac{Stall\,Torque}{Stall\,Current - No\,Load\,Current} \;\; NmA^{-1}$$

$$K_\tau = \frac{0.3107}{5.6 - 0.3} \;\; NmA^{-1}$$

$$K_\tau = 0.0586 \;\; NmA^{-1}$$

From Equation 2.7 in Section 2.3 we assume the Motor Constant,

$$K = 0.0586 \tag{4.8}$$

Armature Resistance, R

The Armature Resistance of DC Motor can be calculated using Ohm's Law. Hence,

$$\begin{aligned} R &= \frac{Rated\,Voltage}{Stall\,Current}\ \ \Omega \\ R &= 2.1429\ \ \Omega \end{aligned} \tag{4.9}$$

Friction Coefficient of Cart, c_c

The DC Motor model described in Section 2.3 Equation 2.17 is a First Order System, with the parameters Armature Resistance R, Motor Constant K, Moment of Inertia of the rotor J and the coefficient of friction of rotor b. While the parameters R and K are calculated in Section 4.2, the parameters b and J cannot be calculated as it depends on various factors of the DC Motor and its internal components. Instead, we can estimate these to parameters with the help of the First-order Transfer function model. The Time Constant τ_m in the transfer function contains all the dynamics of the DC Motor model. Therefore, we can find the Time Constant for the DC Motor and then estimate the unknown parameters using the actual model and the mathematical model. τ_m can be calculated from the settling time of the angular velocity of the motor using the "5 time constant settling time convention" [25], given by

$$\tau_m = 5 * t_s \tag{4.10}$$

where t_s is the settling time of the motor for a given input step signal. This can be estimated by giving a step input to the DC Motor and by measuring the angular velocity of the DC Motor.

In the linearized State Space model of the Inverted Pendulum on a Cart Equation 2.25 we only need the Friction Coefficient of the Cart c_c. When we perform the above-mentioned step test without any wheels connected we can estimate both b and J for the DC motor but we do not require those parameters in the final state space equations.

Hence to estimate the Friction Coefficient of the Cart c_c, we can repeat the step test experiment but now with all the components of the system integrated. Performing the step test on the robot placed on the selected surface will now result in the angular velocity to settle longer than the other and we will get a new Time Constant τ_{m_new} that has both the dynamics of the DC Motor rotor and the Friction Coefficient of the Cart system. In other words, τ_{m_new} should be greater than τ_m as it now includes the friction factor included in the First Order System as shown in Figure 4.4 and Figure 4.5.

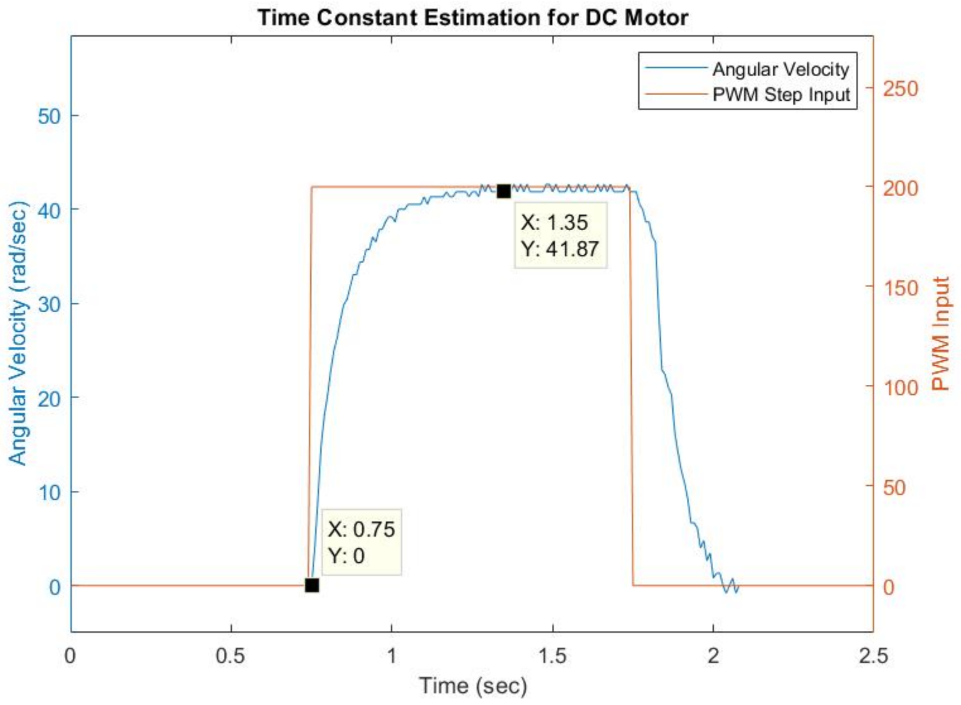

Figure 4.4: Settling Time and Time Constant for Free Running DC Motor

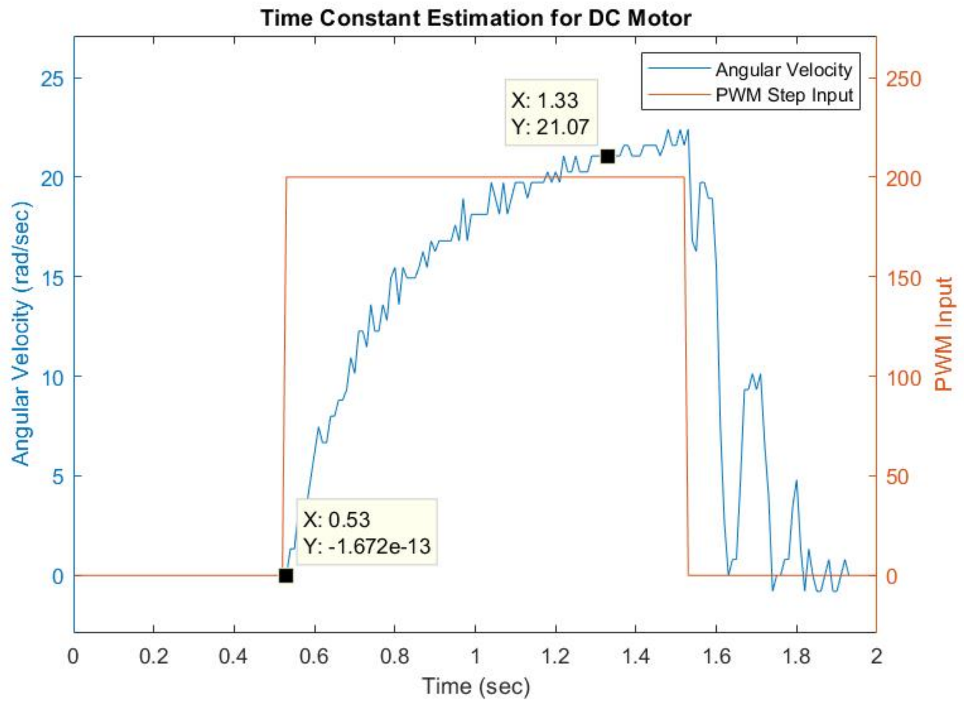

Figure 4.5: Settling Time and Time Constant for DC Motor on Surface

Once we have data collected from the actual system, we can now iterate the Friction Coefficient c_c parameter to get a close estimate based on the Time Domain data between the step input and the angular velocity of the cart system as shown in Figure 4.6. Choosing c_c to be 10 gave the closest match between the mathematical model and the actual model. Therefore, the Friction Coefficient of the Cart was estimated to be

$$c_c \approx 10 \tag{4.11}$$

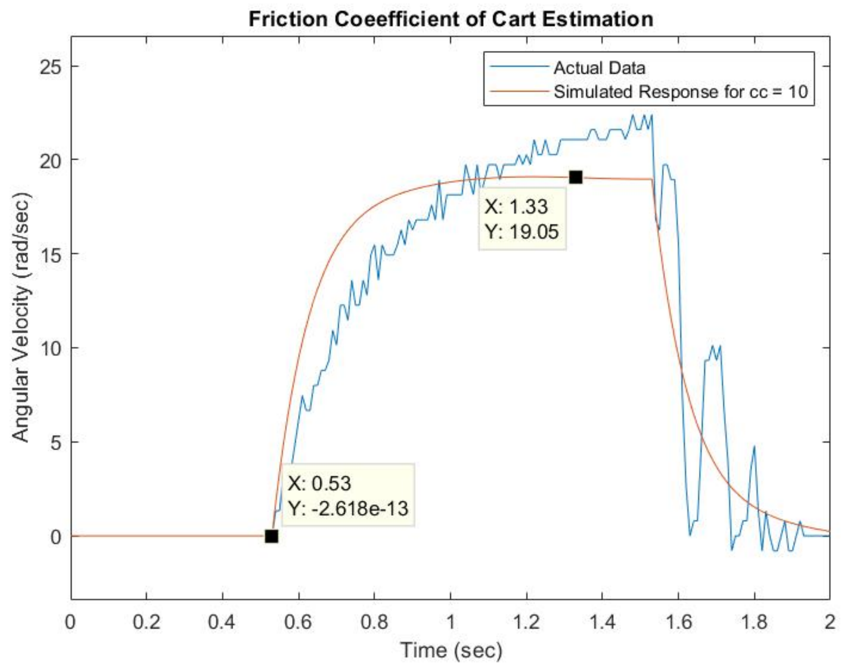

Figure 4.6: Friction Coefficient of Cart Estimation

Friction Coefficient of the Pendulum, c_p

The Pendulum is mounted on to the pendulum structure as mentioned in Subsection 3.2. The T-Shaped pendulum arm is connected to ball bearings at both ends. There are no sensors to measure the friction of the ball bearings. Hence, we can try to estimate the Friction Coefficient of the pendulum c_p by measuring the exponential decay of the pendulum's oscillations. Another way to estimate c_p is that we know the pendulum system is a Second Order system with a Damping constant. So now we can measure the pendulum's oscillation in Time Domain and iterate the Friction Coefficient c_p parameter to get a close estimate. The Pendulum can be actuated effectively by sending a Pseudo-Random Binary Sequence (PRBS) [4] input to the motors. The PRBS signal has fast switching of polarities for a given PWM amplitude which makes the cart switch directions actuating the pendulum significantly.

Choosing c_p to be 0.0005 gave the closest match between the mathematical model and the actual model. Therefore, the Friction Coefficient of the Pendulum was estimated to be

$$c_p \approx 0.0005 \tag{4.12}$$

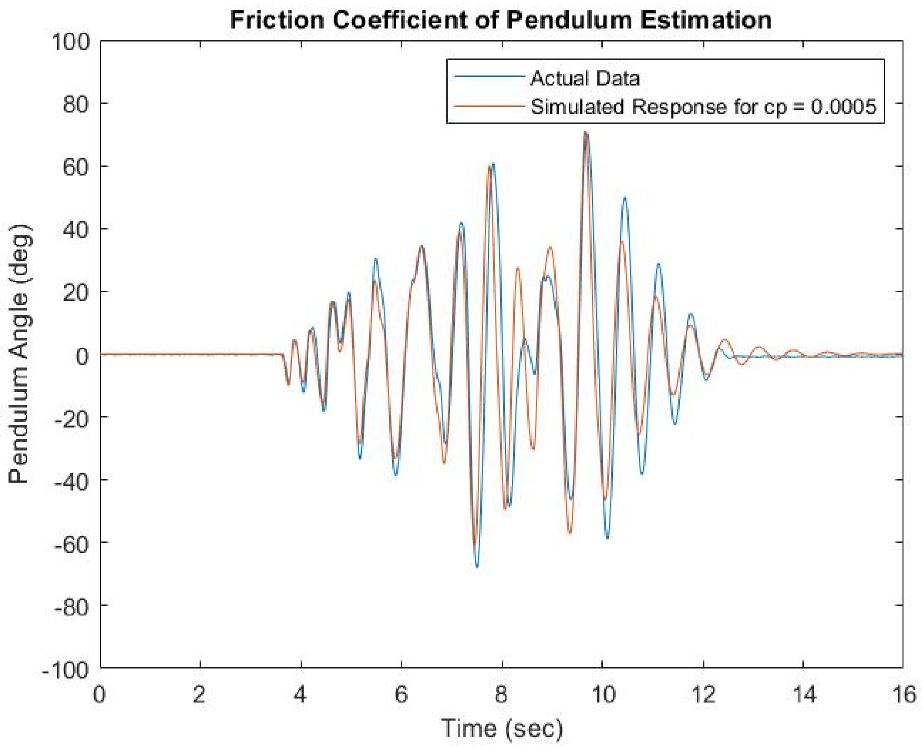

Figure 4.7: Friction Coefficient of Pendulum Estimation

4.3 Results

The parameters described in this chapter are listed in Table 4.1.

Parameter	Value
M	1.21 kg
m	0.020 kg
g	9.81 $\frac{m}{s^2}$
r	0.0375 m
l	0.115 m
I_w	2.8395 $\times 10^{-5}$ $kg\,m^2$
I_p	26.45 $\times 10^{-5}$ $kg\,m^2$
K	0.0586
R	2.1429 Ω
c_c	10
c_p	0.0005

Table 4.1: Parameters of the Inverted Pendulum on a Cart System

Using the values in Table 4.1, we can rewrite Equation 2.26 and Equation 2.27 as

$$\frac{y(s)}{V(s)} = \frac{1.168s^2 + 2.209s - 99.67}{s^4 + 11.74s^3 - 68.1s^2 - 838s}$$

$$\frac{\theta(s)}{V(s)} = \frac{582.1s}{s^3 + 11.74s^2 - 68.1s - 838} \tag{4.13}$$

The above transfer functions, shown in Equation 4.13 are derived with Voltage as input to the system. The input DC Voltage is given as a Pulse Width Modulated (PWM) signal to the motor drivers with a minimum value of 0 which corresponds to 0% duty cycle and a maximum value as 400 which corresponds to 100% duty cycle as per the Motor Driver limitation. (i.e PWM is usually ranged between 0 to 255 which is mapped as 0 to 400 in this case). Assuming that the input DC Voltage and PWM input have a linear relationship, and the fact that control is over the PWM duty cycle and not the voltage itself this factor needs to be included in our model so that we have the actual system with an effective voltage level. The transfer function shown in Equation 4.13 is multiplied with a factor 0.03 derived from the ratio of $\frac{PWM_{max}}{V_{max}}$ to get our final transfer function as shown in Equation 4.14. For Simplicity, the angle of the pendulum system is changed from *rad* to *deg* units scale.

$$\frac{y(s)}{V_{pwm}(s)} = \frac{0.03505s^2 + 0.06626s - 2.99}{s^4 + 11.74s^3 - 68.1s^2 - 838s}$$

$$\frac{\theta(s)}{V_{pwm}(s)} = \frac{17.46s}{s^3 + 11.74s^2 - 68.1s - 838} \tag{4.14}$$

4.4 Conclusion

The parameters of the Inverted Pendulum on a Cart System calculated were discussed based on the construction of the robot system. The moment of inertia for the wheels I_w and the pendulum I_p were calculated. Some experiments were done to estimate the Time Constant τ_m using the "5 time constant settling time convention" [25] as shown in Equation 4.10. Based on the Time Constant, the Friction Coefficient of the Cart c_c was estimated and an experiment with PRBS input signal to the robot was done to estimate the Friction Coefficient of the Pendulum c_c.

Using these parameters, we get the parameterized First Principles Model for the Inverted Pendulum on Cart system shown in Figure 4.1. The Transfer Functions in Equation 4.14, are used to design controllers to Stabilize the pendulum in the Inverted position while maintaining the position of the cart system.

Chapter 5

CONTROLLER DESIGN AND RESULTS

5.1 Introduction

In this Chapter, the design, implementation, structure, and tuning of the different controllers used for the Inverted Pendulum on a Cart system are discussed. There are a total of four controllers implemented on the robot which helps in the control of the system (Yaw rate, Energy, Stabilizing and Position). Each controller is responsible for their control actions which when combined result in the balancing of the pendulum for the system shown in Figure 4.1. This chapter will also discuss the effect of filters and the implementation of controllers on the Arduino Due microcontroller.

5.2 Yaw Rate Control using Ziegler-Nichols Method

The robot is equipped with two DC motors and two caster wheels and the first assumption in this thesis is that the robot must drive straight at any given point of time. When both the DC motor are given the same PWM input, the robot tends to drift apart while in motion rather than maintaining a straight-line path. To make the robot drive straight both the wheels must have the same angular velocity. To ensure that both the motors run with the same angular velocity the Yaw angle θ_{yaw} of the robot is measured and based on the robot Yaw angle the angular velocity of the robot can be controlled. To measure this angle the GY-521 IMU sensor mentioned in Subsection 3.2 was used. The GY-521 IMU sensor gives information of the robot Yaw rate $\dot{\theta}_{yaw}$ using the gyroscope value and uses the Madgwick filter [16, 17] to estimate the Yaw angle θ_{yaw}.

Using the information of the Yaw rate and Yaw angle the robot can be modeled to perform like a two-wheeled differential drive robot which is always intended to drive in a straight line. Since the main objective of this thesis is not the modeling of the differential drive system, the yaw rate control of the system is done using the Ziegler-Nichols method of tuning a PID controller [35].

The Ziegler-Nichols second method of tuning mentioned in [9] helps to design a controller by measuring two main feedback loop parameters Ultimate gain K_u and Ultimate period P_u for a system which is not modeled. K_u is measured when there is a Proportional controller for the system and the gain of the controller is increased from zero to a point where the output of the closed loop system oscillates. Now, when the closed loop system is oscillatory P_u measured from the period of the oscillation. Having, the parameters K_u and P_u, we can design closed loop controllers as shown in Table 5.1 [9, 32].

	P	PI	PD	PID
K_p	$0.5K_u$	$0.45K_u$	$0.8K_u$	$0.6K_u$
K_i	0	$0.54K_u/P_u$	0	$1.2K_u/P_u$
K_d	0	0	$0.1K_uP_u$	$0.075K_uP_u$

Table 5.1: Controller Design using Ziegler-Nichols Method

5.2.1 Yaw Rate Controller Tuning

To estimate the parameters K_u, a proportional controller is implemented in a closed loop and the output of the system $\dot{\theta}_{yaw}$ is evaluated for a step input signal. The gain of the proportional controller is increased from zero to a value where the robot begins to oscillate for a given Yaw rate reference as shown in Figure 5.1.

This value of gain is the Ultimate gain of the system K_u.

$$K_u = 0.4 \tag{5.1}$$

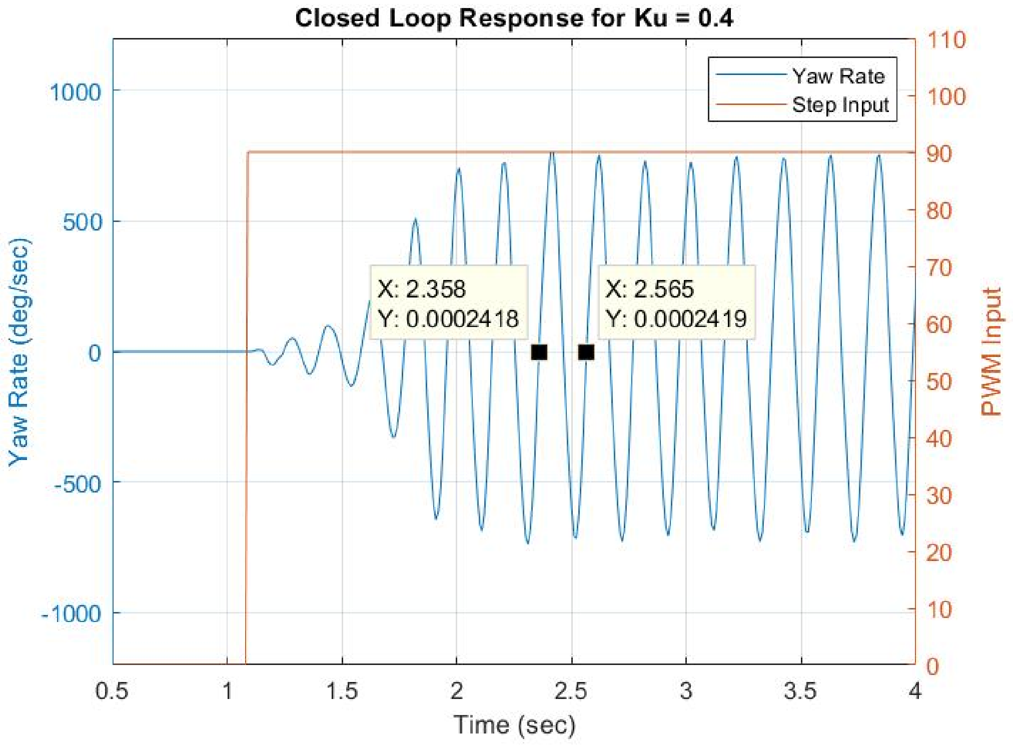

Figure 5.1: Closed Loop Response for a Proportional Control

From the above Figure 5.1 the period of oscillation can be estimated which is the Ultimate period P_u of the system,

$$P_u = 0.2070\,sec \tag{5.2}$$

The control system is designed to maintain a zero Yaw rate $\dot{\theta}_{yaw}$ so that the robot always can drive in a straight line for any Yaw angle θ_{yaw} as shown in Figure 5.2. A PID controller is designed for this system to control the Yaw rate of the robot with a reference of 0 deg/sec. Using Table 5.1 and Equations 5.1 and 5.2 we derive the gains for the PID controller shown in Table 5.2

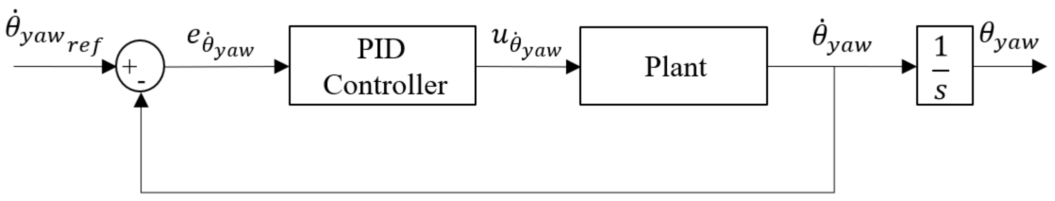

Figure 5.2: Yaw Rate Feedback Control Structure

Gain	Value
K_p	0.2400
K_i	2.3188
K_d	0.0062
τ	0.0050

Table 5.2: PID Gains for Yaw Rate Controller using Ziegler-Nichols Method

The PID controller is implemented as a bi-proper transfer function as shown in Equation 5.3 [1]. The pseudo-pole is chosen to be half the sample time (i.e. $T_s/2$). The controller is sampled at 100 Hz frequency and implemented in discrete time [2] using Tustin or Bi-Linear transforms [31].

$$C(s) = \frac{U(s)}{E(s)} = \frac{s^2K_d + sK_p + K_i}{s(\tau s + 1)} \tag{5.3}$$

This PID controller ensures that the closed loop system always drives the robot in the straight line replicating the two-wheel differential drive robot. This helps in reducing the loss of energy when the pendulum is swinging up or while the pendulum is balanced. The performance of the robot with and without a PID controller for a step input is shown below.

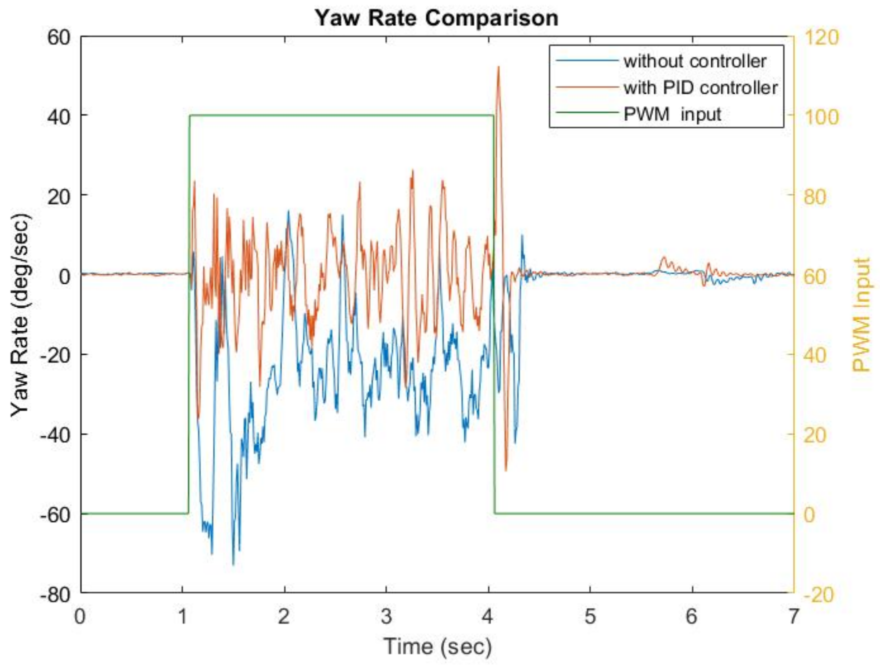

Figure 5.3: Yaw Rate of Robot without Control vs with PID Control

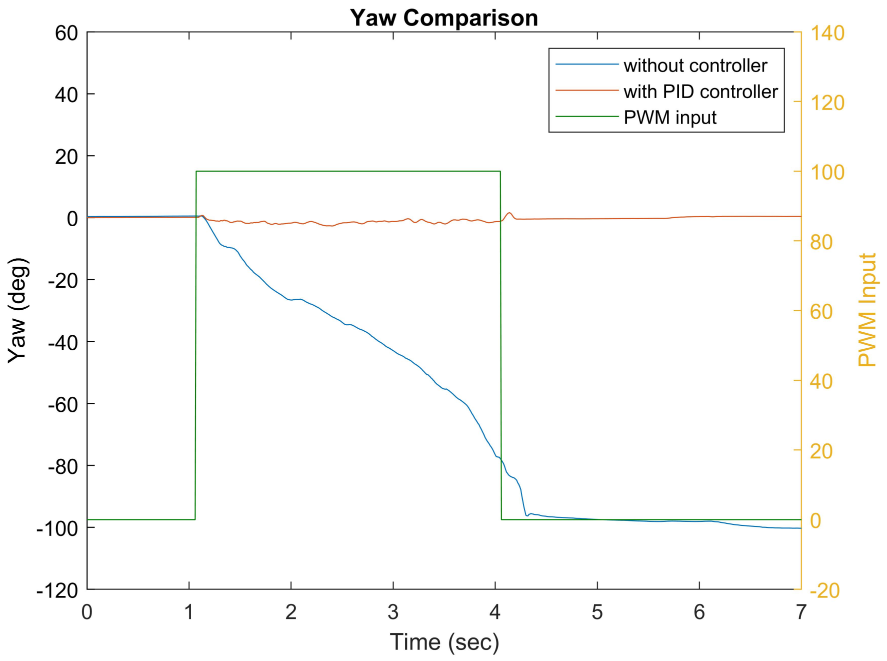

Figure 5.4: Yaw of Robot without Control vs with PID Control

5.3 Energy Controller

An Energy controller is designed to swing the pendulum from its stable equilibrium point to the unstable equilibrium point around which the pendulum is controlled. The swing-up strategy for this system is based on the total energy of the pendulum. Different methods for swinging up a pendulum using energy control are discussed in [3, 22, 7, 20, 33]. The pendulum's total energy is the sum of potential energy and kinetic energy.

$$E = \frac{1}{2} I_p \dot{\theta} + mgl(1 - cos(\theta)) \tag{5.4}$$

The Energy controller should provide a control signal such that the pendulum reaches the energy level in the inverted position where the stabilizing controller takes over to balance the pendulum. The energy of the system in the inverted position (i.e 180° or $\pi\, rad$) is just the potential energy of the system. Assuming that $\dot{\theta} = 0$, the energy at the inverted position is given as,

$$E_\pi = 2mgl \tag{5.5}$$

Examining the pendulum equation from Section 2.4 Equation 2.24, the behavior of the energy E along the solutions of the pendulum with respect to the acceleration of the pivot point should approach to zero if $\dot{\theta} \neq 0$. In general, the Energy error $(E - E_\pi)$ should decrease to zero. Calculating the time derivative of E we get,

$$\dot{E} = \dot{\theta}[ml^2\ddot{\theta} + mglsin(\theta)] \tag{5.6}$$

Substituting Equation 2.24 in above Equation 5.6 we get,

$$\dot{E} = \dot{\theta}[-c_p\dot{\theta} - mlr\ddot{\psi}cos(\theta)] \tag{5.7}$$

The input to the above equation is linear acceleration of the cart $(r\ddot{\psi})$, which also provides the torque to rotate the pendulum about its pivot point. The method of controlling the cart's velocity instead of the acceleration is discussed in [7]. The control input should be chosen such that it drives the pendulum towards the point E_π. As the control input is directly related to the linear acceleration $r\ddot{\psi}$ of the wheels, it can be replaced with the control law shown in Equation 5.8 obtained by choosing a Lyapunov function [3] such that the solutions approach zero.

$$u_E = r\ddot{\psi} \quad = k_e(E - E_\pi)sgn(\dot{\theta}cos(\theta)) \tag{5.8}$$

where, k_e is a design parameter, which determines the behavior of the energy in different regions. The signum (sgn) function is used to determine the polarity of the control signal and sets the control output to be zero when the pendulum is at rest or it is at 90°.

The main motivation for using the energy controller is to swing the pendulum by building energy so that the stabilizing linear controller can balance the pendulum when it is close to the unstable equilibrium point. [3] and [20] discusses a strategy where the energy and linear controller are switched on and off based on the specified region of interest. [7] describes the idea of building energy wells for swing up, velocity and position. However, with all these methods of energy control, the pendulum has zero energy when at rest. Therefore, the pendulum needs an external force to initiate energy control. In this thesis, a new control strategy is proposed to avoid the necessity of external input to initiate the swing up.

5.3.1 Energy Control Algorithm

The Energy control algorithm proposed in this thesis eliminates the condition of external force to initiate the swing up strategy. In this method of Energy control, we provide the limits for the inner loop controller while maintaining the overall nature of the system. This means that both the energy controller and the inner loop controller is always online. When the pendulum is at rest with zero energy, the inner loop controller initiates the actuation of the robot which swings the pendulum. Now, the energy controller dominates the control input as the inner loop controller depends on the control limits provided by the energy controller. When the pendulum is in the inverted position the energy controller becomes negligible and the inner loop controller dominates the control loop based on an allowable linear range given by the energy controller. This method of control avoids the complex switching between controllers which often leads to a bumpy transfer in the control signal. A similar approach is discussed for a torque pendulum in [23]. The control loop of this Energy controller is shown in Figure 5.5 and the integration of the energy controller and the inner loop controller is discussed in Subsection 5.4.2.

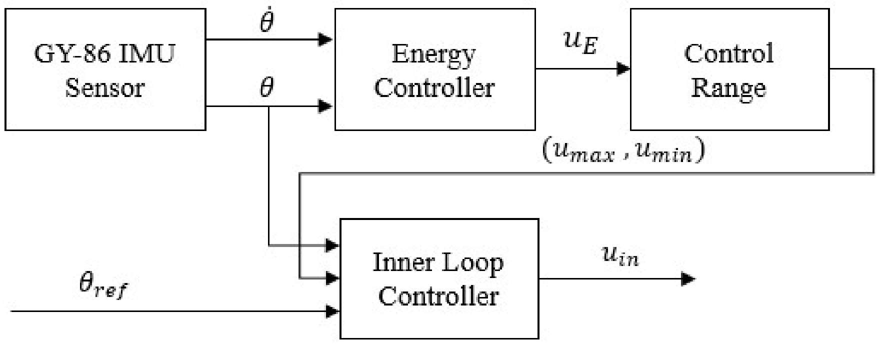

Figure 5.5: Energy Control Loop Structure

5.3.2 Energy Controller Tuning

The Energy controller is tuned based on the parameter k_e which is chosen from manual tuning so that the Energy error converges to zero. The energy control output is saturated based on the maximum PWM range. Hence, the Equation 5.8 becomes,

$$u_E = sat[k_e(E - E_\pi)sgn(\dot{\theta}cos(\theta))] \tag{5.9}$$

The energy control also defines constraints for the inner loop integrator so that windup of the integrator is avoided due to large errors when implemented on hardware. The Integrator winds up when the control input is clipped off due to actuator limitations (i.e the actuators do not respond to signals beyond a certain operating range). In this thesis, actuators are the two DC motor that takes PWM input signals. Hence, the range of PWM signal is the limitation for the actuators. These control limits for the inner loop controller are defined as

$$\begin{aligned} u_{min} &= u_E - \Delta u \\ u_{max} &= u_E + \Delta u \end{aligned} \tag{5.10}$$

Here the parameter Δu is an allowable range for the stabilizing controller which is chosen a priori. Using these as the limits on the inner loop controller, we can have a bumpless transfer between Energy and the Stabilizing controller.

5.4 Stabilizing Linear Controller (Inner Loop)

The stabilizing controller is designed for the plant transfer function shown in Equation 4.14 to stabilize the angle of the pendulum with an inner loop control design. Normally, for a system that needs a good command following capability a Proportional Integral Derivative (PID) type of control is used as discussed in [1]. But, for this particular problem, we aim to stabilize the pendulum rather than to perform command following due to the nature of the system. The pendulum system by itself is an unstable system with a right half plane pole and a zero at the origin. Hence for this type of plant designing a simple PID controller will not suffice. Based on the Root Locus insight shown in Figure 5.6, we need a compensator with a right half plane pole and two lead zeros to make the system stable.

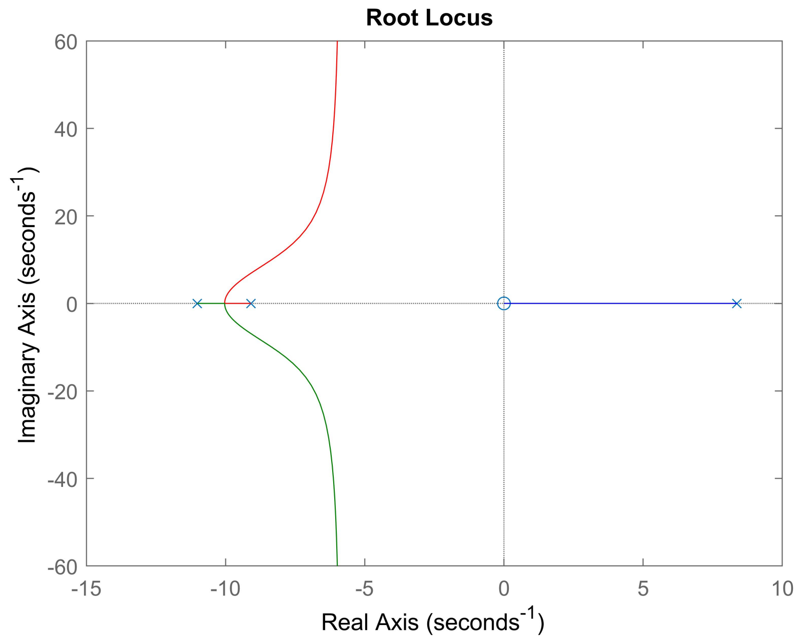

Figure 5.6: Root Locus of Pendulum Transfer function

5.4.1 Modified PID Control

The requirement of an RHP pole in the controller makes this a classical hard control design problem, a normal PID like controller cannot be used for this purpose of balancing the pendulum. Hence, we opted for a modified design of a PID controller to make the closed loop system stable. The modified PID controller has the general form as shown below

$$C(s) = \frac{K(s+z1)(s+z2)}{(s-\epsilon)(\tau s+1)} \tag{5.11}$$

where, $z1, z2$ are the PID zeros, τ is the pseudo pole, and ϵ is the location of the right half plane pole which is added to stabilize the inner loop.

The PID zero's for the controller are derived using a Frequency Loop Shape approach which is explained with details in [12]. Using this approach the controller is tuned to minimize the difference between actual and target loop transfer function using frequency domain convex optimization. The target loop chosen for this design is based on an LQR which is iterated for the given sensitivity bandwidth. The pole location ϵ and bandwidth of the closed loop are the design parameters used for tuning.

The controller design in Equation 5.11, can also be expanded in terms of the three gains used in a general PID controller as,

$$C(s) = K_p + \frac{K_i}{s-\epsilon} + \frac{K_d\, s}{\tau\, s+1} \tag{5.12}$$

The control structure used for the stabilizing controller is similar to a classic feedback controller, with an added constrain limit to the integrator. The inner loop control structure for this design is shown in Figure 5.7

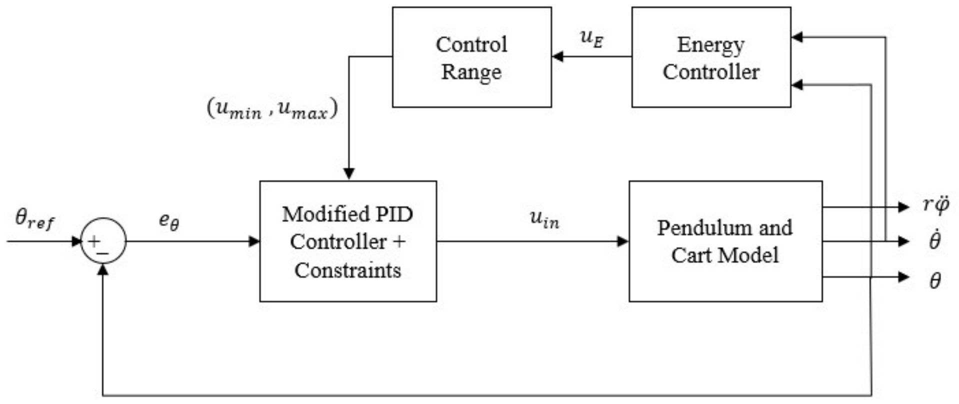

Figure 5.7: Inner Loop Control Structure

5.4.2 Inner Loop Tuning

The Inner loop system is treated as a Single Input Single Output (SISO) system as the position if the cart is not controlled in this feedback loop. The Inner Loop controller is designed for the pendulum transfer function,

$$\frac{\theta(s)}{V_{pwm}(s)} = \frac{17.46s}{s^3 + 11.74s^2 - 68.1s - 838} \tag{5.13}$$

The Transfer function has a zero at the origin (0 *rad/sec*) which increases the magnitude and phase response of the system, and a pole in the right half plane (8.3675 *rad/sec*). This makes the inner loop plant unstable with an infinite gain margin. The bode response of the transfer function is shown in Figure 5.8.

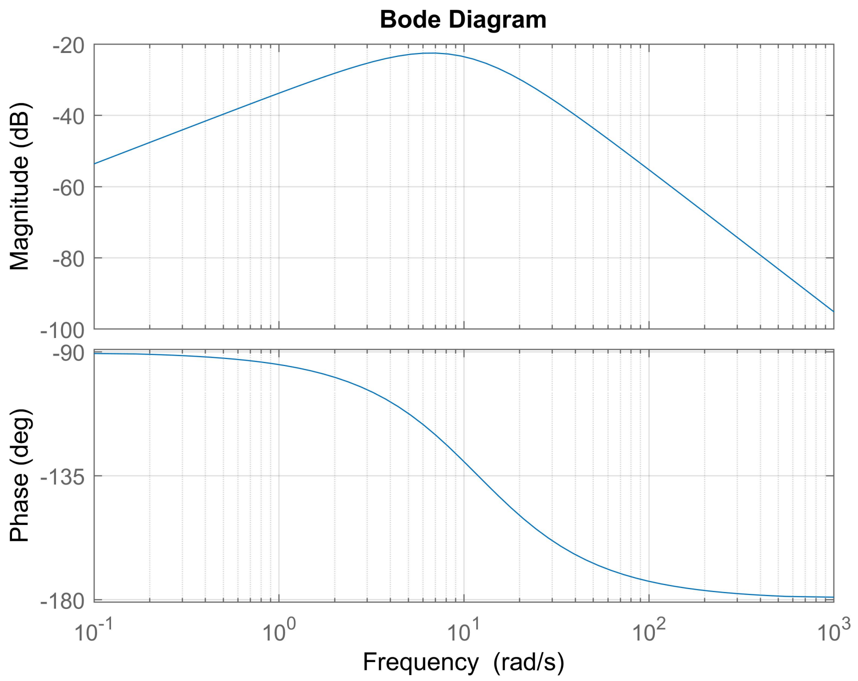

Figure 5.8: Frequency Response of Pendulum Transfer Function

The Frequency Loop Shape (FLS) method discussed in [12, 13], shows the design of a controller using an LQR computation approach based on the design parameters such as bandwidth, pole location, and the pseudo pole of the differentiator. The system is sampled at 100 Hz frequency, so the control bandwidth should be half the system bandwidth to satisfy the Nyquist Sampling Theorem and since the plant has a right half plane pole at 8.3675 rad/sec, the controller bandwidth should be greater than the bandwidth of the right half plane pole. Thus, the controller bandwidth was hence chosen to be 10 rad/sec and the right half plane pole of the controller placed an order of magnitude below the system right half plane pole so that the poles and zeros of the system move to the left half plane making the open loop system stable as shown in Figure 5.9.

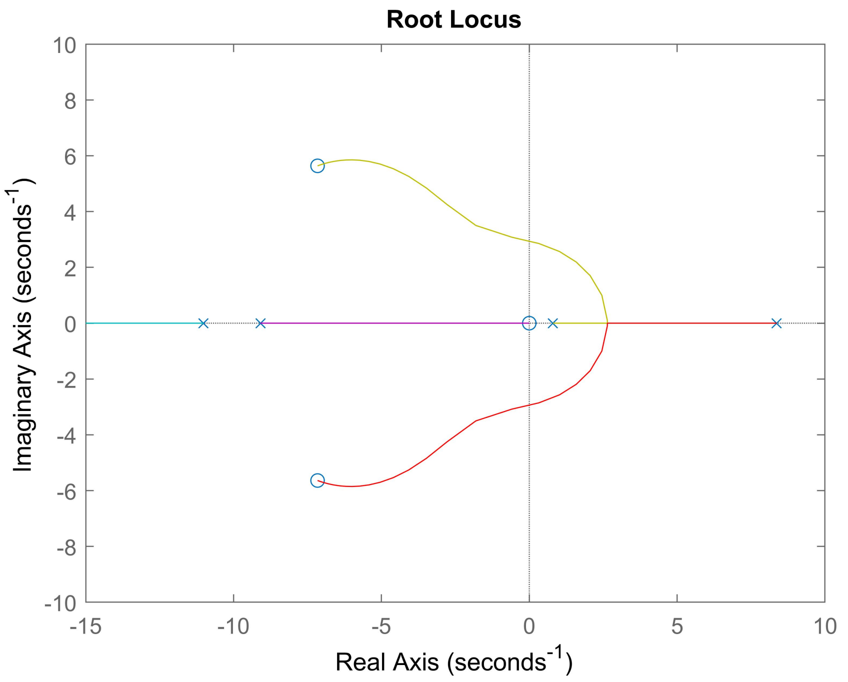

Figure 5.9: Root Locus of Open Loop System

Using these parameters the controller was designed which resulted in an open loop stable system with good sensitivity characteristics at control bandwidth as shown in Figure 5.10

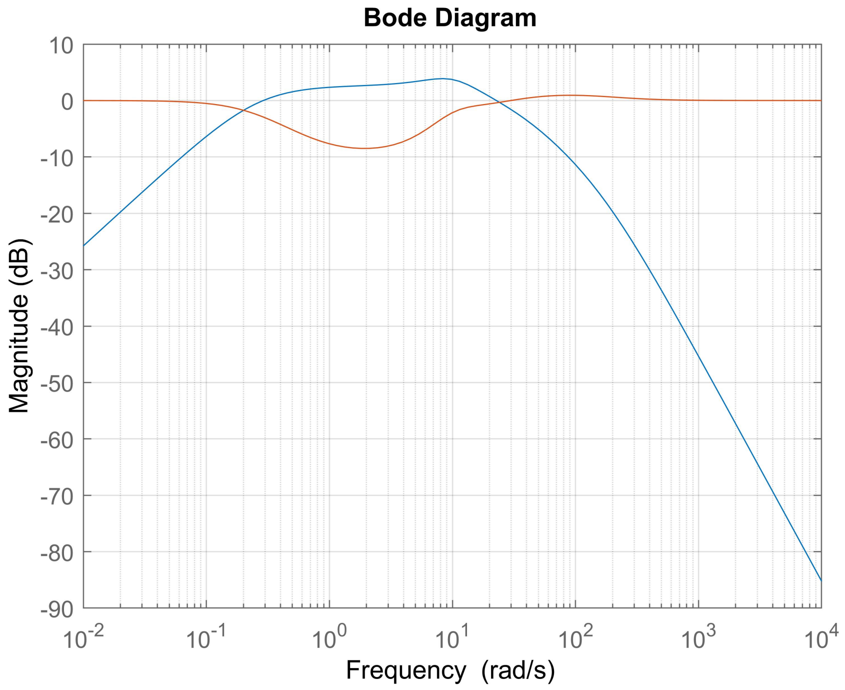

Figure 5.10: Closed Loop Sensitivity and Complementary Sensitivity of Inner Loop

This system cannot deal with constant disturbances and hence the sensitivity is reduced around the crossover frequency. However, this sensitivity response has a good disturbance rejection, it has a very aggressive control action which might damage the hardware. To reduce this effect of control action a low pass filter is added to the plant output so that we get a smoother response but with a trade-off in the sensitivities. A simple low pass filter with a cutoff frequency of 100 rad/sec, higher than the control bandwidth is used to achieve a faster roll-off at high frequencies. Figure 5.11 shows the frequency response of the low pass filter shown in Equation 5.14. The Frequency Response of the open loop system is shown in Figure 5.12 and the closed loop sensitivity and complementary sensitivity plot for the closed loop system after adding the low pass filter F is shown in Figure 5.13

$$F(s) = \frac{1}{0.01s + 1} \tag{5.14}$$

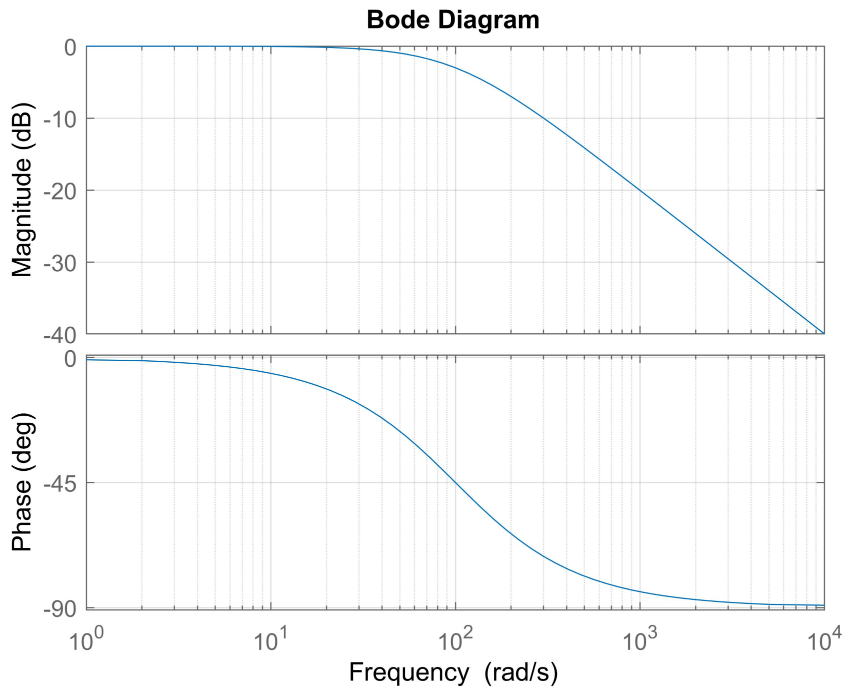

Figure 5.11: Frequency Response of Low Pass Filter

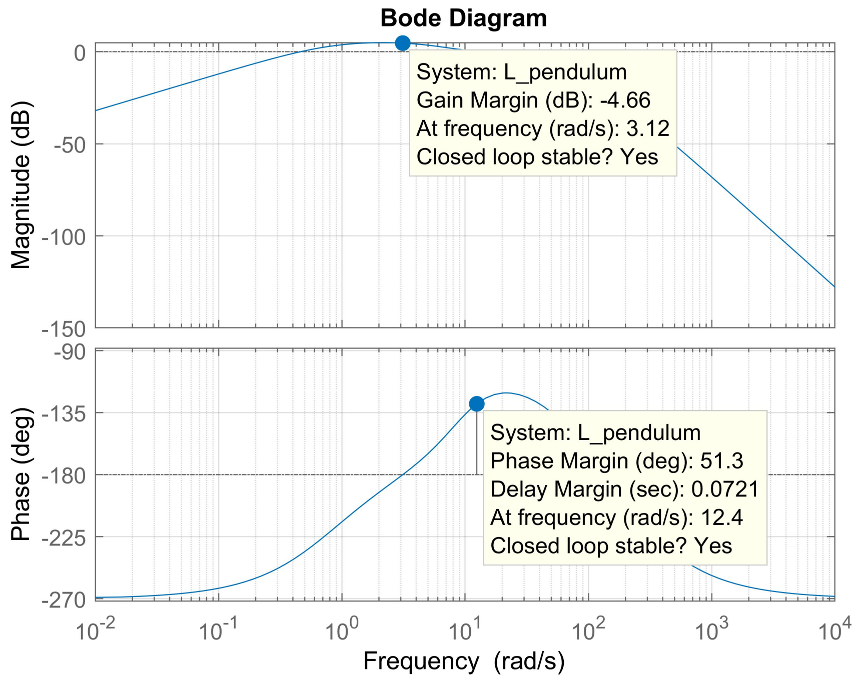

Figure 5.12: Open Loop Frequency Response of Stabilizing Pendulum Control System

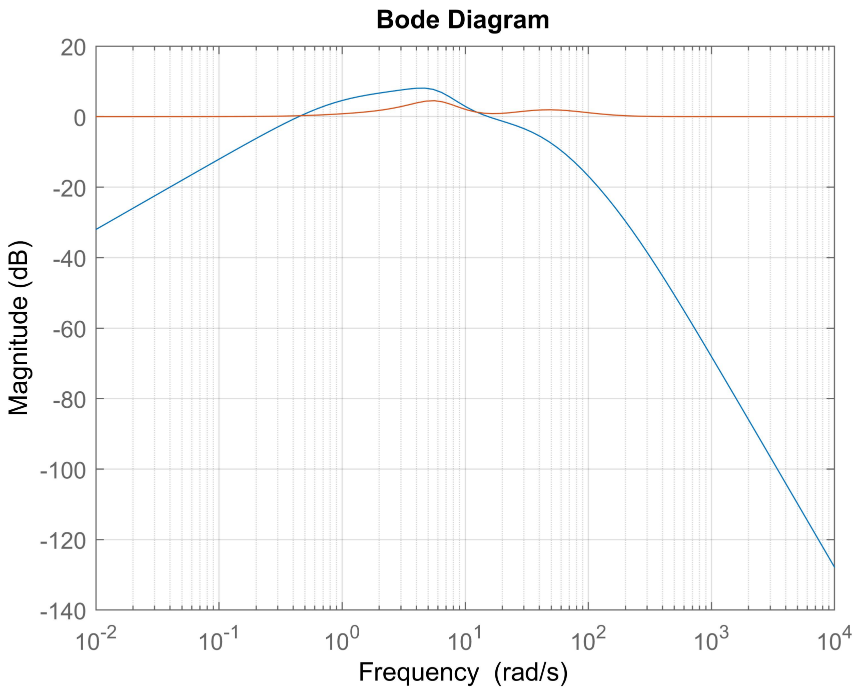

Figure 5.13: Closed Loop Sensitivity and Complementary Sensitivity of Inner Loop with Filter

Both the inner loop controller and the low pass filter mentioned above are in the continuous time domain. This cannot be directly implemented on physical hardware because they operate in the discrete time domain. Thus for mapping the poles and zeros from the complex $j\omega$ plane to be within a unit circle in the complex z plane Tustin or Bi-linear Transforms are used [31]. The Transfer function of the controller is got by substituting Equation 5.15 in Equation 5.12.

$$s = \frac{2}{T_s}\frac{z-1}{z+1} \tag{5.15}$$

where T_s is the sample time of the discrete system in sec.

$$C(z) = K_p + \frac{K_i T_s(z+1)}{z(2-\epsilon T_s)-(2+\epsilon T_s)} + \frac{2K_d(z-1)}{z(2\tau+T_s)-(2\tau-T_s)} \tag{5.16}$$

It can be seen that the Proportional branch of the controller is not affected by the discretization but the Integral and Derivative branches of the controller are changed to discrete time. To avoid Integrator Wind-up due to actuator saturation, the Integral part of the controller is turned off when the actuator reaches their physical saturation (i.e in this case the actuator saturation is from the DC motors). Following the work and method shown in [1] we can write the discrete time difference equation for the modified Integral branch as

$$u_i(k) = \max\left(\min\left(\left[\frac{2+\epsilon T_s}{2-\epsilon T_s}\right] u_i(k-1) + \left[\frac{K_i T_s}{2-\epsilon T_s}\right](e(k)+e(k-1)), u_{max}\right), u_{min}\right) \tag{5.17}$$

where u_{max} and u_{min} are the saturation range or constraints to the inner loop controller derived from the energy controller as mentioned in Equation 5.10.

Similarly, the Derivative branch of the controller can be written in the form of discrete time difference equation as

$$u_d(k) = \left[\frac{2K_d}{T_s+2\tau}\right](e(k)-e(k-1)) - \left[\frac{T_s-2\tau}{T_s+2\tau}\right] u_d(k-1) \tag{5.18}$$

The design parameters used to derive the inner loop controller are shown in Table 5.3

Parameter	Value
BW	10 $\frac{rad}{sec}$
ϵ	0.8 $\frac{rad}{sec}$
τ	0.005 sec

Table 5.3: Design Parameters for Stabilizing Inner loop Controller

Using these design parameters, we design the modified PID controller for the inner loop plant with the augmented low pass filter F as shown in Equation 5.19

$$\frac{\theta(s)}{V_{pwm}(s)} = \frac{17.46s}{s^3 + 11.74s^2 - 68.1s - 838} \frac{1}{0.01s + 1} \tag{5.19}$$

The controller designed for the above transfer function is written in the form of Equation 5.11 as

$$C_i(s) = \frac{1.158s^2 + 16.58s + 96.18}{0.005s^2 + 0.996s - 0.8} \tag{5.20}$$

where the gains of the modified PID controller are given in Table 5.4

Gain	Value
K_p	16.9620
K_i	109.7500
K_d	1.0733
τ	0.0050

Table 5.4: PID Gains for Stabilizing Inner Loop Controller

Since the objective of this controller is to stabilize the pendulum around the unstable equilibrium point, we look at the input disturbance rejection properties of the closed loop system. The Step input disturbance an Impulse input disturbance rejection of the simulated closed loop system is shown in Figure 5.14 and Figure 5.15

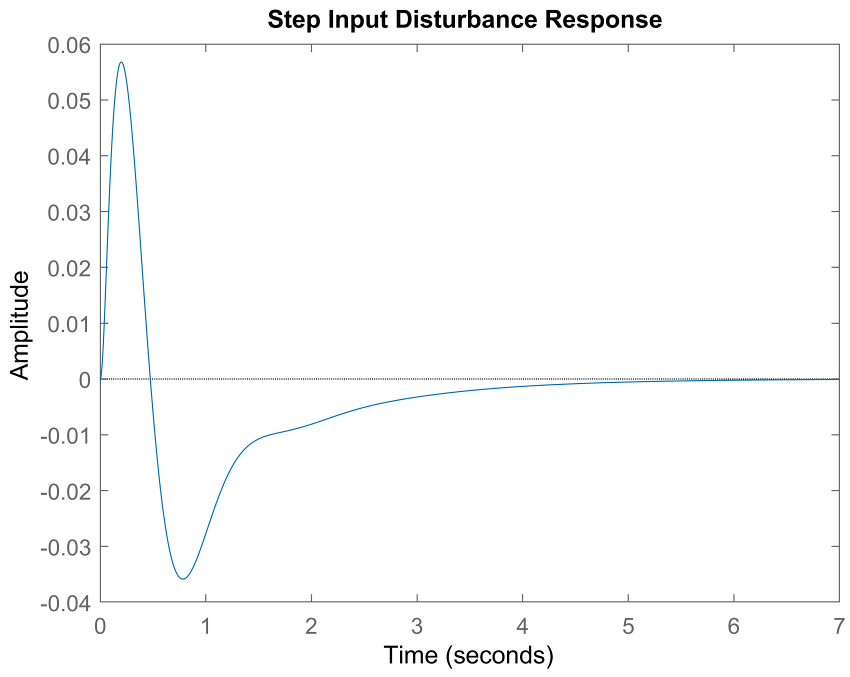

Figure 5.14: Step Response of the Stabilizing Inner Loop Controller to Input Disturbance

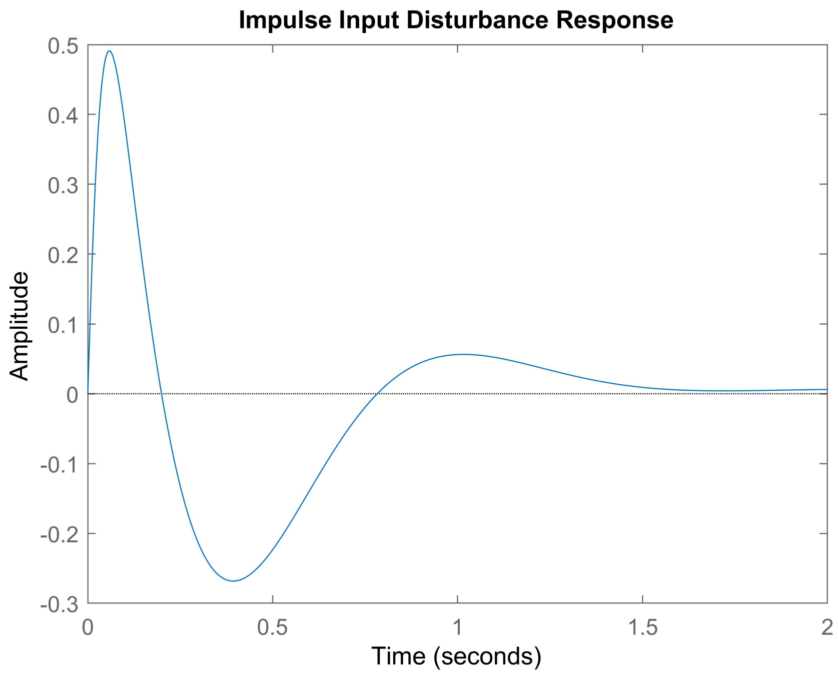

Figure 5.15: Impulse Response of the Stabilizing Inner Loop Controller to Input Disturbance

5.5 Outer Loop Position Control

The outer loop control is used to control the position of the robot while balancing the pendulum. As mentioned in Section 2.2 the system is a Single Input Multi Output (SIMO) system and the only input to the system is PWM input. Since the balancing of the pendulum also is dependent on the same input, we follow the classic cascade control technique. The position plant model has the entire dynamics of the pendulum plant and stabilizing the inner loop controller. Hence, we calculate the outer loop transfer function model based on the control structure shown in Figure 5.16

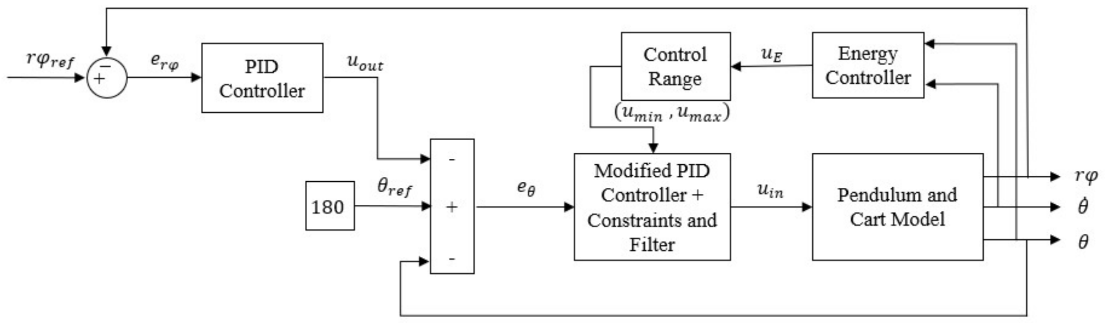

Figure 5.16: Outer Loop Control Structure

The outer loop position transfer function is calculated by solving the algebraic loop as shown in the above control structure and the position transfer function is given as,

$$\frac{y(s)}{V_{pwm}(s)} = \frac{-811.9\,s^4 - 1.316e^4\,s^3 - 2.015e^4\,s^2 + 8.643e^5\,s + 5.752e^6}{s(s^6 + 310.9\,s^5 + 2.321e^4\,s^4 + 5.994e^5\,s^3 + 4.008e^6\,s^2 + 1.812e^7\,s + 1.341e^7)} \tag{5.21}$$

The Frequency response of Equation 5.21 is shown in Figure 5.17 and the corresponding root locus is shown in Figure 5.18

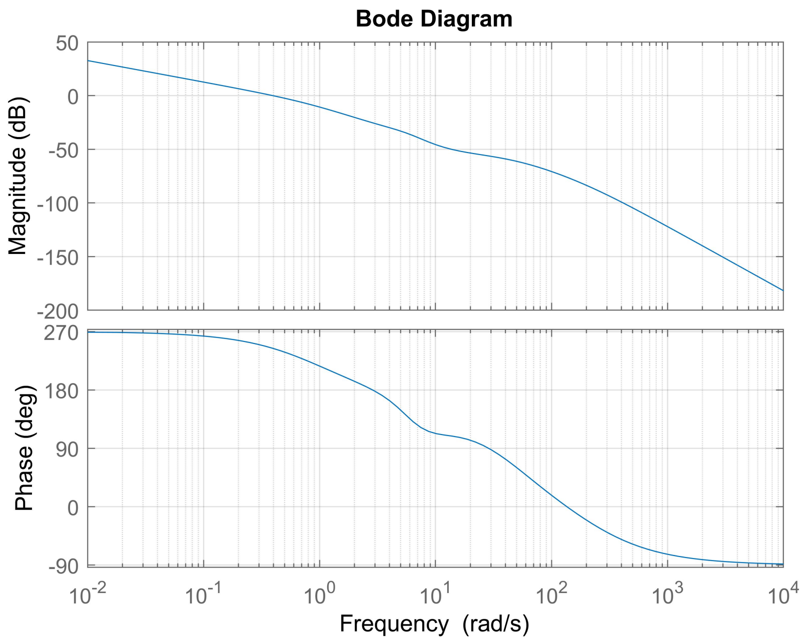

Figure 5.17: Frequency Response of Position Transfer Function

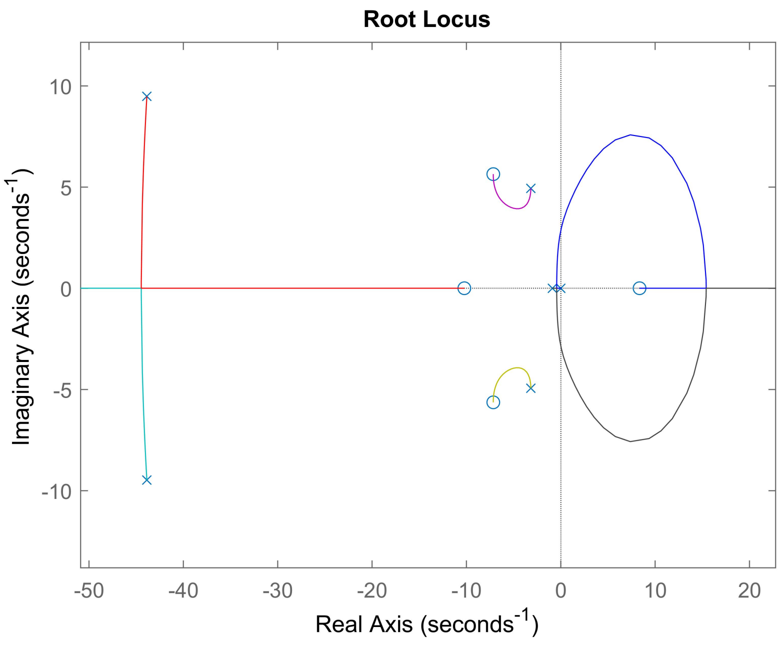

Figure 5.18: Root Locus of Position Transfer Function

From the above Root locus plot, we get an insight that two poles are very close to the origin and a right half plane zero. The location of the right half plane zero is at 8.34 rad/sec and to design any type of control system the bandwidth of the controller must be less than the bandwidth of the unstable zero. Hence, we design the outer loop position controller with low bandwidth and make sure we have a low derivative action to keep the inner loop stable at all times.

5.5.1 Outer Loop Tuning

The Frequency response of the open loop system is shown in Figure 5.19 and the closed loop sensitivity and complementary sensitivity is shown in Figure 5.20

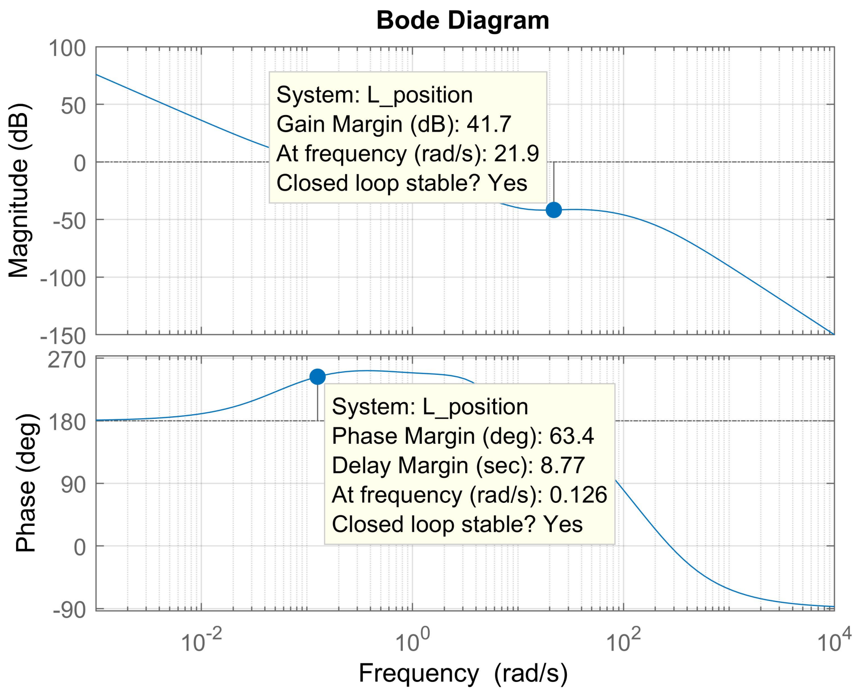

Figure 5.19: Open Loop Frequency Response of the Outer Loop Position control

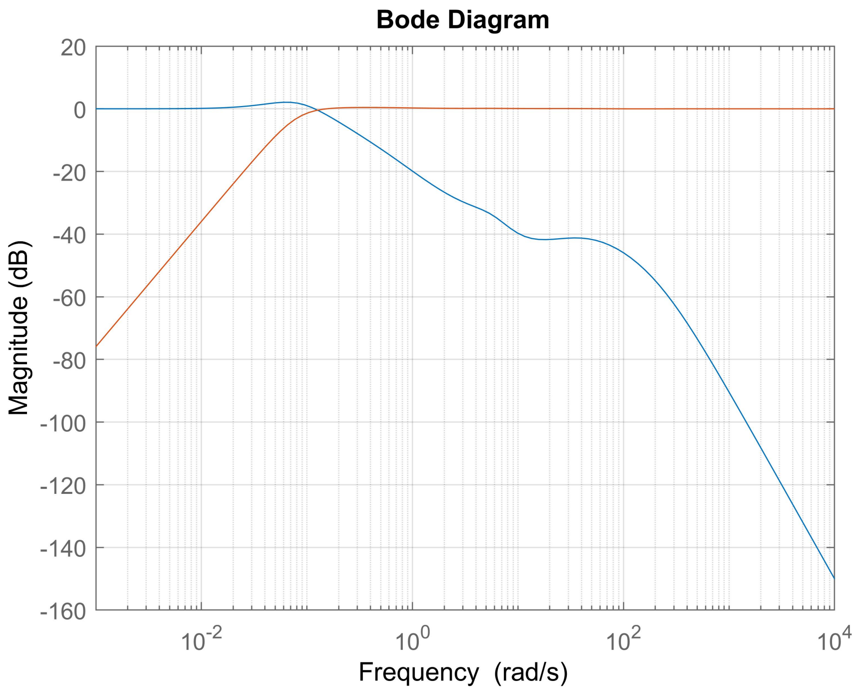

Figure 5.20: Closed Loop Sensitivity and Complementary Sensitivity of Outer Loop Position control

The position transfer function is a stable system and a PID controller is designed for this plant using the same FLS approach [12] shown in Subsection 5.4.2. The PID controller is designed for the parameters shown in Table 5.5

Parameter	Value
BW	$0.1 \ \frac{rad}{sec}$
ϵ	$0 \ \frac{rad}{sec}$
τ	$0.005 \ sec$

Table 5.5: Design Parameters for Outer Loop Position controller

The controller designed for the outer loop transfer function shown in Equation 5.21 is written in the form of Equation 5.11 with the value of $\epsilon = 0$ which mean we have a pure integrator ensuring zero steady-state error for command following. The outer loop controller transfer function is shown in Equation 5.22

$$C_o(s) = \frac{0.1938s^2 + 0.2826s + 0.01461}{0.005s^2 + s} \tag{5.22}$$

where the gains of the outer loop PID controller are given in Table 5.6

Gain	Value
K_p	0.2825
K_i	0.0146
K_d	0.1923
τ	0.0050

Table 5.6: PID Gains for Outer Loop Position Controller

Figure 5.21 shows the simulated step response of the outer loop position control.

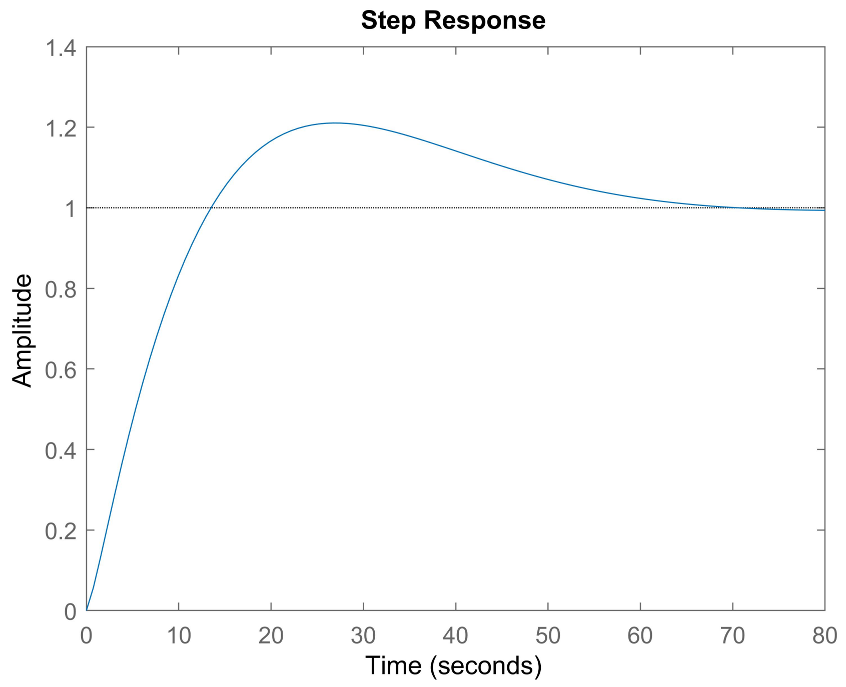

Figure 5.21: Step Response of the Outer Loop Position Control

5.6 Results

The controllers designed in Subsections 5.3.2, 5.4.2 and 5.5.1 are implemented on hardware and the outputs from the system are shown. The swing-up of the pendulum using an Energy controller is shown in Figure 5.22. The Energy controller swings up the pendulum to the inverted position for a gain of $k_e = 6$

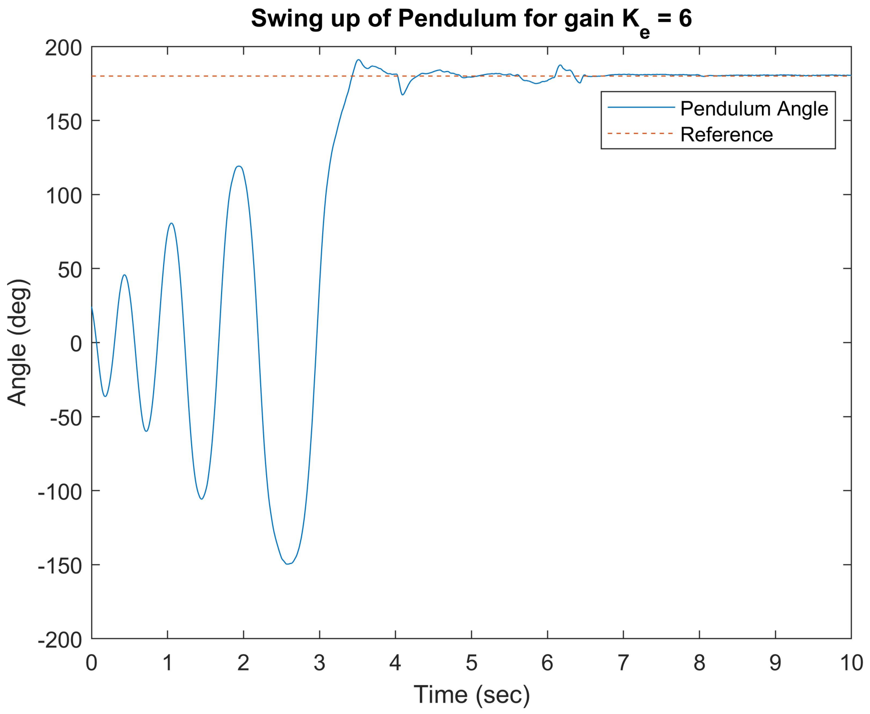

Figure 5.22: Swinging Up of Pendulum using Energy Control

As the pendulum reaches near the unstable equilibrium point the stabilizing inner loop controller takes control and balances the pendulum. This can be seen in Figure 5.23.

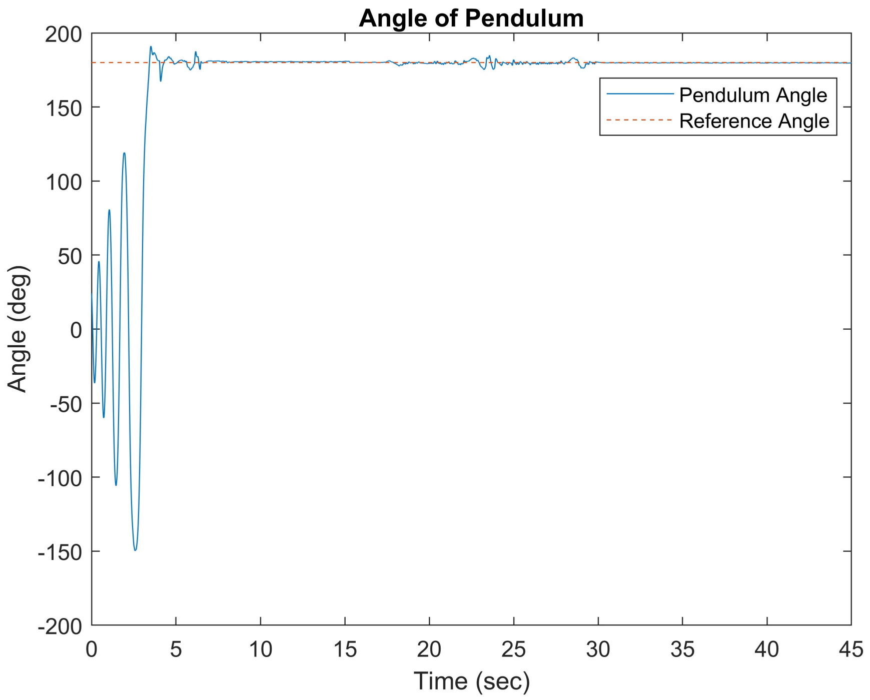

Figure 5.23: Balancing of Pendulum using Stabilizing Inner Loop Control

The position of the cart system is the other output of concern, and the outer loop controller has to keep the position of the cart maintained at 0 meters. This is shown in Figure 5.24.

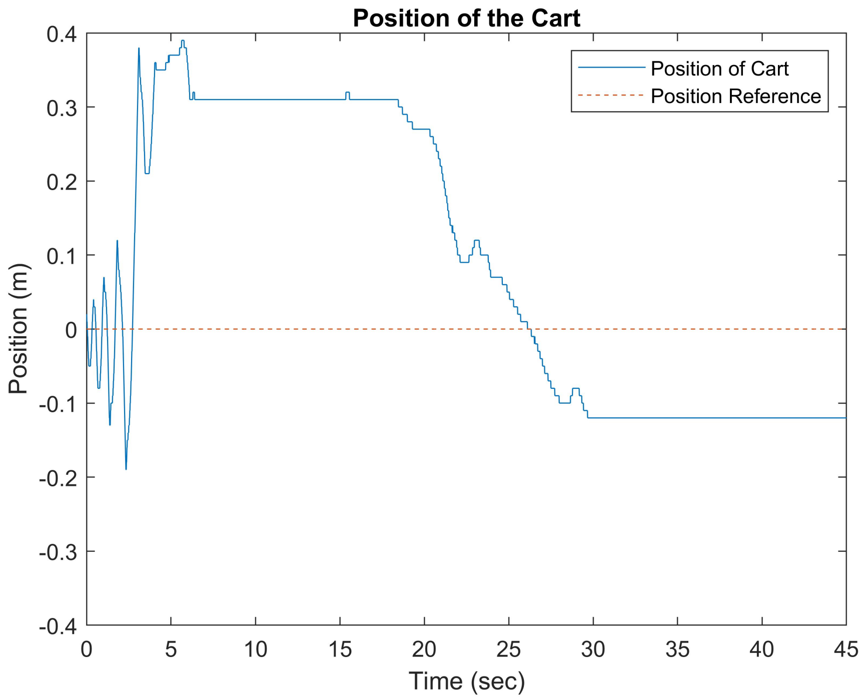

Figure 5.24: Position of Cart using Outer Loop Control

The controller input for the Inverted pendulum on a Cart system is shown with limits provided by the energy controller in Figure 5.25.

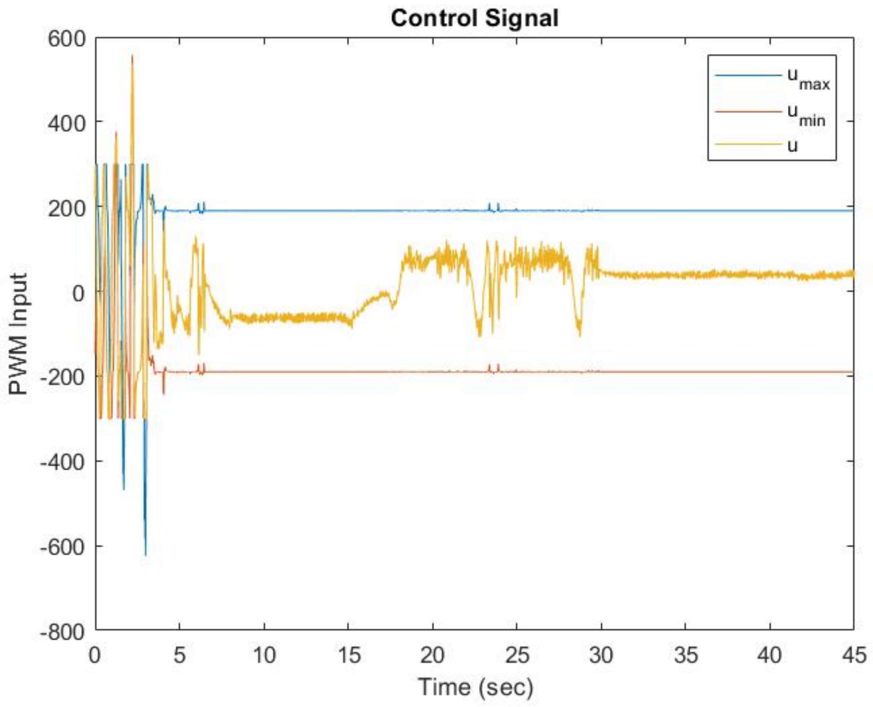

Figure 5.25: Control Signal with the Limitations from Energy Control

Finally, the combined output of the Pendulum angle and Cart position is shown in Figure 5.26 and the response of the system to input disturbance is shown in Figure 5.27.

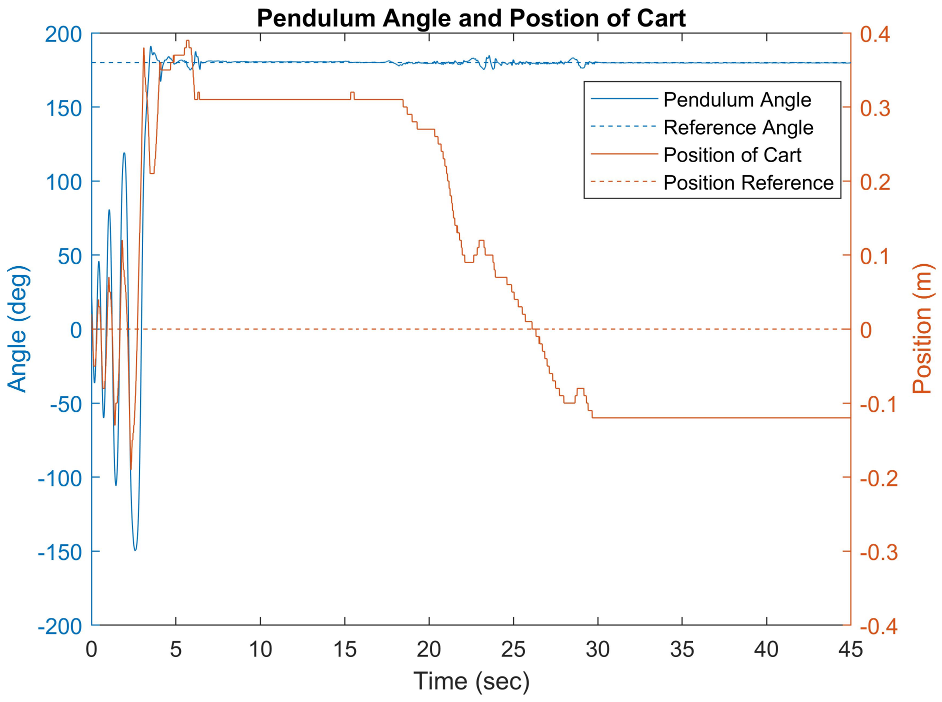

Figure 5.26: Pendulum Angle and Position of Cart

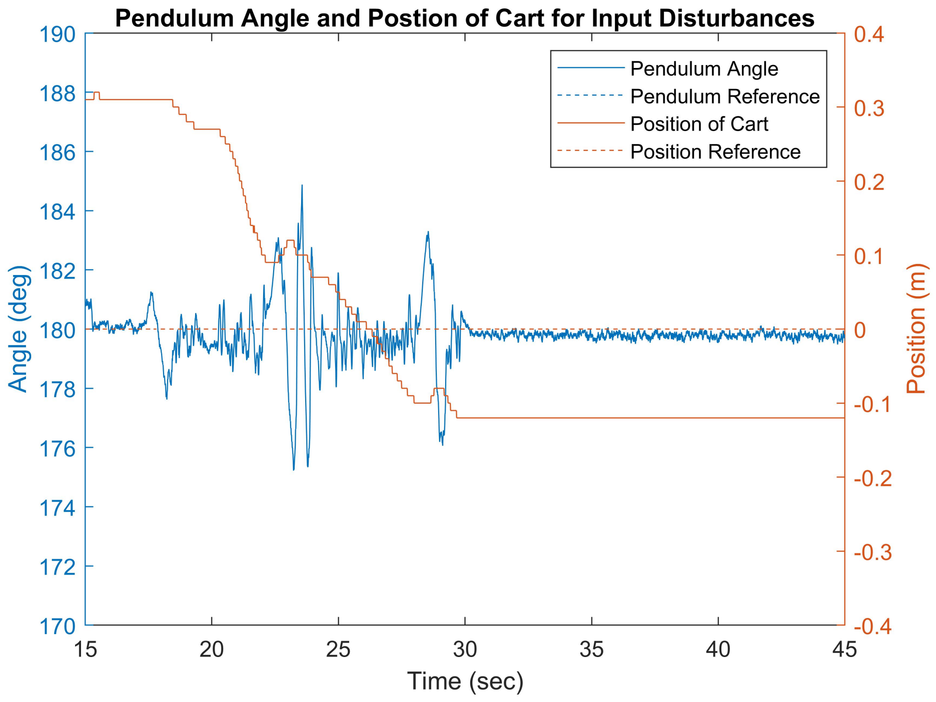

Figure 5.27: Pendulum Angle and Position of Cart for an Input Disturbance

5.7 Conclusions

This chapter briefly described the type of controllers used on the robot to balance the inverted pendulum on a cart. The control structures for each of the control loop were specified and the implementation of the controller in discrete time with integrator anti-windups was discussed.

The design specifications for the yaw rate, the energy, the modified PID, and outer loop PID controllers were discussed and the significance of using different methods to design these controllers was discussed.

Finally, these controllers were designed and implemented on hardware, and the results based on the First Principles model were shown and their performances were discussed.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Summary

The Inverted pendulum on a cart system has been studied for various research applications for a very long time. Most of the systems shown in the literature did not include the dynamics of the DC motors and either had a simulation model of the system or a lab model of the system actuated in a rail track for a fixed length. Most of these systems also had system parameters that were easier to control. For example, the length of the pendulum was usually between $0.5\,m$ and $1\,m$ which makes it easy to balance.

In this thesis, the Inverted Pendulum on a cart system was designed for a robot-like model as shown in Figure 4.1. The modeling of the robot was done using Newton-Euler's first principles approach. The DC motor dynamics were added to the system of equations and the final equations were used to build simulation models and for parameterization. The robot was mostly 3D printed, hence making it very cost-effective and easy to replace. The system was designed as a small robot with a pendulum of length less than 25% of the typical length discussed in most of the literature. Using sensors like the Inertial Measurement Unit (IMU) and encoders to measure the necessary states of the system made it a hard problem. After the construction of the robot the parameter was estimated based on few experiments and all the parameters of the system are listed in Table 4.1

The parameters were substituted into the generic system equations shown in Equations 2.26 and 2.27 to derive the actual plant model transfer function shown in Equation 4.14. Based on the plant model there were four controllers used to balance the inverted pendulum on a cart. Firstly, a yaw rate controller using the Ziegler-Nichols method was used to drive the robot in a straight line so that the yaw of the robot was controlled. The stabilizing and balancing of the pendulum was taken care of by the inner loop modified PID controller which was designed using a frequency loop shaping technique for the unstable system. An outer loop PID controller was added to maintain the position of the system. The tuning of these controllers with design parameters was discussed.

A new Energy control algorithm was discussed in Section 5.3 which provided limits for the inner loop controller based on the energy of the pendulum. This technique enabled both the inner loop controller and the energy controller to be always online resulting in a smooth transfer of control between them. The control input of the inner loop controller was always within the limits provided by the energy controller.

All the controllers were implemented on the physical hardware and the results of the system balancing an inverted pendulum on a cart were shown along with the system response to input disturbances.

6.2 Future Work

The future work that can be done on this robot to further improve this research are listed below:

- Implementation of a robust adaptive controller that includes a system identification component and generates suitable excitation, in order to improve the system response.
- Comparison with a more traditional switching controller strategy where an energy controller swings up the pendulum and then hands-off to a linear controller for stabilization.
- Rotational pendulum, where the pendulum is mounted on a pendulum arm rotating on a horizontal plane (no need for tracks).
- Dual link problem (Difficult problem even as a simulation with much more difficult construction).
- Modern controllers like MPC, MHE can be evaluated in this robot, but at the cost of a new microcontroller that can process at a higher bandwidth.

REFERENCES

[1] Aggarwal, V., *Quadrotor Design, Modeling and Control*, Ph.D. thesis, Arizona State University (2020).

[2] Ashfaque, B. S. and K. Tsakalis, "Discrete-time pid controller tuning using frequency loop-shaping.", IFAC Proceedings Volumes **45**, 3, 613–618 (2012).

[3] Åström, K. J. and K. Furuta, "Swinging up a pendulum by energy control", Automatica **36**, 2, 287–295 (2000).

[4] Braun, M., D. Rivera, A. Stenman, W. Foslien and C. Hrenya, "Multi-level pseudo-random signal design and" model-on-demand" estimation applied to nonlinear identification of a rtp wafer reactor", in "Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)", vol. 3, pp. 1573–1577 (IEEE, 1999).

[5] Cannon, R. H., *Dynamics of physical systems* (Courier Corporation, 2003).

[6] Chapala, S. R., G. S. Pirati and U. R. Nelakuditi, "Determination of coordinate transformations in uavs", in "2016 Second International Conference on Cognitive Computing and Information Processing (CCIP)", pp. 1–5 (IEEE, 2016).

[7] Chatterjee, D., A. Patra and H. K. Joglekar, "Swing-up and stabilization of a cart–pendulum system under restricted cart track length", Systems & control letters **47**, 4, 355–364 (2002).

[8] Chung, C. C. and J. Hauser, "Nonlinear control of a swinging pendulum", automatica **31**, 6, 851–862 (1995).

[9] Copeland, B. R., "The design of pid controllers using ziegler nichols tuning", Internet: http://educypedia. karadimov. info/library/Ziegler_Nichols. pdf (2008).

[10] Due, A. and A. Core, "Arduino due", Retrieved **9**, 16, 2019 (2017).

[11] Grassi, E. and K. Tsakalis, "Pid controller tuning by frequency loop-shaping", in "Proceedings of 35th IEEE Conference on Decision and Control", vol. 4, pp. 4776–4781 (IEEE, 1996).

[12] Grassi, E. and K. Tsakalis, "Pid controller tuning by frequency loop-shaping: application to diffusion furnace temperature control", IEEE Transactions on Control Systems Technology **8**, 5, 842–847 (2000).

[13] Grassi, E., K. S. Tsakalis, S. Dash, S. V. Gaikwad, W. MacArthur and G. Stein, "Integrated system identification and pid controller tuning by frequency loop-shaping", IEEE Transactions on Control Systems Technology **9**, 2, 285–294 (2001).

[14] Joglekar, H., D. Chatterjee and A. Patra, "Swing-up and stabilization of a cart-pendulum system using energy control", in "EAIT Conference Proceedings", pp. 33–37 (2001).

[15] Lundberg, K. H. and T. W. Barton, "History of inverted-pendulum systems", IFAC Proceedings Volumes **42**, 24, 131–135 (2010).

[16] Madgwick, S., "An efficient orientation filter for inertial and inertial/magnetic sensor arrays", Report x-io and University of Bristol (UK) **25**, 113–118 (2010).

[17] Madgwick, S. O., A. J. Harrison and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm", in "2011 IEEE international conference on rehabilitation robotics", pp. 1–7 (IEEE, 2011).

[18] Mark Drela, MIT Aero and Astro, "First-order dc electric motor model", (2007).

[19] Mori, S., H. Nishihara and K. Furuta, "Control of unstable mechanical system control of pendulum", International Journal of Control **23**, 5, 673–692 (1976).

[20] Muskinja, N. and B. Tovornik, "Swinging up and stabilization of a real inverted pendulum", IEEE transactions on industrial electronics **53**, 2, 631–639 (2006).

[21] Nour, M., J. Ooi and K. Chan, "Fuzzy logic control vs. conventional pid control of an inverted pendulum robot", in "2007 International Conference on Intelligent and Advanced Systems", pp. 209–214 (IEEE, 2007).

[22] Otani, Y., T. Kurokami, A. Inoue and Y. Hirashima, "A swingup control of an inverted pendulum with cart position control", IFAC Proceedings Volumes **34**, 22, 395–400 (2001).

[23] Papadopoulos, A., *Swinging Up the Inverted Pendulum by Energy Adaptive Proportional-integral-derivative Control*, Ph.D. thesis, Arizona State University (2002).

[24] Prof. Bill Messner, Prof. Dawn Tilbury, Prof. Rick Hill, Prof. Shuvra Das, JD Taylor, "Controls tutorials for matlab and simulink", URL https://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=SystemModeling, [Online; accessed 03-November-2020] (2019).

[25] Rodriguez, A., *Analysis and Design of Feedback Control Systems* (CONTOL3D, L.L.C., Tempe, Az 85284, 2003).

[26] Schaefer, I. and R. Cannon, "On the control of unstable mechanical systems", Research Review **5**, 5, 11 (1966).

[27] Sinha, N. K., C. D. Dicenzo and B. Szabados, "Modeling of dc motors for control applications", IEEE Transactions on Industrial Electronics and Control Instrumentation **IECI-21**, 2, 84–88 (1974).

[28] Tsakalis, K., S. Dash, A. Green and W. MacArthur, “Loop-shaping controller design from input-output data: application to a paper machine simulator”, IEEE transactions on control systems technology **10**, 1, 127–136 (2002).

[29] Tsakalis, K., J.-J. Flores-Godoy and K. Stoddard, “Temperature control of diffusion/cvd furnaces using robust multivariable loop-shaping techniques”, in “Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304)”, vol. 4, pp. 4192–4197 (IEEE, 1999).

[30] Wang, L., T. Barnes and W. R. Cluett, “New frequency-domain design method for pid controllers”, IEE Proceedings-Control Theory and Applications **142**, 4, 265–271 (1995).

[31] Wikipedia contributors, “Bilinear transform — Wikipedia, the free encyclopedia”, URL https://en.wikipedia.org/w/index.php?title=Bilinear_transform&oldid=936849926, [Online; accessed 03-November-2020] (2020).

[32] Wikipedia contributors, “Ziegler- nichols method — Wikipedia, the free encyclopedia”, URL https://en.wikipedia.org/w/index.php?title=Bilinear_transform&oldid=936849926, [Online; accessed 03-November-2020] (2020).

[33] Yang, J.-H., S.-Y. Shim, J.-H. Seo and Y.-S. Lee, “Swing-up control for an inverted pendulum with restricted cart rail length”, International Journal of Control, Automation and Systems **7**, 4, 674–680 (2009).

[34] Yoshida, K., “Swing-up control of an inverted pendulum by energy-based methods”, in “Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)”, vol. 6, pp. 4045–4047 (IEEE, 1999).

[35] Ziegler, J. G. and N. B. Nichols, “Optimum settings for automatic controllers”, trans. ASME **64**, 11 (1942).

APPENDIX A

SOURCE CODE

```
// Library Files
#include "MadgwickAHRS.h"
#include "MPU6050.h"
#include "I2Cdev.h"
#include "HMC5883L.h"
#include "SPI.h"
#include "Wire.h"
#include "math.h"
#include "PRBS.h"
#include "PID.h"
#include "DualVNH5019MotorShield.h"
#include "Encoder.h"
#include "Diff.h"
#include "firstorderFilter.h"
#include "mPID.h"
#include "prefilter.h"
#include "m1PID.h"


//Pin Definitions
#define statusLED 13
#define RX_LED 11

//Left Motor
#define enaR 24
#define enbR 25

//Right Motor (Pendulum sensor side)
#define enaL 50
#define enbL 51

//Motor Variables
float rpmR = 0, rpmL = 0, posR = 0, posL = 0, pos = 0, rpm = 0;
float r = 0.0375; //radius of wheel in meters
float intMin = -200, intMax = 200; //integrator saturation
float actMin = -400, actMax = 400; //actuator saturation
float MR_in, ML_in;
float dzn = 0, DEADZONE = 50;

//Motor and Encoder Objects
DualVNH5019MotorShield md(2, 4, 9, 10, A0, 7, 8, 6, 12, A1);

Encoder en1(enaR, enbR);
Encoder en2(enaL, enbL);

Diff v1;//Differentiator for velocity measurement
Diff v2;
```

```
// IMU and Madgwick Objects
MPU6050 accelgyro(0x68);//GY86
MPU6050 imu521(0X69); //GY521
Madgwick filter;

//First order Filter
firstorderFilter f1, g1;

//IMU Variables
int16_t ax, ay, az, ax1, ay1, az1;
float AX, AY, AZ, AX1, AY1, AZ1;
int16_t gx, gy, gz, gx1, gy1, gz1;
float GX, GY, GZ, GX1, GY1, GZ1, GZ1_fil;
int16_t mx, my, mz;
float MX, MY, MZ;
float roll, pitch, yaw;

//IMU Calibration variables
volatile int buffersize = 1000;
volatile int mean_gx, mean_gy, mean_gz, mean_gx1, mean_gy1, mean_gz1;

//Angle Measurement Variables
float roll_prev = 0;
float roll_gl = 0;
float roll_new;
int rot = 0; int Counter = 0;
int Counter_prev = 0;
float theta_dot = 0;
float ang = 0;
int n = 0;

//Sampling
uint32_t loopTimer;
float F = 100;
float Ts = 1 / F;
uint16_t f = 100;
float t;
float tau = 1 / (2 * F);

//Control Input Variables
float u, u_R, u_L;

// Yaw Rate Control PID
PID gzPID;
float ref_gz = 0;
float error_gz;
float u_gz;
float Kp_gz, Ki_gz, Kd_gz, tau_gz;

//Pendulum Control PID
m1PID penPID;
float u_in; //innerloop control variable
float ref_pen = 180.00;
float error_pen;
float u_pen;
float Kp_pen, Ki_pen, Kd_pen, tau_pen;
float p;

//Position Control PID
PID posPID;
float u_out;//outerloop control variable
float ref_pos = 0;
float error_pos;
float u_pos;
float Kp_pos, Ki_pos, Kd_pos, tau_pos;

//Energy Control Parameter
float g = 9.81; //acceleration due to gravity
float l = 0.115; //length of pendulum
float m = 0.020; //mass of pendulum
float cp = 0.0005;
float a = m * g * l;
float b = m * l * l;
float ref_e = 2 * a; //reference for energy control
float rad = M_PI / 180; //constant to convert deg to rad
float eMin = -190, eMax = 190; //control input saturation
float ueMax, ueMin; // value to the linear controller
float Kp_e = 6; //energy control input gain
float e_control, u_e, u_gain , error_u; //control variables

void setup() {

pinMode(statusLED, OUTPUT); // status LED
pinMode(RX_LED , OUTPUT);
digitalWrite(RX_LED, LOW);

md.init();// initializing the motor driver pinmodes and Timer 1
v1.setup(tau, Ts); //differentiator for left motor
v2.setup(tau, Ts); //differentiator for right motor

f1.setup(f, Ts); //plant filter
g1.setup(10, Ts); //yaw rate filter
```

```
Wire.begin();

// setup for IMU and controllers
imusetup();
gzControlSetup();
pendControlSetup();
posControlSetup();

//IMU calibration
digitalWrite(statusLED , HIGH);
gyroCalibration();
digitalWrite(statusLED , LOW);

Serial.begin(115200);
Serial1.begin(115200);

filter.begin(100); //Madgwick filter sampling freq

delayMicroseconds(5000000);
loopTimer = micros(); //clock

}

void loop() {

digitalWrite(RX_LED, HIGH); //loop execution indication

//functions for sensor and motor calculations
imuval();
pendAngle();
cartPos();

//functions for energy,inner and outer loop control
pendControl();

// Motor Actuation for Control Input
MotorActuation();

//print0(); // USB debugging
print1(); //SD Card data logging

// sample time overrun condition
if (micros() - loopTimer >= (1000000 / F)) digitalWrite(statusLED, HIGH);
else digitalWrite(statusLED, LOW);
while (micros() - loopTimer < (1000000 / F));
loopTimer = micros();
```

```
t = (micros() / float(1000000)); //time variable

}

void imusetup(void) {

accelgyro.setI2CMasterModeEnabled(false);
accelgyro.setI2CBypassEnabled(true) ;
accelgyro.setSleepEnabled(false);

//Initializing device
Serial.println("Initializing␣I2C␣devices...");
accelgyro.initialize();
imu521.initialize();

// verify connection
Serial.println("Testing␣device␣connections...");
Serial.println(accelgyro.testConnection() ? "MPU6050␣connection␣successful" :
"MPU6050␣connection␣failed");

}

void gyroCalibration() {

long i = 0, buff_gx = 0, buff_gy = 0, buff_gz = 0, buff_gx1 = 0, buff_gy1 = 0,
buff_gz1 = 0;

while (i < (buffersize + 101)) {
accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
imu521.getMotion6(&ax1, &ay1, &az1, &gx1, &gy1, &gz1);

if (i > 100 && i <= (buffersize + 100)) {
buff_gx = buff_gx + gx;
buff_gy = buff_gy + gy;
buff_gz = buff_gz + gz;

buff_gx1 = buff_gx1 + gx1;
buff_gy1 = buff_gy1 + gy1;
buff_gz1 = buff_gz1 + gz1;

}
if ( i == (buffersize + 100)) {
mean_gx = buff_gx / buffersize;
mean_gy = buff_gy / buffersize;
mean_gz = buff_gz / buffersize;

mean_gx1 = buff_gx1 / buffersize;
mean_gy1 = buff_gy1 / buffersize;
mean_gz1 = buff_gz1 / buffersize;
}
i++;
delay(2);
}
}

void imuval(void) {

accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
imu521.getMotion6(&ax1, &ay1, &az1, &gx1, &gy1, &gz1);

AX = ax / float(16384); //1g range for accelerometer
AY = ay / float(16384);
AZ = az / float(16834);

GX = (gx - mean_gx) / float(16.4); // 2000deg range for gyroscope
GY = (gy - mean_gy) / float(16.4);
GZ = (gz - mean_gz) / float(16.4);
GZ1 = (gz1 - mean_gz1) / float(16.4);

GZ1_fil = g1.filteredValue(GZ1);

//Madgwick update functions
filter.updateIMU(GX, GY, GZ, AX, AY, AZ);
roll = filter.getRoll(); //angle of pendulum
theta_dot = GX; //angular velocity of pendulum

}

float pendAngle() {

roll_new = roll - roll_prev;
//condition to map angle from +/- 0 to 360
if (roll_new - roll_prev < -355) {
rot++;
}
else if (roll_new - roll_prev > 355) {
rot--;
}

int Counter_new = rot; //rotation counter

if (roll < 0) n = 1;
else n = 0;
```

```
roll_gl = roll + (Counter_new * 360); // pendulum angle in deg
ang = roll + (n * 360); //pendulum angle 0 to 360
roll_prev = roll;
return roll_gl;

}

void MotorActuation(void) {

MR_in = speedSat(u_R) ; //right motor input
ML_in = speedSat(u_L) ; //left motor input
md.setSpeeds(MR_in, ML_in); //motor driver input function

}

//saturation function
float speedSat(float in) {
float out = max(min(in, intMax), intMin);
return out;
}

//sign function
float sign(float in) {
float out;
if (in > 0) out = 1;
else if (in < 0) out = -1;
else if (in = 0) out = 0;
return out;
}

//cart position and velocity calculation
void cartPos(void) {

posR = (en1.read() * 2 * M_PI * r) / 464.64; // position in meters
posL = (en2.read() * 2 * M_PI * r) / 464.64;

rpmR = v1.getRate(posR); //angular velocity in m/sec
rpmL = v2.getRate(posL); //angular velocity in m/sec

pos = (posR + posL) / 2;
rpm = (rpmR + rpmL) / 2;

}
```

```
void pendControl(void) {

//energy function K.E + P.E
u_e = (0.5 * b * sq(theta_dot)) + (a * (- cos(pendAngle() * rad) + 1));
error_u = u_e - ref_e; //energy error
//control input
u_gain = -1 * (Kp_e * error_u) * sign(theta_dot * cos(pendAngle() * rad));
ueMax = min(u_gain + eMax , intMax + 100);
ueMin = max(u_gain + eMin , intMin - 100);

//error calculations
error_gz = ref_gz - GZ1_fil; //yawrate error
u_gz = gzPID.getControl(error_gz, intMin, intMax); //yawrate control input
error_pos = ref_pos - pos; //outerloop error
u_out = posPID.getControl(error_pos, intMin, intMax); //outerloop control input
error_pen = ref_pen - ang - u_out; //innerloop error

//control inputs
u = penPID.getControl(error_pen, ueMin, ueMax); //innerloop control input
u_in = f1.filteredValue(u);
u_R = u_in + u_gz; //right motor control input
u_L = u_in - u_gz; //left motor control input

}

void gzControlSetup(void) {

Kp_gz = 0.24;
Ki_gz = 2.31;
Kd_gz = 0.006;
tau_gz = 1 / (2 * F);
gzPID.setup(Kp_gz, Ki_gz, Kd_gz, tau_gz, Ts);

}

void pendControlSetup(void) {

Kp_pen = 16.9620;
Ki_pen = 109.7500;
Kd_pen = 1.0733;
tau_pen = 1 / (2 * F);
p = 0.8; // Magnitude value of pole in RHP
penPID.setup(Kp_pen, Ki_pen, Kd_pen, tau_pen, Ts , p);

}
```

```
void posControlSetup(void) {

Kp_pos = 0.2825;
Ki_pos = 0.0146;
Kd_pos = 0.1923;
tau_pos = 1 / (2 * F);
posPID.setup(Kp_pos, Ki_pos, Kd_pos, tau_pos, Ts);

}

//Serial print function for usb debugging
void print0(void) {

Serial.print(t); Serial.print(",");
Serial.print(ref_pen); Serial.print(",");
Serial.print(pendAngle()), Serial.print(",");
Serial.print(roll_gl); //Serial.print(",");
Serial.print(ang); //Serial.print(",");
Serial.print(rot); Serial.print(",");
Serial.print(n);// Serial.print(",");
Serial.print(error_pen); Serial.print(",");
Serial.print(u_in); Serial.print(",");
Serial.print(GZ1); Serial.print(",");
Serial.print(error_gz); Serial.print(",");
Serial.print(u_gz); Serial.print(",");
Serial.print(u_in); Serial.print(",");
Serial.print(u_e); Serial.print(",");
Serial.print(u_gain); Serial.print(",");
Serial.print(error_u); Serial.print(",");
Serial.print(e_control); Serial.print(",");
Serial.print(uMax); Serial.print(",");
Serial.print(uMin); Serial.print(",");
Serial.print(u); Serial.print(",");
Serial.print(u_in); Serial.print(",");
Serial.print(u_R); Serial.print(",");
Serial.print(u_L); Serial.print(",");
Serial.print(posR); Serial.print(",");
Serial.print(posL); Serial.print(",");
Serial.print(rpmR); Serial.print(",");
Serial.print(rpmL);; Serial.print(",");
Serial.print(pos); Serial.print(",");
Serial.print(rpm);
Serial.println();

}
```

```
//Serial print function for datalogging
void print1(void) {

Serial1.print(t); Serial1.print(",");
Serial1.print(ref_pen); Serial1.print(",");
Serial1.print(pendAngle()); Serial1.print(",");
Serial1.print(ang); Serial1.print(",");
Serial1.print(theta_dot); Serial1.print(",");
Serial1.print(error_pen); Serial1.print(",");
Serial1.print(u); Serial1.print(",");
Serial1.print(u_in); Serial1.print(",");
Serial1.print(ref_gz); Serial1.print(",");
Serial1.print(GZ1); Serial1.print(",");
Serial1.print(error_gz); Serial1.print(",");
Serial1.print(u_gz); Serial1.print(",");
Serial1.print(u_e); Serial1.print(",");
Serial1.print(error_u); Serial1.print(",");
Serial1.print(u_gain); Serial1.print(",");
Serial1.print(e_control); Serial1.print(",");
Serial1.print(ueMax); Serial1.print(",");
Serial1.print(ueMin); Serial1.print(",");
Serial1.print(error_pos); Serial1.print(",");
Serial1.print(u_out); Serial1.print(",");
Serial1.print(u_R); Serial1.print(",");
Serial1.print(u_L); Serial1.print(",");
Serial1.print(pos);// Serial1.print(",");
Serial1.print(rpm);
Serial1.println();

}
```