



**DEVELOPMENT OF A CART-MOUNTED INVERTED
PENDULUM TEST BENCH**

Sarwan Shah

Muhammad Abdullah Siddiqui

Adil Ali Khan

HABIB UNIVERSITY

KARACHI, PAKISTAN

2021

**DEVELOPMENT OF A CART-MOUNTED INVERTED PENDULUM TEST
BENCH**

The Capstone Design Project
presented to the academic faculty

by

Sarwan Shah

Muhammad Abdullah Siddiqui

Adil Ali Khan

in partial fulfillment of the requirements for
BS Electrical Engineering
in the
School of Dhanani School of Science and Engineering

Habib University

May 2021



This work is licensed under a Creative Commons Attribution 3.0 License.

DEVELOPMENT OF A CART-MOUNTED INVERTED PENDULUM TEST BENCH

This capstone design project was advised by:

Dr. Basit Memon
Faculty of Electrical Engineering
Habib University

Approved by the Faculty of Electrical Engineering on May 18, 2021.

We read and write poetry because we are members of the human race, and the human race is filled with passion. Medicine, law, business, engineering, these are noble pursuits and necessary to sustain life. But poetry, beauty, romance, love, these are what we stay alive for.

Robin Williams as John Keating, Dead Poet's Society, 1989

We dedicate this project to each other, and to all the sleepless nights, all the stressed out days, all the tears, all our blood (not literally) and sweat (quite literally), and all the time we spent putting this together.

ACKNOWLEDGEMENTS

We would like to thank our supervisor Dr. Abdul Basit Memon for his tireless guidance, support and advice throughout this project. We would not be able to accomplish our goals without his tremendous help and thought provoking insights.

Our immense gratitude to Dr. Owais Talaat, Sir Saad Baig and Sir Junaid Ahmed Memon for their extensive feedback at every point.

We would also like to extend our gratitude to Research Assistant Zeeshn Nafees, Lab Assistant Muhammad Irfan, and Research Assistant Ahmad Bilal for their assistance in designing the project hardware and brainstorming ideas.

TABLE OF CONTENTS

List of Tables	xiv
List of Figures	xx
Chapter 1: Introduction and Background	1
1.1 Introduction	1
1.2 Motivation	5
1.3 Literature Review	6
1.3.1 Commercially Designed Systems	6
1.3.2 Systems Designed for Research & Education	8
1.3.3 Modelling Approaches	11
1.3.4 Control Approaches	12
1.3.5 Sensory Feedback Mechanism	13
1.3.5.1 Potentiometers	13
1.3.5.2 Rotary Encoders	14
1.3.5.3 Computer Vision	15
1.3.5.4 Discussion	18
1.4 Problem Statement/Objective	19
Chapter 2: Design Process	20
2.1 Stakeholders and their Requirements	20
2.1.1 Researchers	20

2.1.2	Students	21
2.1.3	Instructors	22
2.1.4	Institution	23
2.2	Design Requirements and Constraints	24
2.2.1	Requirements	24
2.2.2	Constraints	24
2.3	Design Concepts and Methodology	25
2.3.1	Track, Cart and Locomotion System	25
2.3.1.1	Track	25
2.3.1.2	Cart	27
2.3.1.3	Locomotion Mechanism	27
2.3.2	Motor and Power Module	29
2.3.2.1	DC Motors	30
2.3.2.2	Stepper Motors	30
2.3.2.3	Servo Motors	30
2.3.3	Sensor Feedback	31
2.3.3.1	Cart's Displacement	31
2.3.3.2	Pendulum's Angular Displacement	32
2.4	Society, Economic, and Ethical considerations	34
2.5	Environment and Sustainability considerations	36
2.6	Technical Specifications Highlights	37
2.6.1	Design Choices	37
2.6.2	System Characteristics	41
Chapter 3:	System Modelling	43

3.1	Introduction	43
3.2	DC Gear Motor Modelling	44
3.2.1	Generalized Motor Model	44
3.2.2	Grey Box Motor Model	46
3.2.3	White Box Motor Model	50
3.2.4	Motor Model Validation	52
3.2.5	Discussion	55
3.3	Process Modelling	57
3.3.1	Plant Dynamic Equations	57
3.3.2	Non-linear Plant Model	59
3.3.3	Non-Linear Plant Model with Viscous Friction	59
3.3.4	Linearized Plant Model with Viscous Friction	60
3.3.5	State Space Plant Model	61
3.4	Plant Simulation	63
3.4.1	Open-loop Response of Non-linear Plant Model	63
3.4.2	Open-loop Response of Non-linear Plant incorporating Viscous Friction	65
3.5	Plant Tuning and Validation	67
3.6	System Control	73
Chapter 4: Design Details		79
4.1	Solution Statement	79
4.2	Solution Overview	80
4.3	System Details	82
4.3.1	Power Module	82

4.3.2	Actuator Module	83
4.3.2.1	Motor and Motor Driver	84
4.3.2.2	Locomotion Mechanism	85
4.3.3	Sensing Module	87
4.3.3.1	Conventional Encoder Feedback	88
4.3.3.2	Vision Feedback	89
4.3.4	Microcontrollers	90
4.3.5	Calibration Module	91
4.3.6	User-End Module	92
4.4	Hardware Specifications	95
4.5	Addendum	96
Chapter 5: Prototyping and Testing	97
5.1	The Constructed Prototype	97
5.2	Control Using Encoders	98
5.2.1	System Response When Being Controlled	98
5.2.1.1	Control Successfully Achieved	98
5.2.1.2	Control Unsuccessful	99
5.2.2	Open Loop System Response	101
5.3	Challenges	103
5.3.1	Encoder Error	103
Chapter 6: Vision	107
6.1	Introduction	107
6.2	Methodology and Algorithm	108

6.2.1	Detection	108
6.2.2	Tracking and Localization	109
6.3	Perspective Camera	112
6.3.1	Perspective Projection Model	112
6.3.2	Lens Distortion	113
6.4	Results and Performance	116
Chapter 7: Conclusion and Future Work	121
7.1	Conclusion	121
7.2	Way Forward	122
Appendix A: Commercially Available Setups	125
Appendix B: 3D Models	126
Appendix C: Test-Bench Design	130
Appendix D: Schematics	133
Appendix E: System Modelling and Simulations	135
E.1	MATLAB Code for finding mechanical tau	135
E.2	Non-linear Plant Model Derivation	136
E.3	Non-linear Plant Model incorporating Viscous Friction Derivation	137
E.4	Linearized Plant Model incorporating Viscous Friction Derivation	139
E.5	Plant Model Validation Dataset 2	141
Appendix F: Codes Used for Programming System Microcontrollers	145
F.1	Code Used for Controller Arduino	145

F.2	Code Used for DAQ Arduino	150
Appendix G: Comparison of Encoder and Visual Feedback		155
G.1	Cart is placed to the Left of the Camera	155
G.2	Cart is placed to the Right of the Camera	159
Appendix H: Bill of Materials		163
Appendix I: Gantt Chart		164
Appendix J: User Manual		165
J.1	Introduction	165
J.2	System Specifications	165
J.3	Setting Up the Test-Bench	166
J.3.1	Checklist	166
J.3.2	Tensioning the Timing Belt	167
J.3.3	Coupling Cart to Timing Belt	168
J.3.4	Varying the Track Length	169
J.4	Connecting Microcontrollers to your PC	169
J.5	Programming the Microcontrollers	170
J.6	Testing the Control Algorithm	170
J.7	Frequently Asked Questions	170
Appendix K: Videos		174
K.1	Overview of System Hardware	174
K.2	Open Loop System Response	174
K.3	Controlling the System	174

K.4 Visual Feedback from the System	174
References	179
Vita	180
7.5 Sarwan Shah	180
7.6 Abdullah Siddiqui	180
7.7 Adil Ali Khan	180

LIST OF TABLES

2.1 Feature List	23
3.1 DC Motor Specification	44
3.2 Electrical Characteristics	45
3.3 Electrical Characteristics	46
3.4 Motor Model Parameters	49
3.5 Electrical Characteristics	50
3.6 Motor Model Parameters	52
3.7 Steady State Velocity Errors	55
3.8 Fine Tuned Values for Friction Coefficients	67
3.9 Other Simulation Parameters	67
3.10 Mean Error Values for Cart - Model vs. Actual	71
3.11 LQR Controller Parameters	73
3.12 System Response Characteristic Parameters	75
3.13 Required Motor Voltage Characteristic Parameters	76
3.14 Required Input Force Characteristic Parameters	77
3.15 Required Angular Acceleration Characteristic Parameters	78
4.1 Test-bench Hardware Specifications	95
5.1 Gain value for a PID controller	98

5.2 System Characteristics of the Closed Loop System	99
6.1 Camera Settings	108
6.2 Vision System Specifications	110
A.1 Quanser Inverted Pendulum Setup Specifications	125
A.2 GoogloTech Inverted Pendulum Setup Specifications	125
E.1 Mean Error Values for Cart - Model vs. Actual	144
H.1 Bill of Materials	163

LIST OF FIGURES

1.1 An Inverted Pendulum Diagram	1
1.2 Linear Inverted Pendulum	2
1.3 Reaction Wheel Inverted Pendulum	3
1.4 Rotary Inverted Pendulum	3
1.5 Linear Double Inverted Pendulum	3
2.1 U-Channel Track	25
2.2 Dual Pipe Track	26
2.3 Timing Belt Locomotion Mechanism	27
2.4 Bike Chain Locomotion Mechanism	28
2.5 Rack & Pinion Locomotion Mechanism	29
2.6 Incremental Rotary Encoder	33
2.7 37D Metal Gearmotor	37
2.8 Timing Belt Clamp	38
2.9 SICK Incremental Rotary Encoder	38
2.10 Limit Switch	38
2.11 Arduino ATmega 2560	39
2.12 GW-insteck GPS-3303 Power Supply	39
2.13 Pololu Dual VNH5019 Motor Driver Shield for Arduino	40
2.14 Realtime Graphs of System Sensors	40

2.15	Front View of Platform	41
3.1	Angular Velocity response of DC Polulu Motor	48
3.2	Angular Displacement - Model Validation	53
3.3	Angular Velocity - Model Validation	53
3.4	Angular Displacement - Model Validation	54
3.5	Angular Velocity - Model Validation	54
3.6	Comparison between Angular Displacement models	55
3.7	Comparison between Angular Velocity models	56
3.8	Cart-Mounted Inverted Pendulum Free-Body Diagram	57
3.9	Simulink Non-linear Plant Model	63
3.10	Open-loop Non-linear Plant Model Response - $\theta = 45^\circ$	64
3.11	Open-loop Non-linear Plant Model Response - $\theta = 170^\circ$	64
3.12	Open-loop Non-linear Plant Model Response with Viscous Friction - $\theta = 45^\circ$	65
3.13	Open-loop Non-linear Plant Model Response with Viscous Friction - $\theta = 170^\circ$	65
3.14	Model Displacement Tuning & Comparison - $\theta = 50^\circ$	68
3.15	Model Velocity Tuning & Comparison - $\theta = 50^\circ$	68
3.16	Angular Displacement of Pendulum – Model vs. Actual – $\theta = -71.66^\circ$	69
3.17	Angular Velocity of Pendulum – Model vs. Actual – $\theta = -71.66^\circ$. . .	69
3.18	Error in Angular Displacement of Pendulum - Model vs. Actual	70
3.19	Error in Angular Velocity of Pendulum - Model vs. Actual	70
3.20	Cart Displacement - Model vs. Actual	71
3.21	Cart Velocity - Model vs. Actual	71
3.22	Simulink System Model	74

3.23	Controlled Response - $\theta = 5^\circ$	74
3.24	Angular Response - $\theta = 5^\circ$	75
3.25	Motor Voltage Required - $\theta = 5^\circ$	76
3.26	Input Force Required - $\theta = 5^\circ$	77
3.27	Angular Acceleration Required - $\theta = 5^\circ$	77
4.1	System Block Diagram	81
4.2	Block Diagram representing the Power Module	83
4.3	Block Diagram representing the Actuator Module	84
4.4	DC Gearmotor coupled to the Timing Belt using a Pulley	85
4.5	Free Pulley coupled to the Timing Belt and also to a Rotary Encoder	86
4.6	A Linear Bearing coupled to the Cart via a U-Clamp	87
4.7	Block Diagram representing the Sensing Module	88
4.8	The Pendulum coupled to the Cart via a Rotary Encoder	88
4.9	The ZED 2 camera set up to observe the Test-bench	90
4.10	Block Diagram showing the relationship between the DAQ Module and the Controller	91
4.11	A Limit Switch coupled to the Pipe with a Hose Clamp	92
4.12	Real-time Graphs of the System Sensors and Animation of the System	93
4.13	Real-time visual feedback obtained by ZED 2 camera.	94
5.1	The Test-Bench Prototype	97
5.2	Side View of the Test-Bench	97
5.3	System Response when successfully controlled	99
5.4	System Response when being controlled	100
5.5	System Response when being controlled (Close-Up)	100

5.6	System Response when being controlled (Further Close-Up)	100
5.7	Open Loop System Response	102
5.8	Open Loop System Response (close-up)	102
5.9	Pulse Output by the Encoder	104
5.10	Different ways of encoding the Encoder Signal	105
6.1	Blue mask applied on image retrieved from camera.	109
6.2	Real-time visual feedback obtained by ZED 2 camera.	110
6.3	Central projection model showing image plane and discrete pixels . . .	112
6.4	Comparison between encoder and visual feedback - with lag	117
6.5	Comparison between encoder and visual feedback - compensated for Lag	117
6.6	Horizontal error bars for the angular response of the vision system . . .	118
6.7	Histogram of error values - compensated for Lag	118
6.8	Plot of error versus time - compensated Lag	119
6.9	Plot of error versus time - with Lag	119
6.10	Plot of error versus latency	120
B.1	Labelled System Model	126
B.2	Front View	126
B.3	Top View	127
B.4	Bottom View	127
B.5	Backside View	127
B.6	Left Side View	128
B.7	Right Side View	128
B.8	Skewed View	129

C.1	Cart Top View	130
C.2	Cart Closeup	130
C.3	Platform Circuit	131
C.4	Left Platform View	131
C.5	Far End Base with Free Pulley and Encoder	132
C.6	Motor and Circuit Closeup	132
D.1	Circuit Schematic	133
D.2	PCB Schematic	134
E.1	Angular Displacement of Pendulum – Model vs. Actual – $\theta = -141.25^\circ$	141
E.2	Angular Velocity of Pendulum – Model vs. Actual – $\theta = -141.25^\circ$	141
E.3	Error in Angular Displacement of Pendulum - Model vs. Actual	142
E.4	Error in Angular Velocity of Pendulum - Model vs. Actual	142
E.5	Distribution of Errors in Angular Displacement of Pendulum	142
E.6	Distribution of Errors in Angular Velocity of Pendulum	143
E.7	Cart Displacement - Model vs. Actual	143
E.8	Cart Velocity - Model vs. Actual	143
G.1	Comparison between encoder and visual feedback - with lag	155
G.2	Horizontal error bars for the discrepancy between encoder and visual feedback	156
G.3	Vertical error bars for the discrepancy between encoder and visual feed-back	156
G.4	Histogram of error values - compensated for Lag	157
G.5	Histogram of error values - with Lag	157
G.6	Plot of error versus time - compensated Lag	157

G.7	Plot of error versus time - with Lag	158
G.8	Comparison between encoder and visual feedback - compensated for Lag	158
G.9	Plot of error versus latency	158
G.10	Comparison between encoder and visual feedback - with lag	159
G.11	Horizontal error bars for the discrepancy between encoder and visual feedback	159
G.12	Vertical error bars for the discrepancy between encoder and visual feedback	160
G.13	Histogram of error values - compensated for Lag	160
G.14	Histogram of error values - with Lag	161
G.15	Plot of error versus time - compensated Lag	161
G.16	Plot of error versus time - with Lag	161
G.17	Comparison between encoder and visual feedback - compensated for Lag	162
G.18	Plot of error versus latency	162
I.1	Gantt Chart	164
J.1	An Inverted Pendulum Diagram	165
J.2	Close-Up of the C-Clamp coupling the R.H.S Block to Desk	167
J.3	Close-Up of the backside of the Cart	168
J.4	Close-Up of the Limit Switch and Hose Clamp	169
J.5	Close-Up of the Linear Bearing and U-Clamp located under the Cart	171
J.6	Close-Up of the Motor connected to a Pulley and Timing Belt	173
J.7	Close-Up of the Pendulum coupled to Encoder	173

EXECUTIVE SUMMARY

This thesis report discusses the construction of a cart-mounted inverted pendulum test bench for use in the control lab at Habib University. The goal of the project was to design a reliable, robust, and cost-effective platform that could be used by students to explore diverse aspects of control system design, as well as to investigate the challenges in vision-based real-time control of inverted pendulum systems.

The test-bench was designed to be modular, scalable, and open-source to allow flexibility in interacting with it, and modifying different components. The hardware was completely designed and constructed in-house, which means that it can easily be replicated if needed. The complete system was modelled and simulated via MATLAB to predict system behaviour with varying input parameters.

The test-bench operation was tested by attempting to stabilize the pendulum about its inverted state. Robust control could not be achieved within the given time-frame, primarily due to the limitations of the DC gear motor available in the lab. However, significant progress was achieved in the attempts to control the system, and the authors believe that a better motor (with a higher torque) would readily allow the system to achieve stabilization.

The primary novelty of this project lies in exploring vision to generate feedback from the system. An algorithm was designed to track the cart center and the pendulum bob to calculate angular and linear displacements using the ZED2 Stereo Camera. The results obtained were promising in terms of accuracy once the time lag is compensated. However, further work is required to minimize latency in the image processing algorithms by either optimizing the algorithms or utilizing higher-end CPUs/GPUs before the system is capable enough to attempt control based on visual feedback alone.

KEYWORDS: Inverted Pendulum, Control, Feedback, Vision

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Introduction

The inverted pendulum is a classical control problem which involves indirectly actuating a pendulum – which is essentially any rod that is free to move about a fixed point – in such a manner that the actuation leads to the pendulum attaining an upright inverted position with its center of mass directly above its pivot. This position is in contrast to its default orientation. Ideally, the pendulum is actively actuated or controlled to maintain this position despite the disturbances it may face. Figure 1.1 shows the setup of a cart-based inverted pendulum setup with a pendulum that is in the process of being actuated to its inverted state. The goal is to maintain $\theta = 180^\circ$ by controlling the force, F .

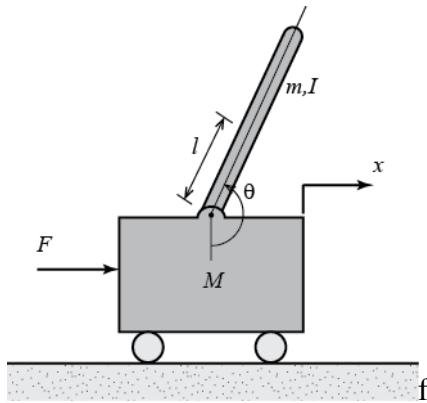


Figure 1.1: An Inverted Pendulum Diagram [1]

The classical control problem has been a popular benchmark for the last 50 years for testing control algorithms. It has been used in control education and research, with over 300 references in literature [2], reflecting its wide resource base and rich utility. This is attributed to its simple structure, implementation, and application of its dynamics in a variety of domains. Several physical laws such as the law of conservation, collision laws, and the value of acceleration due to gravity were established using the dynamics

of a pendulum by Newton and Hooke during the 17th century. In more recent times, it has been utilized to model and simulate control of several robotic systems such as cranes, rocket motion, and legged-locomotion [2].

Furthermore, the classical control problem can also be realized in many different variants of varying complexity, making it challenging and attractive. These variants are based on the methods of actuation that can be utilized, such as a linear motion for the cart-mounted (See Figure 1.2) and wheeled inverted pendulum or rotational motion with the rotary (See Figure 1.4) or reaction wheel (See Figure 1.3) variants of the inverted pendulum. The reaction wheel inverted pendulums can also be further classified into single and double reaction wheel configurations. These have found applications in modelling spacecraft thruster systems for attitude control applications [3].



Figure 1.2: Linear Inverted Pendulum [4]

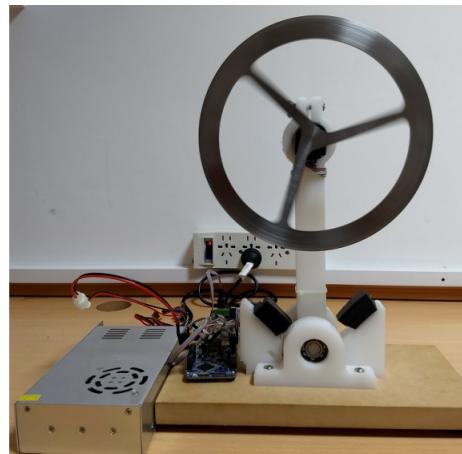


Figure 1.3: Reaction Wheel Inverted Pendulum [5]



Figure 1.4: Rotary Inverted Pendulum [6]

Additional distinction can be made in terms of degrees of freedom of the system, for instance by adding additional pendulum links to achieve the double, triple or quadruple inverted pendulum configurations. A linear double inverted pendulum is shown in Figure 1.5.



Figure 1.5: Linear Double Inverted Pendulum

One of the constraints that make the inverted pendulum problem a challenge is that

it is an under-actuated system where the number of actuators is less than the degrees of freedom. This is made difficult by the non-linear behavior of the system since the placement of the actuators categorize it as an under-actuated system. Furthermore, additional constraints exist due to the sensors that provide feedback, which are limited in terms of their sampling rate, resolution, and sensitivity [7].

The problem involved designing a cost-effective cart-mounted inverted pendulum. The intention is to utilize the test-bench in the Control Lab at Habib University, with the purpose of enriching the understanding of students by providing them with a first-hand experience of engaging with this benchmark control problem.

The proposed solution consists of a hardware setup and a software interface on LabVIEW/MATLAB to help students engage with the system. The hardware setup consists of a pendulum and a rotary encoder mounted on a cart, which is attached to a track and is coupled to a DC motor and another rotary encoder via a belt for actuation. The software consists of a GUI that will allow the user to interact with the system parameters and view the system response in real-time.

Beyond this, the possibility of using Computer Vision replacing traditional sensors such as a rotary encoder is also explored within this project.

1.2 Motivation

The motivation for pursuing this project lies in the authors' interest in control theory and computer vision, both of which were bolstered through their respective course projects. Through the Principles of Feedback Control course, they were introduced to the inverted pendulum problem on a theoretical level. The authors believe that having the students engage with such an interesting theoretical control problem on a hardware level early in the course through demonstrations or labs would help kindle their curiosity towards the course content and strengthen their understanding of it. Additionally, the availability of such a platform could also motivate students to pursue research in the rich domain of control theory.

Furthermore, the awareness that the boom in imaging technology and increase in computing power over the past decade has enabled the integration of Computer Vision's utility in a variety of disciplines - ranging from medical engineering and autonomous vehicle systems to vision-based control in robotics - inspired and motivated the students to explore its potential in the domain of control theory, and more specifically towards the inverted pendulum problem.

It was also identified that as a relatively new institution, it is important for Habib University to be able to offer its students such research avenues. However, acquiring commercially available solutions for each research tangent would add a significant financial strain on the institution, as is discussed ahead, and this was avoided through this project which can be adapted to different research purposes, hence, serving the mutual interest.

1.3 Literature Review

This section provides an overview of the existing commercial and non-commercial solutions for cart-mounted inverted pendulum systems and related research. This helps establish the novelty and motivation for pursuing the presented solution, as well as justifies some design choices made in the solution presented in Chapter 4.

The literature on cart-mounted inverted pendulum systems can be broadly divided into the following categories:

- Commercially designed systems
- Systems designed for research & education
- Modelling approaches
- Control approaches
- Sensory feedback mechanisms

1.3.1 Commercially Designed Systems

Two industry-leading solution providers for inverted pendulum test-benches are Quanser and Leybolds. They provide a range of inverted pendulum configurations including the cart-mounted inverted pendulums, and their solutions are specifically tailored for education and research purpose. Both systems provide excellent build quality, ensuring the longevity of their systems. Quanser inverted pendulum setups can be seen in Figures 1.2, 1.4, and 1.5.

The Quanser system depicted in Figure 1.2 has the actuation and sensing mechanism mounted directly onto the cart, including a lightweight motor, gearing mechanisms, and quadrature encoders. Some of the system specifications for Quanser's system are specified in Appendix A, Figure A.1. Their design in terms of dimensions and weights offers

a very compact and minimal setup, allowing for a hassle-free user experience. In addition, the system comes with a Data Acquisition Board with two possible configurations: USB or PCI. These allow for the easy extraction of sensor data from the hardware setup and its integration with specifically developed software environments on both: MATLAB and NI LabVIEW, which allow the user to interact with the system.

From a control perspective, the Quanser system also offers a reasonable degree of modularity in terms of increasing the control problem's complexity. It comes with a variety of add-ons, such as the ability to interchange between pendulums of different lengths, connecting two pendulums, or adding a see-sawing mechanism to the track base. The latter two help in increasing the control problem complexity by introducing more degrees of freedom to the system. Despite these liberties, the system is not completely modular and is limited by the lack of control over factors such as track length, motor type, and specification, or sensor choice, which can also greatly influence the control problem at hand. [8]

Leybold's solution utilizes a timing belt coupled with gears to move the cart in contrast to Quanser's platform. It does not offer much open source information of its system specifications or the available modularity options. However, there is strong reason to believe it might be similarly limited across factors of modularity and access, given that it is also designed as a commercial product and is only available online in one particular configuration. Nonetheless, provides a Data Acquisition Board for its system's integration with a dedicated MATLAB environment. [9]

Both solutions provide a developed mathematical model of their systems that can be utilized for control, or for simulation purposes. These are a part of the rich educational courseware provided by Quanser and Leybold's, which consists of detailed guidelines to use the equipment and to carry out structured experiments. These can be utilized by educators to engage their students with various aspects of this intriguing control prob-

lem.

The downsides associated with these commercial solutions is their associated expensive price tag: 6000 and 15000 dollars for the Quanser and Leybold systems respectively. Modularity appears to be an attractive feature with the Quanser systems, however, each add-on incurs a significant additional cost. Another disadvantage is that their systems do not provide open access to the system parameters i.e. the parameters can't be independently varied by a user. This not only limits the variability that can be achieved on the control problem, as stated above but also limits the research potential of such a test-bench by locking access to many integral system components and disallowing the addition or replacement of these components. There are other localized solution providers across the globe that design platforms upon request, but the drawback in these cases is the significantly added cost. The details of one such provider, Googoltech, can be found in Appendix A, Table A.2.

In contrast, the solution presented in this report provides a competitive cart-mounted inverted pendulum test-bench which costs less than 200 dollars. Furthermore, it is designed as an open-system, allowing the user to interact with low-level system parameters such as track length, cart mass, motor type & specification, or feedback sensor choice, making room for a broader control experience.

1.3.2 Systems Designed for Research & Education

Beyond these commercial solution providers, plethora of solutions presented in various research papers and theses are aimed at designing systems for research and education purposes. Some of these solutions are summarized below:

A master's degree project from Cornell University focused on making a single cart mount inverted pendulum hardware setup and software at minimal cost for research purposes [7]. The goal was to implement LQR control over two degrees of freedom:

the cart displacement and the pendulum. The system used a dc motor and bike chain for actuation and a low-cost Atmega32 microcontroller for digital control. The thesis also involved a detailed discussion of the constraints, requirements, and design choices that went into designing the mechanical system, as well as the various options of control approaches such as classical, modern robust, and model predictive control. The processing ability of the Atmega32 limited computation that could be performed and the motor rated speed and torque values served as limits for the actuation mechanism. This limited the controllability to only a short period of time about the linearized position, before largely deviating from the desired set point.

A paper titled “*Design and Control of an Inverted Pendulum System for Intelligent Mechatronics System Control Education*” presents a cart-mounted inverted pendulum system specifically designed for fitting an educational kit [10]. The system was designed to be cost-effective, portable, and compact. System parameters such as track length were very limited, and cannot be scaled. The system used a dc gear motor with an encoder along with a belted mechanism for the actuation of the cart. A DSP 2812 MCU board was utilized as the control unit, while an FPGA was utilized as the data acquisition unit (DAQ). The paper implemented a unique neuro-fuzzy controller logic, which involved utilizing a neural network along with fuzzy logic to train a controller for the non-linear dynamic system.

Another paper presented a unique design, involving a wheeled autonomous cart-mounted inverted pendulum system [11]. It identified its interest in limiting the user’s access to system parameters to increase student engagement and learning at lower levels. The complete system was mounted on a wheeled cart, consisting of a motor for actuation, an encoder for feedback, and the processing unit. The system is designed to appear as a closed-system for students, such that they did not have to worry about factors such as programming or interfacing of the various components, and can directly relate the system’s behavior to the control concept studied in class. To simulate this

experience, the system used a battery and an on-cart myRIO board for data acquisition and interfacing, which was also connected to the PC over Wi-Fi, with the PC acting as the main controller unit.

Another paper [12] presented the design of a cart-mounted inverted pendulum system as an assembly kit, targeted for educational intuitions. The work developed and provided CAD models of components that could be printed/laser cut on a fiberboard, and then assembled to form the setup for an inverted pendulum system. The only additional components that were required to be added were a motor, an encoder, and a microcontroller interfacing unit. The system utilized a stepper motor for actuation via a belt mechanism and an Atmega2560 for interfacing and control. In addition, the work also involved designing MATLAB firmware that allows for communication with and control of the system. While portability and ease of assembly and implementation are advantages, the system lacks considerably in-terms of scalability.

A low-cost solution using a dot-matrix printer assembly for designing a cart-mounted inverted pendulum system was presented in [13]. The base was 3D-printed. It used a dc encoder motor driven by a linear current-control servo amplifier and an incremental encoder for angular feedback. However, the system lacked a data acquisition interface and utilized a National Instruments (NI) Data Acquisition Board (DAQ) for interfacing the system with the PC environment of MATLAB/Simulink.

All of these papers have detailed reasonably justified working solutions, offered valuable insights on how the system may be designed, and outlined considerations to be made for such a design. However, most have still lacked the fundamental element of scalability and an open-system approach for maximizing user-interaction with low-level system parameters. These factors were identified to be crucial for a richer control experience in a lab setting, and hence, are novel components of our system design.

1.3.3 Modelling Approaches

Mathematically modeling the inverted pendulum was identified as an important component of most of the designed systems as it provided the user with information about the system's dynamics and behavior. This, in turn, is critical for control experimentation and performing research on the system. The prevalent modeling approaches in literature are: Newton-Euler method and Euler-Lagrange method. The former focuses on establishing the dynamics through force balance or Newtonian mechanics, while the latter focuses on achieving the same results through differences of kinetic and potential energies of the system between various states. The Euler-Lagrange approach is preferred in literature as per [2] for its ease of modelling more complicated dynamic systems.

While most papers stress on achieving accurate models for easier controller design, this paper [7] pointed out the importance of not overestimating system complexity as that would lead to similar drawbacks, especially with parameters such as friction, which were better traded-off under the assumption of being negligible. The paper utilized the Newton-Euler approach for modeling the system dynamics, then considered the state-space approach for its representation and manipulation. It suggested that unmodelled dynamics such as friction could also be compensated by inheriting them as a part of some system parameters.

Similarly, [10], [14], and [11] also used the Newton-Euler approach, modelling the dynamics of their respective systems. All exhaustive explanations of the system model included linearisation as an important practice to reduce the complexity of the model, so that it may operate as a linear system in a limited range [15].

There were a few different attempts to model the motor apart from the standard approach. For example [7] utilized a brute force approach, obtaining the voltage-velocity relations, that were then used to identify lumped parameters of a motor's dynamic model. The approach in [16] provides guidelines for obtaining a motor model using

both its dynamics and its electrical characteristics. This was utilized in [14]. It was also found to be an approach consistent with the industry, which provided similar model parameters and relations in their data-sheets [17].

1.3.4 Control Approaches

Several control approaches were identified in literature including the PID Control, Robust Control, and Model Predictive Control.

It was identified that classical control utilized techniques and tools such as Bode Plots, PID, Root Locus, and Nyquist Plots for system analysis and control [16]. Such a controller was operated in the frequency domain. However, classical control was found by [16] to be only well-suited to SISO (Single Input Single Output) systems, and scalability to MIMO (Multiple-Input and Multiple-Out) systems was not found to be an optimal choice. Furthermore, while PID control has been explored for the single inverted pendulum configurations [18] [19], no literature was found that utilized it in isolation for the case of higher-order configurations of the double or triple inverted pendulum.

A few papers dealt with MIMO systems and represented the system's behavior in the time-domain [7], and hence utilized the state-space representation of system dynamics. It focused on minimizing a cost-equation and using state observers for achieving control. The Linear Quadratic Regulator (LQR) was often utilized in literature [20][11][21], but was identified to be insufficient for controlling the inverted pendulum in a robust manner, where robustness was defined as the controller's ability to be stable for a large random sample of system models with parameters that fall within a certain uncertainty range. [7].

For more robust control, the LQR cost equation was coupled with a state-observer in the form of a Kalman Filter estimator as well as a deterministic observer. This was

known as the Linear Quadratic Gaussian (LQG) controller [7] and allowed for more robust control. The downsides to this method were found to be the added computational complexity.

Similar to modern control techniques, model predictive control was identified to be good for tracking behavior relying on the discrete-time system representation. This technique worked on the principle of identifying the next N-states for determining its control effort, but only implementing it for the immediate next step [22].

1.3.5 Sensory Feedback Mechanism

Feedback sensors play a critical part in any control system, and this remains true for cart-mounted inverted pendulum systems as-well. They provide displacement and angular feedback for such systems. In this section, some of the popular choices of feedback sensors in literature and each of their limitations are discussed.

1.3.5.1 Potentiometers

A master's thesis from the Naval Postgraduate School, Monterey, California, 1981, utilized the potentiometer sensor in its cart-mounted inverted pendulum system [23]. The system utilized a potentiometer for both, the measurement of feedback for cart-displacement and the measurement of feedback for angular-displacement of the pendulum. The thesis also worked on deriving transfer functions for these potentiometers and extensively calibrating them to their specific utility. The system was concluded to not work as well as expected. This was attributed to the possibility of friction and drag components due to the potentiometers, which were not modeled for. Furthermore, the thesis also suggested the use of higher precision potentiometers, highlighting the limitations of potentiometers.

Another research paper [24] from 2012 explored the use of a potentiometer feedback mechanism and PD controller for a cart-mounted inverted pendulum. The paper

discussed its mathematical modeling and controller design in great detail. The system was able to achieve the required control goals of stabilization and swing-up but was not considered robust by the authors. The friction and accuracy of a potentiometer was identified as a limitation, and it was suggested to use more accurate and robust sensors such as encoders or gyroscopes.

A bachelor thesis from the Royal Institute of Technology in Stockholm, Sweden, 2020, observed a similar stance on its use of potentiometers for feedback [25]. The thesis explored the relationship between the minimum required bandwidth for stability, the pendulum center of mass, and the feedback sampling frequency. The results were concluded to be limited due to potentiometer constraints such as resolution, noise, lack of precision.

1.3.5.2 Rotary Encoders

In recent years, the most popular choice of a sensor in literature for an inverted pendulum system was found to be rotary encoders in [10], [11], [21], [26], and [14]. This is because of their high accuracy, precision, availability in a wide variety of specifications, and reasonable price range. Within the domain of encoders, there exists the choice of utilizing incremental or absolute encoders. Incremental rotary encoders have found more popular use due to their cheaper cost, and simpler usability for most research purposes.

This paper [26] from 2015 on the ‘Control of Double Inverted Pendulum First Approach’ utilized a rotary encoder for measuring the angle of only the first pendulum. The paper found the encoder to be susceptible to noise due to high-powered electronics in proximity but was otherwise satisfied with its working.

Similarly, another 2019 research paper [27] from DHA Suffa University, Karachi also utilized an incremental rotary encoder in their Control Design and Implementation

of an inverted pendulum and found its use to be satisfactory.

Despite their well-documented use, this paper [28] identifies challenges with using incremental quadrature rotary encoders concerning the loss of signals due to noise or interference. This leads to a drift in the reading. Furthermore, it also highlights important considerations that need to be made while programming for the acquisition of data from encoders and the need to use more robust measurement approaches, such as utilizing interrupts for both signal edges: rising and falling.

1.3.5.3 Computer Vision

Another avenue in sensing growing in popularity is the use of computer vision as a feedback mechanism on an inverted pendulum system. This is a direction that this project is also adopting, and to this end, literature relating to inverted pendulums and the use of vision was extensively reviewed to help establish a sense of direction and novelty. The problems identified with most of these systems were limitations in terms of sampling rate, resolution, and latency when compared with traditional encoder systems, which offers sampling rates up to hundreds of kHz, very high resolution and virtually zero latency in contrast [29].

A paper [15] from 2000 explored the potential of using computer-vision as a visual encoder for an inverted pendulum system. The setup utilized a camera with a sampling rate of 85 Hz at a resolution of 640×480 pixels for image acquisition, a specialized IMAQ PCI 1424 digital frame-grabber card for direct image transfer to PC, a pattern matching algorithm for determining angular position, and an LQR based controller. All these elements were implemented, intensively processed, and interfaced via the LabVIEW environment on a PC. The paper pointed out that typical systems using rotary encoders having sampling rates averaging at 100 Hz, and identified that using analog camera-based systems only a sampling rate of 30 Hz was achievable, which lead to very 'sluggish' control, pointing out a limitation of such systems. The paper suggested the

use of non-interlaced digital cameras for higher sampling rates and spatial resolution. It also pointed out that a high sampling rate translated to a higher data transmission rate, which was only achievable with PCI systems. Another limitation is also identified in the form of limited resolution, inaccurate calibration of the algorithm, time delays during image acquisition, and drifting light conditions, which did not allow the system to accurately estimate parameters such as velocity. As such, the system utilized an observer and a rotary encoder for swing-up and stabilization and is subsequently switched to vision on demand.

Another paper [30] also attempted to use computer vision as an angular feedback mechanism in a hybrid setting, where the continuous position is determined by a rotary encoder, and the vision-based system only provides feedback at delayed and sampled intervals. The motivation was supervision required in remote control systems, which offered more flexible contact-less systems with an improved SNR. The vision system had a sampling rate of 25 Hz at a resolution of 640×480 . The control achieved is limited to only small cart displacements. The paper also pointed out that no significant success has been achieved in implementing control using only vision-based feedback systems when the paper was published in 2008. The key limitation identified to this was sampling rates and delays in the acquisition of data from images (i.e. latency).

Written in 2014, this paper on real-time embedded computer vision systems also identified the latency and computational power as a limitation for utilizing computer vision as a feedback mechanism on low-powered mobile systems [31]. It used an amalgamation of FPGAs (due to their parallel processing ability) and mobile CPUs, along with estimation algorithm to tackle these issues for mobile robotic systems. It used a stereo camera with a sampling rate of 60 Hz, and a resolution of 752×480 pixels. Their system was able to gather feedback information of a dense 3D environment with a delay of only 2 milliseconds. However, no further processing or extraction of data was done from the image.

This master's thesis on low latency vision-based control for robotics [32] also identified the challenges around vision-based robotic systems. It focused on minimizing the latency of an FGPA coupled with a dual-core ARM processor for the purpose of a high-speed tracking and interception system. This system involved the control of a robotic Foosball game system. The results showed an 100% interception rate with the highly optimized vision-based feedback system, with latencies as low as 25 ms at a frame-rate of 60 Hz for a resolution of 640×480 . The thesis also extensively reviewed and discussed the options that were available for designing a vision-based system in terms of cameras, FPGAs, and image processing techniques, reviewing and contrasting their pros and cons.

Similarly, a bachelor's thesis from 2015 on the Control of Double Inverted Pendulum First Approach' [21] also utilized a vision-based system for the control of a double pendulum by using it as a feedback mechanism for the angle of the second pendulum only. The camera utilized offered a sampling rate of 30 Hz and a resolution of 640×480 . The latency identified to process an image section of 458×458 with an angular resolution of 0.25 degrees as estimated to be 315 ms, reflecting the high latency of the system.

The findings from a research conducted at Habib University, Karachi, in 2020 were also explored [33]. The work focused on utilizing computer vision to estimate the angular displacements velocities of a double inverted pendulum. It also identified the limitations faced by rotary encoders in terms of noise and calibration issues. The paper utilized Canny edge detection in conjunction with Hough transform to estimate the required parameters and focused on latency-accuracy-memory trade-off attempting to find the least computationally expensive approach.

1.3.5.4 Discussion

As established through this literature review, traditional sensors used in inverted pendulum systems such as rotary encoders are often sufficient for feedback, but their downside often exists in the form of hardware integration, calibration issues, sensor noise, and remote control. Furthermore, these sensors are often bulky and can interfere with the system dynamics in cases where additional links (pendulums) need to be added. On the other hand, a potentiometer suffers from an inherent lack of accuracy and precision, proving to be an undesirable choice for sensing.

The literature showed that it is desirable to explore the use of computer vision as a feedback mechanism as it has the potential of overcoming some of the issues faced by rotary encoders while offering reasonable performance. However, the introduction of computer vision and the associated image processing brings in issues concerning sampling rate, resolution, and latency. Modern cameras and technology have enabled us to minimize most of these. However, latency remains a concern for vision-based systems.

Furthermore, no successful attempts were identified in using an independent vision-based feedback mechanism for control of an inverted pendulum.

1.4 Problem Statement/Objective

To develop a reliable, robust, and cost-effective cart-mounted inverted pendulum test-bench for the Control lab at Habib University with the aim of maximizing students' opportunities to explore diverse aspects of control system design, as well as to explore the challenges in vision-based real-time control of inverted pendulum systems.

CHAPTER 2

DESIGN PROCESS

2.1 Stakeholders and their Requirements

The stakeholders of this test-bench platform are primarily the researchers in the domain of control theory, the students using the test-bench, the instructors utilizing the platform to teach new concepts, and the institution i.e. Habib University where the test-bench is being used and with which all the aforementioned parties are affiliated. Each stakeholder has certain requirements that define the constraints for the overall product.

Note that there is considerable overlap between the requirements by researchers, students, and instructors since students and instructors comprise the researchers' category.

2.1.1 Researchers

The system requirements for different research projects involving inverted pendulums would invariably differ. Therefore, the system must be modular and open-source so that different parameters may be varied and components detached/replaced as needed by a prospective researcher. These researchers will most likely be students who have taken the basic 'Feedback and Control Systems' course and are pursuing their interests in the domain.

Variable parameters are ones such as length of the track, length of the pendulum, control effort by motor, etc. Interchangeable components are ones such as motor, encoder, pendulum link, etc. depending upon the requirements. The test-bench should also allow for expansion into more complex inverted pendulum configurations, particularly the double and the triple configurations as the research domain for these problems

is considerably wider.

There should be a minimal effort that goes into operating the setup beyond writing the control algorithm as that will be the focus of researchers attempting to optimize control techniques. As such, it is required that the interface be user friendly and system performance be reliable.

2.1.2 Students

It is anticipated that the platform can be used by many different categories of students.

Students taking the 'Feedback and Control Systems' course could have the setup as a part of their lab component to see actual applications of the theory that they have studied. These students would require that the system model and control algorithm be simple to understand and implement. They will also only be interested in the basic functionality of the platform. Repeatability is an important concern for these stakeholders so that lab experiments can be easily reproduced. The system operation should remain the same under the same parameter values for a reasonably long time. The system needs to have minimal wear and tear that may directly affect its operation. This need directly stems from the fact that lab manuals will be designed that should be easily read and implemented by the students on the system. This would not be possible if system parameters changed from the ones initially calculated and outlined in the manuals. A good GUI would also increase ease of use for the students allowing them to focus on the control aspects of the platform.

However, the system will also be used by students taking advanced control courses and pursuing research tangents. These students would be interested in a more in-depth look at the system parameters and working, and at the details of individual components. They may also wish to configure or replace components as per their requirements. There needs to be some sort of layered interface catering to both types of students.

In addition, to control students, the system can also be used for demos to students of courses like 'Computer Vision' and 'Digital Image Processing' to showcase how visual feedback can be implemented in real-world cases. More generally, the students would also benefit from the option to engage with the test-bench in terms of understanding how individual components are working. This would not be possible in an industry grade product which acts as a black box. The test-bench should allow a student to be able to understand and implement the operation of individual components in a standalone fashion as well.

2.1.3 Instructors

The instructor would require that the system be durable and system operation repeatable so that a course component can be designed around the usage of the test-bench through the semesters.

For the beginner level control courses, the instructor would require that higher-order details of the platform components and non-linearities be hidden from the students allowing them to focus on the basics of the control algorithm. At the same time, if the system is being used for higher-level courses or research, these non-linearities should be available to the user. There needs to be a smart interface allowing the user to switch between these two modes.

There should be a minimal effort that goes into operating the setup beyond writing the control algorithm as that will be the focus of students applying their course learning outcomes via the platform.

The student manuals, system models, and control techniques should ideally tie in with the theory studied in the control courses so that there is a minimal gap for the instructor to cover when introducing students to the test-bench.

2.1.4 Institution

The institution, in this case, is of course Habib University, or more specifically the Feedback and Control Lab. The primary concern in the context of the institution is cost minimization. The test-bench must be cost-effective when compared to the industry-grade alternatives in the market for example those by Quanser and Leybold.

It is important to note here that not only is the one-time purchase cost orders of magnitudes lower than the competitors, but the cost for replacing any components or expanding the system to more complex configurations is also much cheaper than the market rate.

The table 2.1 summarizes the list of features and defines their relevance to each stakeholder.

Table 2.1: Feature List

Feature	Researchers	Students	Instructors	Institution
Modularity	✓	✓	✓	
Cost Effectiveness	✓			✓
Repeatability	✓	✓	✓	
Durability	✓	✓	✓	✓
User Friendly Interface	✓	✓	✓	
Low Level Control	✓		✓	
Comprehensive Manuals	✓	✓	✓	

2.2 Design Requirements and Constraints

There are some important points that must be factored in when exploring and selecting from the design alternatives. These requirements and constraints are outlined below.

2.2.1 Requirements

- The system designed must be modular i.e. easy to dismantle and put back together.
- The components used should be cheap and readily available in the local markets in case they need to be replaced at some point.
- The test-bench performance should be consistent over time. This means that the mathematical modeling and related parameters should hold valid for a reasonably long time.
- The platform should not be so noisy that it's distracting in a lab environment.
- The platform must be self-contained and portable.
- The cart should be allowed movement in only one direction i.e. one degree of freedom. It should not be wobbly.

2.2.2 Constraints

- The maximum length of the track/setup is constrained to the length of the lab work benches.
- The maximum length of the pendulums is similarly constrained to the height of the lab workbenches.

2.3 Design Concepts and Methodology

The cart-mounted inverted pendulum system can be broken down into a variety of subsystems. There is a certain degree of flexibility in how each subsystem can be implemented. Note that while each subsystem module can be implemented in its own way, there is inter-dependence between modules which helps define the optimal product. The different design choices were obtained through an extensive literature review.

2.3.1 Track, Cart and Locomotion System

2.3.1.1 Track

Two popular ways to design the track are

- U-Channel:

Here the cart is nestled inside a U-Channel track. The cart is supported upwards by the base of the channel, while the sides of the channel ensure that there is no lateral movement. Wheels/Bearings are attached to allow easy movement.

This method often suffers from increased friction. It is also mechanically complex in terms of constructing a precise track over a long distance and integrating multiple bearings for smooth movement.

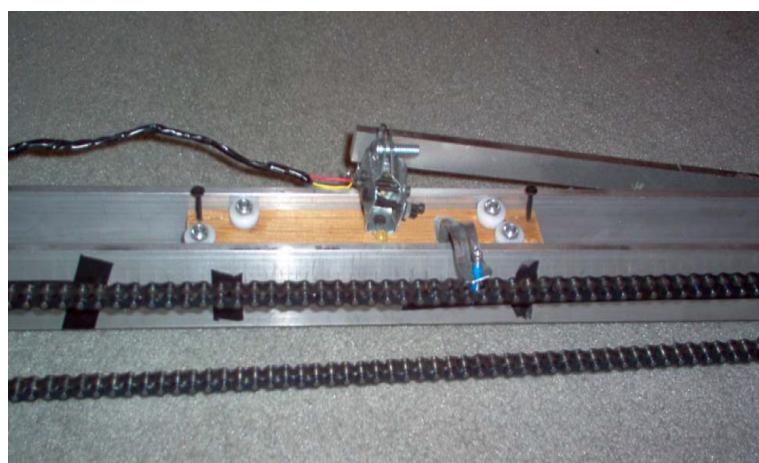


Figure 2.1: U-Channel Track [7]

- Dual Pipe:

Here the cart rests on two (some variants may have one or even three pipes) pipes such that there are bearings attached to the bottom of the cart that are fitted around the pipe forming the track. This has the same affect of preventing lateral movement as the sides of the U-channel provide.

This mechanism suffers fewer friction problems and is mechanically less complex as only two cylindrical bearings will be needed for smooth movement.

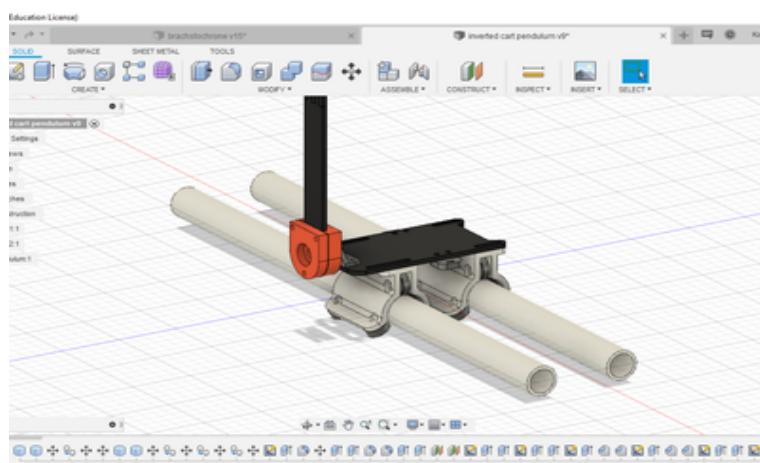


Figure 2.2: Dual Pipe Track [34]

The length of the track could be varied using different methods. Two ideas that were explored for this purpose were

- Drilling through holes in the track support at the system end without the motor. This would mean that the far end support would be screwed to the track, and could be moved along the track to restrict its length. However, the authors realized that this method would be unwieldy to implement and inefficient.
- Using limit switches clamped to either end of the track to define its endpoints. These limit switches are placed in holders which can be moved along the track length to change it. This is an efficient and logically convenient solution comparative to the first one.

2.3.1.2 Cart

The cart merely needs to be large enough to allow placement of the rotary encoder and allow suspension of pendulum links. It may be made of different light-weight materials e.g. Aluminum, Acrylic, etc. depending on availability. It needs to be solidly coupled to the track so that there is no lateral movement when it sides or when the pendulum swings generate inertia.

2.3.1.3 Locomotion Mechanism

There are two main ways in which the cart may be moved using a single motor.

- Timing Belt:

A timing belt is subtended between a motor placed on one end and a free moving pulley at the other end of the track. Both ends of the timing belt are clamped tightly to the cart. The motor can rotate clockwise or anticlockwise to move the cart left and right.



Figure 2.3: Timing Belt Locomotion Mechanism [34]

- Bicycle Chain:

A bicycle chain could be used similarly to the timing belt. This is a cheap solution. However, such a mechanism would suffer from slippage between gear and chain links increasing motor response time. It would also increase the load on the

motor due to the large weight of the chain. The chain would also be bulky and difficult to manage.

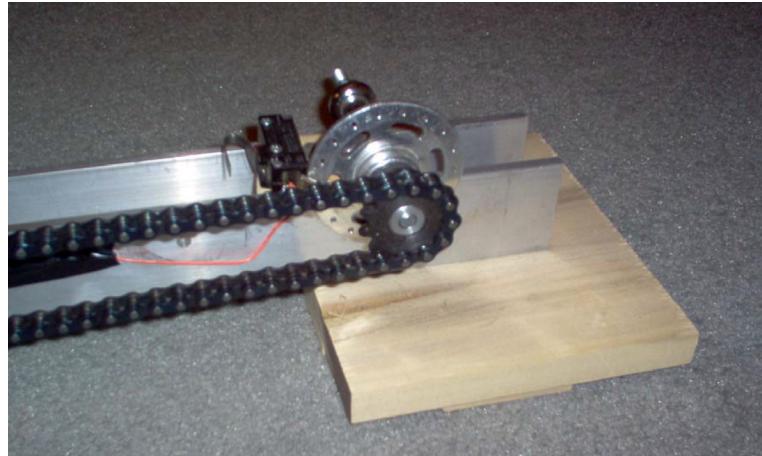


Figure 2.4: Bike Chain Locomotion Mechanism [7]

- Rack and Pinion Mechanism:

As the name suggests, the motor is mounted on the cart. A separate toothed track is made for locomotion purposes. The gear at the end of the motor shaft is attached to the toothed track. Motor rotation rotates the gear on the toothed track and moves the cart as required. The circular gear forms the pinion while the linear gear or track forms the rack. This is the method which is employed in the Quanser inverted pendulum system.

This method suffers from jitters and vibrations due to the motor's presence that increases stress on the cart, and this needs a high degree of refinement and optimization to be resolved. It was also difficult to obtain precisely toothed track of the requisite length. With the motor on the track, there will also be additional wire management issues.

One variation of this method can be that instead of implementing a track, the system is simply built to have a cart with motorized wheels that allow for movement. Some gear mechanism will have to be implemented to drive wheels simultaneously. This method would remove track limitations and make the system com-

pact, but introduce other complications. Cart and Pendulum may have more than 1 and 2 DOF resp. which will make the control problem much more difficult. It will also be harder to measure cart displacement without a reference point.

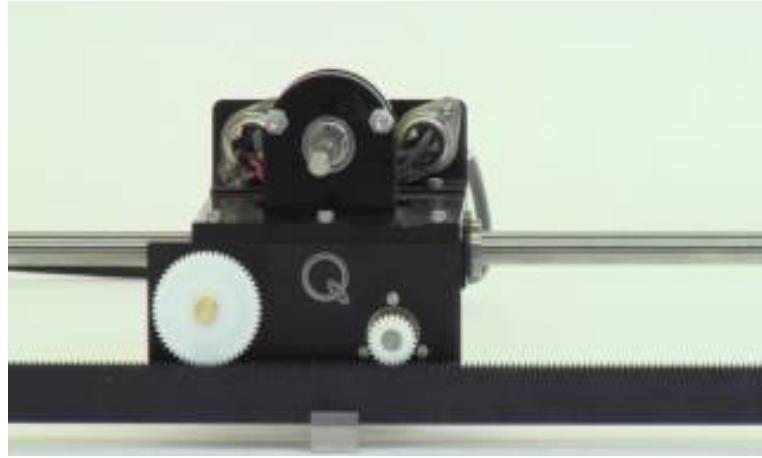


Figure 2.5: Rack & Pinion Locomotion Mechanism [4]

2.3.2 Motor and Power Module

The three types of motors that may be used to control cart movement are the DC, Stepper, and Servo motors. The factors governing the choice of motors are torque and speed. For the purposes of this project, the requirement was a high torque, high-speed motor to achieve pendulum swing up and stabilization. Torque is needed for the cart to change direction quickly. Speed is required so the cart can move faster than the pendulum can fall.

We need to design a power module specially for the motor. This is because, during the control operation, the motor's speed and direction drastically change repeatedly. The large power consumed by the motor and the accompanying voltage spikes may damage any other circuit connected to the same power supply. This necessitates the use of optoisolators to safeguard all circuit components.

2.3.2.1 DC Motors

DC Motors can have high torque and high speed, but these features come with increased cost. The cost is both monetary and in terms of the power required to run the motor. A motor driver circuitry (typically H-Bridge) will need to be attached to the motor to control its speed and direction.

DC Gear Motors are variants of DC Motors which are particularly suited to high torque and high-speed applications.

2.3.2.2 Stepper Motors

Stepper Motors are suitable for high torque applications but not for high-speed ones. For movement in both directions, a bipolar stepper motor would be required. These may be suited to stabilization of an inverted pendulum but are not optimal for swing up functionality. They are also quite expensive.

2.3.2.3 Servo Motors

Servo Motors are the opposite of Stepper Motors in the sense that they are suitable for high speed but not for high torque applications. Another practical limitation with servo motors is that they are usually restricted to 360 degrees rotation. Accommodating this restriction can lead to mechanical complications e.g. if the size of the drive wheel is increased such that it corresponds to the length of the track for one rotation, the torque provided is greatly reduced.

Servo Motors are also not suitable for this application as the control effort applied to a servo motor controls its position or angular displacement, not its speed. This would make the controller design complicated.

2.3.3 Sensor Feedback

We need to collect data about the pendulum's angular displacement and the cart's horizontal displacement. Both of these can be achieved using different sensors.

2.3.3.1 Cart's Displacement

- Potentiometer:

A potentiometer could be attached to the drive or the free pulley. The signal from the potentiometer could then be used to mathematically derive the displacement of the cart. This is a simple solution to implement but suffers a few drawbacks e.g. output voltage may be non-linear, a potentiometer may not be precise enough, the potentiometer transfer function may add complexities, etc.

- Motor Encoder:

This method would involve calibration of the motor encoder using the track limit switches to define a range of values corresponding to the track length and subsequently allow the controller to read the displacement at any given instant.

- Ultrasonic Sensors:

These could be placed at either end of the track and calibrated to give the cart displacement. However, these sensors are not highly precise and also have the problem of giving incorrect reading if anything appears between them and the cart.

- Radar and Sonar Sensors:

These sensors may be highly accurate but are too expensive to merit consideration in this platform application. They also take a large amount of processing time which is detrimental to the control techniques.

- Vision Sensors:

A camera can be mounted in front of the platform to record the cart's displacement. This is a novel method in comparison to the other feedback sensing meth-

ods and is gaining favor with advances in image processing technologies. One complication of this method is that you need to decide camera placement such that the entire set up is viewable. It may also suffer from latency issues if the image processing techniques or camera specifications are non-optimal. On the other hand, it has the added advantage of not suffering from mechanical wear and tear, and that one sensor can be sufficient to record both cart displacement and pendulum's angular displacement if configured properly.

2.3.3.2 Pendulum's Angular Displacement

- Gyroscope:

The gyroscope can be placed at the end of the pendulum. This sensor notes down the values of yaw, pitch, and roll and can calculate angular displacement from the rate of change of coordinates. However, it is difficult to calibrate, can easily go out of sync, and creates the considerable problem of wire management in a rotating pendulum.

- Potentiometer:

The potentiometer could be attached to the shaft to which the pendulum link is attached. The voltage output could give the angular value, however, these sensors are often imprecise because output voltage is often non-linear.

- Incremental Rotary Encoder:

These encoders are highly accurate in terms of measuring the angle and direction of rotation. These can be attached to the shaft connecting the pendulum link to the cart. However, these are costly. They may have a significant weight that impacts the motor's ability to quickly actuate the cart. They may also have a bulky or unwieldy size which means that they can not be implemented in double or triple pendulum configurations efficiently.



Figure 2.6: Incremental Rotary Encoder [35]

- Vision Sensors:

A camera can be mounted in front of the platform to record the pendulum's angular displacement. This is a novel method in comparison to the other feedback sensing methods and is gaining favor with advances in image processing technologies. One complication of this method is that you need to decide camera placement such that the entire track is observable. It may also suffer from latency issues if the image processing technique is non-optimal. On the other hand, it has the added advantage of not suffering from mechanical wear and tear, and that one sensor can be sufficient to record both cart displacement and pendulum's angular displacement if configured properly. Similarly, one camera may be sufficient to observe and control various systems working in close proximity.

2.4 Society, Economic, and Ethical considerations

The inverted pendulum design is a classic control problem. It is often used as a benchmark to test novel control methodologies (e.g. PID control, state-space representation, neural networks, fuzzy control, genetic algorithms, etc.). It can also be used to represent a variety of real-world problems.

The applications of the inverted pendulum model are wide and varied. It can be representative of a class of altitude control problems whose goal is to maintain the desired vertically oriented position at all times. It can be used to model a rocket propulsion system or a missile guidance system. Self-balancing personal transporters such as Segway, self-balancing hoverboards, and self-balancing unicycles also take inspiration from the inverted pendulum model. It is extensively used in robotics e.g. in biped robot upright balance and locomotion [2].

The wide array of applications mean that there is ample research potential in the domain. The system needs to be robust, modular, and durable so that future students and researchers can find novel ways to improve real-world applications that borrow from the inverted pendulum problem. The design and provision of such a cost-effective platform will also have a positive impact on the engineering education in the country since institutions will no longer have to rely on importing industry-grade test-benches in order to facilitate student understanding of real-world concepts.

The fact that this platform has so many practical applications means that there is a huge potential for positively impacting society by using this platform to optimize control techniques for the aforementioned applications. This in turn would serve to make these applications more cost-effective and economical.

Since inverted pendulum modeling has extensive applications in the domain of

robotics, it can also raise some ethical qualms. Even though the platform utilization is merely concerned with improving the hardware capabilities of different robots, an argument could be made that hardware improvements lead to progress in the field of robotics. Partially or fully self-aware robots have a lot of potential for positively (e.g. facilitating humans by taking initiatives, performing menial tasks, etc.) as well as negatively impacting society (e.g. replacing the human workforce, making humans lazy, etc.)

2.5 Environment and Sustainability considerations

The system is modular in the sense that it consists of many different components that are replaceable depending upon the requirement. There are no significant specialized components required for the platform. This is beneficial from a sustainability point of view as it means that the test-bench can (and in fact has) incorporate recycled or used materials. In other words, if a component is replaced it can be used for some other project and does not have to be discarded.

Similarly, another aim of the project is that the final product is power efficient to ensure that there is no energy wastage. This is important since excess energy production almost always has an adverse affect on the environment.

The motors and sensors used in the system are not brand new either. They have been obtained from either lab inventory or recycled industrial products. Apart from the sensors and motors; aluminum pipes, stainless steel pipes, and acrylic sheets have been utilized to construct the hardware. Stainless steel and aluminum are frequently recycled materials. However, acrylic is a plastic that is not readily recyclable or biodegradable. Therefore, care will have to be taken when discarding the acrylic components of the system.

2.6 Technical Specifications Highlights

The system design choices were made after an extensive literature survey of existing design alternatives. The final design choices are outlined in this section. 3D models were designed at the start of the design process. These models can be found in Appendix B, Figures B.1-B.8.

2.6.1 Design Choices

The cart was mounted on a dual-pipe track subtended between two blocks acting as the base of the platform. A 37D metal gearmotor rated at 150 RPM for 12V with encoder was mounted on one base to act as the actuator. The encoder of the gearmotor was not utilized due to low resolution (64PPR) [17]. Figure 2.7 shows this kind of motor.



Figure 2.7: 37D Metal Gearmotor [17]

A locomotion mechanism consisting of a GT2 timing belt was subtended between a pulley mounted on the gearmotor shaft on one end and a free pulley on the other end. The timing belt was coupled to the cart using 3D printed clamps. The clamps are shown in Figure 2.8

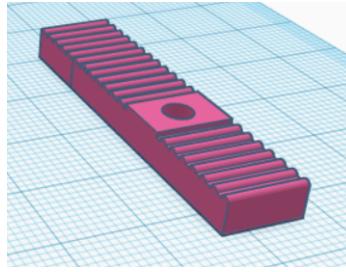


Figure 2.8: Timing Belt Clamp

Two incremental rotary encoders of 2500 PPR resolution (SICK DBS36E-S3EK02500) were used as sensors for this platform [36]. This type of encoder is shown in Figure 2.9. One encoder was attached to the pendulum shaft to measure the angular displacement and the other encoder was coupled to the free pulley to measure the linear displacement.



Figure 2.9: SICK Incremental Rotary Encoder [36]

Two limit switches were mounted on either end of the track to act as the track end limits. These were designed to be capable of sliding along the track to change track length as required. There are many different varieties of limit switches available in the market and the ones utilized in this test-bench are shown in Figure 2.10.



Figure 2.10: Limit Switch [37]

The Arduino ATmega 2560 which is shown in Figure 2.11 was the microcontroller used to interface the sensors to the motor for this system. It is planned that the controller will also be implemented on this device.

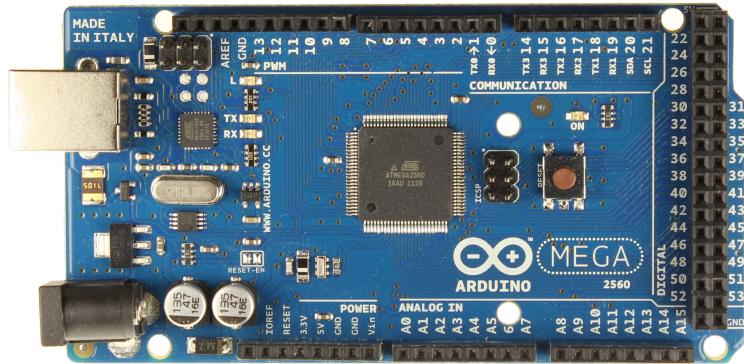


Figure 2.11: Arduino ATmega 2560 [38]

A GW-insteek GPS-3303 lab power supply as shown in Figure 2.12 was operated at 24V/3A to act as the power source.



Figure 2.12: GW-insteek GPS-3303 Power Supply [39]

A Pololu dual VNH5019 motor driver shield for Arduino was used to interface the Arduino, power supply and garmotor together. Figure 2.13 displays this motor driver.

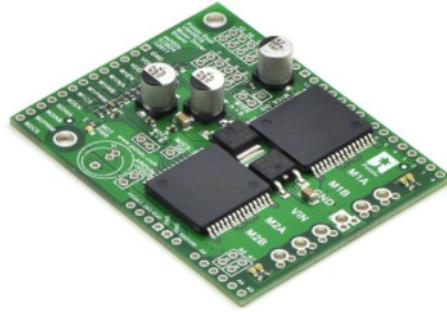


Figure 2.13: Pololu Dual VNH5019 Motor Driver Shield for Arduino [40]

An interface was created between the Arduino serial monitor and MATLAB which allowed a real-time display of sensor data on a PC. Figure 2.14 shows three plots displaying *linear displacement versus time*, *angular displacement versus time*, and *cart-pendulum animation*.

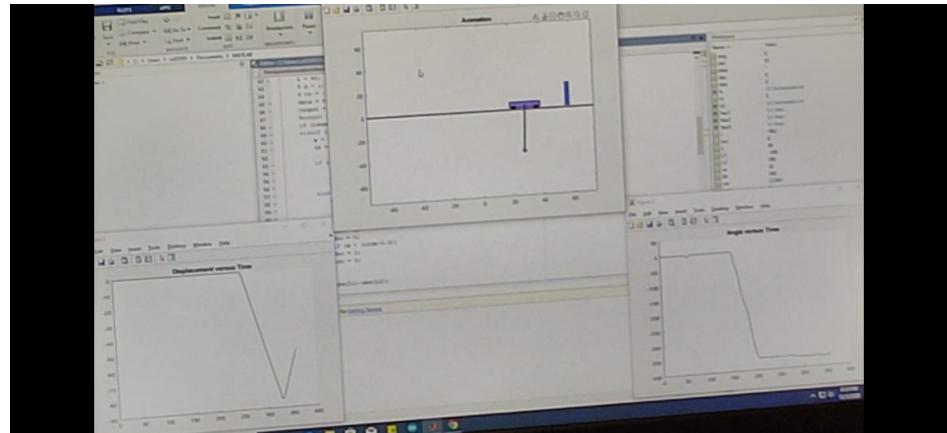


Figure 2.14: Realtime Graphs of System Sensors

Figure 2.15 shows an overview of the cart-mounted inverted pendulum test-bench designed for this project. More closeups of the system can be found in Appendix C, Figures C.1-C.6.

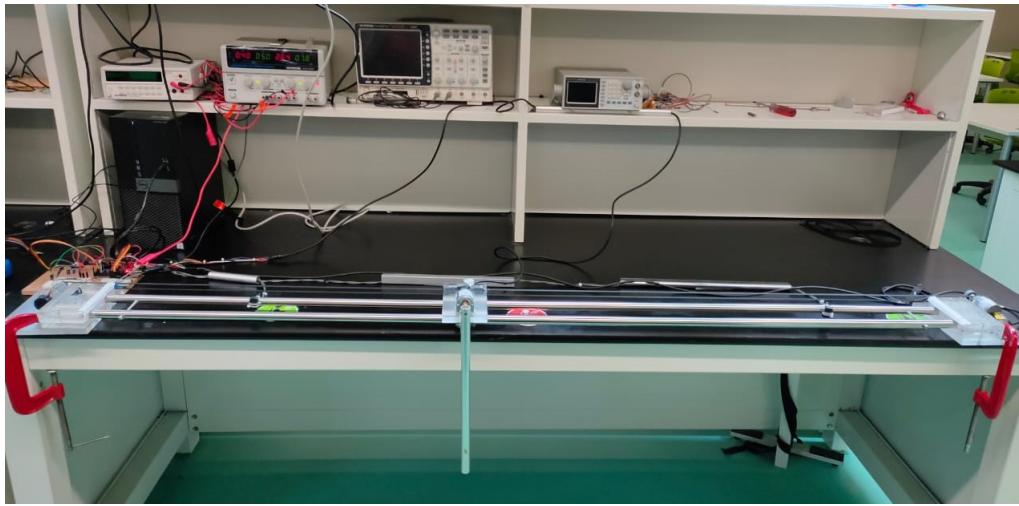


Figure 2.15: Front View of Platform

2.6.2 System Characteristics

The system was designed to be low-cost, scaleable, modular, and open-source which sets it far apart from the competitive products in the market. The products by companies such as Quanser and Leybold's are not only orders of magnitudes more expensive, they are also black-boxes which the user cannot actively interact with. Replacement or addition of any part required is an expensive task. Similarly, it was also intended that the designed platform display comparable behavior with systems employed in research when the same control strategy is deployed.

The aim is to implement stabilization in our inverted pendulum system. Based on extensive research, the authors concluded that our system should be able to achieve this goal for the following specifications.

The system should have a settling time of 1 - 2 seconds for small disturbances in the range of 1 - 3 degrees, as was observed in [10][30][41][11][42], which utilized various control methodologies including: neuro-fuzzy control, hybrid-vision control, PID control and more popularly LQR. For larger disturbances in the range of 5 - 15 degrees, settling times of no greater than 5 seconds were observed, as seen in [10][13][41].

The system should not have an overshoot exceeding the initial disturbance in any case, as was observed in the literature cited above. [14] designed its system under the constraints of a maximum overshoot of 20%.

The deviation in cart position should be in a range of 8 - 10 cm for smaller disturbances [43][10], and no greater than 20 cm for larger ones [13], comparing to full track lengths from 70 cm to 125 cm.

CHAPTER 3

SYSTEM MODELLING

3.1 Introduction

This chapter details development of the mathematical model and simulations carried out to achieve a better understanding of the design and implementation of the physical system. The models and simulations outlined in this chapter not only served to enrich the authors' understanding of the control problem at hand, but also helped to provide a baseline using which the authors were able to design the physical system, and understand its limitations and constraints before constructing it.

3.2 DC Gear Motor Modelling

The motor utilized to actuate the inverted pendulum system is a Pololu DC Gear Motor 4754. The specifications of the motor, as provided by the manufacturer, are stated below:

Table 3.1: DC Motor Specification

Parameter	Value
Diameter	37 mm
Shaft Diameter	6 mm
Rated Voltage	12 V
No-load Current	0.2 A
Rated Stall Current	5.5 A
No-load RPM	150
Stall Torque	2.65 Nm
Gear Ratio	70:1

It was important to mathematically model this motor as it enabled the controller to translate the angular feedback, θ_p , of the pendulum to the control effort in terms of motor voltage, E_a .

3.2.1 Generalized Motor Model

The first step of motor modelling involves referring to the generalized electromechanical model of a DC motor which has the following transfer function:

$$\frac{\theta_m}{E_a} = \frac{\frac{K_t}{R_a J_m}}{s[s + \frac{1}{J_m}(D_m + \frac{K_t K_b}{R_a})]} \quad (3.1)$$

Where θ_m is the motor shaft angle,

E_a is the armature voltage,

K_b is the motor-back-emf constant,

K_t is the motor torque constant,

D_m is the viscous armature damping,

J_m is the armature inertia,

R_a is the armature resistance,

L_a is the armature inductance.

The values for R_a and L_a were obtained by connecting the motor to an impedance analyzer. These are given in Table 3.2.

Table 3.2: Electrical Characteristics

Parameter	Value	Units
R_a	4.5562	Ohms
L_a	1.87	mH

Furthermore, generalized equations describing the relation between torque-speed and torque-current for such a motor can be expressed as follows:

$$T_m = \frac{K_t}{R_a} e_a - \frac{K_b K_t}{R_a} w_m \quad (3.2)$$

$$w_m = \frac{e_a}{K_b} - \frac{R_a}{K_b K_t} T_m \quad (3.3)$$

From the above equations, two further equations were extracted based on the stall condition i.e. $w_m = 0$, and no-load condition i.e. $T_m = 0$. These are given below:

$$T_m = \frac{K_t}{R_a} e_a \quad (3.4)$$

$$w_m = \frac{e_a}{K_b} \quad (3.5)$$

Additionally, the relation between the torque and current can be generally expressed as:

$$I_a = I_{no-load} + \frac{1}{K_t} T_m \quad (3.6)$$

The aforementioned equations provide the necessary background and tools that

helped the authors in modelling the motor. The derived motor model is a lumped model that takes into account the inertia and friction due to the actuation mechanism, as the motor's mechanical response was derived from the encoder while it remained coupled with the actuation system. Two approaches were utilized to model the motor, which were grey-box modelling and white-box modelling respectively. This helped cross-validate each of the models, and choose the one closest to the actual response.

3.2.2 Grey Box Motor Model

Determining back-emf constant K_b

It is known that the armature voltage and angular velocity are related by the following equation:

$$e_a = K_b w_m \quad (3.7)$$

From the above equation, since $e_a = E_a - I_a R_a$,

$$K_b = \frac{E_a - I_a R_a}{w_m} \quad (3.8)$$

Subsequently, a digital multi-meter was used to measure the voltage and current drawn by the motor. The no-load speed was also estimated by using a slow-motion camera capturing the revolutions completed by the shaft in a fixed duration. This was then multiplied by a conversion factor to obtain the value of speed in radians per second. The table below summarizes the findings:

Table 3.3: Electrical Characteristics

Parameter	Value	Units
E_a	12	V
I_a	0.17	A
R_a	4.5562	Ohm
w_m	16.02	rad/s

The value of K_b can therefore be obtained as follows:

$$K_b = \frac{12 - (0.17)(4.5562)}{16.02} = 0.7007 \text{ Vs/rad} \quad (3.9)$$

Determining motor-torque constant K_t

Two approaches were utilized for finding K_t , which consequently lead to the generation of two models. This helped in the cross-validation of the models.

Approach One - Experimental Assumption:

$$K_{t1} = K_b = 0.7007 \text{ Nm/A} \quad (3.10)$$

Approach Two - Experimental Determination:

It was known that the motor stall torque is 2.65 Nm , thereby using(3.4), K_t was experimentally determined to be:

$$K_{t2} = \frac{(T_{stall})(R_a)}{E_a} = \frac{(2.65)(4.5562)}{12} = 1.006 \text{ Nm/A} \quad (3.11)$$

Determining the Damping Constant D_m

Similarly, two values for D_m were also determined based on the following relationship:

$$D_{m1} = \frac{K_t I_m}{w_m} = \frac{(0.7007)(0.17)}{16.02} = 7.4356 \times 10^{-3} \text{ Nm s/rad} \quad (3.12)$$

$$D_{m2} = \frac{K_t I_m}{w_m} = \frac{(1.006)(0.17)}{16.02} = 1.0675 \times 10^{-2} \text{ Nm s/rad} \quad (3.13)$$

Determining the motor inertia J_m

Ideally, determining the motor inertia requires information about the motor's internal components such as the shaft length and shaft mass. However, these were unavailable for the motor under consideration. Therefore, an alternative strategy was used which made use of the following relationship:

$$J_m = \frac{t_{mech}(D_m R_a + K_b K_t) - D_m L_a}{R_a} \quad (3.14)$$

Here, t_{mech} is the mechanical response time of the motor. It is defined as the time required by the motor to reach 63.63% of its steady-state speed. This was determined using the angular velocity response of the motor by coupling a rotary encoder with the motor. The response in Figure 3.1 was then acquired by MATLAB via an Arduino. The data was then fitted appropriately using an exponential fit, and the relevant parameters were extracted. The code employed for this is given in the appendix E.1.

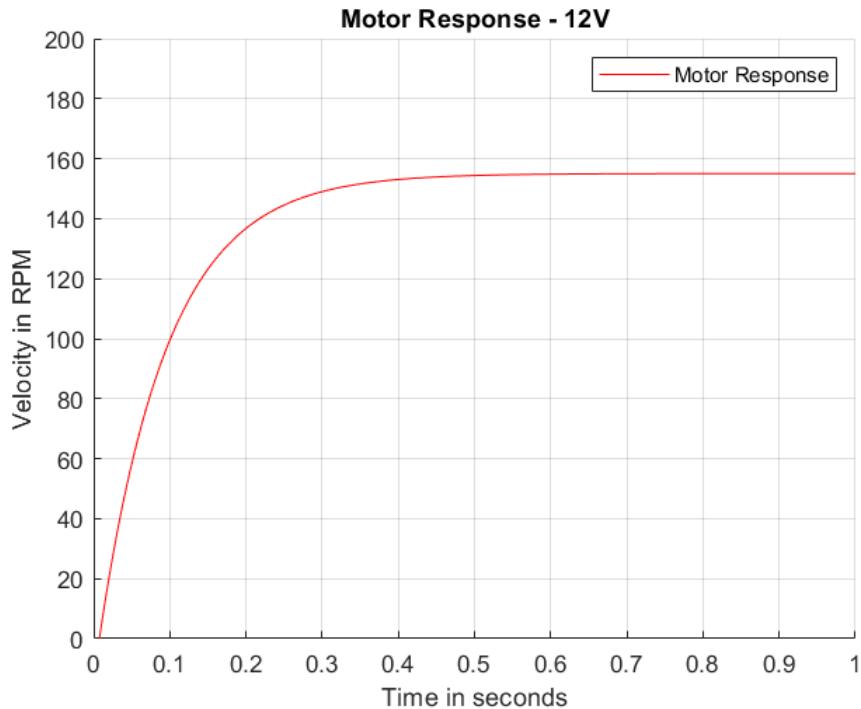


Figure 3.1: Angular Velocity response of DC Polulu Motor

The following value for t_{mech} is acquired through interpolation.

$$t_{mech} = 0.0986 \text{ seconds} \quad (3.15)$$

Based on this value, and with the aid of(3.14), two values for J_m were obtained as follows:

$$J_{m1} = 0.011355 \text{ Nm}^2 \quad (3.16)$$

$$J_{m2} = 0.016303 \text{ Nm}^2 \quad (3.17)$$

Motor Models

The parameter values determined via grey-box modelling are summarized in table 3.4.

Table 3.4: Motor Model Parameters

Parameter	Value	Units
K_b	0.7007	Vs/rad
t_{mech}	0.0986	sec
K_{t1}	0.7007	Nm/A
K_{t2}	1.006	Nm/A
D_{m1}	7.4356×10^{-3}	Nms/rad
D_{m2}	1.0675×10^{-2}	Nms/rad
J_{m1}	0.011355	Nm^2
J_{m2}	0.016303	Nm^2

Using (3.1), the following two frequency domain motor models were acquired:

$$\frac{\theta_{m1}}{E_{a1}} = \frac{13.544}{s(s + 10.1450)} \quad (3.18)$$

$$\frac{\theta_{m2}}{E_{a2}} = \frac{13.543}{s(s + 10.1446)} \quad (3.19)$$

It was observed that both motor models were approximately the same. The models verified that the assumption of $K_b = K_t$ was in-fact valid for such small-scale DC Gear Motors. From this, the angular velocity response is obtained by differentiating (3.18) and/or (3.19), which simply involved multiplying it with s in the frequency domain as shown below:

$$\frac{w_m}{E_a} = \frac{13.543}{s + 10.1446} \quad (3.20)$$

3.2.3 White Box Motor Model

To perform white box modelling, the following equations were obtained from speed-torque and current-torque curves from data-sheet of the Pololu DC Gear Motor 4754 [17]. These equations were converted and expressed in units of radians per second and N m respectively.

$$w_m = 15.708 - (5.443 \times 10^{-4})T_m \quad (3.21)$$

$$I_a = 0.088 + 1.8355T_m \quad (3.22)$$

Determining back-emf constant K_b

The following data was acquired from the motor data-sheet as well:

Table 3.5: Electrical Characteristics

Parameter	Value	Units
E_a	12	V
I_a	0.15	A
R_a	4.5562	Ohm
w_m	15.708	rad/s

Using (3.8), the value for K_b is calculated as follows:

$$K_b = \frac{12 - (0.15)(4.5562)}{15.708} = 0.720 \text{ Vs/rad} \quad (3.23)$$

Determining motor-torque constant K_t

Comparing with table 3.5 and (3.6), then:

$$K_{t2} = \frac{1}{1.8355} = 0.545 \text{ Nm/A} \quad (3.24)$$

Determining the Damping Constant D_m

Similarly, two values for D_m were also determined based on the following relationship:

$$D_m = \frac{K_t I_m}{w_m} = \frac{(0.545)(0.15)}{15.708} = 5.204 \times 10^{-3} \text{ Nm s/rad} \quad (3.25)$$

Determining the motor inertia J_m

The value of t_{mech} remains the same in this case, and was determined using the angular velocity obtained in figure (3.1).

$$t_{mech} = 0.0986 \text{ seconds} \quad (3.26)$$

Based on this value, and with the aid of(3.14), the value of J_m was calculated:

$$J_m = 0.009 \text{ Nm}^2 \quad (3.27)$$

Motor Models

The parameter values determined via white-box modelling are summarized in the table below:

Table 3.6: Motor Model Parameters

Parameter	Value	Units
K_b	0.720	Vs/rad
t_{mech}	0.0986	sec
K_t	0.545	Nm/A
D_m	5.204×10^{-3}	Nms/rad
J_m	0.009	Nm^2

Using (3.1) the following frequency domain motor model was acquired:

$$\frac{\theta_m}{E_a} = \frac{13.291}{s(s + 10.1476)} \quad (3.28)$$

From this, the angular velocity response was obtained by differentiating (3.28) such that:

$$\frac{w_m}{E_a} = \frac{13.291}{s + 10.1476} \quad (3.29)$$

3.2.4 Motor Model Validation

Grey Box Model Validation

Using the model described by (3.19) and (3.20), and the actual response obtained using a rotary encoder, the following response graphs were plotted which successfully validated the model:

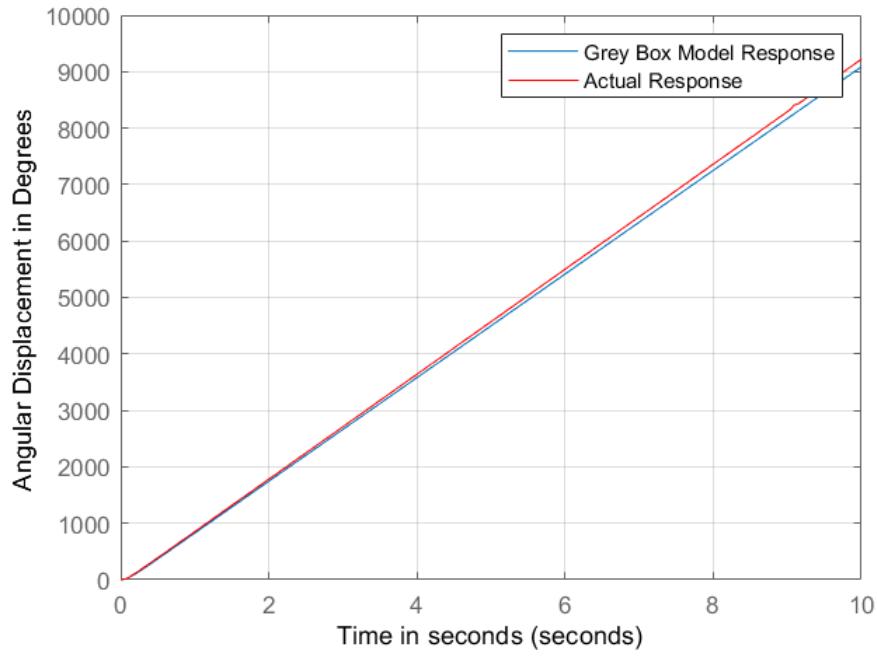


Figure 3.2: Angular Displacement - Model Validation

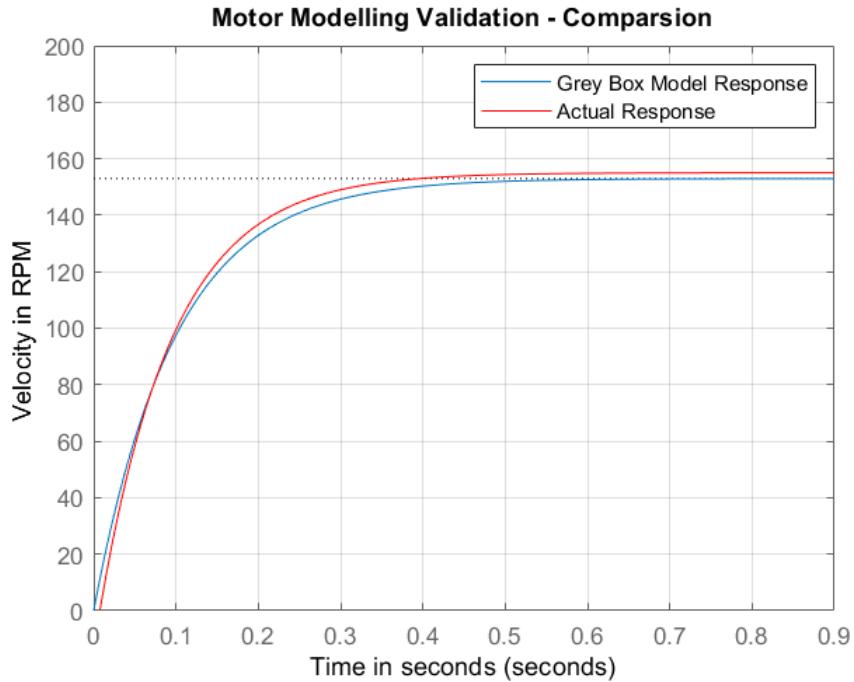


Figure 3.3: Angular Velocity - Model Validation

It was observed from Figures 3.2 and 3.3 that the grey-box motor model is a close match with the actual response of the motor, with only a 3.24% error in the steady state velocity. However, Figure 3.2 does show that the error in displacement keeps on increasing as time increases and will approach infinity as time approaches infinity.

White Box Model Validation

Using the model described by (3.28) and (3.29), and the actual response obtained using an encoder, the following graphs were plotted that validated the model:

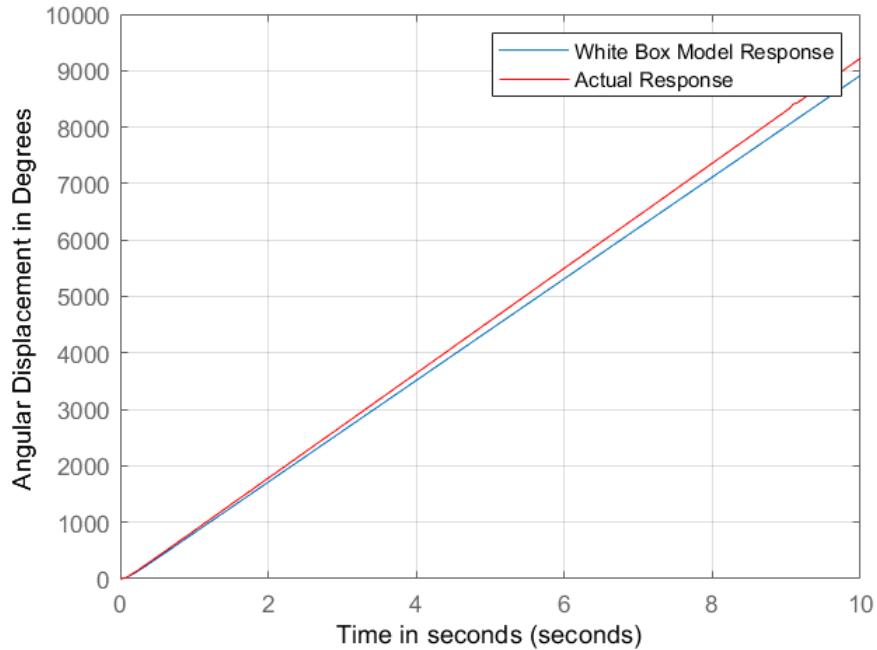


Figure 3.4: Angular Displacement - Model Validation

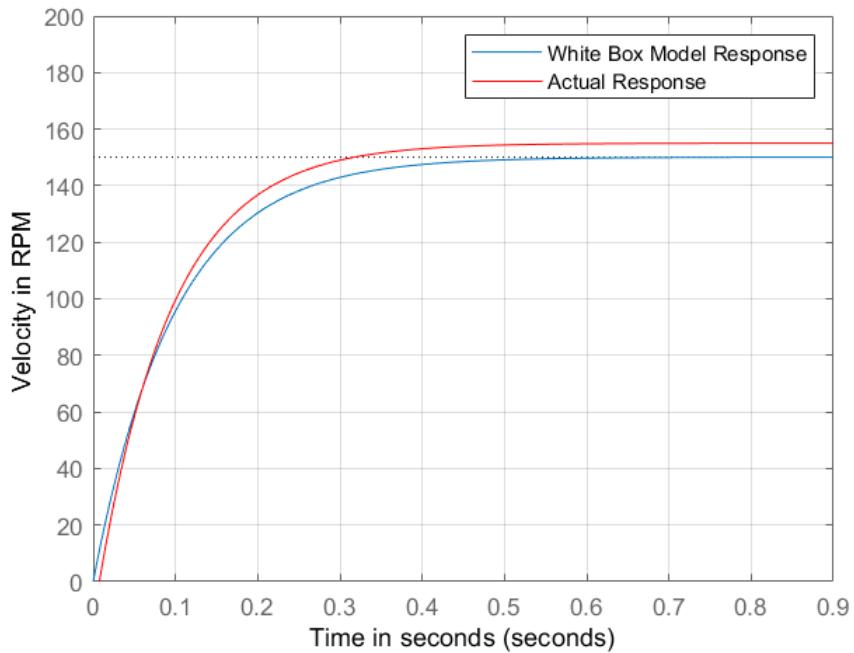


Figure 3.5: Angular Velocity - Model Validation

It was observed from Figure 3.4 and 3.5 that the white-box motor model fairly

matches with the actual response of the motor with a 5.84% error in the steady state velocity.

3.2.5 Discussion

Model Comparison

A comparison of the grey-box and white-box model is presented below. It can be observed that the grey-box model is closer to the actual response than the white-box model, as indicated by the errors in their steady state velocities.

Table 3.7: Steady State Velocity Errors

Model	% Error
Grey Box	3.24
White Box	5.84

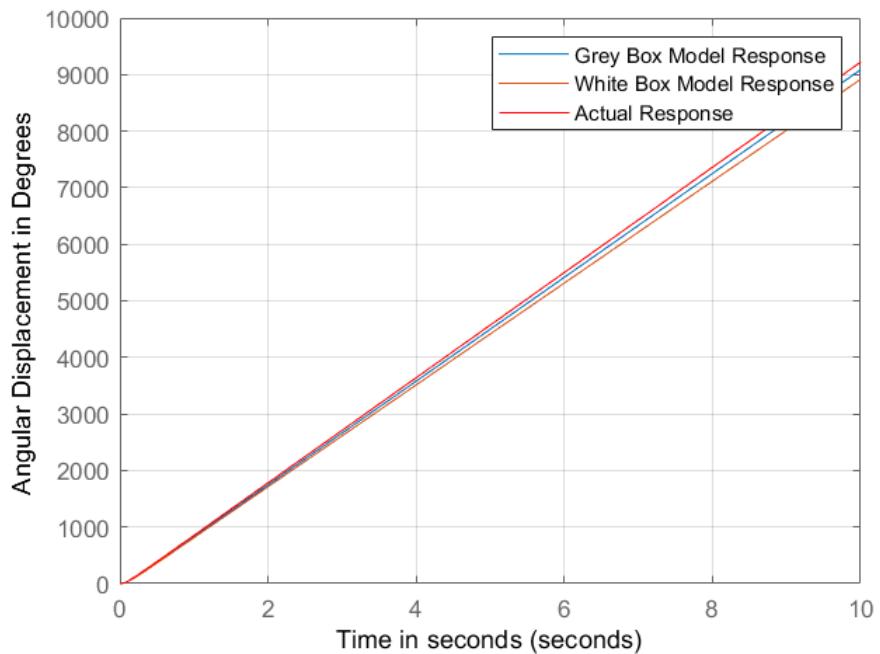


Figure 3.6: Comparison between Angular Displacement models

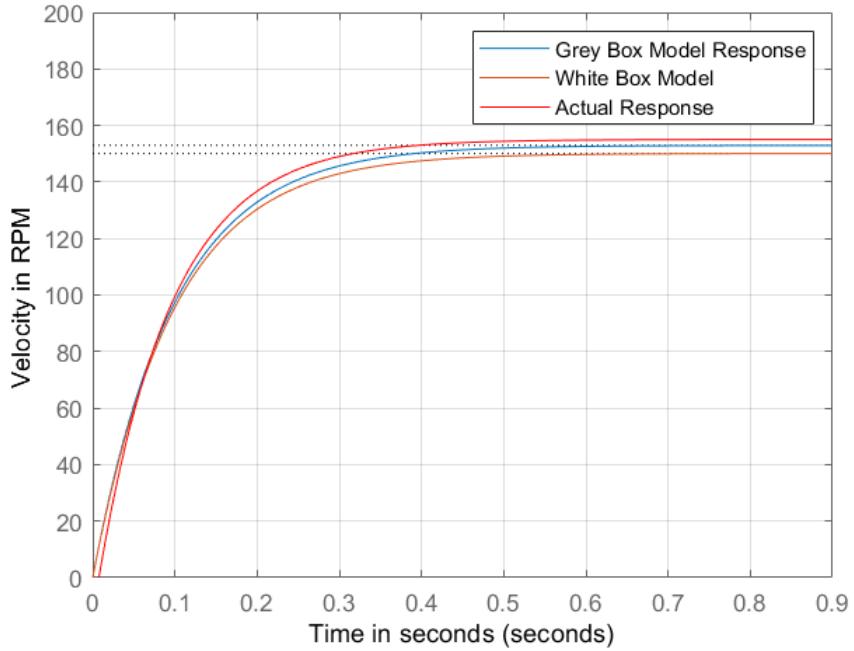


Figure 3.7: Comparison between Angular Velocity models

It was deduced that the grey-box model offered a better approximation of the motor's response since the parameters in the modelling process were extracted directly from experiments conducted on the motor itself. This automatically took into account any variations in behavior of the motor due to aging or due to deviations from the generalized specifications provided by the manufacturer, which are inherent to the manufacturing process. For a white box model, these factors were neglected as this modelling approach relies on data retrieved from the manufacturer's specifications, and hence the resulting model deviates more from the true response of the motor.

3.3 Process Modelling

3.3.1 Plant Dynamic Equations

The authors utilized the first principles approach to develop a mathematical model for the dynamics of the plant which consists of a cart-mounted inverted pendulum. Figure 3.8 shows the free-body diagram of this system.

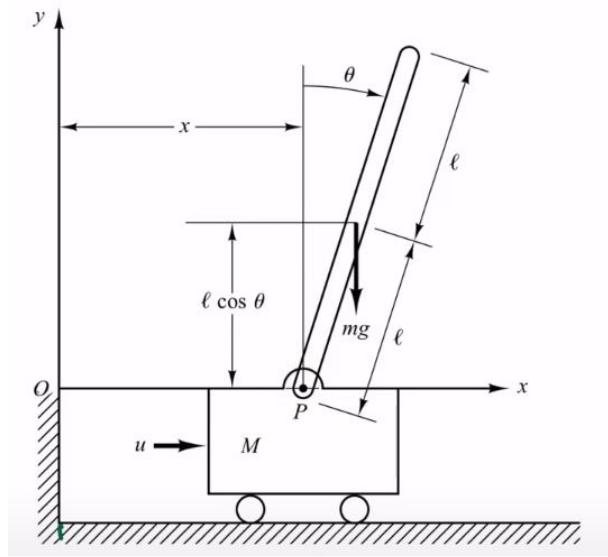


Figure 3.8: Cart-Mounted Inverted Pendulum Free-Body Diagram [44]

Using Newton's 2nd Law of Motion, the horizontal and vertical forces, V_f and H_f , acting at the center of gravity of the pendulum can be described by the following two differential equations:

$$V_f - mg = m \frac{d^2 y_g}{dt^2} \quad (3.30)$$

$$H_f = m \frac{d^2 x_g}{dt^2} \quad (3.31)$$

Where $x_g = x + l \sin \theta$, $y_g = l \cos \theta$, and θ is a function of time. Upon differentiating, the following equations are obtained:

$$V_f = mg - ml\ddot{\theta} \sin \theta - ml\dot{\theta}^2 \cos \theta \quad (3.32)$$

$$H_f = m\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta \quad (3.33)$$

Using (3.32) and (3.33) from above, the force balance on the cart can be described as:

$$F - H_f = M \frac{d^2(x)}{dt^2} \quad (3.34)$$

$$F = (M + m)\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta \quad (3.35)$$

Where F is the externally applied force on the cart. This produces the first dynamic equation, which provides the relationship between an externally applied force, plant state variables and the cart acceleration \ddot{x} . Another dynamic equation was derived considering the torque balance on the pendulum as a result of the vertical and horizontal forces as follows:

$$\tau_{total} = \tau_v + \tau_h \quad (3.36)$$

$$I\ddot{\theta} = V_f l \sin \theta - H_f l \cos \theta \quad (3.37)$$

Where I is the inertia due to the mass and geometry of the pendulum. Utilizing (3.36) & (3.37), as well as (3.30) & (3.32) from above, the second dynamic equation is derived as follows:

$$(I + ml^2)\ddot{\theta} + m\ddot{x}l \cos \theta - mgl \sin \theta = 0 \quad (3.38)$$

The equations above provide a mathematical relation between the angular acceleration $\ddot{\theta}$ and other state variables. Finally, the following two non-linear equations describing the dynamic model of the inverted pendulum are obtained as follows:

$$F = (M + m)\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta \quad (3.39)$$

$$(I + ml^2)\ddot{\theta} + m\ddot{x}l \cos \theta - mgl \sin \theta = 0 \quad (3.40)$$

3.3.2 Non-linear Plant Model

To derive a non-linear plant model, the pair of dynamic equations are manipulated such that the higher order state variable are made the subject in the equation. Using (3.39) and (3.40), the following two state equations for the non-linear plant are obtained as:

$$\ddot{x} = \frac{F}{a} - \frac{m^2 gl^2 \sin \theta \cos \theta}{a(I + ml^2)} + \frac{ml \sin \theta \dot{\theta}^2}{a} \quad (3.41)$$

Where $a = M + m - \frac{m^2 l^2 \cos \theta^2}{I + ml^2}$

$$\ddot{\theta} = \frac{F}{b} - (M + m) \frac{g \sin \theta}{b \cos \theta} + \frac{ml \dot{\theta}^2 \sin \theta}{b} \quad (3.42)$$

Where $b = ml \cos \theta - (M + m) \frac{(I + ml^2)}{ml \cos \theta}$

A detailed derivation for the above expressions can be found in appendix E.2.

3.3.3 Non-Linear Plant Model with Viscous Friction

The previously derived model provided a baseline for the control problem. However, it was still very idealized in its behavior and did not incorporate any form of damping due to friction. To make the model richer and a better approximation of a real-world physical system, another model was derived using the following dynamic equations incorporating viscous friction:

$$F = (M + m)\ddot{x} - B_1 \dot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta \quad (3.43)$$

$$(I + ml^2)\ddot{\theta} - B_2 \dot{\theta} + m\ddot{x}l \cos \theta - mgl \sin \theta = 0 \quad (3.44)$$

Where B_1 and B_2 are coefficients of viscous friction for the cart and pendulum respectively. Using(3.43) and (3.44), the following state equations were obtained:

$$\ddot{x} = \frac{F}{a} + \frac{B_1 \dot{x}}{a} - \frac{m^2 g l^2 \sin \theta \cos \theta}{a(I + ml^2)} + \frac{ml\dot{\theta}^2 \sin \theta}{a} \quad (3.45)$$

Where $a = M + m - \frac{m^2 l^2 \cos \theta^2}{I + ml^2}$

$$\ddot{\theta} = \frac{F}{b} - \frac{(M + m)B_2 \dot{\theta}}{bml \cos \theta} - \frac{(M + m)g \sin \theta}{b \cos \theta} + \frac{ml\dot{\theta}^2 \sin \theta}{b} \quad (3.46)$$

Where $b = ml \cos \theta - \frac{(M + m)(I + ml^2)}{ml \cos \theta}$

A detailed derivation for the above expressions can be found in appendix E.3.

3.3.4 Linearized Plant Model with Viscous Friction

Since the goal of the design was to demonstrate control against disturbances about the inverted position, the non-linear plant model incorporating viscous friction was linearized about that inverted position. The following assumptions were made to linearize the equations at $\theta = 180$ degrees:

$$\sin \theta \approx \theta$$

$$\cos \theta \approx -1$$

$$\theta\dot{\theta}^2 \approx 0$$

This lead to the following pair of dynamic equations:

$$F = (M + m)\ddot{x} - B_1 \dot{x} - ml\ddot{\theta} \quad (3.47)$$

$$(I + ml^2)\ddot{\theta} - B_2 \dot{\theta} - ml\ddot{x} - mg l \theta = 0 \quad (3.48)$$

Where B_1 and B_2 are coefficients of viscous friction for the cart and pendulum respectively. Using (3.47) and (3.48), the following linearized state equations were

derived:

$$\ddot{x} = \frac{F}{a} + \frac{mlB_2\dot{\theta}}{a(I+ml^2)} + \frac{m^2gl^2\theta}{a(I+ml^2)} + \frac{B_1\dot{x}}{a} \quad (3.49)$$

Where $a = M + m - \frac{m^2l^2}{I+ml^2}$

$$\ddot{\theta} = \frac{F}{b} + \frac{(M+m)g\theta}{b} + \frac{(M+m)B_2\dot{\theta}}{mlb} + \frac{B_1\dot{x}}{b} \quad (3.50)$$

Where $b = \left(\frac{(M+m)(I+ml^2)}{ml} - ml \right)$

A detailed derivation for the above expressions can be found in appendix E.4.

3.3.5 State Space Plant Model

In order to simulate the linearized plant model that incorporates viscous friction, the state equations are utilized to form a state space model. The state variables are defined as:

$$\mathbf{x}_1 = \theta$$

$$\mathbf{x}_2 = \dot{\theta}$$

$$\mathbf{x}_3 = x$$

$$\mathbf{x}_4 = \dot{x}$$

This could further be expressed as:

$$\dot{\mathbf{x}} = \frac{d}{dt} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \\ \dot{\mathbf{x}}_3 \\ \dot{\mathbf{x}}_4 \end{bmatrix}$$

Hence, utilizing (3.49) and (3.50), the following state-space model is deduced:

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \\ \dot{\mathbf{x}}_3 \\ \dot{\mathbf{x}}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(M+m)g}{b} & \frac{(M+m)B_2}{mlb} & 0 & \frac{B_1}{b} \\ 0 & 0 & 0 & 1 \\ \frac{m^2l^2g}{a(I+ml^2)} & \frac{mlB_1}{a(I+ml^2)} & 0 & \frac{B_1}{a} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{b} \\ 0 \\ \frac{1}{a} \end{bmatrix} \mathbf{F}$$

Where $a = M + m - \frac{m^2l^2}{I + ml^2}$ and $b = \left(\frac{(M+m)(I+ml^2)}{ml} - ml \right)$

3.4 Plant Simulation

It was important to simulate the mathematical models of the system in order to observe the behaviour and determine correctness. For this purpose the authors' developed a detailed MATLAB-Simulink model to aid the process.

3.4.1 Open-loop Response of Non-linear Plant Model

The Simulink model shown in Figure 3.9 was developed to simulate the open-loop response of the non-linear plant model, which consisted of a plant model that was described as a set of differential equations.

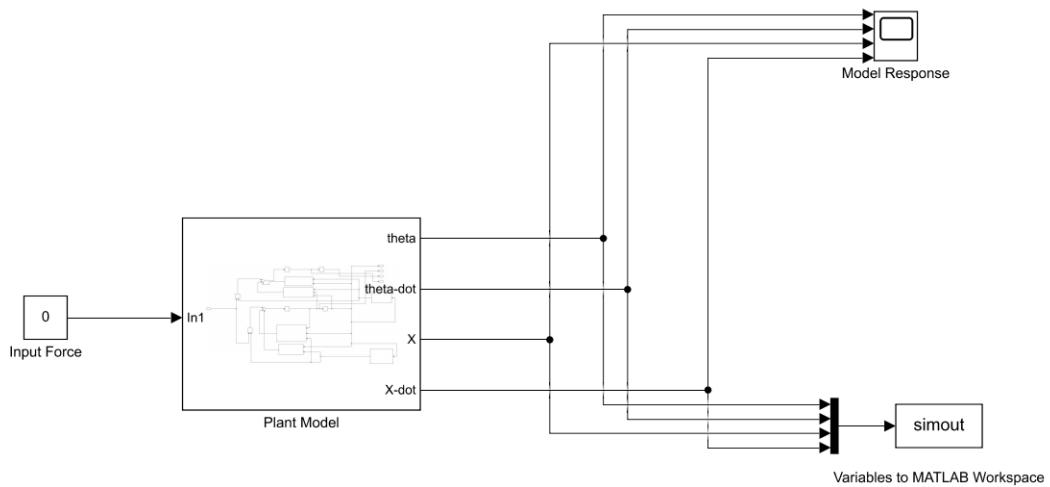


Figure 3.9: Simulink Non-linear Plant Model

MATLAB was utilized to initialize the system's physical parameters such as the pendulum's mass, length, inertia etc, after which the Simulink model was run. The response obtained is shown in Figures 3.10 and 3.11 respectively.

For clarity it is worth noting that in Figures 3.10 to 3.13 the horizontal axis shows time in seconds, whereas the vertical axis shows the angular displacement (yellow), cart displacement (red), angular velocity (blue), and cart velocity (green) in units of radians, radians per second respectively.

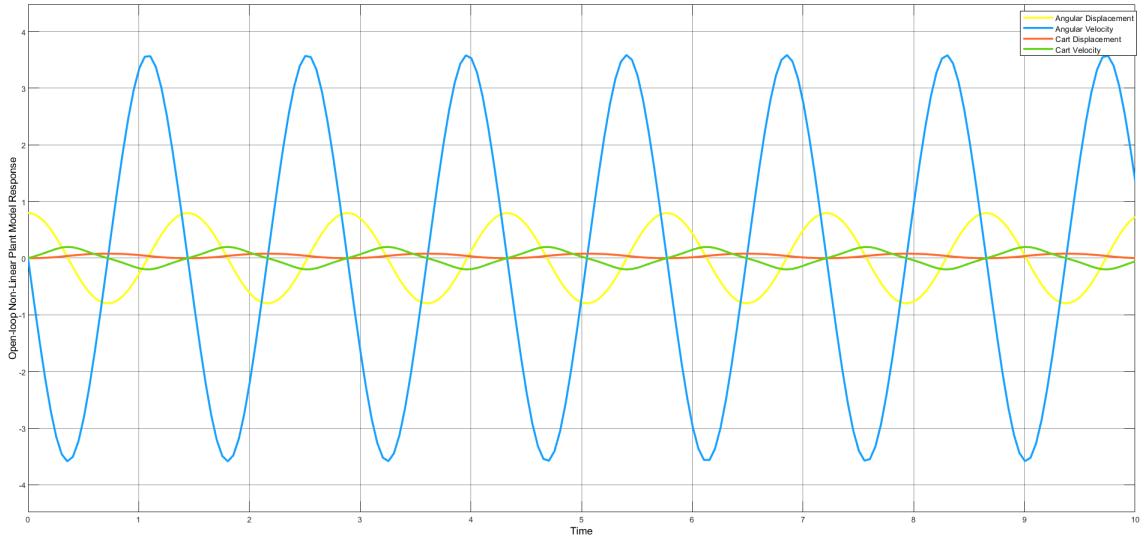


Figure 3.10: Open-loop Non-linear Plant Model Response - $\theta = 45^\circ$.

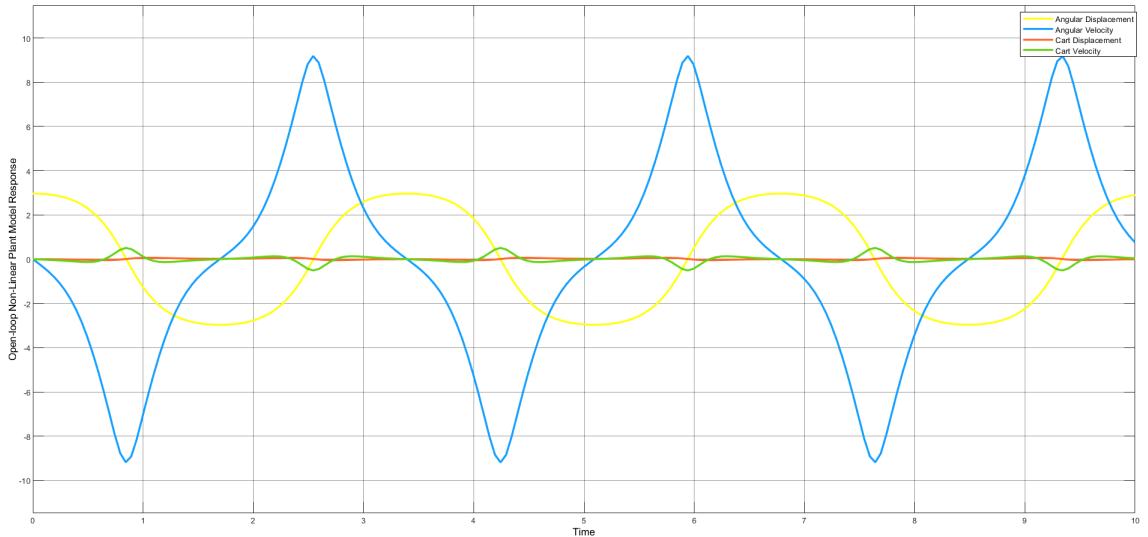


Figure 3.11: Open-loop Non-linear Plant Model Response - $\theta = 170^\circ$

The model's response demonstrated a behaviour that is reflective of the swinging of an ideal pendulum. It was observed that the pendulum's angular displacement and velocity oscillate without any damping, as expected, since the simulation does not incorporate friction. Furthermore, as the initial displacement angle is increased, the maximum velocity attained by the pendulum also increases. Another interesting observation is that oscillations in the pendulum's motion cause minor variations in the cart's displacement and velocity demonstrating the physical connection between the two components.

3.4.2 Open-loop Response of Non-linear Plant incorporating Viscous Friction

Viscous friction was incorporated into the Simulink model developed in the previous section, and in a similar fashion as above, the model was simulated to observe the changes in the plant's response due to friction. The result for this can be seen in Figures 3.12 and 3.13.

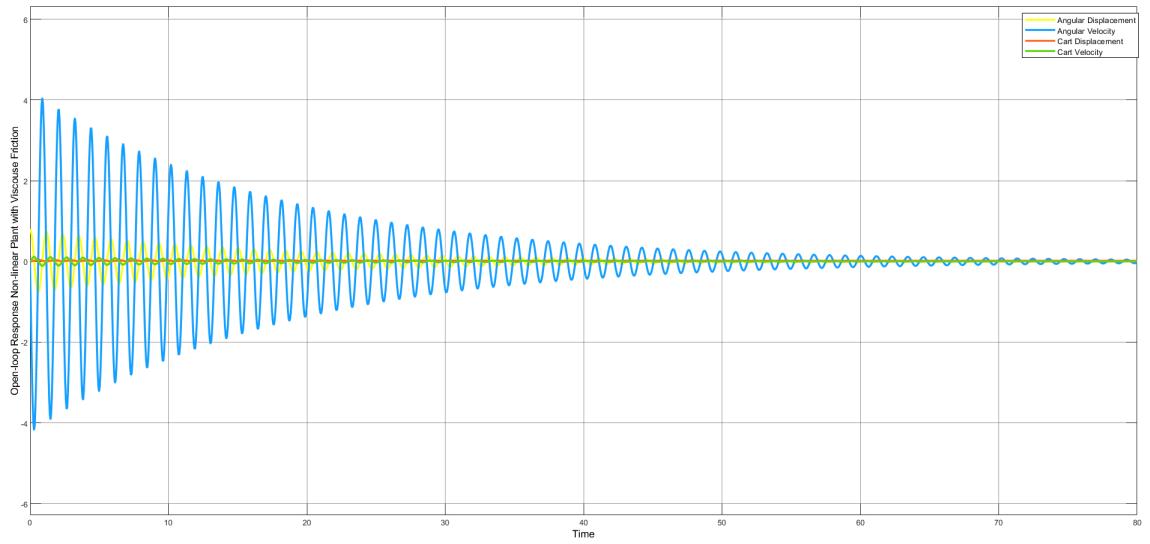


Figure 3.12: Open-loop Non-linear Plant Model Response with Viscous Friction - $\theta = 45^\circ$

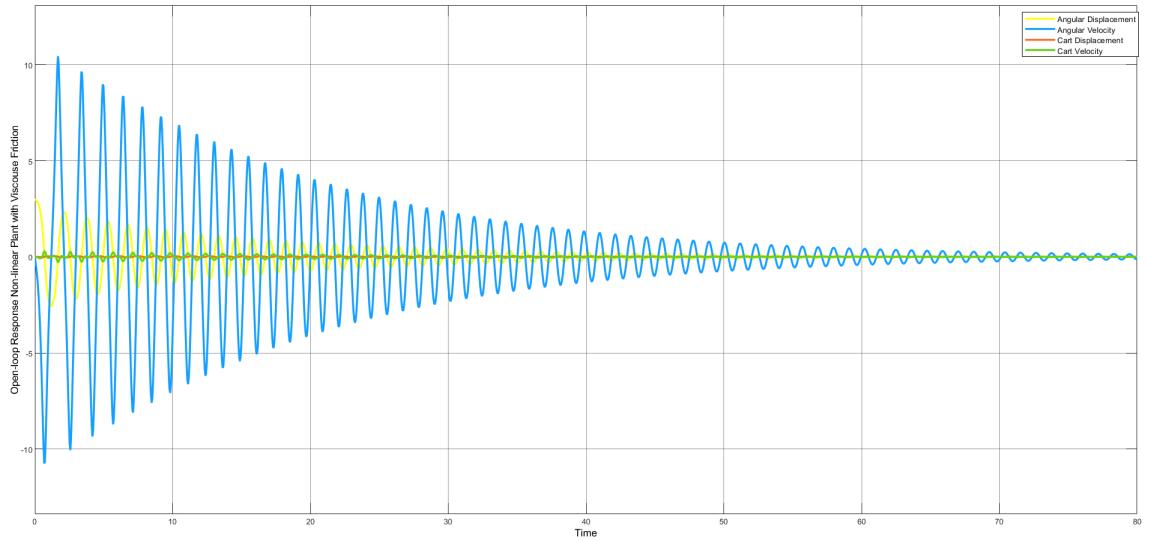


Figure 3.13: Open-loop Non-linear Plant Model Response with Viscous Friction - $\theta = 170^\circ$

It was observed that the pendulum's angular displacement and velocity eventually

settled to zero as a result of the damping due to viscous friction. The respective settling times in each of the above cases were 38 and 57 seconds respectively.

3.5 Plant Tuning and Validation

After modelling the plant, the obtained plant model was tuned via the simulations. This involved extensive tweaking of the viscous friction coefficients B_1 and B_2 by comparing them with the plant's actual response obtained through encoders. The responses for the pendulum's angular displacement and velocity are shown in Figures 3.14 and 3.15.

Table 3.8: Fine Tuned Values for Friction Coefficients

Parameter	Value
B_1	0.001
B_2	0.0008

Table 3.9: Other Simulation Parameters

Parameter	Value
Cart Mass	0.637 kg
Shaft Mass	0.023 kg
Coupler Mass	0.049 kg
Pendulum Mass	0.085 kg
Pendulum Length	0.517 m
Gravitational Acceleration	9.81 kg ms ⁻²
System Inertia	0.005 kg m ⁻²

For clarity it is worth noting that in Figures 3.14 to 3.21 the horizontal axis shows time in seconds, whereas the vertical axis for graphs concerning displacement and velocity has units in radians and radians per second.

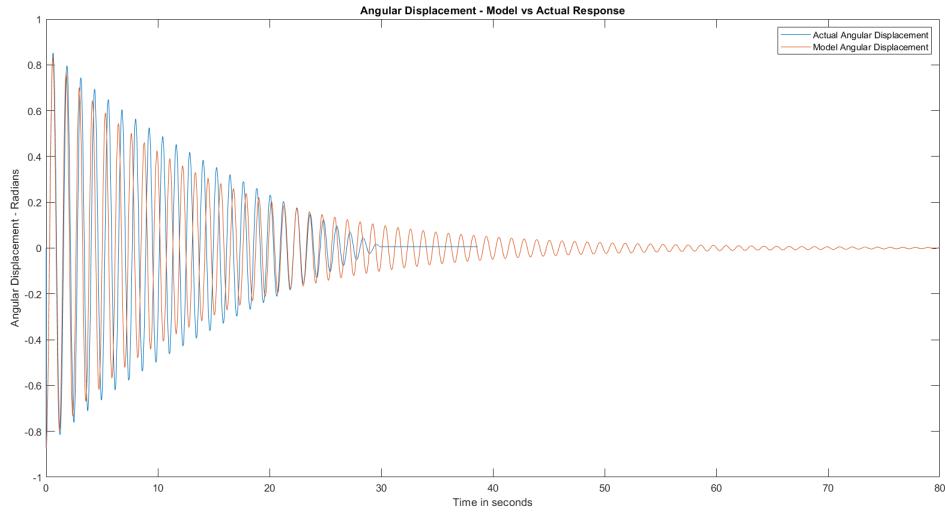


Figure 3.14: Model Displacement Tuning & Comparison - $\theta = 50^\circ$

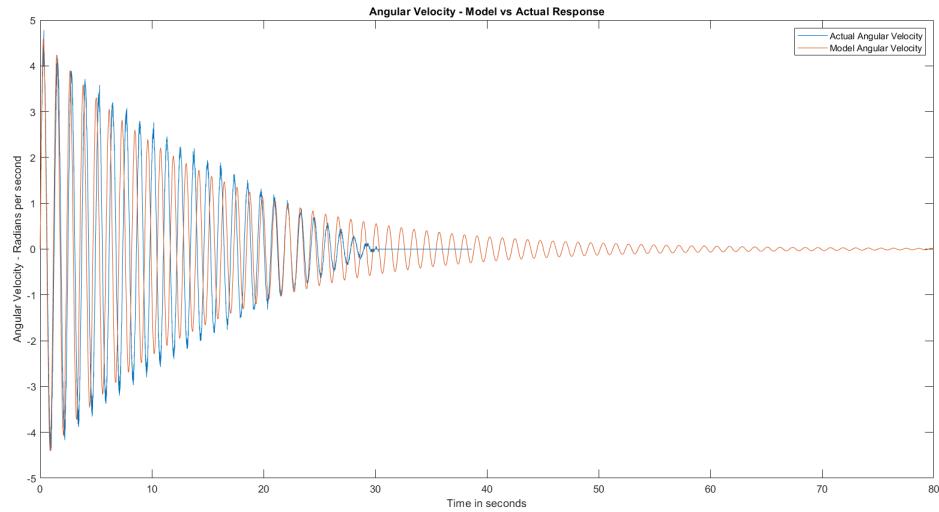


Figure 3.15: Model Velocity Tuning & Comparison - $\theta = 50^\circ$

The Figures 3.14 and 3.15 above show only the standalone behaviour of the pendulum. For a more realistic approximation of the system to further tune the friction parameters, both the pendulum and cart motion were taken under consideration. This is demonstrated in the Figures E.1 - E.8.

Figures 3.16 and 3.17 shows a comparison between the values of the simulated and actual response for the pendulum's angular displacement and velocity. One interesting takeaway from Figures 3.16 and 3.17 is that as time progresses, the model response

gradually starts to lead the actual response, which is undesirable.

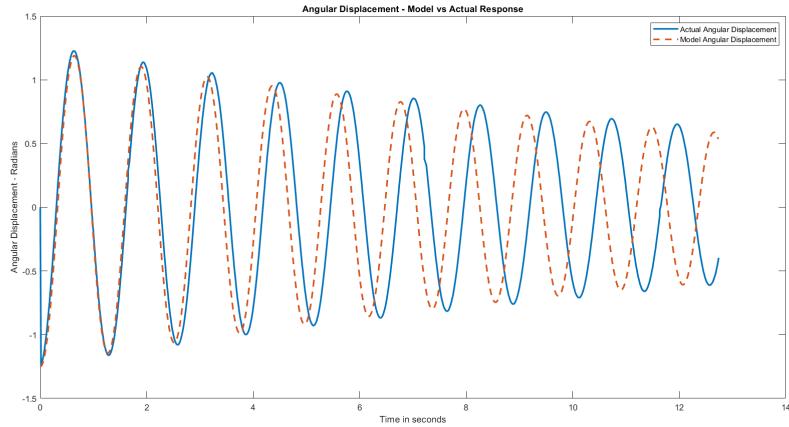


Figure 3.16: Angular Displacement of Pendulum – Model vs. Actual – $\theta = -71.66^\circ$

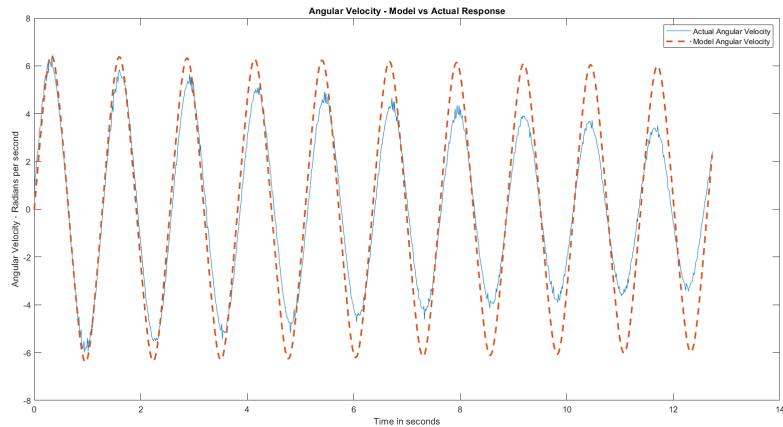


Figure 3.17: Angular Velocity of Pendulum – Model vs. Actual – $\theta = -71.66^\circ$

Figures 3.18 and 3.19 show the error magnitudes between the actual and the modelled response for the pendulum's angular displacement and velocity values. It can be easily observed that as time progresses, the error in both the figures starts to increase gradually, giving rise to a lead/lag problem in Figures 3.16 and 3.17.

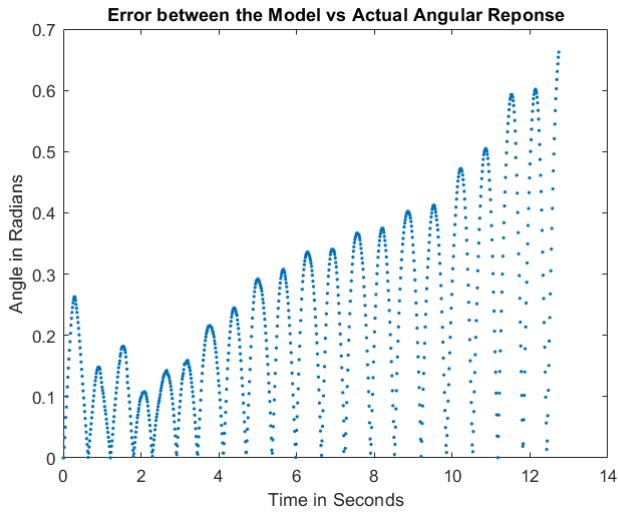


Figure 3.18: Error in Angular Displacement of Pendulum - Model vs. Actual

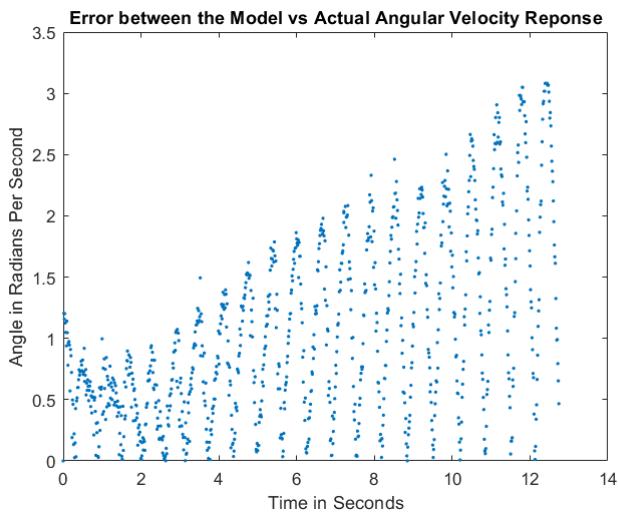


Figure 3.19: Error in Angular Velocity of Pendulum - Model vs. Actual

Figures 3.20 and 3.21 show the comparison between the modelled and actual response values for the cart's linear displacement and velocity values. It can be observed that in both cases, the two curves follow each other closely and the error is minimal as shown in table 3.10.

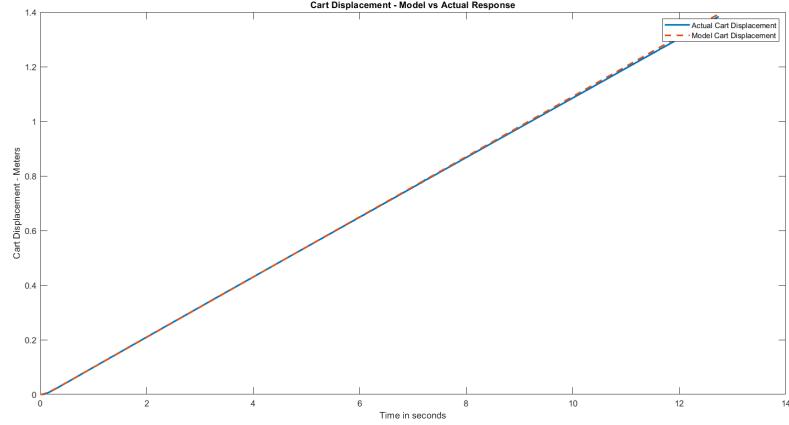


Figure 3.20: Cart Displacement - Model vs. Actual

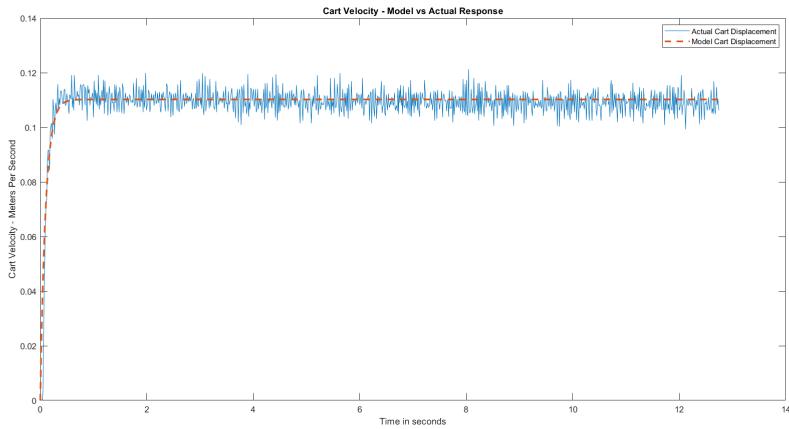


Figure 3.21: Cart Velocity - Model vs. Actual

Table 3.10: Mean Error Values for Cart - Model vs. Actual

Parameter	Value
Displacement Error	0.0410 m
Velocity Error	0.0066 ms^{-1}

Hence, the authors observed that there still existed some discrepancy between the actual response and model response of the plant. This was understood to be likely because the plant's model did not sufficiently model other sources of damping in the system, such as the damping due to the turning point created about the cart and belt-pulley mechanism. However, the authors believe that this model response was sufficient

to design a controller around it, with the expectation that the controller would cater for the inconsistencies between the actual plant and the modelled plant.

3.6 System Control

After developing the motor and plant models, the complete system was simulated and a controller was designed around it. The purpose being to identify limiting variables in the system which would help improve the design of the physical system and allow the authors to finalize its specifications, such as the mass and length of the pendulum, or the type of motor that would be required.

MATLAB was used to simulate control by initializing the system's physical parameters such as the pendulum's mass, length, inertia etc, along with state vector of the system's initial state. Furthermore, the state space matrices developed in section 4.3.5 were also initialized. Then the resulting state space system was checked that whether it was controllable by observing the state space system's rank. Once controllability was verified, an LQR controller was designed using MATLAB's built-in commands and passed to the Simulink workspace. The LQR parameters were obtained as shown in table 3.11.

Table 3.11: LQR Controller Parameters

Coefficientr	Value
K_1	16.0464
K_2	2.8593
K_3	-0.0100
K_4	-0.1346

The following Simulink model was utilized to simulate control of the overall system:

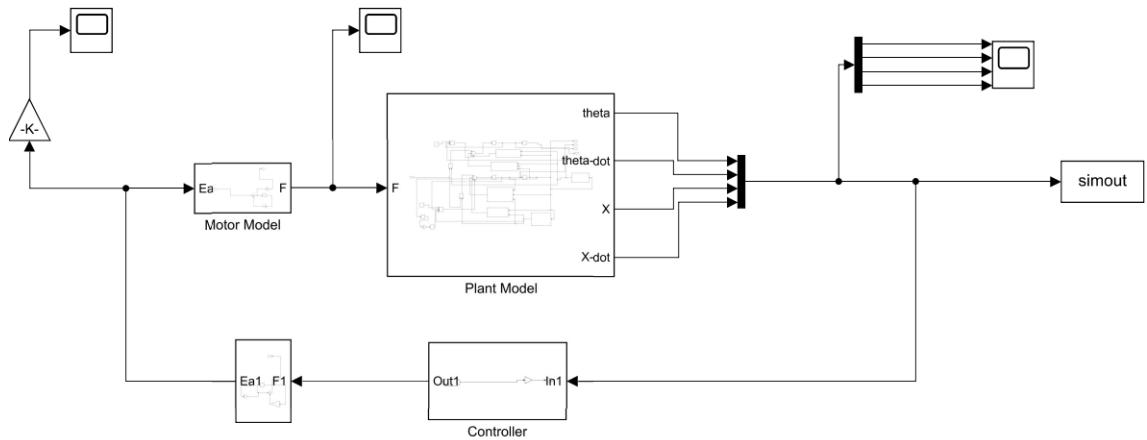


Figure 3.22: Simulink System Model

The following response graphs were obtained upon simulating the designed controller on the aforementioned system:

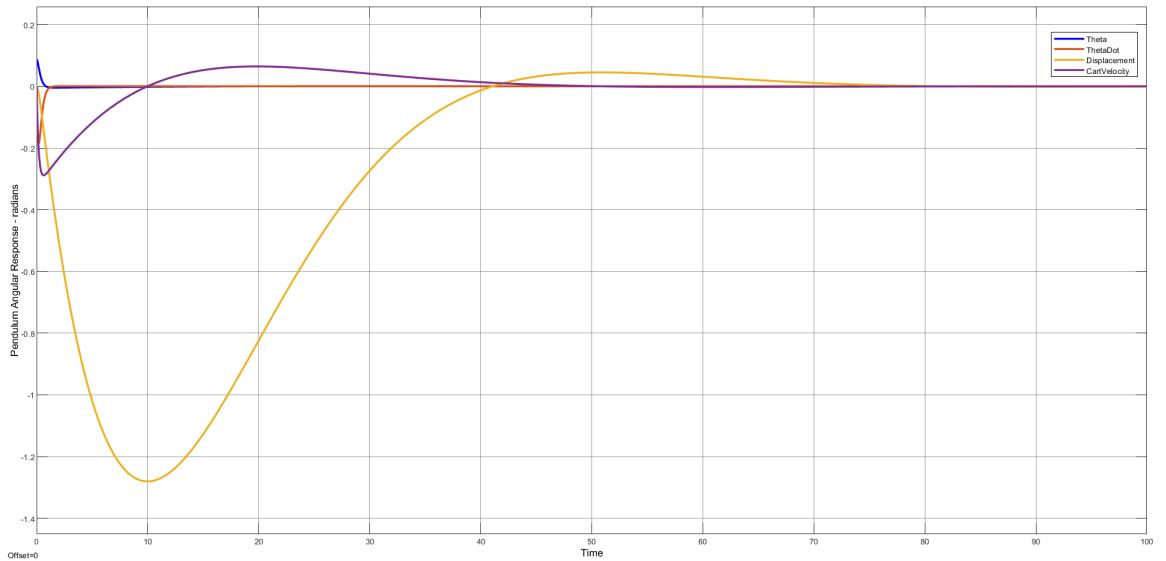


Figure 3.23: Controlled Response - $\theta = 5^\circ$

Figure 3.23 shows that the pendulum retrieves the set point from an initial deviation of 5° successfully. The response for the pendulum's angular displacement can be seen more clearly in Figure 3.24.

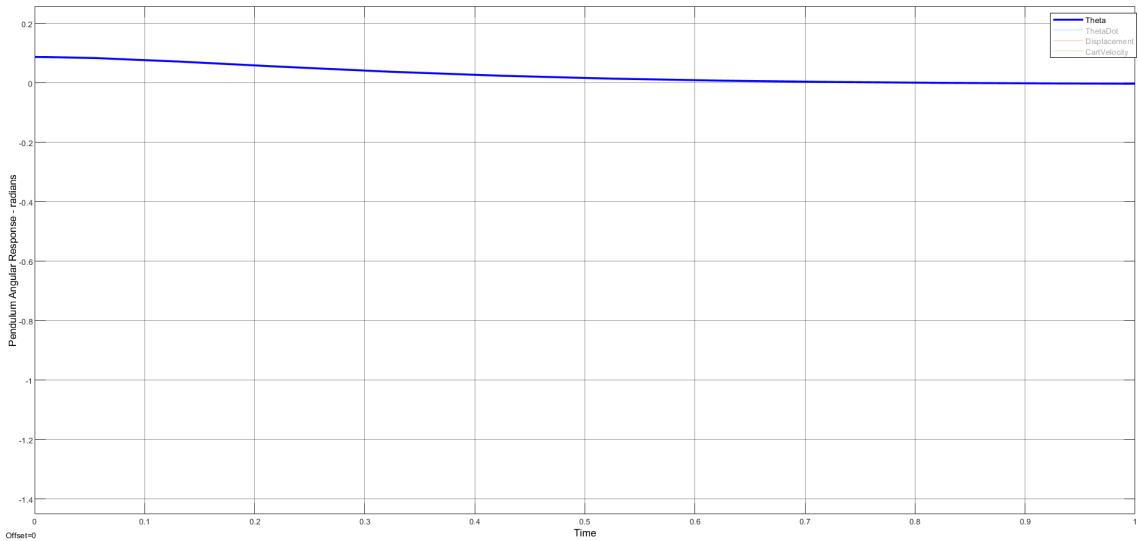


Figure 3.24: Angular Response - $\theta = 5^\circ$

The characteristic parameters of the response shown in Figures 3.23 and 3.24 can be found in table 3.12.

Table 3.12: System Response Characteristic Parameters

Parameter	Value
Pendulum Angular Response Settling Time	0.65 seconds
Pendulum Velocity Peak Value	0.1844 rad/s
Pendulum Velocity Settling Time	0.8 seconds
Cart Displacement Peak Value	1.2 m
Cart Displacement Settling Time	52 seconds
Cart Velocity Peak Value	0.2886 m/s
Cart Velocity Settling Time	45.7 seconds

Since the fact that the system was controllable has been established, the next task was to determine whether the control effort required in the simulation could be provided in the real world. Figure 3.25 shows us the motor voltage required to achieve the desired response, which can be seen to be about 34V.

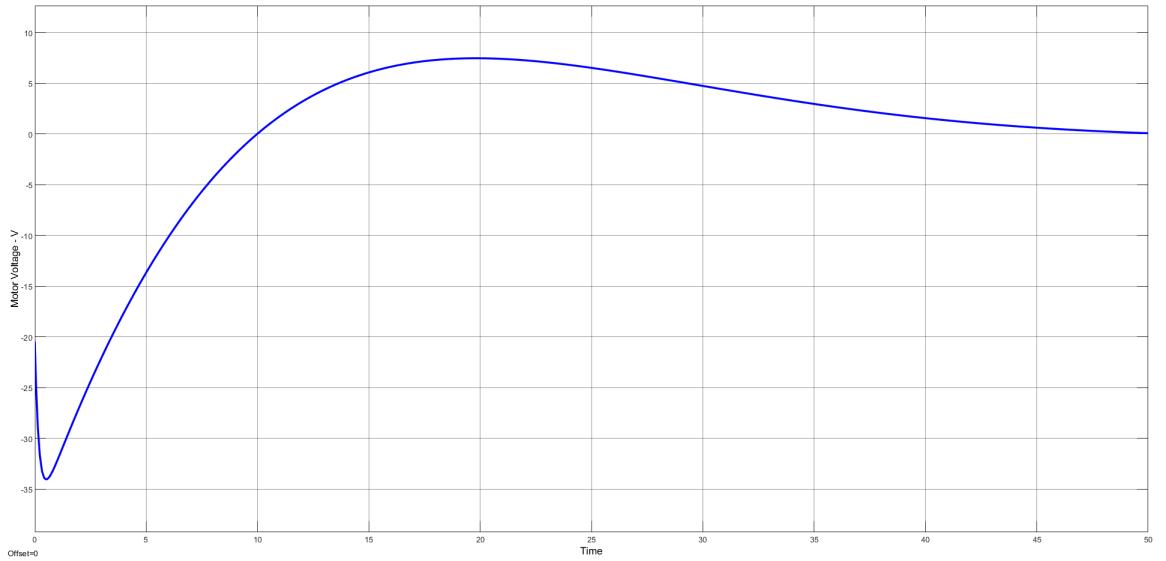


Figure 3.25: Motor Voltage Required - $\theta = 5^\circ$

The required motor control effort characteristics extracted from Figure 3.25 can be seen in table 3.13. The values of these parameters well exceed the capabilities of the motors readily available to us at the time of working on this project.

Table 3.13: Required Motor Voltage Characteristic Parameters

Parameter	Value
Absolute Peak Value	34.05 V
Settling Time	43 seconds

Figure 3.26 and table 3.14 show the magnitude of the input force required to control the system to be $1.4N$. This translates to a load torque magnitude of $0.0084Nm$.

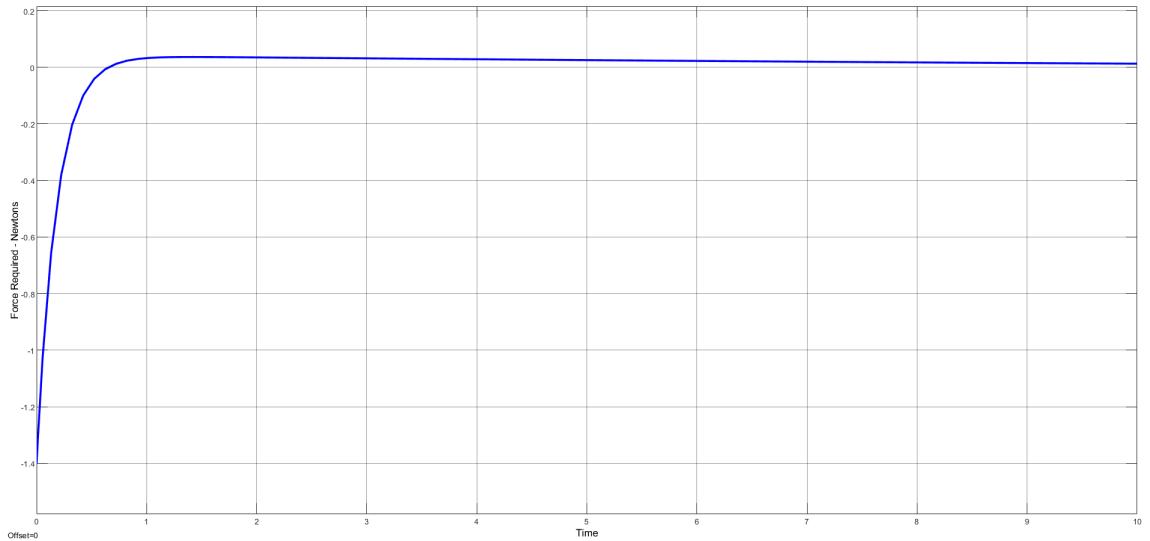


Figure 3.26: Input Force Required - $\theta = 5^\circ$

Table 3.14: Required Input Force Characteristic Parameters

Parameter	Value
Absolute Peak Value	1.4 N
Settling Time	14.4 seconds

It can be seen from Figure 3.27 and table 3.15 that the angular acceleration required to control the system is 184rad/s^2 . This in turn corresponds to an acceleration torque of 0.9195Nm .

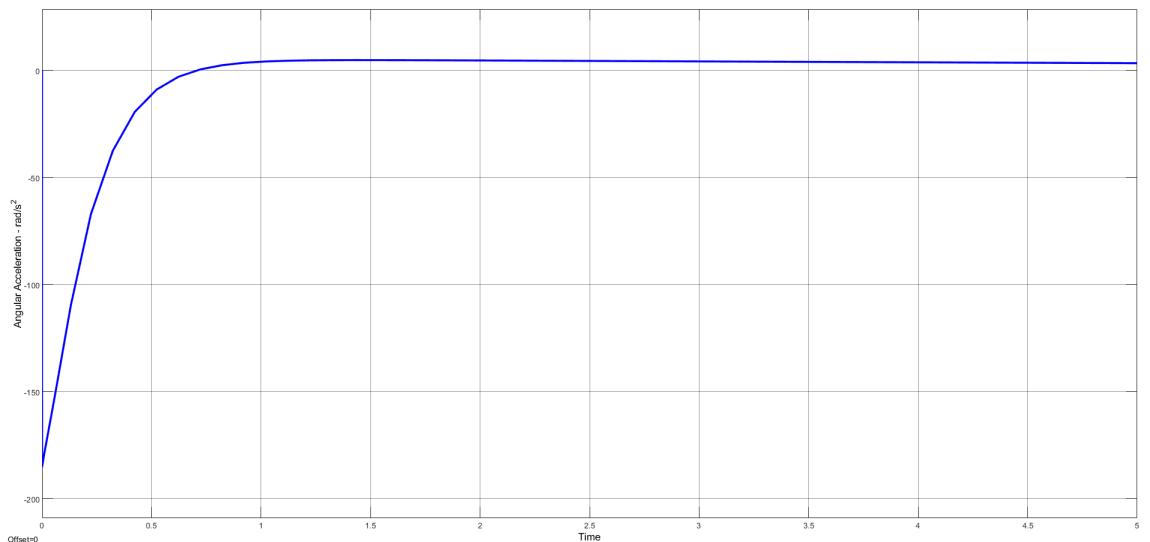


Figure 3.27: Angular Acceleration Required - $\theta = 5^\circ$

Table 3.15: Required Angular Acceleration Characteristic Parameters

Parameter	Value
Absolute Peak Value	183.9 rad/s ²
Settling Time	4.4 seconds

$$\tau = \tau_{load} + \tau_{acceleration} \quad (3.51)$$

It is apparent from 3.51 that the total torque required to control the system comes out to be $0.928Nm$. This is far beyond the ability of the motor which is rated at a torque of $0.3Nm$.

It was observed that the controller developed by MATLAB was in-fact quite aggressive in terms of its control characteristics. Since an LQR controller is an optimal controller by nature of its design, the aggressive behaviour is expected. However, it is also worth noting that the control effort required by this controller in term of motor voltage and motor torque well exceed beyond $12V$ and $0.3Nm$, which are the maximum rated voltage and torque at which the motor can operate, respectively.

Hence, the carried out simulations reveal that the system needs to incorporate a stronger motor in the physical system to make control possible. Another possible change is to vary the system's physical parameters such as the mass of the pendulum and/or the length of pendulum, so that a smaller control effort is required and control can be achieved.

CHAPTER 4

DESIGN DETAILS

4.1 Solution Statement

The cart-mounted inverted pendulum system we have designed is ideally suited for both academic and research purposes. It stands out amongst its competitors on account of it being cost-effective, scalable, and modular. The test-bench is useful because it provides students with opportunities to explore diverse aspects of control system design, and also to explore the challenges in vision-based real-time control of inverted pendulum systems.

4.2 Solution Overview

This section discusses the design overview of all system components in meticulous details. The system was designed after careful thought and consideration of the design alternatives discussed in chapter 2 and simulation results obtained in chapter 3.

Figure 4.1 shows the system block diagram displaying the relationship between various system modules. The system can be broken down into the power module, the actuator module, the controller, the plant module, the sensing module, the CPU, the DAQ module, the calibration module, and the user-end module.

The power module here consists of a power supply that is providing the requisite power to the system. The actuator module consists of the motor driver energizing the motor which is driving the belt and pulley mechanism which is controlling the cart position. The sensing module relies on encoders as well as a camera for input. The CPU performs the image processing function to extract displacement values. Then, there is the DAQ module which is extracting sensor values and communicating them to the controller. Finally, the user-end module consists of the firmware linking the microcontrollers to the computer, as well as a graphical user interface allowing the user to interact with the system. The dotted arrow between the firmware and the controller shows a soft connection through which the controller may be programmed, but not a hard connection that exists when the system is operating.

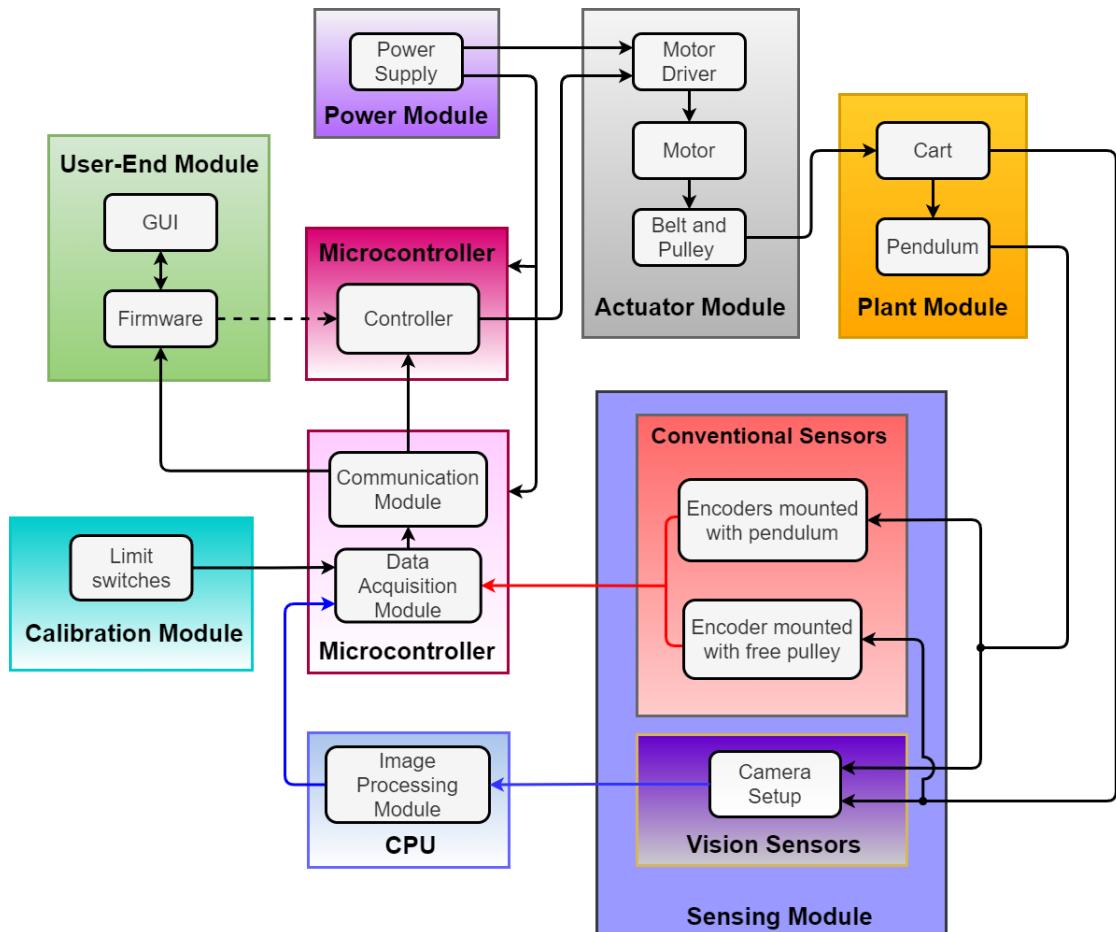


Figure 4.1: System Block Diagram

4.3 System Details

4.3.1 Power Module

The power module consists of power supplies energizing the VNH5019 motor driver, the micro-controllers and the optocoupler circuitry. Two possible power supply configurations are compatible with the circuit.

The first configuration utilizes the GW-insteek GPS-3303 power supply available in the labs at Habib University as shown in Figure 2.12. This power supply can provide a fixed power supply of 5V/1A as well as two variable power supplies of upto 30V/3A. For the specified setup, a 12V/6A connection is required for the motor driver, a 5V/0.4A connection is required for the two Arduinos, and a 5V/0.1A connection for the optocouplers. The current requirement for each Arduino Mega2560 is 0.2A, and as they are connected in parallel so the maximum total requirement is 0.4A. This was the only method initially used to power the system when the circuit was being implemented on a breadboard.

The other possible way to energize the circuit is by using two AC-DC adapters connected to an external AC connection. The first AC-DC adapter supplies a 12V/6A line to the motor driver. A branch is drawn out from the line supplying the motor driver for the optocoupler circuitry. The optocouplers are energized at a rating of 5V/1A using the LM7805 voltage regulator IC. The second AC-DC adapter is used for energizing the microcontrollers in the system together with another LM7805 voltage regulator IC, and supplies power at a rating of 5V/1A.

It is also important to note that the maximum rated value of the input voltage for the LM7805 voltage regulator IC is 35V while the range of acceptable voltage values for the motor driver is 5.5 – 24V at a continuous current rating of 12A (30A maximum). Therefore, the maximum allowable rating for the AC-DC adapter being used

to energize the motor driver and the optocouplers is 24V/12A for our circuit, and this is the maximum power supply that can be used with a more demanding motor for the current motor driver and system configuration. This method of supplying the power using AC-DC adapters may be more suitable in certain circumstances, and was added when the circuit was moved to a Veroboard. The method of energizing circuitry using the lab power supply remains valid as well. Figure 4.2 shows a visual summary of the relationships within the power module as discussed above.

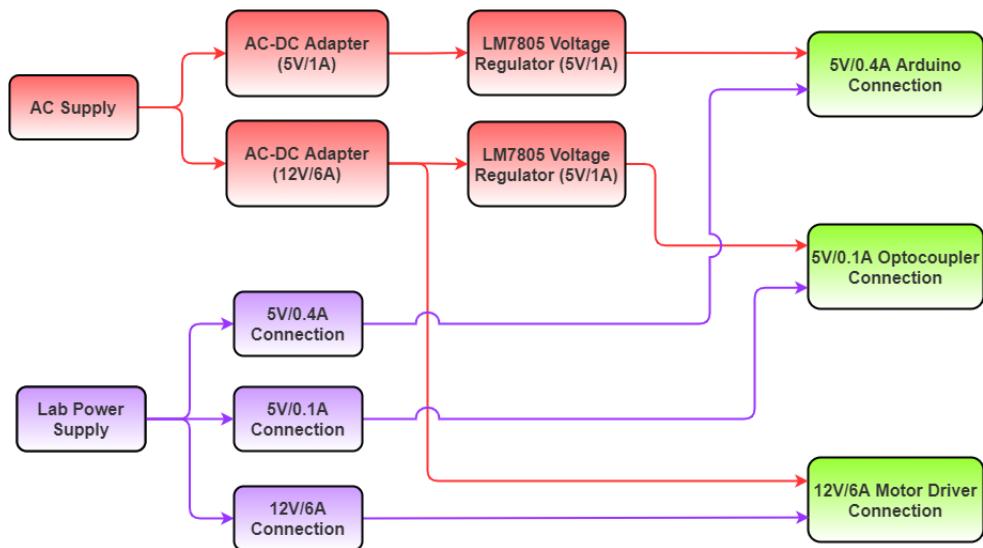


Figure 4.2: Block Diagram representing the Power Module

4.3.2 Actuator Module

Figure 4.3 shows the key relationships between different elements of the actuator module.

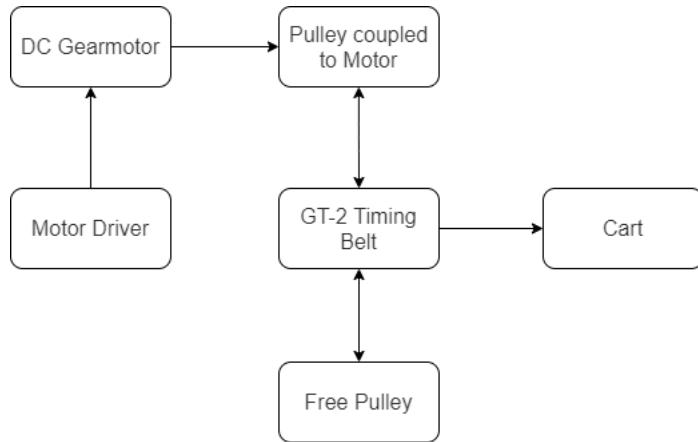


Figure 4.3: Block Diagram representing the Actuator Module

4.3.2.1 *Motor and Motor Driver*

The VNH5019 motor driver as shown in Figure 2.13 is used to drive the DC motor in the setup. The motor driver is connected in the configuration where the driver and the micro-controller connected to it are powered separately. In this configuration, the motor driver can operate in a voltage range of 5.5 – 24V.

The motor driver is isolated from the microcontroller Arduino Mega2560 using an optocoupler circuit. The optocoupler IC used for this purpose is the PC817 chip. Five such chips are used in the connections between the Arduino Mega2560 and the motor driver. The PC817 ICs are powered using a 5V supply isolated from the 5V supply provided to the Arduino Mega2560. This is done to ensure isolation between the grounds of the circuits on either side of the optocouplers.

The motor used in the system is a Pololu 37D Metal DC GearMotor. The motor is rated for operation at 12V with a no-load current of 0.2A and a stall current of 5.5A. The motor has a gear ratio of 70 : 1, a no-load speed of 150 RPM, and an extrapolated stall torque of $27 \text{ kg} \cdot \text{cm}$.

4.3.2.2 Locomotion Mechanism

The locomotive mechanism consists of a belt and pulley system. A pulley is mounted on to a DC motor and is attached to one end of a loop of GT-2 timing belt. There is a free pulley attached to the other end of the timing belt loop. The timing belt is clamped tightly to the cart using 3D printed clamps as shown in Figure 2.8. The GT-2 timing belt used is about 3.6m long, with a 2mm belt pitch and 6mm belt width. The two pulleys used alongside it have an inner diameter of 8mm and 16 teeth each.

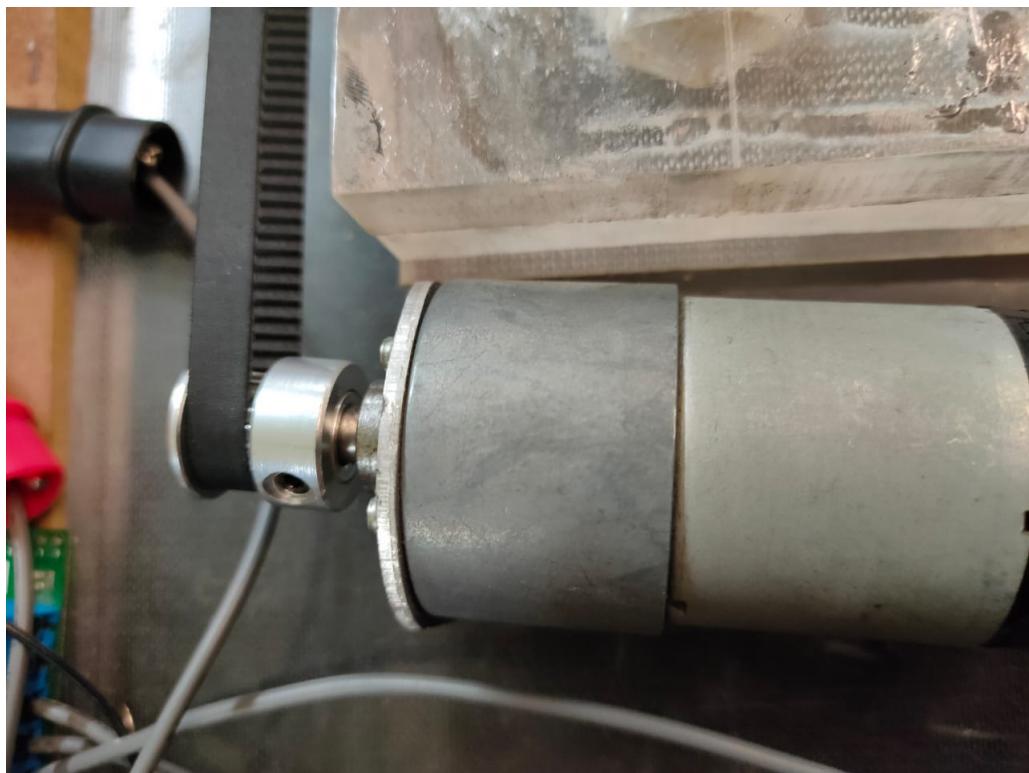


Figure 4.4: DC Gearmotor coupled to the Timing Belt using a Pulley

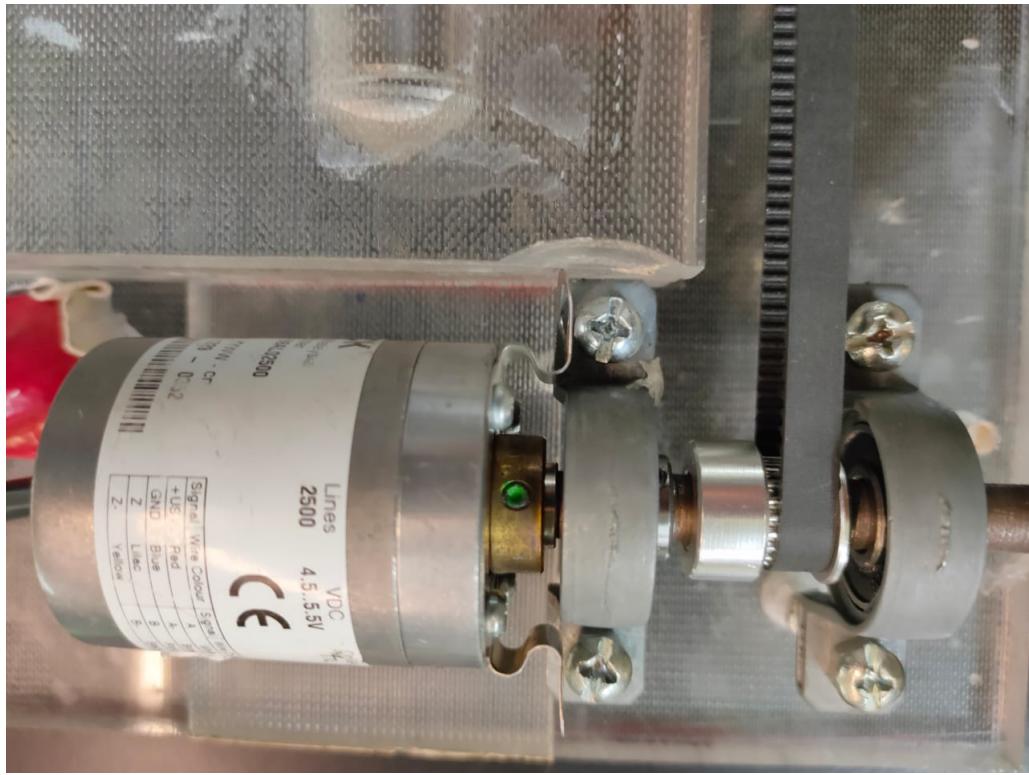


Figure 4.5: Free Pulley coupled to the Timing Belt and also to a Rotary Encoder

Two pipes are used to design the track on which the cart must move. Two pipes are selected because two anchor points ensure that there will be no lateral movement or rotation of the cart about the track. Hollow stainless steel pipes are selected for this purpose since they afford greater tensile strength and have minimal sag over long lengths. The pipes used are 2m long each with an outer diameter of 20mm. The maximum allowable track length is 1.81m which is defined as the length between the limit switches at either end of the track.

The cart is designed as a rectangular piece of acrylic of dimensions $18\text{cm} \times 12\text{cm}$. Two linear bearings are attached to the bottom of the cart using U-Clamps screwed on to the cart. Rubber is inserted in between the U-Clamps and the linear bearings to ensure that there is a tight grip between the cart and the bearings. The linear bearings have an internal diameter of 20mm which matches the outside diameter of the hollow steel pipes making up the track. This ensures that the cart can slide smoothly along the length of the track without any lateral movement. Upon looking closely at Figure 4.6,

it can be observed that black rubber has been inserted between the U-Clamp and the linear bearing in order to provide a tight fit.

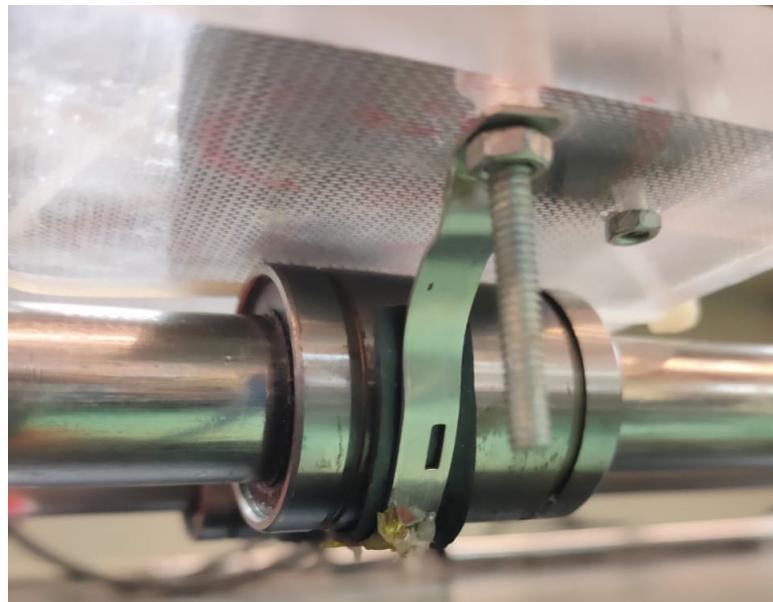


Figure 4.6: A Linear Bearing coupled to the Cart via a U-Clamp

4.3.3 Sensing Module

Figure 4.7 shows the key relationships between different elements of the sensing module. The purple arrows track the flow of information from the plant to the DAQ via incremental encoders while the red arrows track the flow of information via the camera setup.

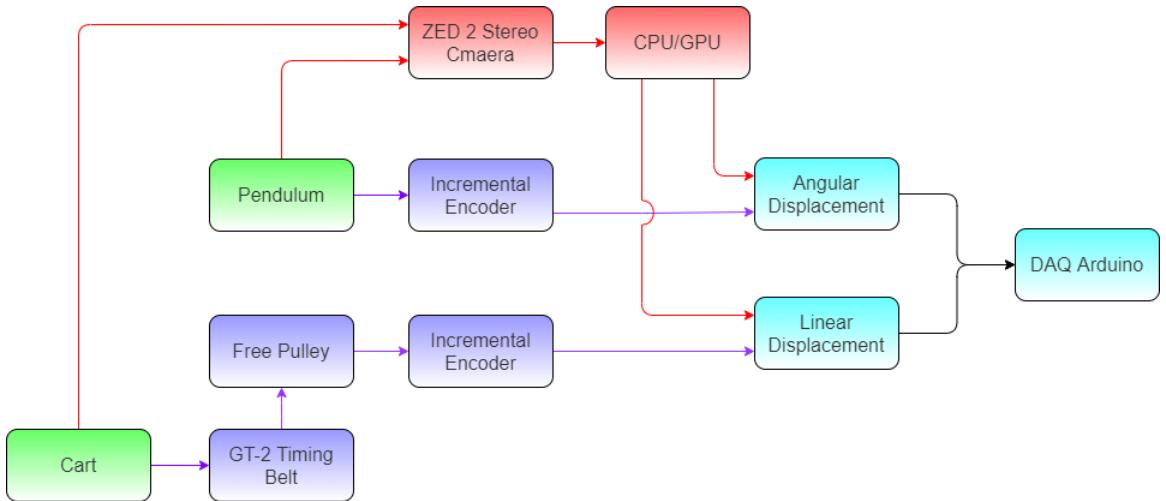


Figure 4.7: Block Diagram representing the Sensing Module

4.3.3.1 Conventional Encoder Feedback

Two incremental rotary encoders are used to generate feedback regarding the plant for use by the controller. One encoder is coupled to the free pulley at the far end of the setup to measure the linear displacement of the cart. This encoder can be observed in Figure 4.5. Another encoder is coupled to the freely rotating pendulum to measure its angular displacement values. This encoder can be observed in Figure 4.8 below.



Figure 4.8: The Pendulum coupled to the Cart via a Rotary Encoder

The encoders used are the SICK DBS36 Core incremental rotary encoders. They

are both identical and are rated at 2500 pulses per revolution. They each require a power connection of $5V/0.1A$ to operate and are powered using the DAQ Arduino.

It is important to note that both signals A and B generated by the encoder are processed by the DAQ module, and the effective resolution comes out as 10000 pulses per revolution using 4-bit encoding. The pulse count can be used to obtain the angular displacement as shown in (4.1).

$$angle = \frac{count}{resolution} \times 360^\circ, \quad (4.1)$$

where resolution = 10000 PPR. Refer to subsection 5.3.1 for further detail about encoder resolution and operation.

Further processing is required to obtain the linear displacement. It is known that the outer diameter of the pulleys used is $12mm$. So the angular displacement of the pulleys and associated encoder can be converted to linear displacement of the cart using the formula shown in (4.2).

$$distance = angle \times \frac{\pi}{180^\circ} \times radius, \quad (4.2)$$

where radius = $6mm$.

4.3.3.2 Vision Feedback

The ZED 2 stereo camera is used to obtain visual feedback of the system which is then processed using image processing and computer vision algorithms to track the pendulum and the cart. The bob at the end of the pendulum and the cart center were painted blue in order to allow for better recognition and tracking. The tracking algorithms are implemented on Python using the OpenCV library.

The ZED 2 camera is used in the setup at a resolution of $1280 \times 720p$ at 60 frames per second. It is also particularly suited for the purpose since it offered a 120° wide angle view which meant that it did not have to be placed a large distance away to encompass the entire platform in its field of view. In fact it was placed about $1.8m$ away from the platform when operating. Figure 4.9 below shows the ZED 2 camera placed on the desk towards the right while the test-bench is mounted on the desk towards the left.



Figure 4.9: The ZED 2 camera set up to observe the Test-bench

The ZED 2 camera is connected via USB port to a CPU and/or GPU which processes each frame obtained in real-time, calculates the angular and linear displacements, and serial writes them to the DAQ Arduino for transmission to the controller.

4.3.4 Microcontrollers

Two microcontrollers are being used in the system. They are both the Arduino Mega2560. The reason for using the Arduino Mega instead of the Arduino Uno is that the implementation required multiple interrupt pins which is a facility offered by the Arduino Mega. Out of the two Arduinos used, one primarily performs the task of a DAQ i.e. it is a data acquisition module while the other primarily performs the task of a controller.

The DAQ Arduino receives sensor data from the limit switches, and the encoders via interrupt pins. It also receives the processed image data from the CPU via serial communication. The required data pre-processing is completed on this device. It then

sends the data at a pre-specified baud rate to the other Arduino using the I2C communication protocol.

The controller Arduino constantly receives processed data from the DAQ Arduino. It can sample the data being received at different sampling rates as per requirement. Timers are used for sampling to ensure consistent time measurements, which can be disturbed by triggered interrupts if on-board timers aren't used. The controller is implemented on this Arduino, and it utilizes the angular and linear displacement input values to generate a corresponding output motor speed which is communicated to the DC gear-motor through the motor driver, in an attempt to achieve stabilization of the pendulum about the inverted position.

Figure 4.10 shows the flow of data from the sensors to the controller.

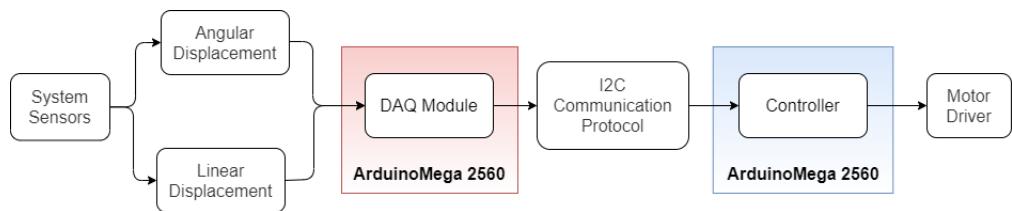


Figure 4.10: Block Diagram showing the relationship between the DAQ Module and the Controller

4.3.5 Calibration Module

The calibration module consists of two limit switches as shown in Figure 2.10. These are mechanical switches that are attached to the pipe on either end of the track using hose clamps. The limit switches are connected to the microcontroller that is responsible for controlling the system. They are configured as interrupts which means that as soon as the cart hits either end of the track i.e. either limit switch, an interrupt will be generated which will cause the motor to brake immediately, and prevent the cart from crashing into the blocks at either edge of the track. They also serve the added functionality of allowing the system to calibrate the track length. If a particular option is selected in the code, then upon initialization the cart will first move left until it strikes

a limit switch, reverse direction and move right till it strikes a limit switch again, and will then return to the center of the track. Hitting both the limit switches enables the system to record the allowable track length between them. This can add an additional constraint to the problem whereby the control algorithm actively tries to ensure that this track length is not exceeded. This can also allow the system to come to a 'smart-stop' (if the option is enabled) when the cart is about to hit the limit switch at the end of the track during normal operation, and the system needs to shut down before actually triggering the limit switch itself, to prevent any potential wear & tear.



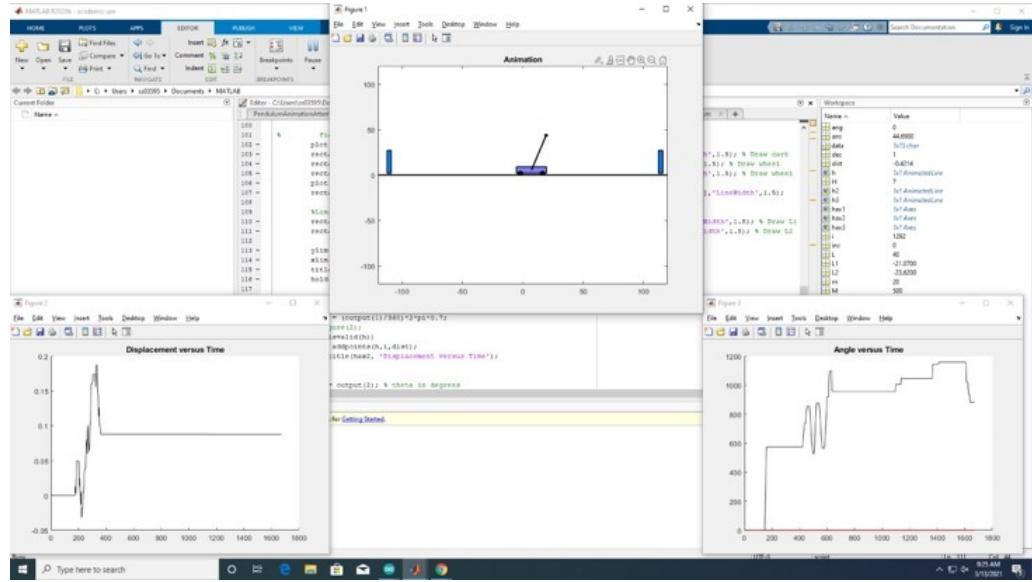
Figure 4.11: A Limit Switch coupled to the Pipe with a Hose Clamp

4.3.6 User-End Module

The user-end module has been implemented using MATLAB, Arduino IDE and Python on a computer. It can communicate information about the system to the user on an active basis.

The information regarding the pendulum angular displacement and the cart linear displacement values are written to the serial monitor by the Arduino Mega2560 per-

forming the function of a DAQ module and containing a communication module. The data on the serial monitor can be read by a MATLAB program which can then display real-time graphs of angular displacement versus time, linear displacement versus time, and an animation showcasing the movement of the pendulum, cart, and the location of the limit switches. This is shown in Figure 4.12 below.



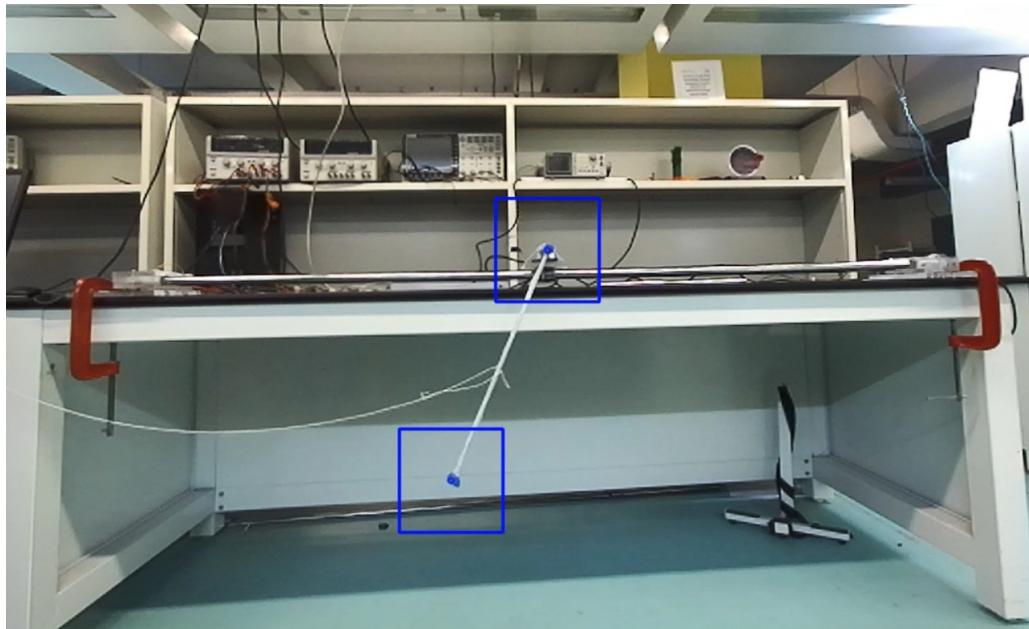


Figure 4.13: Real-time visual feedback obtained by ZED 2 camera.

There are two blue boxes in Figure 4.13. The upper blue box is centered around the cart, while the lower blue box is centered around the bob at the end of the pendulum. The boxes move in real time as the pendulum/cart move to confirm that the algorithm is successfully tracking the pendulum and the cart center. Refer to appendix K.4 for video demo of the algorithm.

4.4 Hardware Specifications

The specifications of the system hardware are shown in table 4.1 below. The track effective length is defined as the length between the two limit switches at either end of the track. The encoder effective resolution is four times the true resolution since it has been quadrupled using software pre-processing as discussed in the section 4.3.3.

Table 4.1: Test-bench Hardware Specifications

Parameter	Value
Track Total Length	1.81 m
Track Effective Length	1.48 m
Cart Mass	0.7087 kg
Pendulum Length	0.517 m
Pendulum Mass	0.08514 kg
Pendulum Bob Mass	0.05125 kg
Motor Voltage	12 V
Max. Motor Current	5.5 A
Max. Motor Speed	150 RPM
Gear Ratio	70
Encoder True Resolution	2500 PPR
Encoder Effective Resolution	10000 PPR

4.5 Addendum

Extensive electrical connections exist between the external power supply, the microcontrollers, the motor driver, the optocouplers, and more components in the system. These are best represented through a circuit diagram and can be found in appendix D.

The construction of the hardware for the project cost around PKR 27,000 in total. The complete bill of material can be found in appendix H.

CHAPTER 5

PROTOTYPING AND TESTING

5.1 The Constructed Prototype

The hardware prototype for the test-bench was meticulously constructed meeting all the requirements outlined in the start of the project i.e. modularity, cost-effectiveness, scalability etc. The complete prototype can be viewed in Figure 5.1.



Figure 5.1: The Test-Bench Prototype

Figure 5.2 below shows another view of the test-bench, where the camera observing the platform is also visible.



Figure 5.2: Side View of the Test-Bench

Further close-ups of different system components can be viewed in the different sections within Chapter 3: Design Details as well as in Appendix C.

5.2 Control Using Encoders

This sections details the efforts by the authors to achieve control over the pendulum about its inverted state. The results shown in the following sections are produced by a PID controller with the parameter values shown in table 5.1.

Table 5.1: Gain value for a PID controller

Parameter	Value
Proportional Gain (K_p)	30.04
Derivative Gain (K_d)	0.0452
Integral Gain (K_i)	63.9

5.2.1 System Response When Being Controlled

5.2.1.1 Control Successfully Achieved

The graph in Figure 5.3 shows the only recorded case where the pendulum was able to withstand a disturbance of a very tiny magnitude and regain its inverted state.

Figure 5.3 shows the pendulum stabilized at about -0.1° initially. Then a disturbance applied makes the pendulum swing in the anti-clockwise direction reaching a maximum value of about 0.55° . The cart starts to move almost immediately as the pendulum angular displacement value crosses zero towards the left causing the pendulum to reverse its direction and it stops soon as the pendulum obtains the mean position again.

We can calculate some key parameters relevant to control problems from the graph in Figure 5.3. The time on the x-axis is measured such that each unit of time equals 5ms. Rise time or t_r is defined as the time taken for the response value to go from 10% to 90% of the maximum magnitude. Settling time or t_s is defined as the time taken for

the system to be bounded within 95% of the mean position again, and maximum overshoot is the maximum magnitude of the response once it surpasses to the mean position.

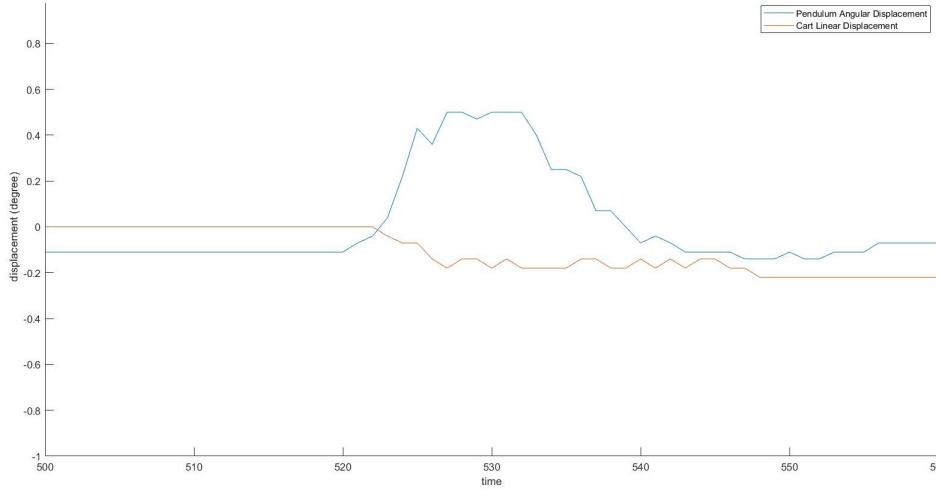


Figure 5.3: System Response when successfully controlled

We can evaluate the performance of the prototype platform using the closed loop system characteristic values as key performance indicators (KPIs). For this particular case, we obtain the characteristic set of values of the closed loop system as shown in table 5.2.

Table 5.2: System Characteristics of the Closed Loop System

Parameter	Value
Rise Time (t_r)	21ms
Settling Time (t_s)	110ms
Maximum Overshoot	0.5 °

5.2.1.2 Control Unsuccessful

Figure 5.4-5.6 show the system response graphs and their close-ups for one particular instance where control could not be achieved. Graphs for only one such instance are shown even though similar results were observed in almost all the test cases run, excepting the one shown in Figure 5.3.

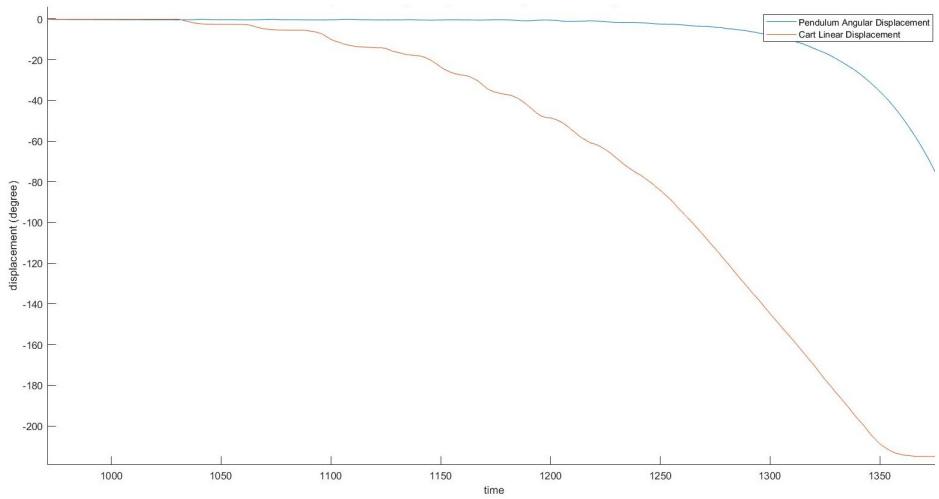


Figure 5.4: System Response when being controlled

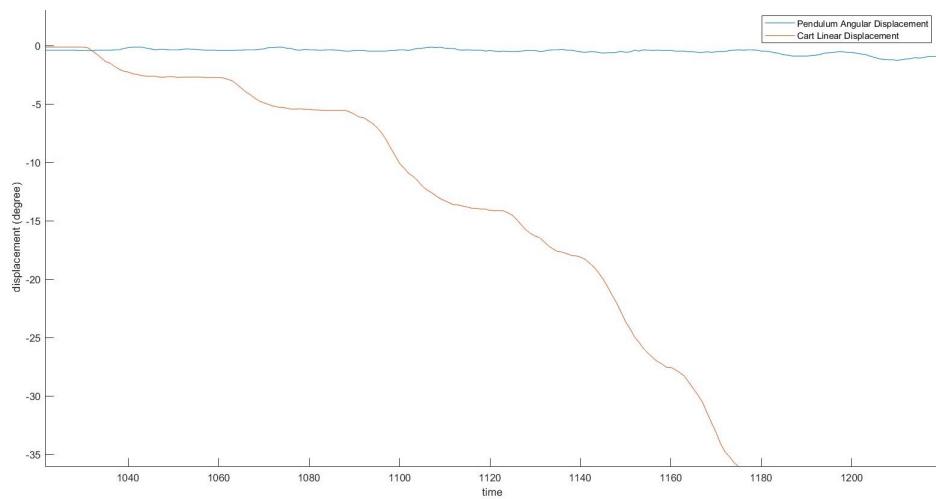


Figure 5.5: System Response when being controlled (Close-Up)

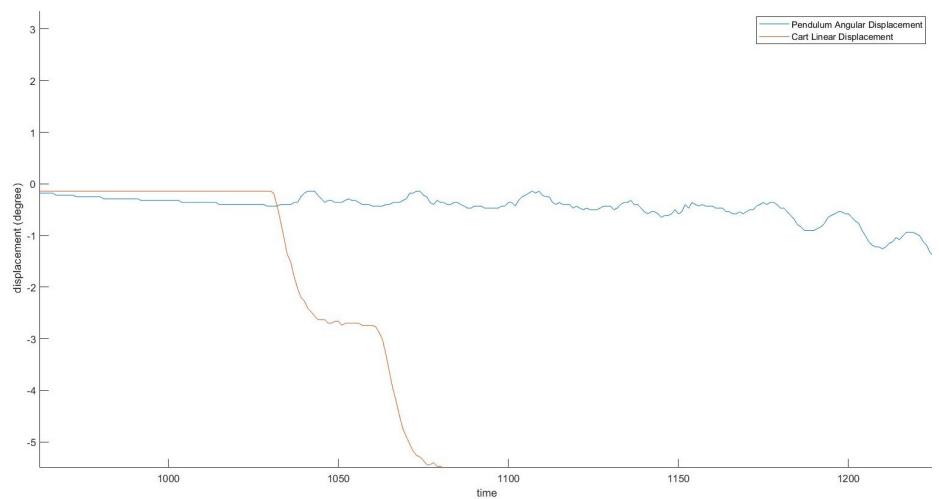


Figure 5.6: System Response when being controlled (Further Close-Up)

It can be observed from the above plots, particularly Figure 5.6, that the cart starts to move as soon as there is a significant change in the pendulum's angular displacement. Once the cart is moving, the angular displacement plot does show sinusoidal like behaviour suggesting that the system is attempting to regain control. However, the cart motion is insufficient to retrieve the set-point. The angular displacement curve does not intersect with the zero-point again, and ultimately deviates altogether from the set-point until any semblance of control is lost, and the pendulum completely loses equilibrium.

5.2.2 Open Loop System Response

The open loop system response was recorded after multiple attempts to control the system robustly had failed. It was hypothesized that the inability to control the pendulum was not on account of sensor latency or algorithmic error, but motor limitation. It was judged that if the open loop test would display the pendulum falling in one direction only, that meant that the motor was insufficient for the purposes of this project.

Figure 5.7 and Figure 5.8 show the open loop response of the system. The code has been configured such that the cart kicks into action soon as the pendulum angular displacement value crosses 0.4° and moves into the requisite direction with maximum speed until the pendulum angular displacement value reaches 30° at which point the cart stops since stabilized position can not possible be regained. The pendulum continues to fall and stabilizes gradually about its mean position.

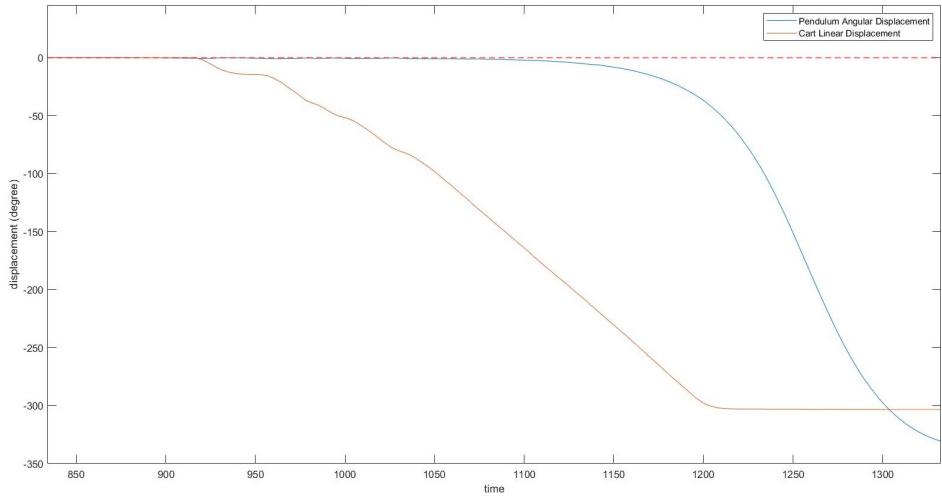


Figure 5.7: Open Loop System Response

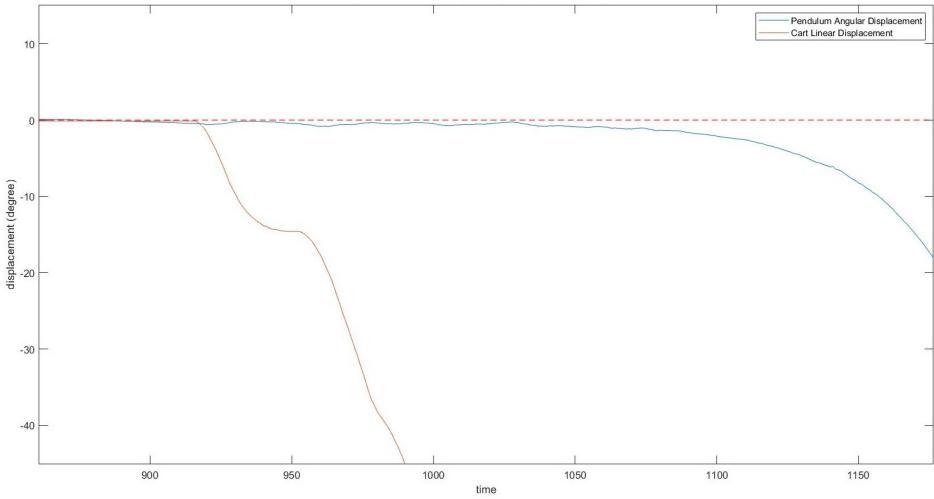


Figure 5.8: Open Loop System Response (close-up)

Figure 5.7-5.8 clearly show that cart movement does not produce a significant impact on the pendulum. It manages to slow down the speed of pendulum falling but does not cause it to change its direction or even maintain the inverted state for a noticeable period of time. Also note that the behaviour shown in Figures 5.4-5.6 is different from the behaviour in Figures 5.7-5.8 since the pendulum's angular displacement curve shows a more sinusoidal response in the former as the cart moves in an attempt to stabilize the pendulum. This is according to expectation as the former graphs demonstrate a controller in action.

5.3 Challenges

5.3.1 Encoder Error

One of the most cumbersome error faced by the authors during the testing phase was a drift in the value of the encoder. This was characterized by a random error that would appear in the angular displacement values of the pendulum when the cart was moving. This error was significant because it caused the controller to lose track of the mean position, and made it impossible to sustain the pendulum in its inverted state. The maximum magnitude of this error was recorded at about 70° when the setup was allowed to run for 5 minutes, in several trials. It is also important to note that the error was only observed when the cart was in motion, which led the authors to conclude that the motion was somehow responsible for the error.

Several steps had to be taken before the error could be resolved. The angular values were plotted with the cart in motion and it was observed that the error in mean position appeared only when switching occurred.

The first hypothesis was that this was caused by back emf generated by the DC motor upon switching. To fix this, an optocoupler circuit connecting the micro-controller Arduino Mega2560 and the motor driver VNH5019 was introduced. This also served the additional purpose of partitioning the circuit and safeguarding components in case of power supply or motor failure. The optocoupler implementation was very successful in reducing the error drift to large values. However, it did not eliminate the error entirely. At this stage the error was measured to reach a maximum of 15° when the system was operational for 5 minutes.

The persistent error that remained and appeared at random was still significant enough to prevent the controller from achieving any reasonable semblance of control. To further diagnose this problem, the angular readings taken from the encoder were

plotted and observed as to how the values varied when the motor switches direction. It is important to remember that an incremental encoder records an angle in the form of up and down counts. Since the pendulum swings in a sinusoidal fashion, the behaviour is represented by a series of alternating up and down counts. A break in that pattern suggests that there is inherent inaccuracy in the measured values of the encoder.

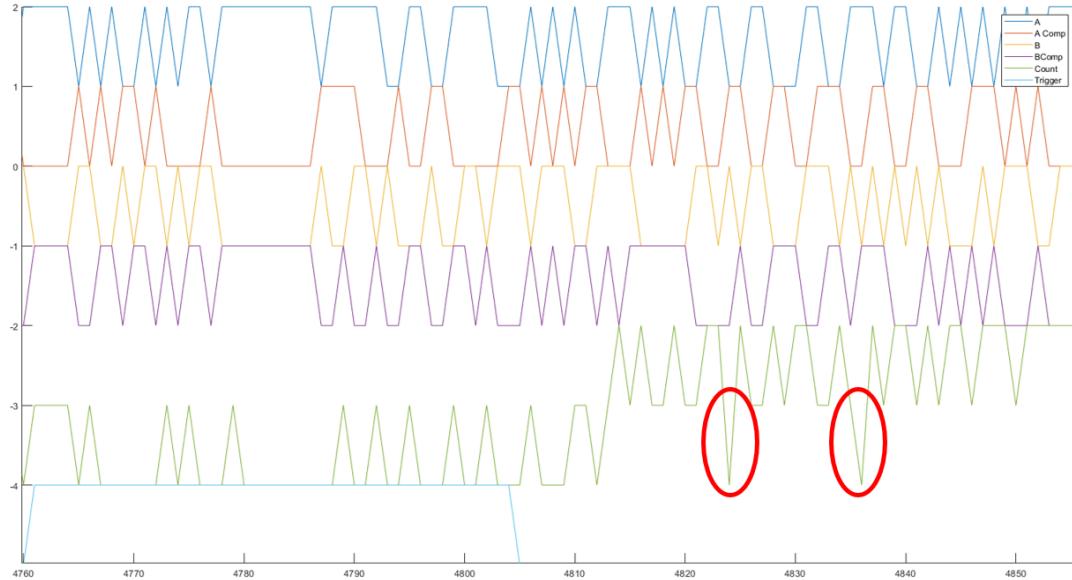


Figure 5.9: Pulse Output by the Encoder

Figure 5.9 shows a snapshot of the pulse output by the encoder at an instance in time. There are six wave-forms shown in the plot. From top to bottom, they are as follows; blue waveform is the signal A, red waveform is the signal A complemented, yellow waveform is the signal B, purple waveform is the signal B complemented, green waveform is the count registered by the algorithm, and the blue waveform represents the trigger. The count registers a positive value when we have an up count i.e. pendulum moving in the anti-clockwise direction and a negative value when we have a down count i.e. pendulum moving in the clockwise direction. The trigger waveform is high at the instance when the cart changes direction and low otherwise.

Upon closely analysing the wave-forms obtained, a few key observations were obtained. There were occasional breaks in the pattern of count emerging which could

not possibly represent the real world motion of the pendulum. Two of these errors are highlighted in Figure 5.9 within red circles. These errors take the form of a stray down count while the algorithm should be up counting i.e. pendulum is moving in the anti-clockwise direction. Another important observation was that these erroneous readings would only appear immediately after the moving cart changed its direction. This is apparent from the trigger waveform in Figure 5.9 because the error values appear just after the trigger value temporarily shows a high, representing a change in direction.

Several readings were taken and the plots obtained were closely analysed. Similar discrepancies were found in the readings when the encoder count values were observed. Thereafter, the authors decided to switch over to 4-bit encoding [45].

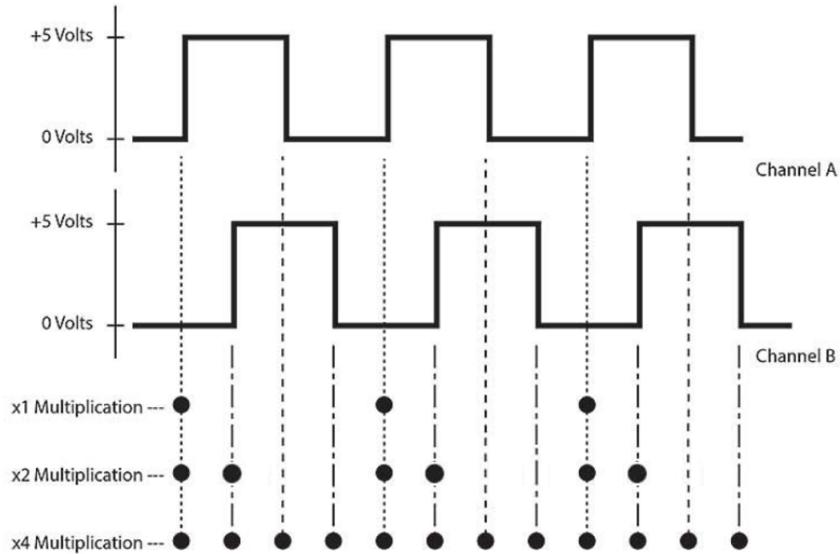


Figure 5.10: Different ways of encoding the Encoder Signal [45]

Figure 5.10 shows the three main ways in which an obtained encoder signal can be encoded and read. The authors decided to switch to the x4 Multiplication or the 4-bit encoding method, because it increased the system resolution four times, to effectively 10,000 PPR [46]. More importantly, it was deduced that the erroneous count values being obtained as shown in Figure 5.9 existed due to breaks in the A and B signals that could not be catered by the x2 Multiplication or 2-bit encoding method which was being used until that point in time. Essentially, 2-bit encoding wasn't sufficient

to remove the noise inherent in the A and B pulse signals being received from the encoder. Once the system was switched over to 4-bit encoding, a drastic change in the encoder performance was noted. The error obtained in the values for the pendulum angular displacement was recorded at a maximum of 0.14° after a continuous operation of about 15 minutes. This was deemed sufficiently accurate for the authors to proceed to the next stage of the project i.e. control.

CHAPTER 6

VISION

6.1 Introduction

This section explores the feasibility of using Computer Vision as a feedback mechanism for the Inverted Pendulum test bench, contributing to the research tangent of this project. It puts forth the algorithms and methodology used to isolate and detect the points of interest on the pendulum in an efficient manner which are then used to calculate the required angles and how to reduce the error of calculated angles using perspective projection model. This is followed by discussion on the performance of the vision system and its limitations, after which the section concludes with the recommended course of action for future work in the domain.

6.2 Methodology and Algorithm

6.2.1 Detection

Visual feedback was obtained by using a ZED 2 camera to obtain real-time video of system operation. The camera was operated at the specifications summarized in table 6.1.

Table 6.1: Camera Settings

Parameter	Value
Resolution	1280x720
FPS	60

An algorithm was implemented using the OpenCV and SDK libraries in order to track the cart center and pendulum bob in real-time, as detailed in Figure 4.13. The algorithms explored for this purpose included the Hough Line Transform, Harris Corner Detector, and the Shi-Tomasi Corner Detector.

The Shi-Tomasi corner detector was the one best suited for detecting the points of interest since it allowed room to specify the exact number of corners that needed to be detected. This was convenient since the geometries of the points that needed to be tracked were fixed and could be specified for the algorithm to track.

Furthermore, the pendulum bob and the coupler (connecting the pendulum to the cart) were painted blue to make them clearly distinguishable from other features in the background. For this purpose a blue color mask of the corresponding HSV value was applied on the image retrieved from the ZED 2 Camera, which helped make the points of interest stand out as distinct. This phenomenon can be seen in Figure 6.1. The blue dot located at the center of the image pinpoints the location of the coupler while the blue dot at the bottom of the image pinpoints the location of the pendulum bob.

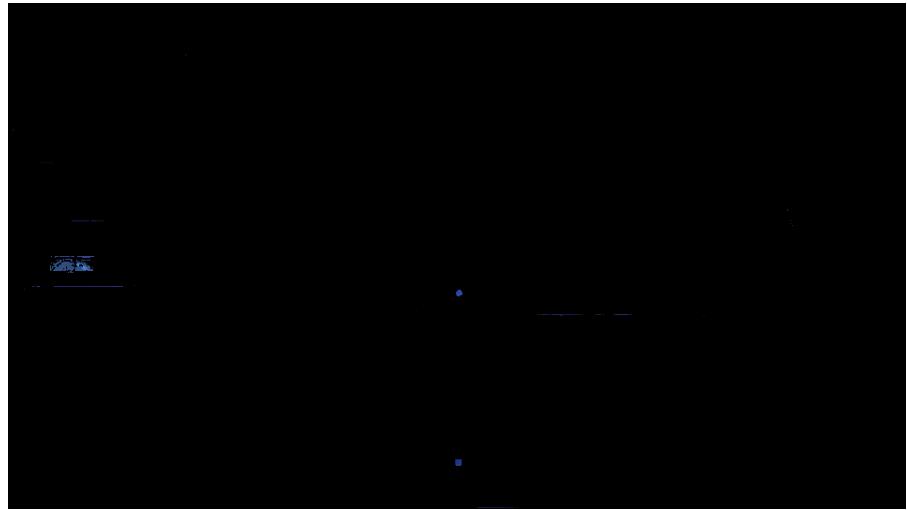


Figure 6.1: Blue mask applied on image retrieved from camera.

6.2.2 Tracking and Localization

A challenging aspect of the vision-problem was to ensure a way for the system to be able to continuously track the points of interest while the pendulum is in motion. The system's ability to do so was constrained by the ability of the processor, the velocity of the pendulum, the image frame size, the FPS (Frames Per Second) of the frame stream, and background noise.

Two constraints that could be reduced successfully were the image frame size and the background noise. After the initial detection, the algorithm created a 100×100 pixel localization region in the image around the the pendulum's bob and the cart's coupler, as represented in Figure 6.2 by the two blue bounding boxes. A secondary region was also generated which was shifted slightly ahead of the first region centered about the bob in the direction of the pendulum's motion. This basically ensured that if the interest points were not found in the first localization region, they could be found in the second region. In the worst case scenario that the pendulum was found in neither region, the algorithm would search the whole image for the points of interest.

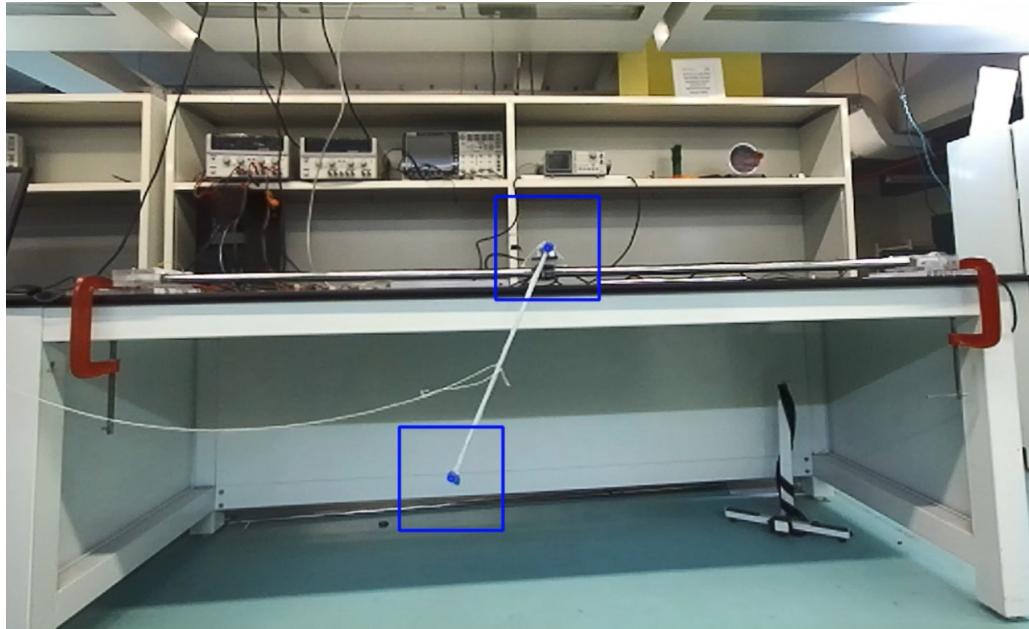


Figure 6.2: Real-time visual feedback obtained by ZED 2 camera.

These improvements visibly improved the robustness of the vision system's tracking ability by reducing the time required to process each frame. The table 6.2 summarizes the effective frame rate which the algorithm was able to process, the resolution it achieved, and the maximum pendulum velocity that the vision algorithm was able to track with these measures in place.

Table 6.2: Vision System Specifications

Parameter	Value
Effective Frame Rate	22
Resolution	2.72°
Maximum Pendulum Velocity	110° per second

The resolution here signifies the average step size between consecutive readings obtained from the visual feedback. This is currently quite large since it is about 3°. However, note that the effective frame rate is quite small as well. The frame rate and the resolution have an inverse relationship. A frame rate of 100 FPS approximately would correspond to a visual feedback resolution of about 0.5°. The ideal goal would

be to utilize a camera with a frame rate of 500 FPS that can approximately track objects with a resolution of 0.1° . For reference, the minimum angular deviation the encoders can measure is 0.036° . However, for practical purposes, a resolution of 0.25° or less is sufficient for system operation, so a camera with a frame rate of 240 FPS or more would be adequate for precise feedback.

6.3 Perspective Camera

6.3.1 Perspective Projection Model

One challenge that is specially pronounced when faced with a moving object is that of skewed lengths and distances. In the test-bench setup, the camera is fixed at a particular distance from the platform while the cart and the pendulum are free to move about. It stands to reason that the image formed by the camera will have slightly warped coordinate values for the cart and pendulum as they change their location. This problem is addressed by incorporating the perspective projection model in the calculations. The perspective projection model allows the algorithm to relate object locations in the real world to the corresponding pixel locations in the image [47].

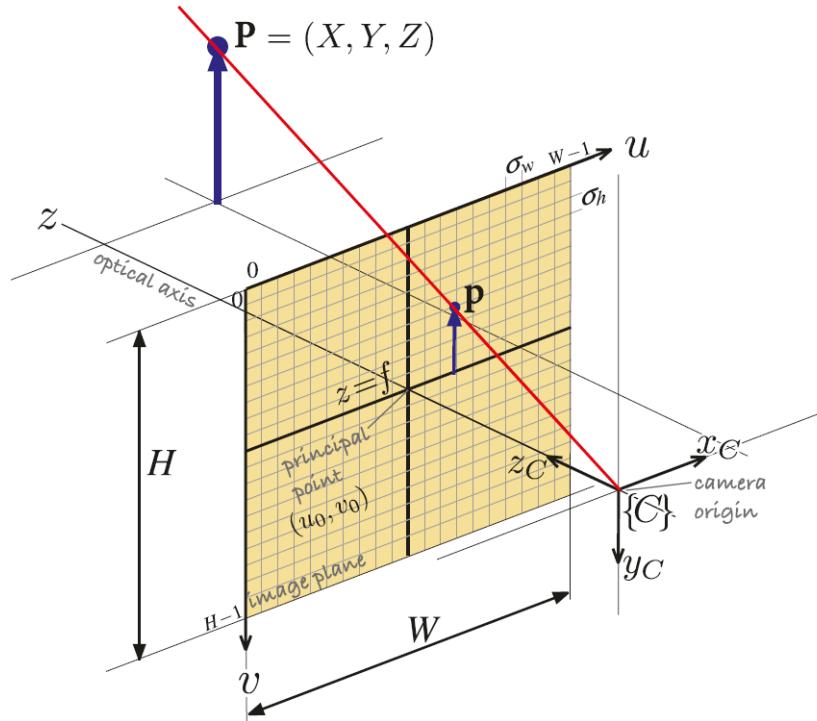


Figure 6.3: Central projection model showing image plane and discrete pixels

Figure 6.3 shows how image coordinates (u, v) on an image plane relate to the real-world coordinates (X, Y, Z) of an object. The origin of an image plane is at its top left pixel. Therefore, the pixel coordinates are related to the image-plane coordinates by

$$u = \frac{x}{p_w} + u_0, v = \frac{y}{p_h} + v_0$$

where p_w and p_h are the width and height of each pixel while (u_0, v_0) is the principal point.

The combined camera projection model [47] can be obtained in the general form as follows:

$$p = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \begin{bmatrix} \frac{f}{p_w} & 0 & u_0 \\ 0 & \frac{f}{p_h} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & t \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (6.1)$$

where

$$u = \frac{\tilde{u}}{\tilde{w}}, v = \frac{\tilde{v}}{\tilde{w}}, w = \tilde{w}$$

One interesting point that should be noted here is that the perspective projection model isn't extremely relevant for calculating pendulum angular displacement values. That is because the model is linear, so it merely scales the input value and shifts them by a particular magnitude. This in turn means that the angle calculated via trigonometry remains same. The formula for obtaining the angle is:

$$\theta = \tan^{-1}\left(\frac{x}{y}\right)$$

where x is the horizontal displacement between the cart center and the pendulum bob while y is the vertical displacement between them.

6.3.2 Lens Distortion

Lenses in modern cameras are highly advanced but even they have certain shortcomings. Geometric imperfections is the most problematic effect that is relevant for applications to an inverted pendulum test-bench. The most prominent geometric imperfections cause distortions of a radial and/or tangential nature. Tangential is usually less significant than

its counterpart and occurs at right angles to the radii. Radial distortion causes image points to be translated along radial lines from the principal point [48].

The coordinates of the point (u, v) after distortion are given by the following formulae:

$$u^d = u + \delta_u, v^d = v + \delta_v$$

where the displacement is

$$\begin{bmatrix} \delta_u \\ \delta_v \end{bmatrix} = \begin{bmatrix} u(k_1r^2 + k_2r^4 + k_3r^6 + \dots) \\ v(k_1r^2 + k_2r^4 + k_3r^6 + \dots) \end{bmatrix} + \begin{bmatrix} 2p_1uv + p_2(r^2 + 2u^2) \\ p_1(r^2 + 2v^2) + 2p_2uv \end{bmatrix} \quad (6.2)$$

Here, (u^d, v^d) are the distorted coordinates which are extracted from the image.

One way to obtain the actual object coordinates is by simply neglecting the tangential component and factoring in the radial components since that is more significant in terms of distortion [48] [49].

$$u = \frac{u^d}{(k_1r^2 + k_2r^4 + k_3r^6)} \quad (6.3)$$

$$v = \frac{v^d}{(k_1r^2 + k_2r^4 + k_3r^6)} \quad (6.4)$$

Another way to calculate the actual coordinates from distorted coordinates can be found in literature in [50]. First, new constants are obtained from the current camera parameters as follows.

$$b_1 = -k_1 \quad (6.5)$$

$$b_2 = 3k_1^2 - k_2 \quad (6.6)$$

$$b_3 = 8k_1k_2 - 12k_1^3 - k_3 \quad (6.7)$$

and then the constants are plugged back into the equation to obtain the actual coordinates as follows.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u^d \\ v^d \end{bmatrix} + \begin{bmatrix} u^d(b_1r^2 + b_2r^4 + b_3r^6) \\ v^d(b_1r^2 + b_2r^4 + b_3r^6) \end{bmatrix} \quad (6.8)$$

The coefficients for the ZED 2 stereo camera are available in the setup files as discussed in [51]. The values obtained for the particular camera used in the inverted pendulum platform are as follows.

$$fx = 528.26 \quad (6.9)$$

$$fy = 527.99 \quad (6.10)$$

$$cx = 645.525 \quad (6.11)$$

$$cy = 356.9645 \quad (6.12)$$

$$k1 = -0.0421708 \quad (6.13)$$

$$k2 = 0.0113528 \quad (6.14)$$

$$k3 = -0.00532588 \quad (6.15)$$

$$p1 = 5.9933 \times 10^{-5} \quad (6.16)$$

$$p2 = -0.000815951 \quad (6.17)$$

The above values are then inserted into 6.2 and/or 6.8 to account for the nonlinearities caused by lens distortion. However, it was surprising to note that the correction in the values by accounting for lens distortion was minimal and did not reduce the error between the angular displacement values as calculated by the camera and by rotary encoders by any significant amount.

Further exploration is needed in this aspect. One promising technique is to manually calibrate the camera instead of relying on the parameter values provided in the setup code files, and then revisit the same formulae. Another technique may be to account for distortion caused by the wide-angle nature of the ZED 2 camera.

6.4 Results and Performance

In order to gauge the performance of the vision-system, the results obtained by the ZED 2 camera were compared with the angular feedback provided from the incremental rotary encoder, where the encoder values served as a benchmark. The results extracted from both sensor were timestamped, allowing for easy synchronization during analysis. The readings for pendulum's angular response were gathered for multiple cart positions: directly in front of the camera, to the left of the camera and to the right of the camera.

This section only displays and discusses the results obtained for the readings obtained with the cart directly in front of the camera. The analysis for the remaining set of readings is similar, and can be found in appendix G.

As discussed in section 6.2.2, one of the constraints on a vision system was the processor's ability to process frames fast enough. The results in Figure 6.4 reflect this limitation as a time lag or latency of $314.3ms$ between when the feedback is received from the vision system compared to the benchmark of encoder values. This is given for an angular response having a time period of 1.38 seconds.

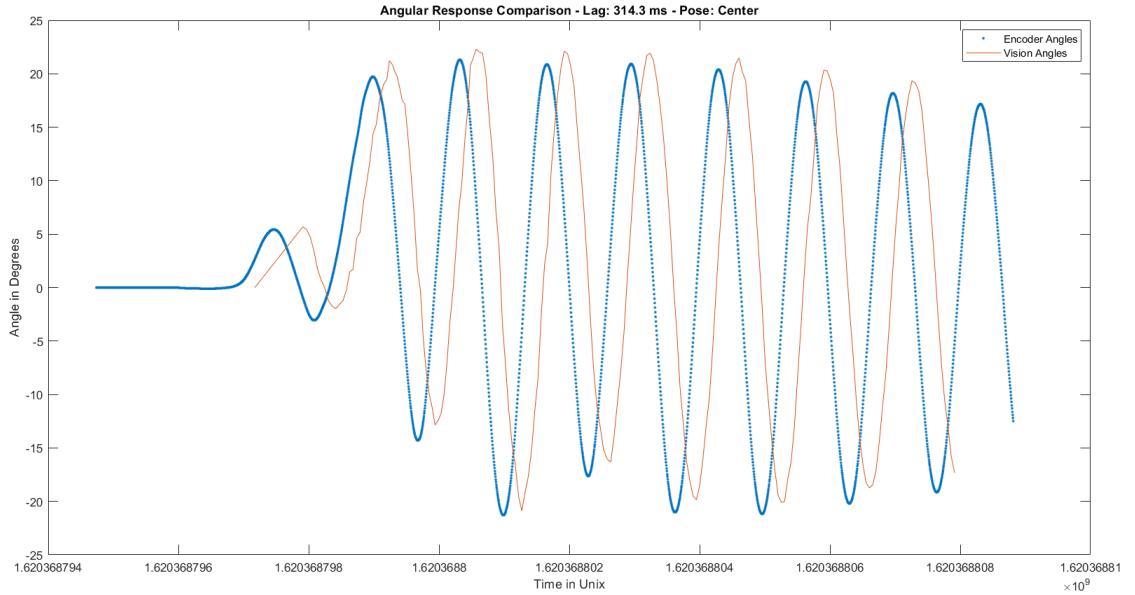


Figure 6.4: Comparison between encoder and visual feedback - with lag

While this constraint was not controllable in the vision system, in order to fairly measure the error in the angles obtained, lag between the curves was removed and they were plotted again as seen in Figure G.8. This is required since without said compensation, computing the error would naturally lead to large magnitudes of error as shown in Figure 6.9.

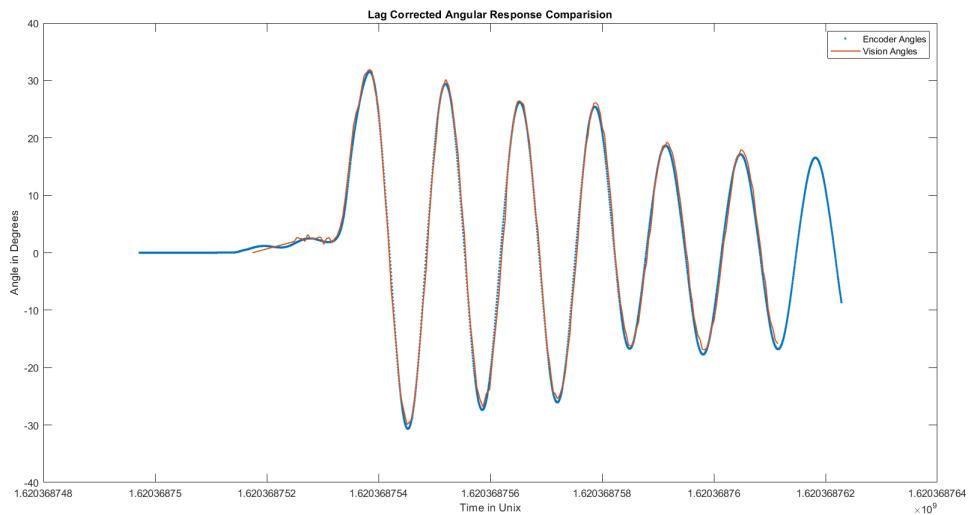


Figure 6.5: Comparison between encoder and visual feedback - compensated for Lag

Figure 6.6 shows the error recorded between the visual and encoder feedback values as error bars on the angular response curve for the vision system.

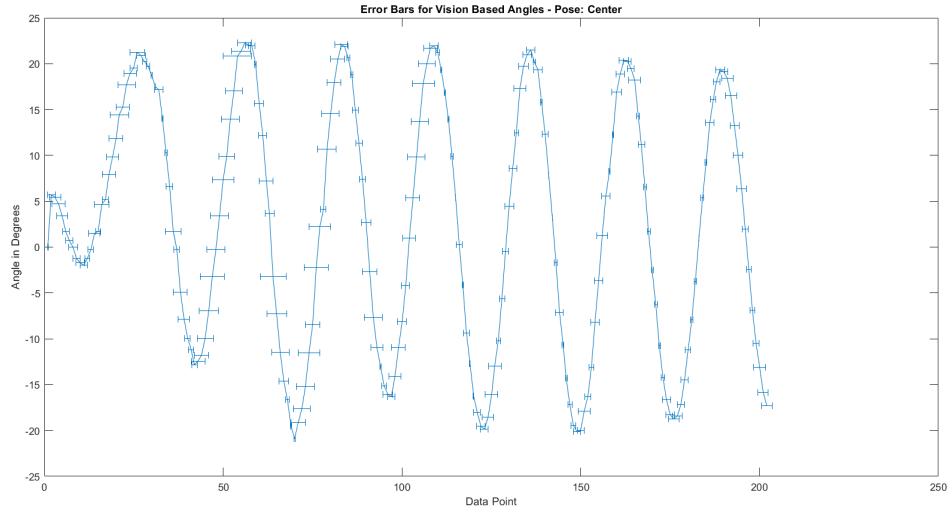


Figure 6.6: Horizontal error bars for the angular response of the vision system

The frequency distribution of different error magnitudes can be seen in 6.7. Note that the error values are concentrated to be within a value of 2° .

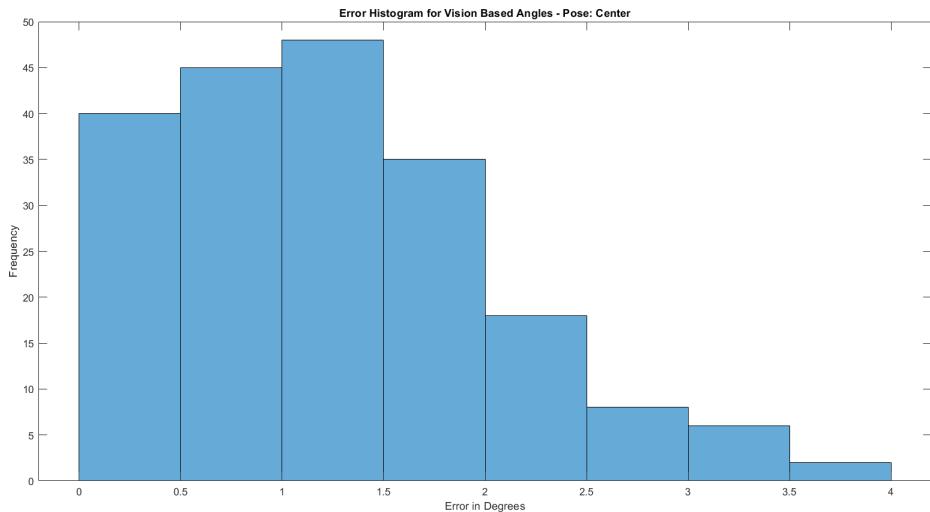


Figure 6.7: Histogram of error values - compensated for Lag

Figure 6.8 shows how the error between visual and encoder feedback varies with time. This curve is plotted once the time lag between the two curves has been eliminated. This ensures that the error values are reflective of the error present due to lens distortion, camera perspective etc. and not due to slow processor capability.

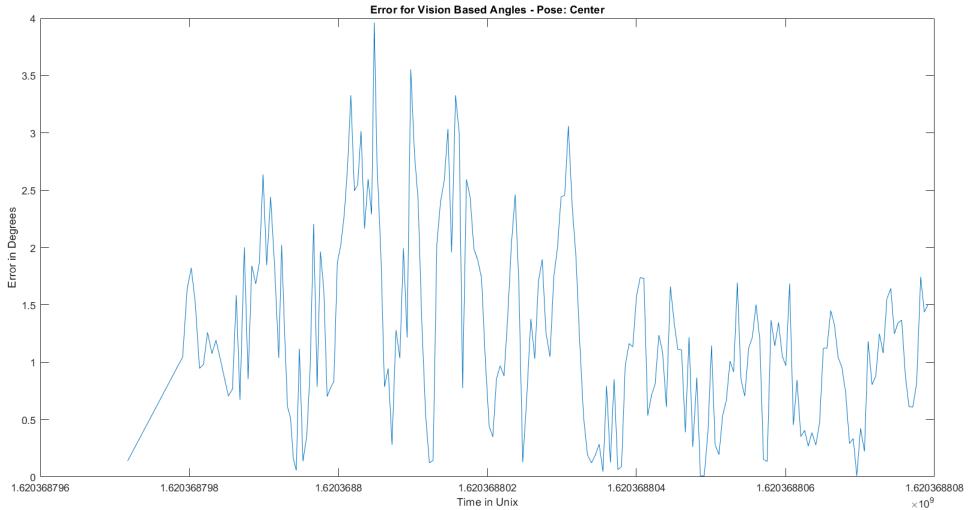


Figure 6.8: Plot of error versus time - compensated Lag

If the lag between the two curves is not eliminated before plotting the error, the results obtained would be meaningless because of the extremely high values of error recorded. This can be seen in Figure 6.9 below.

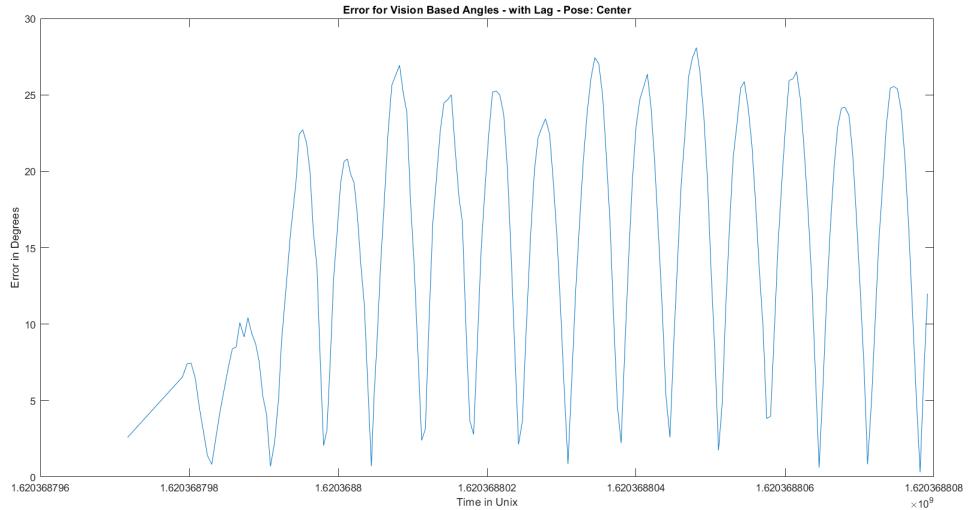


Figure 6.9: Plot of error versus time - with Lag

Figure 6.10 demonstrates how latency impacts the degree of error in the vision system, serving to be a major constraint on the vision system's ability as a viable feedback mechanism. The system is currently operating at about $350ms$ for an average error of about 17° . However, bringing down the latency to less than $50ms$ can reduce the error to being within 2° . Note that this is the error without camera distortion accounted for. So it is expected that once a faster processor is utilized and camera distortion is

compensated, the error values obtained will be negligible.

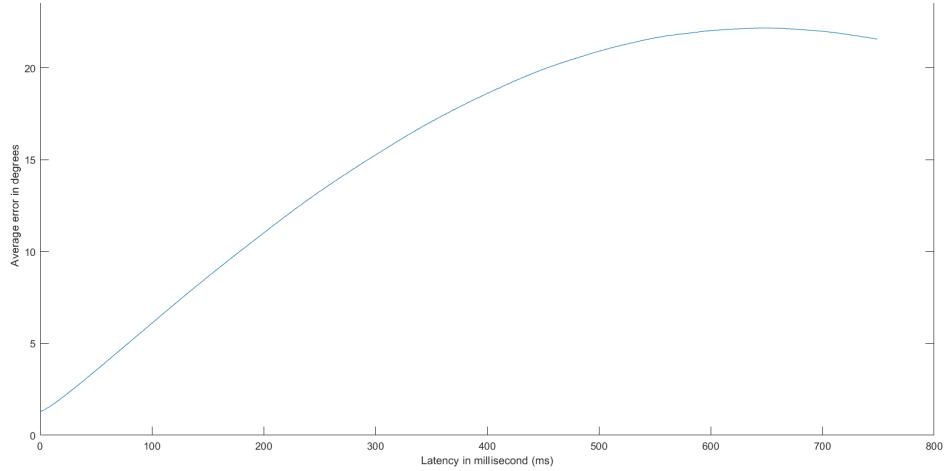


Figure 6.10: Plot of error versus latency

It can be observed from these error plots that the vision feedback system provides values that have a fair degree of accuracy when compared to the encoder feedback which is known to be completely accurate. The mean error value for this particular dataset comes out to be 1.24° with the majority of errors recorded as below 2° . While it is difficult to concretely claim whether this amount of accuracy would be sufficient to achieve control on the inverted pendulum test bench without testing it on the physical system, the results obtained do successfully demonstrate that there is significant potential in using a vision system as a feedback mechanism once the perspective projection model is further improved, and latency is minimized through use of a more powerful CPU/GPU or even a dedicated FPGA architecture.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The Capstone Project was successful in designing the comprehensive hardware required for an inverted pendulum platform while also incorporating the requirements of scalability, modularity, and cost-effectiveness. Vision was implemented successfully on the test-bench in order to calculate the pendulum's angular displacement and track pendulum motion in real-time using the ZED 2 Stereo Camera. LQR and PID control algorithms were implemented on the test-bench via feedback from rotary encoders and actuation by a DC Gear Motor. However, robust control could not be achieved on the platform using either algorithm. The shortcomings faced in the control domain were due to the limitations of the motor available in the lab inventory as discussed extensively in chapter 5.

Furthermore, this thesis report and the accompanying appendix ably documents all the relevant details that are required to understand, repair, or replicate the project as required.

7.2 Way Forward

The first priority of the authors moving forward would be to change the motor used to a DC Gear Motor with a higher stall torque, or even to a Stepper Motor in order to achieve control. Changing to a DC Gear Motor would be simpler since it would be the only factor that would have to change. It would also leave room for the implementation of the swing-up algorithm on the test-bench if the RPM of the DC Gear Motor is high enough. On the other hand, a Stepper Motor is more precise and accurate w.r.t position control and thus suitable for stabilization, but would require major changes in the programmed code, the motor driver, and limit system operation to linearized stabilization control only.

However, the primary potential of working with this test-bench is in the vision domain which is relatively unexplored in literature. Two especially promising directions are optimizing the image processing algorithm to minimize feedback latency and increase accuracy, and shifting image processing to an FPGA to contrast performance with a CPU/GPU.

In particular, there is room to further work on and improve the current image processing algorithm and cater to the non-linear skew introduced in visual feedback due to lens distortion because the current algorithm does not result in a significant correction. There is also a significant latency observed when comparing visual feedback values to encoder values which can be improved using a variety of techniques.

Additionally, once the motor constraint has been fixed, more sophisticated control algorithms, for example the LQR controller or the LQG controller, can be implemented on the test-bench. Similarly, once vision has successfully been implemented as a feedback mechanism to a desired degree of precision and minimum latency value, the system can be controlled via a vision-encoder hybrid approach or even relying on vision

alone. The system response under different feedback schemes can subsequently be investigated. The platform can even be scaled up to a double or triple inverted pendulum to vary the difficulty of the control problem if required.

Appendices

APPENDIX A

COMMERCIALLY AVAILABLE SETUPS

Table A.1: Quanser Inverted Pendulum Setup Specifications [8]

Parameter	Value
Track length	1.02 m
Cart Mass	0.57 kg
Short Pendulum Length	0.2 m
Short Pendulum Mass	0.097 kg
Medium Pendulum Length	0.337 m
Medium Pendulum Mass	0.127 kg
Long Pendulum Length	0.641 m
Long Pendulum Mass	0.230 kg
Motor Voltage	6V
Max. Motor Current	1A
Max. Motor Speed	6000 RPM
Gear Ratio	3.71
Apparent. Motor Speed	1600 RPM
Encoder Resolution	4096

Table A.2: GoogloTech Inverted Pendulum Setup Specifications [52]

Parameter	Value
Swinging rod length	400 mm
Swinging rod mass	0.105 kg
Maximum acceleration	10m/s ²
Maximum speed	1.5m/s
Pitch	10 mm
Linear motor encoder resolution	10,000 pulse/pitch
Rotary encoder resolution	2400 pulse/r
Control refresh rate	200us
Dimensions	620 × 95 × 415mm
Gliding block mass	1.7 kg

APPENDIX B

3D MODELS

The 3D model was constructed using TinkerCAD. Different angles of the preliminary model are shown below in Figs. B.1-B.8.

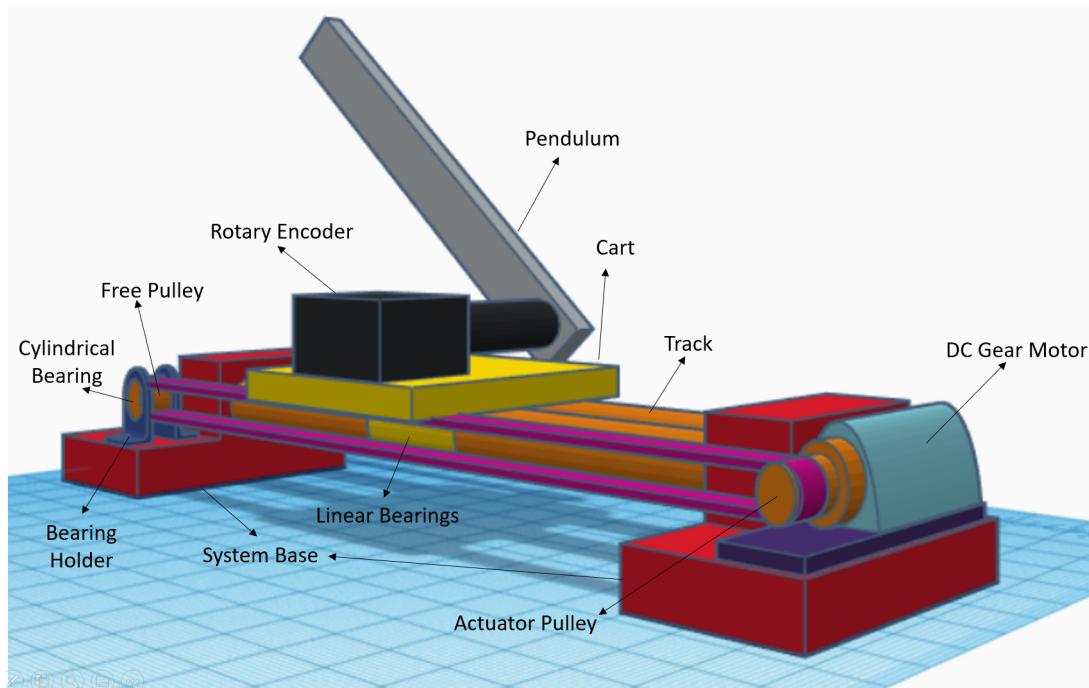


Figure B.1: Labelled System Model

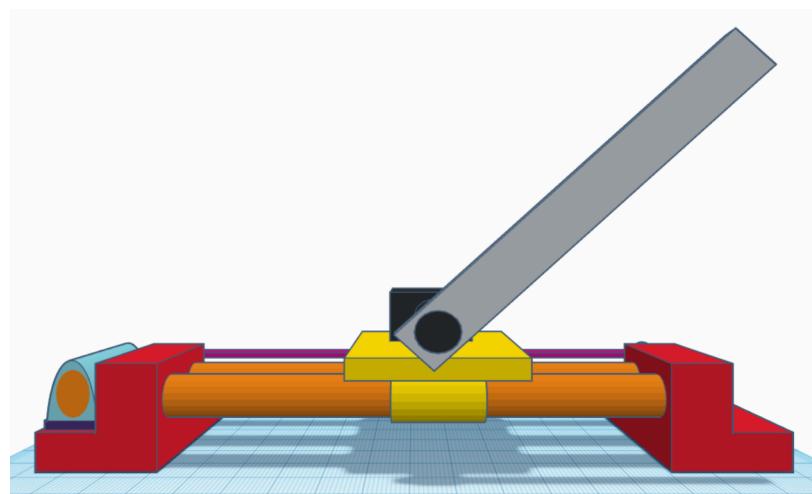


Figure B.2: Front View

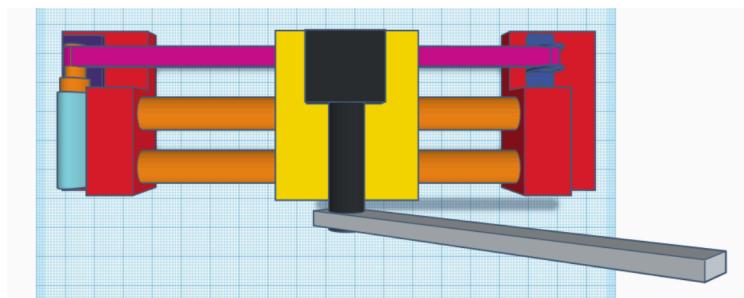


Figure B.3: Top View

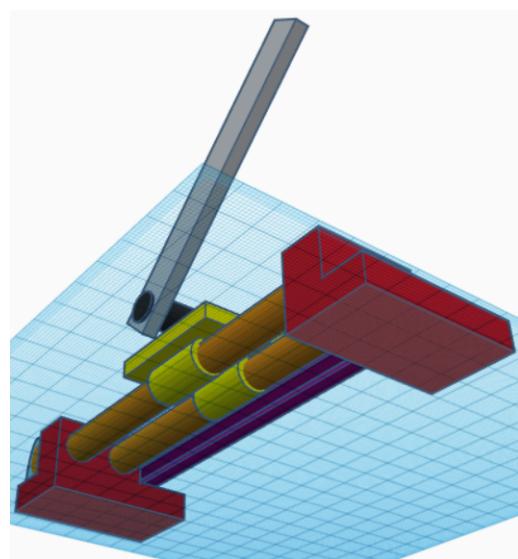


Figure B.4: Bottom View

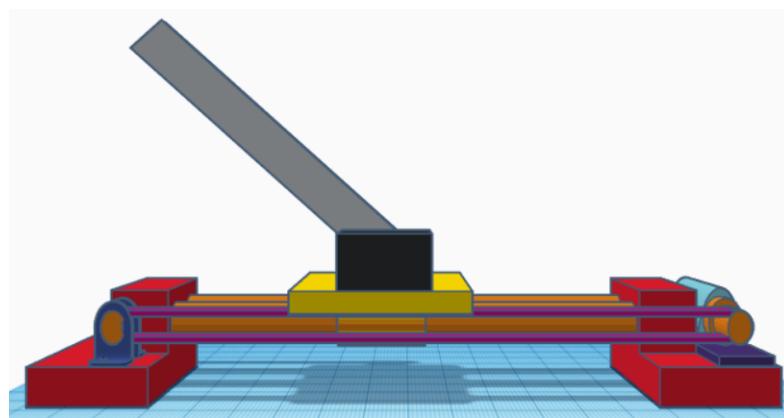


Figure B.5: Backside View

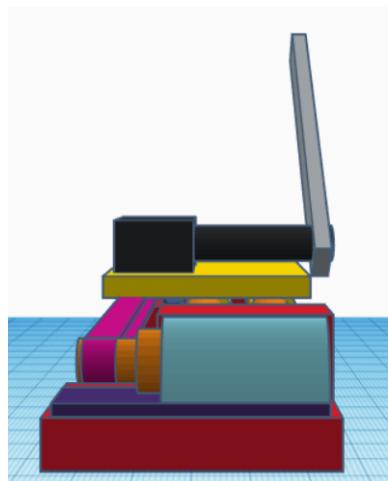


Figure B.6: Left Side View

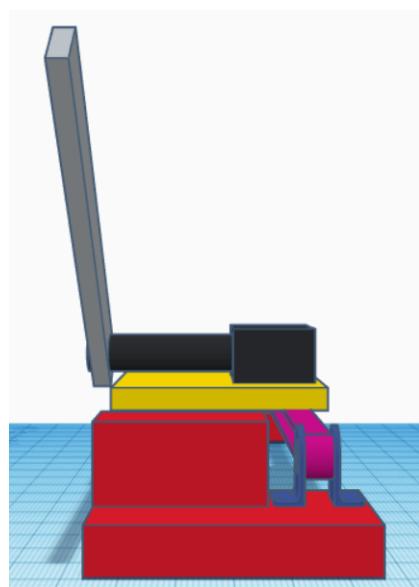


Figure B.7: Right Side View

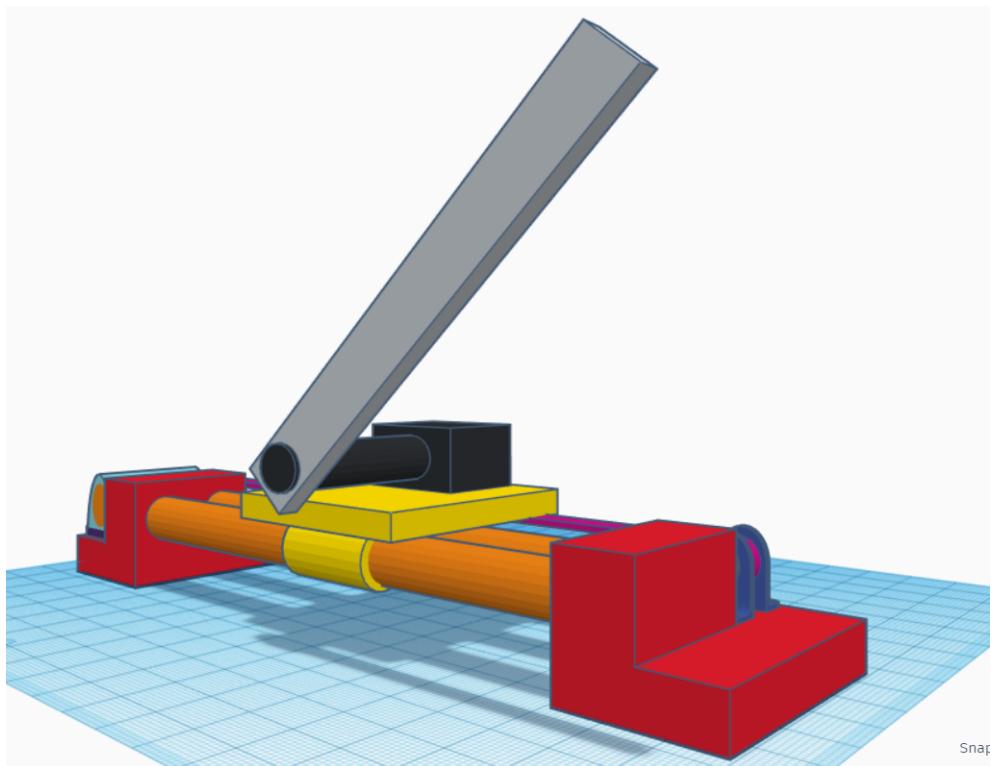


Figure B.8: Skewed View

APPENDIX C

TEST-BENCH DESIGN

The Figs. C.1-C.6 show different closeups of the test-bench constructed.

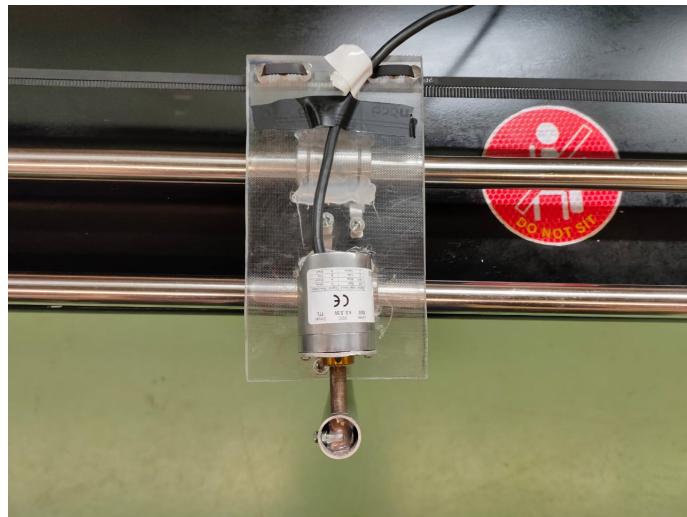


Figure C.1: Cart Top View

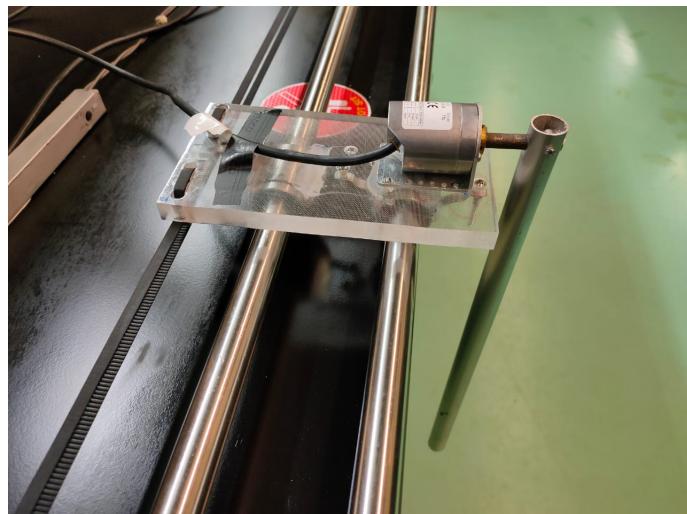


Figure C.2: Cart Closeup

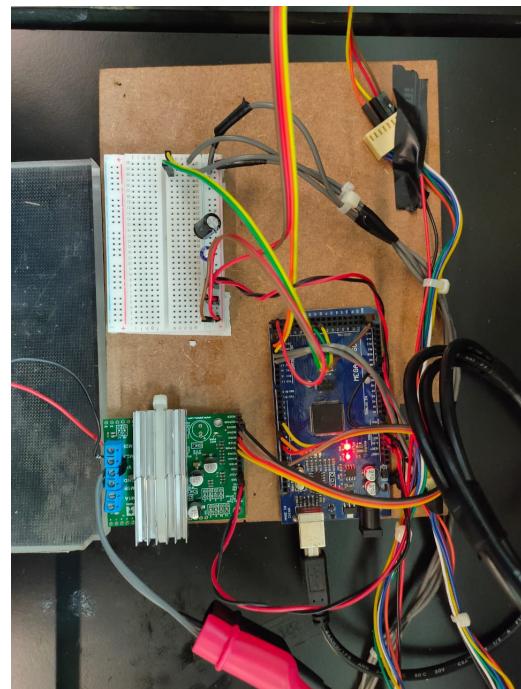


Figure C.3: Platform Circuit

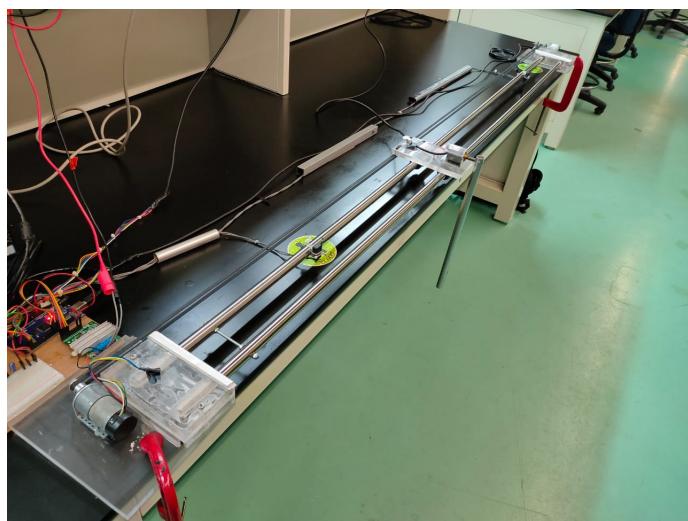


Figure C.4: Left Platform View

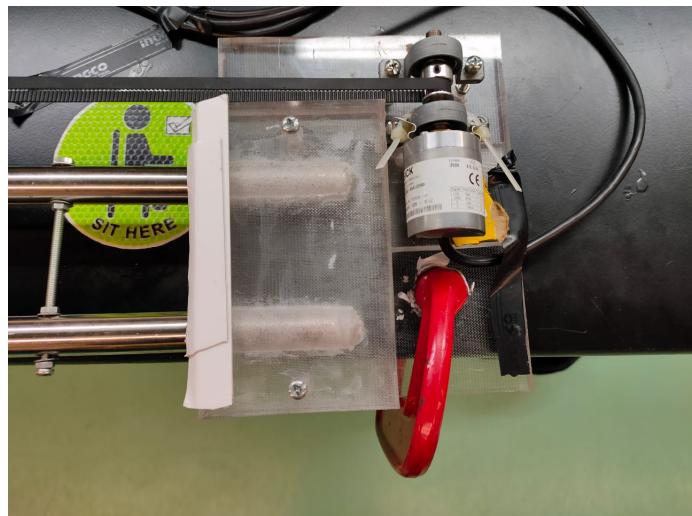


Figure C.5: Far End Base with Free Pulley and Encoder

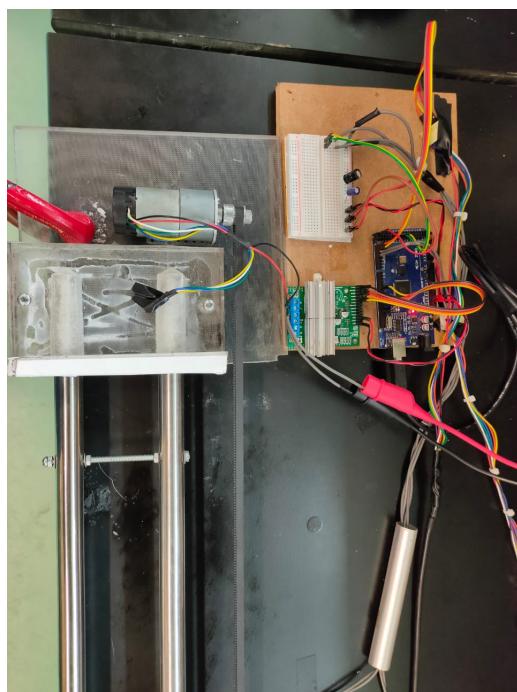


Figure C.6: Motor and Circuit Closeup

APPENDIX D

SCHEMATICS

Fig. D.1 shows the circuit connections of the electrical components present in the system and Fig. D.2 shows the same connections after they have been routed for PCB design.

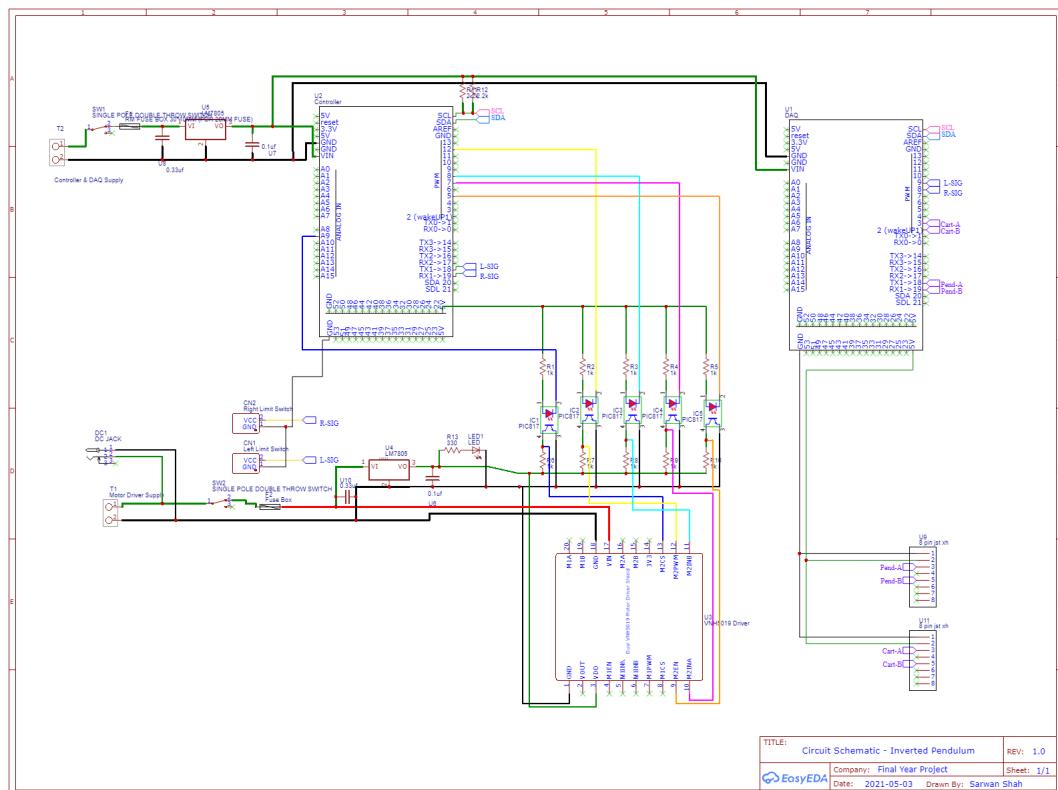


Figure D.1: Circuit Schematic

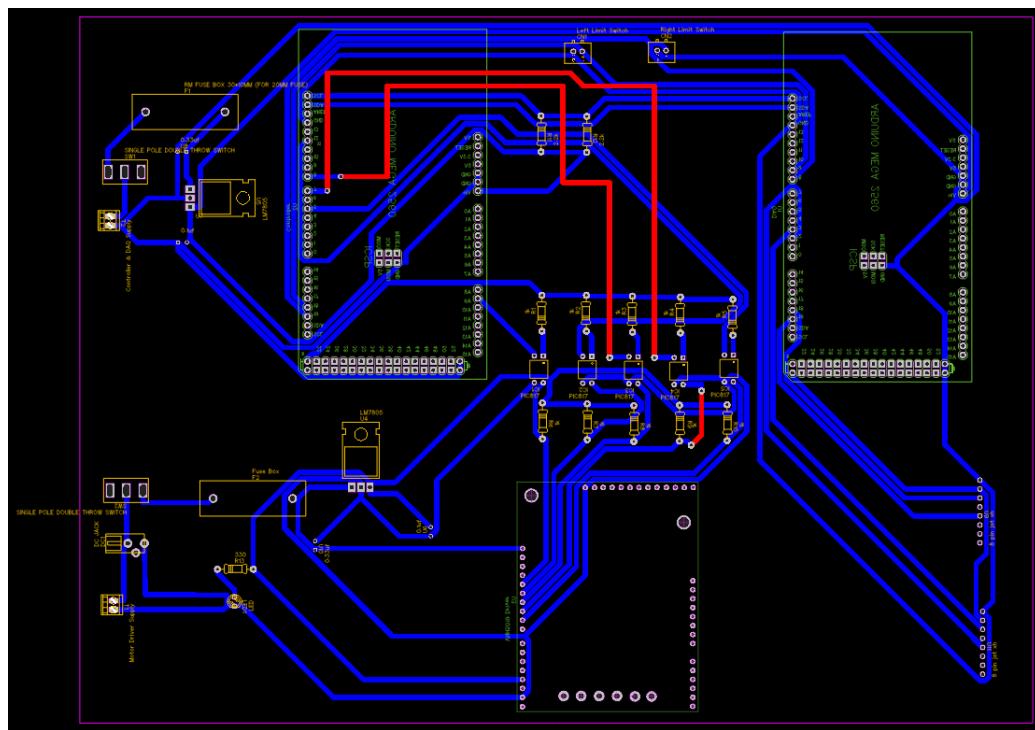


Figure D.2: PCB Schematic

APPENDIX E

SYSTEM MODELLING AND SIMULATIONS

E.1 MATLAB Code for finding mechanical tau

```
1 time = rmmissing(TimeMilliSec1./1000); % Removing Nan's
2 velo = rmmissing(Velocity);
3 time = time - time(1) + (time(2) - time(1));
4 time = [0; time];
5 velo = [0; velo];
6
7 %%sys = tf([20.878,0],[1,15.26,0]);
8 %step(sys*9.55*24);
9 hold on;
10
11 plot(fittedmodel);
12 grid on;
13 hold off;
14 ylim([0 200])
15 xlim([0 1])
16 title("Motor Response - 12V");
17 xlabel("Time in seconds");
18 ylabel("Velocity in RPM");
19 legend('Motor Response');
20
21 % Finding equation of line to find t_mech
22 x = [point1.Position(1), point2.Position(1)]
23 y = [point1.Position(2), point2.Position(2)]
24 c = [[1; 1] x(:)]\y(:);
25 slope_m = c(2);
26 intercept_b = c(1);
27
28 % steady-state value
29 steadystate_val = fittedmodel(5)*0.6363
30
31 % Using inverse function of line to determine t_mech
32 tmech = (steadyval - intercept_b)/slope_m
33
34 v1 = [];
35
36 for i = 1:length(time)
37     v1 = [v1 fittedmodel(time(i))];
38 end
```

```

39
40 raw_response= stepinfo(velo , time )
41 fitted_response = stepinfo(v1 , time )

```

E.2 Non-linear Plant Model Derivation

We have the the following two non-linear equations describing the dynamic model of the inverted pendulum:

$$F = (M + m)\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta \quad (\text{E.1})$$

$$(I + ml^2)\ddot{\theta} + m\ddot{x}l \cos \theta - mgl \sin \theta = 0 \quad (\text{E.2})$$

Making $\ddot{\theta}$ and \ddot{x} subject from (E.2), we get:

$$\ddot{\theta} = \frac{mgl \sin \theta}{I + ml^2} - \frac{ml \cos \theta}{I + ml^2} \ddot{x} \quad (\text{E.3})$$

$$\ddot{x} = \frac{mgl \sin \theta}{ml \cos \theta} - \frac{(I + ml^2)\ddot{\theta}}{ml \cos \theta} = \frac{g \sin \theta}{\cos \theta} - \frac{(I + ml^2)\ddot{\theta}}{ml \cos \theta} \quad (\text{E.4})$$

Substituting (E.3) into (E.1), we get:

$$\begin{aligned} F &= (M + m)\ddot{x} + ml \cos \theta \left(\frac{mgl \sin \theta}{I + ml^2} - \frac{ml \cos \theta}{I + ml^2} \ddot{x} \right) - ml \sin \theta \dot{\theta}^2 \\ F &= (M + m)\ddot{x} + \frac{m^2 gl^2 \sin \theta \cos \theta}{I + ml^2} - \frac{m^2 l^2 \cos \theta^2}{I + ml^2} \ddot{x} - ml \sin \theta \dot{\theta}^2 \\ (M + m - \frac{m^2 l^2 \cos \theta^2}{I + ml^2})\ddot{x} &= F + ml \sin \theta \dot{\theta}^2 - \frac{m^2 gl^2 \sin \theta \cos \theta}{I + ml^2} \end{aligned}$$

Let $a = M + m - \frac{m^2 l^2 \cos \theta^2}{I + ml^2}$, the final equation for \ddot{x} becomes:

$$\therefore \ddot{x} = \frac{F}{a} - \frac{m^2 gl^2 \sin \theta \cos \theta}{a(I + ml^2)} + \frac{ml \sin \theta \dot{\theta}^2}{a} \quad (\text{E.5})$$

Now, substituting (E.4) into (E.1):

$$F = (M + m) \frac{g \sin \theta}{\cos \theta} - (M + m) \frac{(I + ml^2) \ddot{\theta}}{ml \cos \theta} + ml \ddot{\theta} \cos \theta - ml \dot{\theta}^2 \sin \theta$$

$$F = (ml \cos \theta - (M + m) \frac{(I + ml^2)}{ml \cos \theta}) \ddot{\theta} + (M + m) \frac{g \sin \theta}{\cos \theta} - ml \dot{\theta}^2 \sin \theta$$

Let $b = ml \cos \theta - (M + m) \frac{(I + ml^2)}{ml \cos \theta}$, the final equation for $\ddot{\theta}$ becomes:

$$\therefore \ddot{\theta} = \frac{F}{b} - (M + m) \frac{g \sin \theta}{b \cos \theta} + \frac{ml \dot{\theta}^2 \sin \theta}{b} \quad (\text{E.6})$$

E.3 Non-linear Plant Model incorporating Viscous Friction Derivation

$$F = (M + m) \ddot{x} - B_1 \dot{x} + ml \ddot{\theta} \cos \theta - ml \dot{\theta}^2 \sin \theta \quad (\text{E.7})$$

$$(I + ml^2) \ddot{\theta} - B_2 \dot{\theta} + m \ddot{x} l \cos \theta - m g l \sin \theta = 0 \quad (\text{E.8})$$

From (E.8), making $\ddot{\theta}$ and \ddot{x} subject:

$$\ddot{\theta} = \frac{m g l \sin \theta - m l \ddot{x} \cos \theta + B_2 \dot{\theta}}{I + ml^2} \quad (\text{E.9})$$

$$\ddot{x} = \frac{m g l \sin \theta - (I + ml^2) \ddot{\theta} + B_2 \dot{\theta}}{m l \cos \theta} \quad (\text{E.10})$$

Substituting (E.9) in (E.7):

$$F = (M + m) \ddot{x} - B_1 \dot{x} + ml \left(\frac{m g l \sin \theta - m l \ddot{x} \cos \theta + B_2 \dot{\theta}}{I + ml^2} \right) \cos \theta - ml \dot{\theta}^2 \sin \theta$$

$$F = (M + m - \frac{m^2 l^2 \cos \theta^2}{I + ml^2}) \ddot{x} - B_1 \dot{x} + \frac{m^2 g l^2 \sin \theta \cos \theta}{I + ml^2} + \frac{ml B_2 \dot{\theta} \cos \theta}{I + ml^2} - ml \dot{\theta}^2 \sin \theta$$

Again, let $a = M + m - \frac{m^2 l^2 \cos \theta^2}{I + ml^2}$, and make \ddot{x} subject:

$$\therefore \ddot{x} = \frac{F}{a} + \frac{B_1 \dot{x}}{a} - \frac{m^2 g l^2 \sin \theta \cos \theta}{a(I + ml^2)} + \frac{ml \dot{\theta}^2 \sin \theta}{a} \quad (\text{E.11})$$

Now, substituting (E.10) in (E.7), we get:

$$F = (M + m)\left(\frac{mgl \sin \theta - (I + ml^2)\ddot{\theta} + B_2\dot{\theta}}{ml \cos \theta}\right) - B_1\dot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta$$

$$F = \left(ml \cos \theta - \frac{(M + m)(I + ml^2)}{ml \cos \theta}\right)\ddot{\theta} + \frac{(M + m)B_2\dot{\theta}}{ml \cos \theta} + \frac{(M + m)g \sin \theta}{\cos \theta} - ml\dot{\theta}^2 \sin \theta$$

Here, let $b = ml \cos \theta - \frac{(M + m)(I + ml^2)}{ml \cos \theta}$. Hence, making $\ddot{\theta}$ subject, we get:

$$\therefore \ddot{\theta} = \frac{F}{b} - \frac{(M + m)B_2\dot{\theta}}{bml \cos \theta} - \frac{(M + m)g \sin \theta}{b \cos \theta} + \frac{ml\dot{\theta}^2 \sin \theta}{b} \quad (\text{E.12})$$

E.4 Linearized Plant Model incorporating Viscous Friction Derivation

We make the following assumptions to linearize our equations at point $\theta = 180$ degrees:

$$\sin \theta \approx \theta$$

$$\cos \theta \approx -1$$

$$\theta \dot{\theta}^2 \approx 0$$

Then, our system equations become:

$$F = (M + m)\ddot{x} - B_1\dot{x} - ml\ddot{\theta} \quad (\text{E.13})$$

$$(I + ml^2)\ddot{\theta} - B_2\dot{\theta} - ml\ddot{x} - mgl\theta = 0 \quad (\text{E.14})$$

From (3.48), we make $\ddot{\theta}$ and \ddot{x} subject:

$$\ddot{\theta} = \frac{mgl\theta + ml\ddot{x} + B_2\dot{\theta}}{I + ml^2} \quad (\text{E.15})$$

$$\ddot{x} = \frac{(I + ml^2)\ddot{\theta} - mgl\theta - B_2\dot{\theta}}{ml} \quad (\text{E.16})$$

Substituting (E.15) in (E.13):

$$\begin{aligned} F &= (M + m)\ddot{x} - B_1\dot{x} - ml\left(\frac{mgl\theta + ml\ddot{x} + B_2\dot{\theta}}{I + ml^2}\right) \\ F &= (M + m - \frac{m^2l^2}{I + ml^2})\ddot{x} - \frac{mlB_2\dot{\theta}}{I + ml^2} - \frac{m^2gl^2\theta}{I + ml^2} - B_1\dot{x} \end{aligned}$$

Let $a = M + m - \frac{m^2l^2}{I + ml^2}$, such that the linearized expression for \ddot{x} becomes:

$$\therefore \ddot{x} = \frac{F}{a} + \frac{mlB_2\dot{\theta}}{a(I + ml^2)} + \frac{m^2gl^2\theta}{a(I + ml^2)} + \frac{B_1\dot{x}}{a} \quad (\text{E.17})$$

Now, substituting (E.16) in (E.13):

$$F = (M + m) \frac{(I + ml^2)\ddot{\theta} - mgl\theta - B_2\dot{\theta}}{ml} - B_1\dot{x} - ml\ddot{\theta}$$

$$F = \left(\frac{(M + m)(I + ml^2)}{ml} - ml \right) \ddot{\theta} - \frac{(M + m)(mgl\theta)}{ml} - \frac{(M + m)B_2\dot{\theta}}{ml} - B_1\dot{x}$$

Let $b = \left(\frac{(M + m)(I + ml^2)}{ml} - ml \right)$, so that the linearized expression for $\ddot{\theta}$ becomes:

$$\therefore \ddot{\theta} = \frac{F}{b} + \frac{(M + m)g\theta}{b} + \frac{(M + m)B_2\dot{\theta}}{mlb} + \frac{B_1\dot{x}}{b} \quad (\text{E.18})$$

E.5 Plant Model Validation Dataset 2

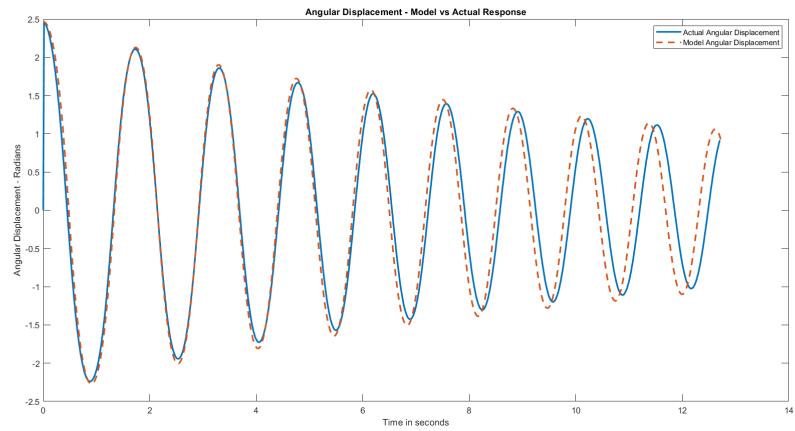


Figure E.1: Angular Displacement of Pendulum – Model vs. Actual – $\theta = -141.25^\circ$

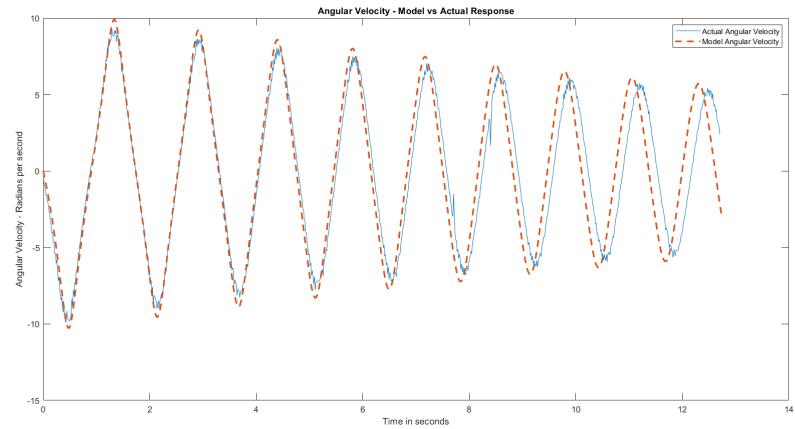


Figure E.2: Angular Velocity of Pendulum – Model vs. Actual – $\theta = -141.25^\circ$

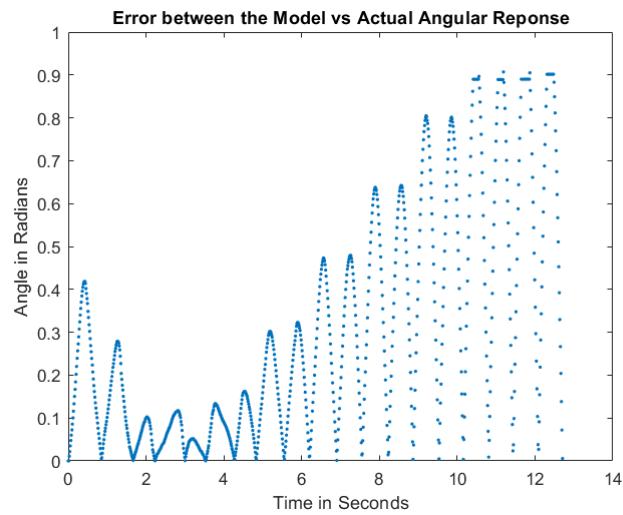


Figure E.3: Error in Angular Displacement of Pendulum - Model vs. Actual

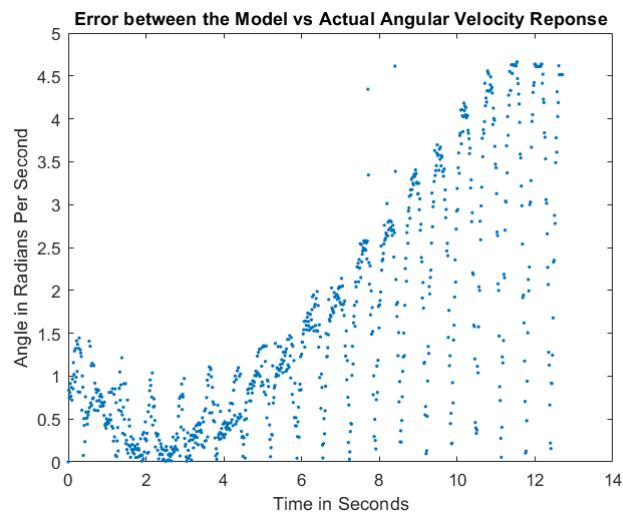


Figure E.4: Error in Angular Velocity of Pendulum - Model vs. Actual

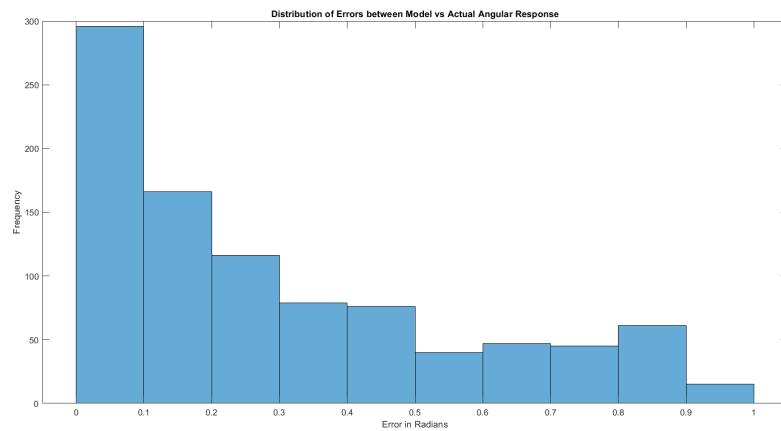


Figure E.5: Distribution of Errors in Angular Displacement of Pendulum

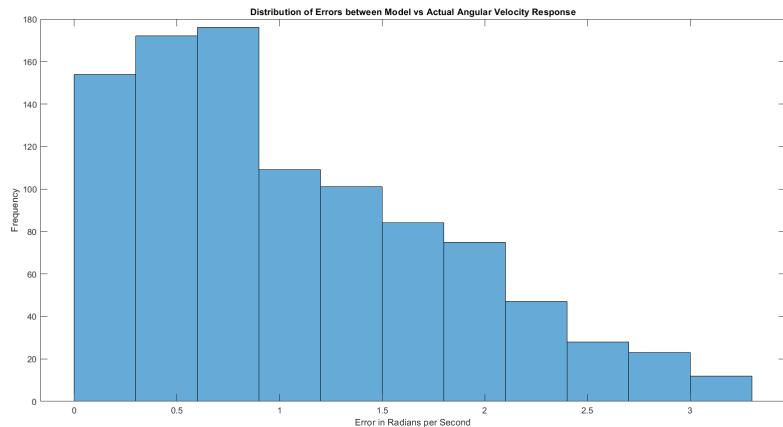


Figure E.6: Distribution of Errors in Angular Velocity of Pendulum

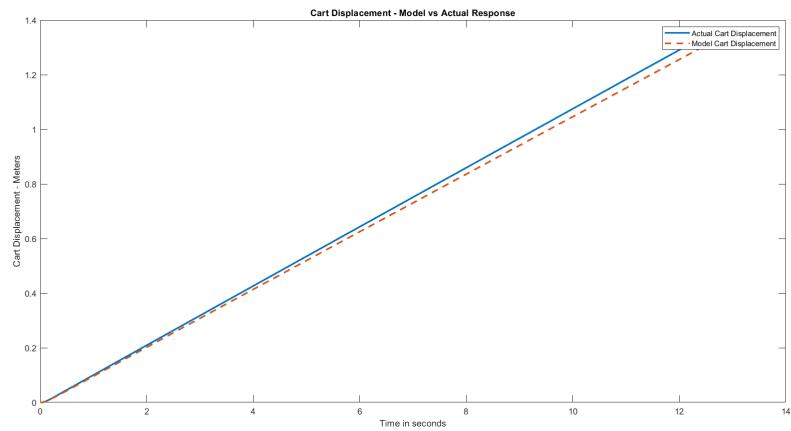


Figure E.7: Cart Displacement - Model vs. Actual

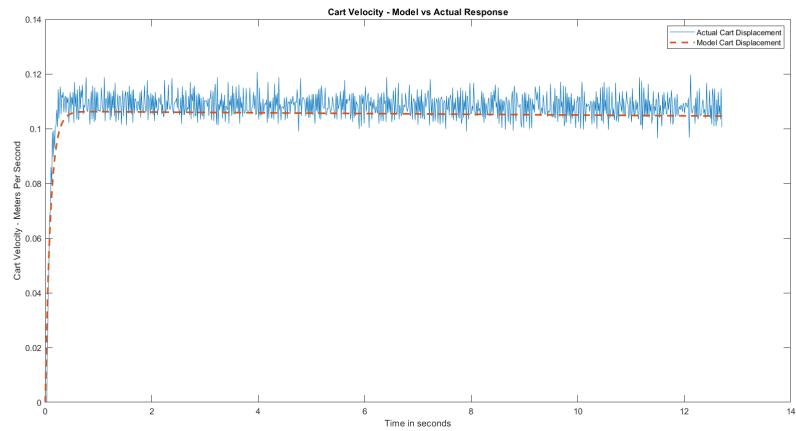


Figure E.8: Cart Velocity - Model vs. Actual

Table E.1: Mean Error Values for Cart - Model vs. Actual

Parameter	Value
Displacement Error	1.3416 m
Velocity Error	0.2121 ms^{-1}

APPENDIX F

CODES USED FOR PROGRAMMING SYSTEM MICROCONTROLLERS

F.1 Code Used for Controller Arduino

The following code was uploaded to the Arduino which had the controller for the system implemented upon it.

```
1
2 #include <Wire.h>
3 #include "DualVNH5019MotorShieldMega.h"
4 DualVNH5019MotorShield md;
5
6 int Lswitch1 = 19;
7 int Lswitch2 = 18;
8 bool MovingLeft = false;
9 bool MovingRight = false;
10 bool initial = true;
11
12 unsigned long lastTime = 0;
13
14 const int SCLpin = 21;
15 const int SDApin = 20;
16
17
18 int L1 = 0;
19 int prevL1 = 0;
20 int L2 = 0;
21 int prevL2 = 0;
22
23 //----- STATE VARIABLES -----//
24
25 volatile float posePendulum = 0;
26 volatile float poseCart = 0;
27 volatile float differentialPosePendulum = 0;
28 volatile float differentialPoseCart = 0;
29 volatile float posePendulumOld = 0;
30 volatile float poseCartOld = 0;
31 volatile float velocityPendulum = 0;
32 volatile float velocityCart = 0;
33
34 float poseInverted = 0;
35 volatile bool control = false;
```

```

36
37 //----- CONTROL VARIABLES -----// 
38
39
40 float setpoint = 0;
41 float Kp = -30.04;
42 float Kd = -0.0452;
43 float Ki = -63.9;
44 float error = 0;
45 float cum_error = 0;
46 float diff_error = 0;
47 float last_error = 0;
48
49
50 volatile float voltage = 0;
51 volatile float Speed = 0;
52 volatile float gain = 0;
53
54 bool forward = true;
55 long int t = 0;
56 float divisor = 1000000;
57
58
59 //Configured to stop the system if fault in motor
60 void stopIfFault()
61 {
62     if (md.getM2Fault ()) {
63         Serial.println("M2 fault");
64         while (1);
65     }
66
67 }
68
69
70 void setup()
71 {
72     Wire.begin(10);
73     Serial.begin(115200);
74
75     pinMode(Lswitch1, INPUT_PULLUP);
76     pinMode(Lswitch2, INPUT_PULLUP);
77
78     md.init();
79
80     GTCCR = (1<<TSM) | (1<<PSRASY) | (1<<PSRSYNC);
81     // Stop all timers
82     setTimer2();                                // Start all timers
83     GTCCR &= 0x00;

```

```

84
85     TCNT2 = 386;
86     TIMSK2 |= (1<<TOIE2); //Enable timer 2 interrupts
87
88 }
89
90
91 void loop()
92 {
93     stopIffFault();
94     LimitSwitchCheck();
95
96     if (control) {
97         Serial.print(posePendulum);
98         Serial.print(",");
99         Serial.print(poseCart);
100        Serial.print(",");
101        Serial.print(velocityPendulum);
102        Serial.print(",");
103        Serial.println(velocityCart);
104
105        requestPose();
106        control = false;
107        poseInverted = 180 - posePendulum;
108
109        //Activate control if the angular deviation is within
110        //      30 degrees but more than 0.4 degrees
111        if (poseInverted >=-0.4 && poseInverted <= 0.4)
112        {
113            cum_error = 0;
114            error = 0;
115            diff_error = 0;
116            md.setM2Brake(400);
117        }
118
119        else if (poseInverted > -30 && poseInverted < 30)
120        {
121            gain = getSpeed(poseInverted, 0.008);
122            md.setM2Speed(gain);
123        }
124
125        else
126        {
127            md.setM2Brake(400);
128        }
129    }
130

```

```

131
132 //I2C communication with DAQ Arduino
133 void requestPose(){
134     Wire.requestFrom(8, 8);
135     while(Wire.available()) {
136         I2C_readAnything(posePendulum);
137         I2C_readAnything(poseCart);
138     }
139 }
140
141
142 //Calculate speed from the error values
143 float getSpeed(float pose, float elapsed_time) {
144     error = setpoint - pose;
145     cum_error = cum_error + (elapsed_time*error)/divisor
146         ;
147     diff_error = ((error - last_error)*divisor)/((
148         elapsed_time));
149     last_error = error;
150
151     voltage = constrain((Kp*error + Kd*diff_error + Ki*
152         cum_error), -12, 12);
153     Speed = map(voltage, -12, 12, -400, 400);
154
155
156 // ----- TIMER ISR and Functions -----//
```

157

158

```

159 ISR(TIMER2_OVF_vect) {
160     TCNT2 = 386;
161     differentialPosePendulum = posePendulum -
162         posePendulumOld;
163     velocityPendulum = differentialPosePendulum/0.032;
164     differentialPoseCart = poseCart - poseCartOld;
165     velocityCart = differentialPoseCart/0.032;
166     posePendulumOld = posePendulum;
167     poseCartOld = poseCart;
168     control = true;
169
170
171 void setTimer2() {
172     ASSR |= (0<<AS2);
173     //Clear bit to use I/O or system clock i.e. 16 Mhz
174 }
```

```

175 //reset TC2 register
176 TCCR2A = 0x00;
177 TCCR2B = 0x00;
178 TCNT2 = 386;
179
180 //Set timer 2 in PWM Phase-Correct Mode i.e. TCNT
181 // counts up to OCRA and then down to zero at which
182 // overflow flag is triggered
183 TCCR2A |= (0<<WGM20);
184 TCCR2A |= (0<<WGM21);
185 TCCR2B |= (0<<WGM22);
186
187 //Set timer 2 with 1024 Prescaler = 15.625 Khz
188 TCCR2B |= (1<<CS20);
189 TCCR2B |= (1<<CS21);
190 TCCR2B |= (1<<CS22);
191
192 }
193
194
195
196 void LimitSwitchCheck() {
197 L1 = digitalRead(Lswitch1);
198 L2 = digitalRead(Lswitch2);
199
200 if ((prevL1 == 0 && L1 == 1) && (MovingLeft == true ||
201 initial == true)){
202 initial = false;
203 MovingRight = false;
204 MovingLeft = true;
205 } else if ((prevL1 == 1 && L1 == 1) && (MovingLeft ==
206 true || initial == true)){
207 md.setM2Brake(400);
208 delay(50);
209 //Serial.println("Ouch");
210 MovingRight = true;
211 MovingLeft = false;
212 md.setM2Speed(-400);
213 }
214
215 if ((prevL2 == 0 && L2 == 1) && (MovingRight == true ||
216 initial == true)){
217 MovingRight = true;
218 MovingLeft = false;

```

```

218     initial = false;
219 }
220 else if ((prevL2 == 1 && L2 == 1) && (MovingRight ==
221     true)) {
222     md.setM2Brake(400);
223     delay(50);
224     //Serial.println("Yoo");
225     md.setM2Speed(400);
226     MovingRight = false;
227     MovingLeft = true;
228 }
229
230 prevL1 = L1;
231 prevL2 = L2;
232 }
233
234 template <typename T> unsigned int I2C_writeAnything (
235     const T& value)
236 {
237     Wire.write((byte *) &value, sizeof (value));
238     return sizeof (value);
239 } // end of I2C_writeAnything
240
241 template <typename T> unsigned int I2C_readAnything(T&
242     value)
243 {
244     byte * p = (byte*) &value;
245     unsigned int i;
246     for (i = 0; i < sizeof value; i++)
247         *p++ = Wire.read();
248     return i;
249 } // end of I2C_readAnything

```

F.2 Code Used for DAQ Arduino

The following code was uploaded to the Arduino which was responsible for receiving sensor data and communicating them to the controller.

```

1 #include <Wire.h>
2
3 #define SCLpin 21
4 #define SDApin 20
5
6 const int encoderA = 18;
7 const int encoderB = 19;

```

```

8 const int encoderA2 = 3;
9 const int encoderB2 = 2;
10 const int16_t Address = 0x08;
11
12 int16_t auxilliary = 0;
13 bool interrupt = false;
14 unsigned long timeDelay = 0;
15
16 int A = 0;
17 int B = 0;
18 int A2pos = 0;
19 int B2pos = 0;
20 int Acomp = 0;
21 int Bcomp = 0;
22 int countState = 0;
23 int trig = 0;
24
25 int BatArise = 0;
26 int AatBrise = 0;
27 int BatAfall = 0;
28 int AatBfall = 0;
29
30 volatile float posePendulum = 0;
31 volatile float poseCart = 0;
32 volatile float differentialPosePendulum = 0;
33 volatile float differentialPoseCart = 0;
34 volatile float posePendulumOld = 0;
35 volatile float poseCartOld = 0;
36 volatile float velocityPendulum = 0;
37 volatile float velocityCart = 0;
38
39 volatile float counter, counter_comp, counter2 = 0;
40 //This variable will increase or decrease depending on
   the rotation of encoder
41
42 void setup()
43 {
44   Serial.begin(38400);
45   Wire.begin(Address);
46   Wire.onRequest(requestEvent);
47
48   pinMode(encoderA, INPUT_PULLUP);
49   // internal pullup input pin 2
50   pinMode(encoderB, INPUT_PULLUP);
51   // internal pullup input pin 3
52   pinMode(encoderA2, INPUT_PULLUP);
53   // internal pullup input pin 2
54   pinMode(encoderB2, INPUT_PULLUP);

```

```

55 // internal pullup input pin 3
56
57 attachInterrupt(digitalPinToInterruption(encoderA), A_isr,
58   CHANGE);
58 attachInterrupt(digitalPinToInterruption(encoderB), B_isr,
59   CHANGE);
59 attachInterrupt(digitalPinToInterruption(encoderA2),
60   A2_isr, CHANGE);
60 attachInterrupt(digitalPinToInterruption(encoderB2),
61   B2_isr, CHANGE);
61 }
62
63
64 //Calculate pose values for the pendulum and cart
65 void loop()
66 {
67   posePendulum = ((counter/(10000))*360.0);
68   poseCart = ((counter2/(10000))*360.0)
69   *0.007*0.0174533;
69 }
70
71 void requestEvent() {
72   I2C_writeAnything(posePendulum);
73   I2C_writeAnything(poseCart);
74 }
75
76
77 // ----- ENCODER ISR's ----- //
78
79 void A_isr(){
80   A = digitalRead(encoderA);
81   B = digitalRead(encoderB);
82
83   if (A==HIGH)
84   {
85     if (B==HIGH) { counter--; }
86     else if (B==LOW) { counter++; }
87   }
88   else if (A==LOW)
89   {
90     if (B==HIGH) { counter++; }
91     else if (B==LOW) { counter--; }
92   }
93 }
94
95 void B_isr(){
96   A = digitalRead(encoderA);
97   B = digitalRead(encoderB);

```

```

98
99     if  (B==HIGH)
100    {
101        if  (A==HIGH)      { counter++;   }
102        else if (A==LOW) { counter--;   }
103    }
104    else if (B==LOW)
105    {
106        if  (A==HIGH)      { counter--;   }
107        else if (A==LOW) { counter++;   }
108    }
109 }
110
111 void A2_isr(){
112     A2pos = digitalRead(encoderA2);
113     B2pos = digitalRead(encoderB2);
114
115     if  (A2pos==HIGH)
116     {
117         if  (B2pos==HIGH)      { counter2--;   }
118         else if (B2pos==LOW) { counter2++;   }
119     }
120     else if (A2pos==LOW)
121     {
122         if  (B2pos==HIGH)      { counter2++;   }
123         else if (B2pos==LOW) { counter2--;   }
124     }
125 }
126
127 void B2_isr(){
128     A2pos = digitalRead(encoderA2);
129     B2pos = digitalRead(encoderB2);
130
131     if  (B2pos==HIGH)
132     {
133         if  (A2pos==HIGH)      { counter2++;   }
134         else if (A2pos==LOW) { counter2--;   }
135     }
136     else if (B2pos==LOW)
137     {
138         if  (A2pos==HIGH)      { counter2--;   }
139         else if (A2pos==LOW) { counter2++;   }
140     }
141 }
142
143
144 template <typename T> unsigned int I2C_writeAnything (
    const T& value)

```

```
145  {
146  Wire.write((byte *) &value, sizeof (value));
147  return sizeof (value);
148 } // end of I2C_writeAnything
149
150 template <typename T> unsigned int I2C_readAnything(T&
151 value)
152 {
153     byte * p = (byte*) &value;
154     unsigned int i;
155     for (i = 0; i < sizeof value; i++)
156         *p++ = Wire.read();
157 } // end of I2C_readAnything
```

APPENDIX G

COMPARISON OF ENCODER AND VISUAL FEEDBACK

The following sections detail plots comparing the pendulum angular displacement feedback obtained from the rotary encoders to the feedback obtained from the ZED 2 camera, for situations in which the cart is located to the left and right of the camera in its field of view, respectively.

G.1 Cart is placed to the Left of the Camera

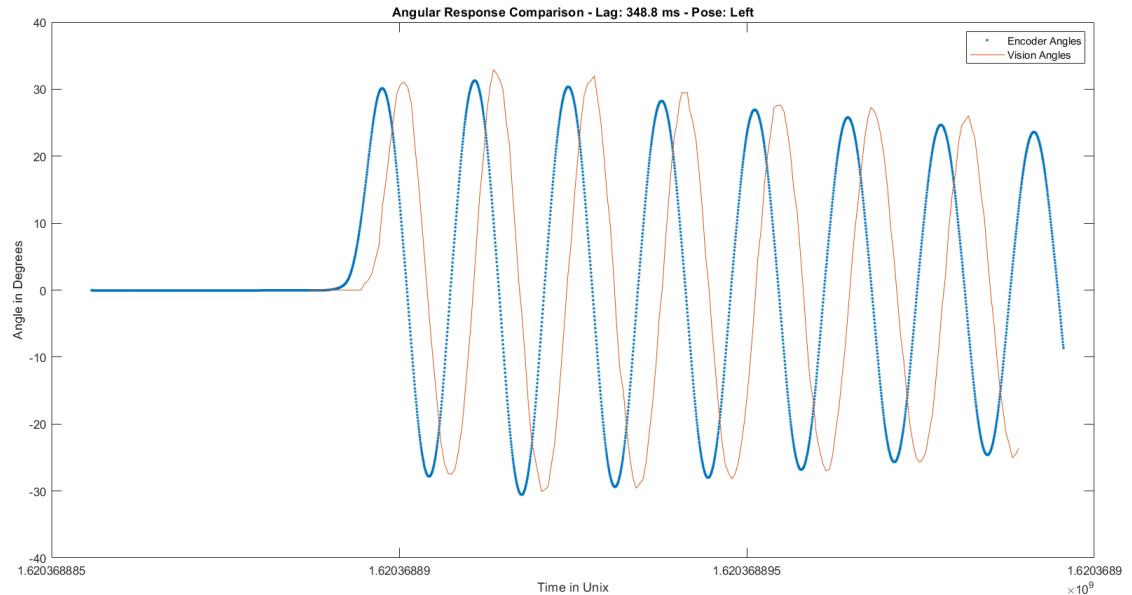


Figure G.1: Comparison between encoder and visual feedback - with lag

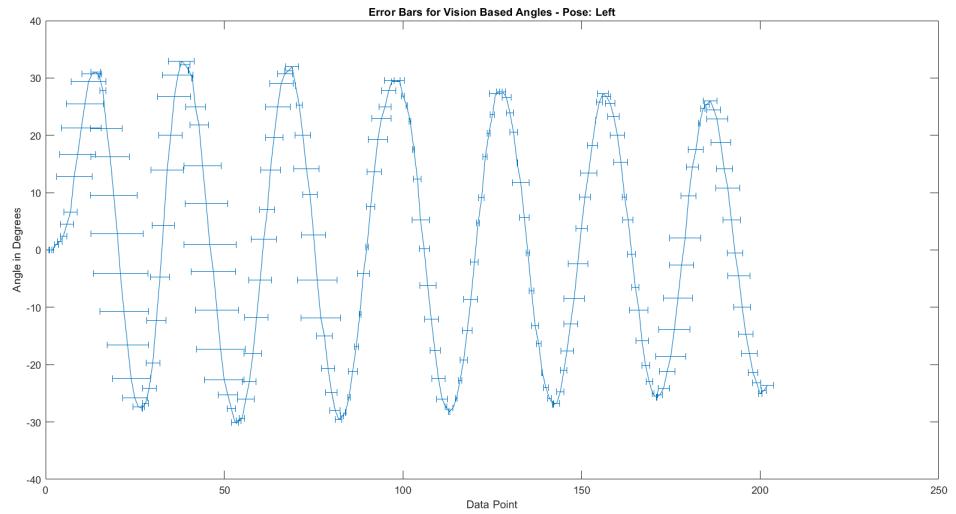


Figure G.2: Horizontal error bars for the discrepancy between encoder and visual feedback

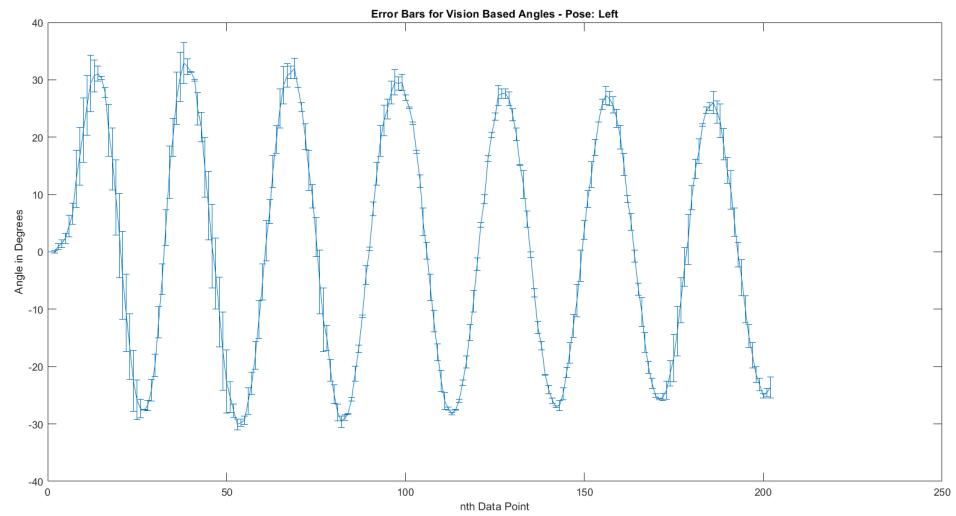


Figure G.3: Vertical error bars for the discrepancy between encoder and visual feedback

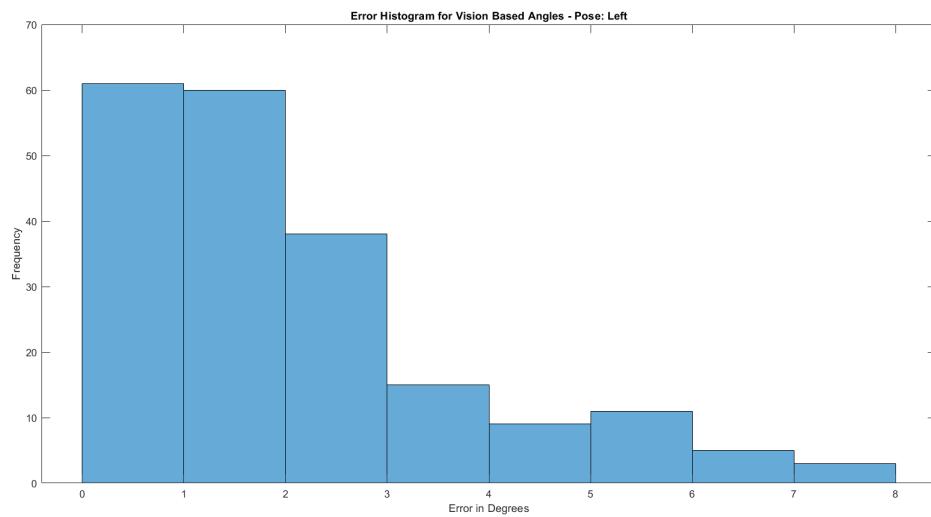


Figure G.4: Histogram of error values - compensated for Lag

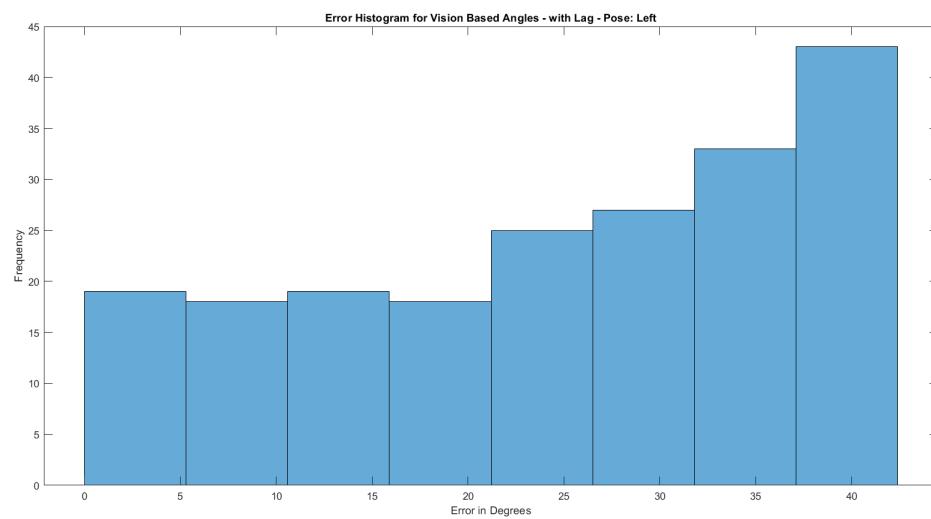


Figure G.5: Histogram of error values - with Lag

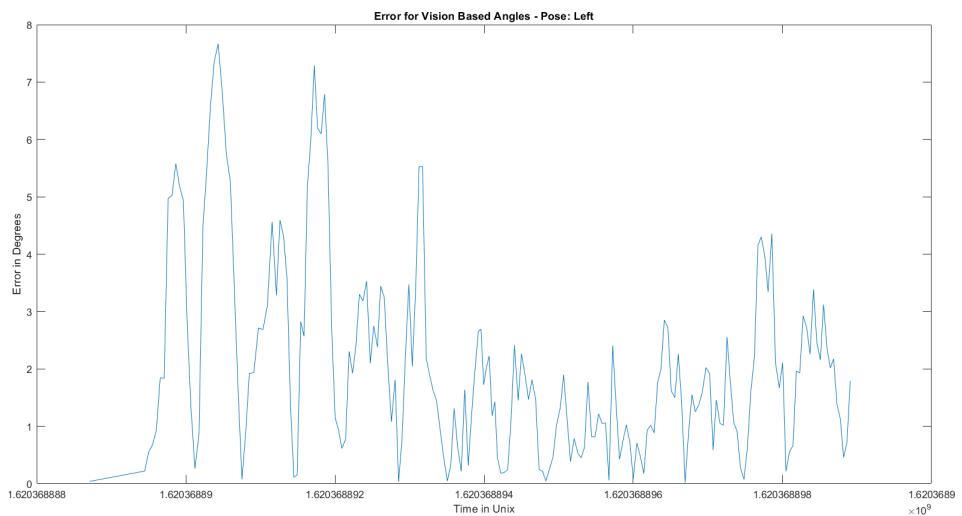


Figure G.6: Plot of error versus time - compensated Lag

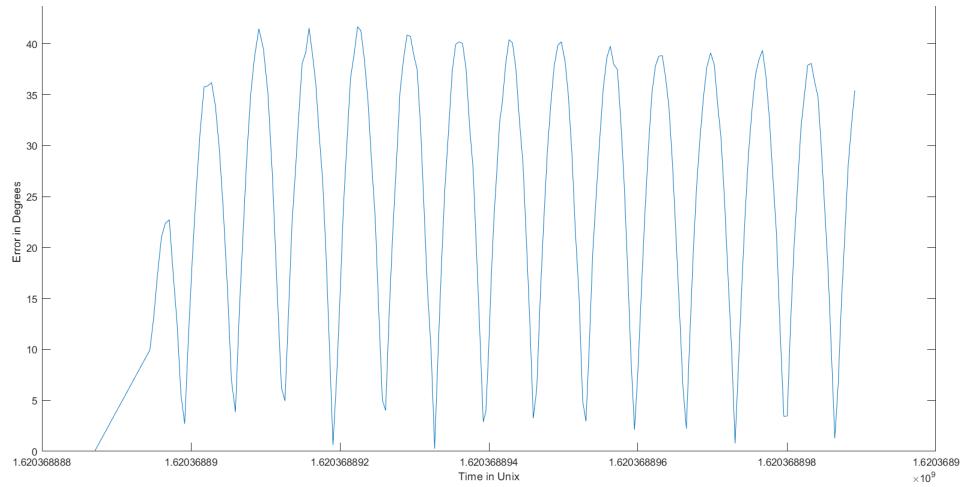


Figure G.7: Plot of error versus time - with Lag

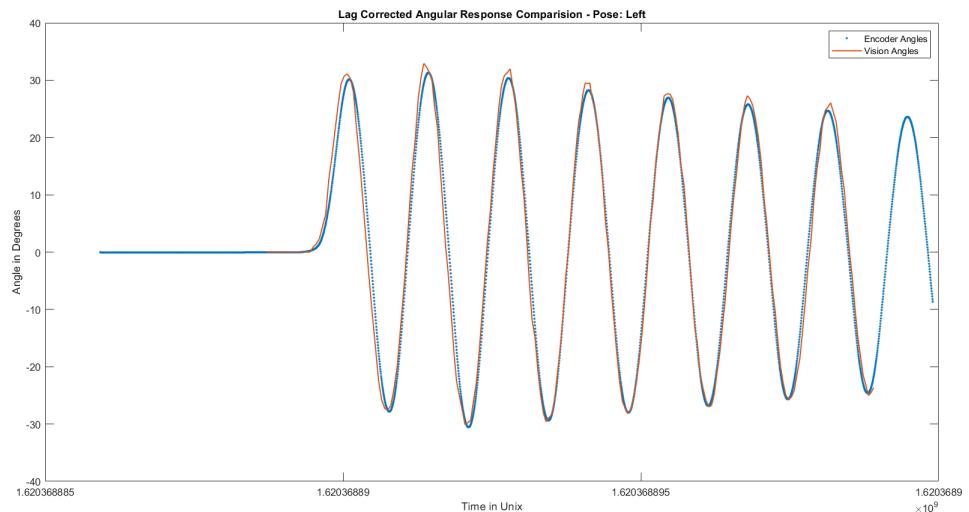


Figure G.8: Comparison between encoder and visual feedback - compensated for Lag

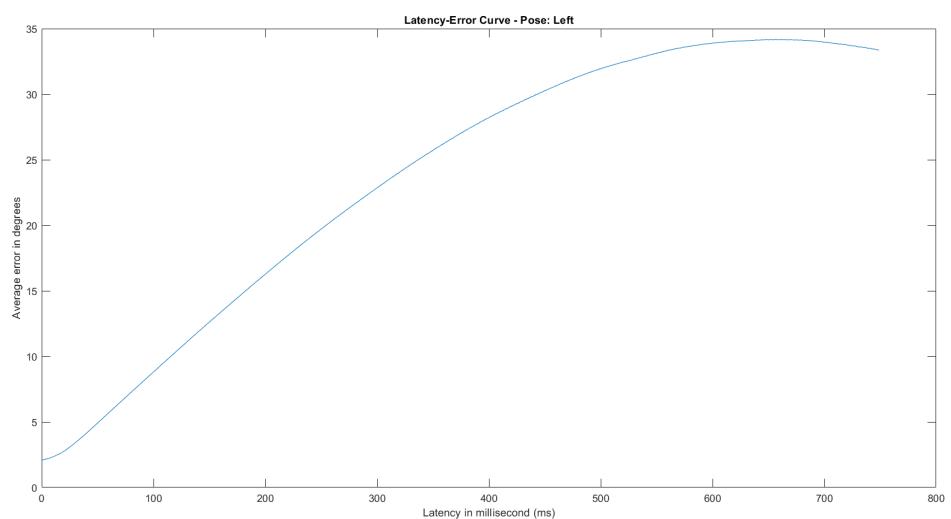


Figure G.9: Plot of error versus latency

G.2 Cart is placed to the Right of the Camera

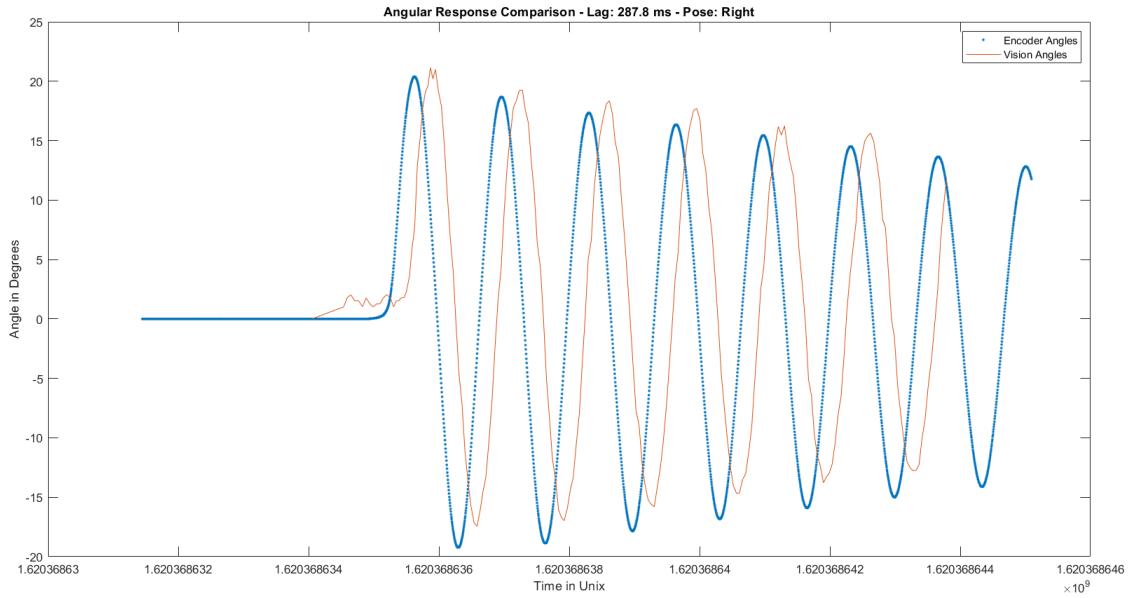


Figure G.10: Comparison between encoder and visual feedback - with lag

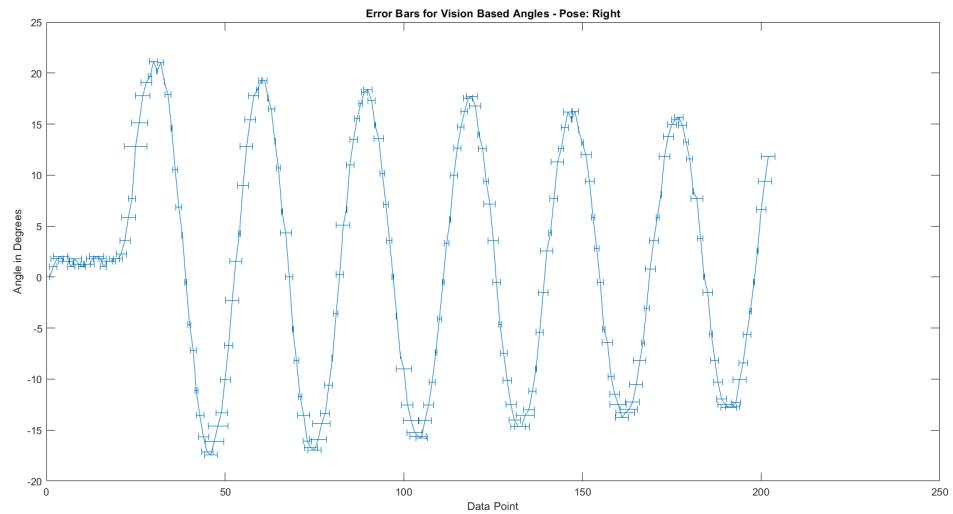


Figure G.11: Horizontal error bars for the discrepancy between encoder and visual feedback

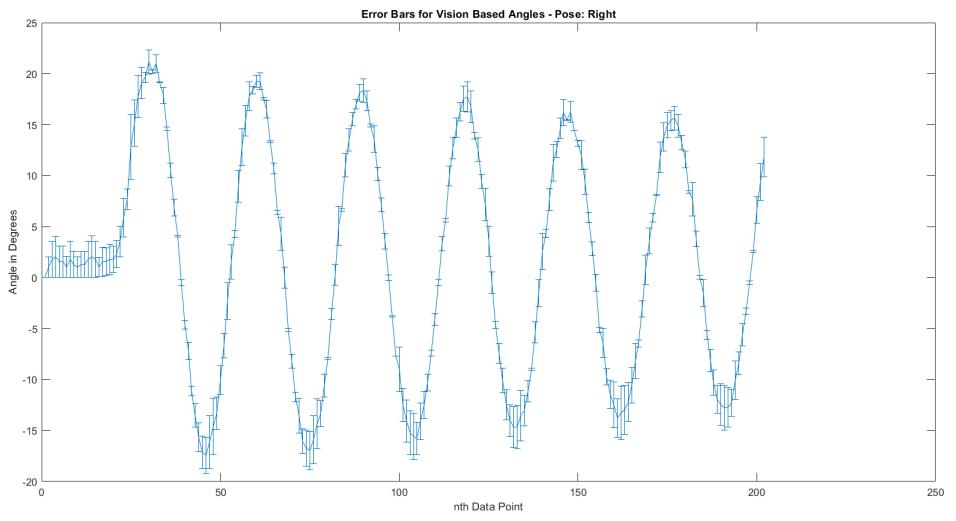


Figure G.12: Vertical error bars for the discrepancy between encoder and visual feedback

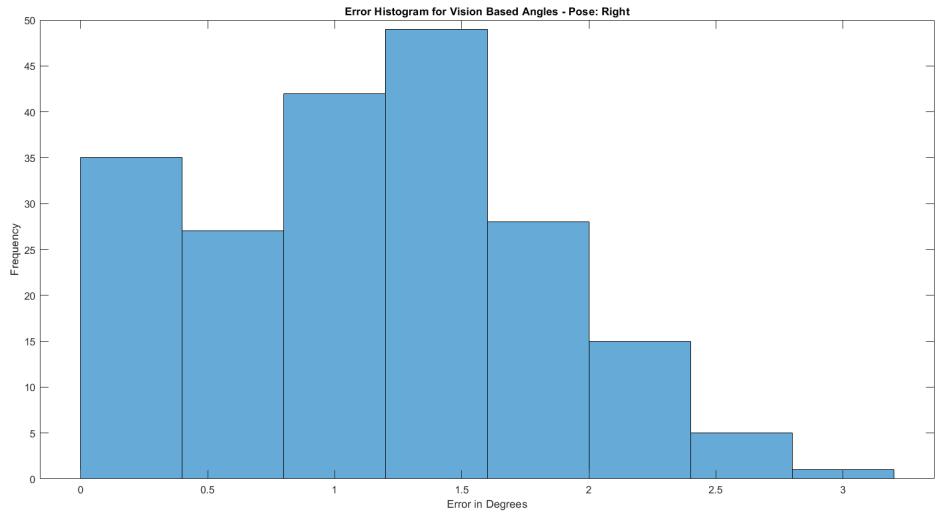


Figure G.13: Histogram of error values - compensated for Lag

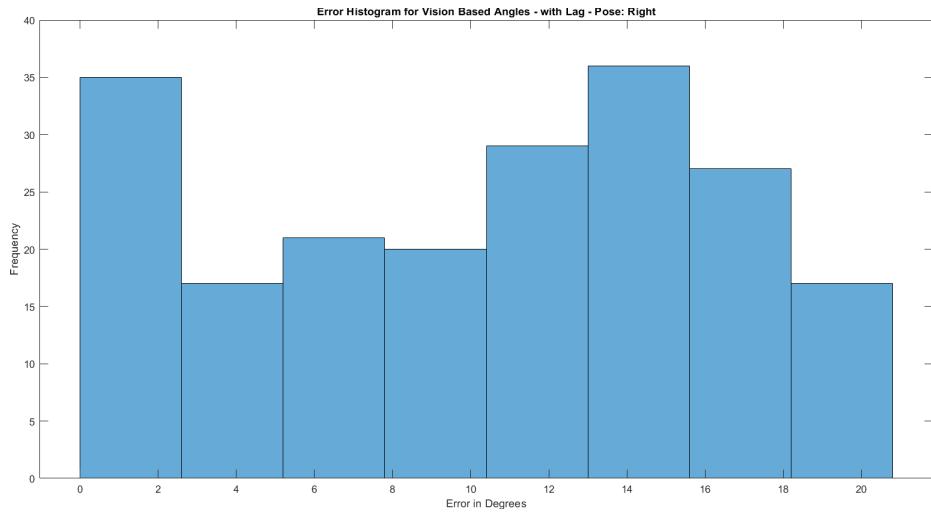


Figure G.14: Histogram of error values - with Lag

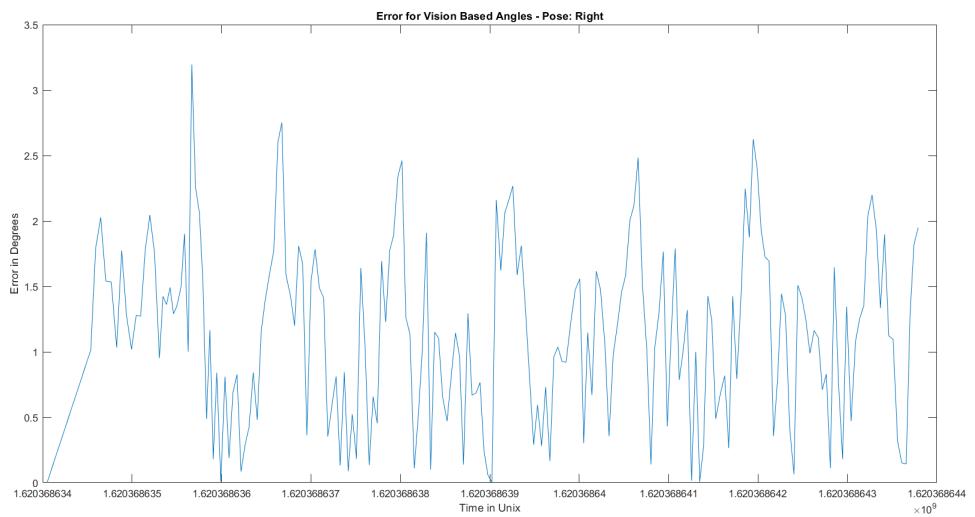


Figure G.15: Plot of error versus time - compensated Lag

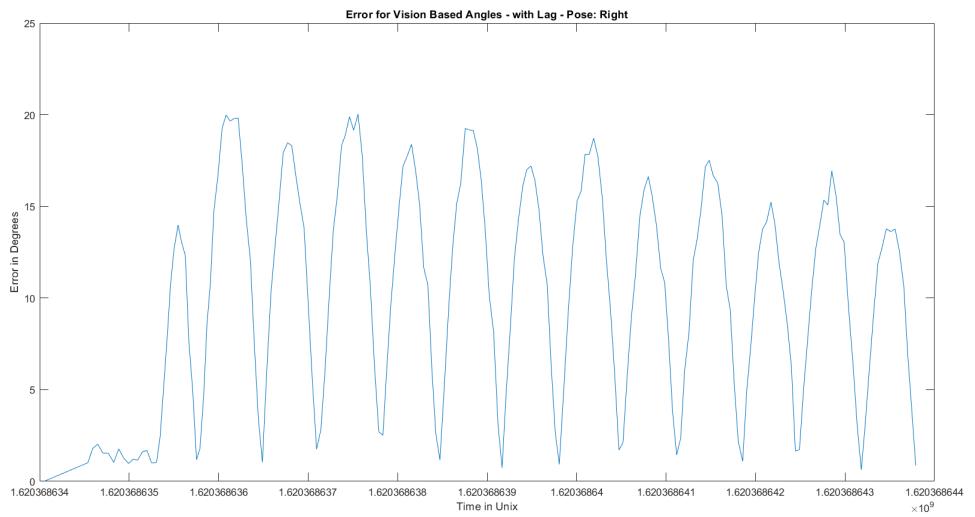


Figure G.16: Plot of error versus time - with Lag

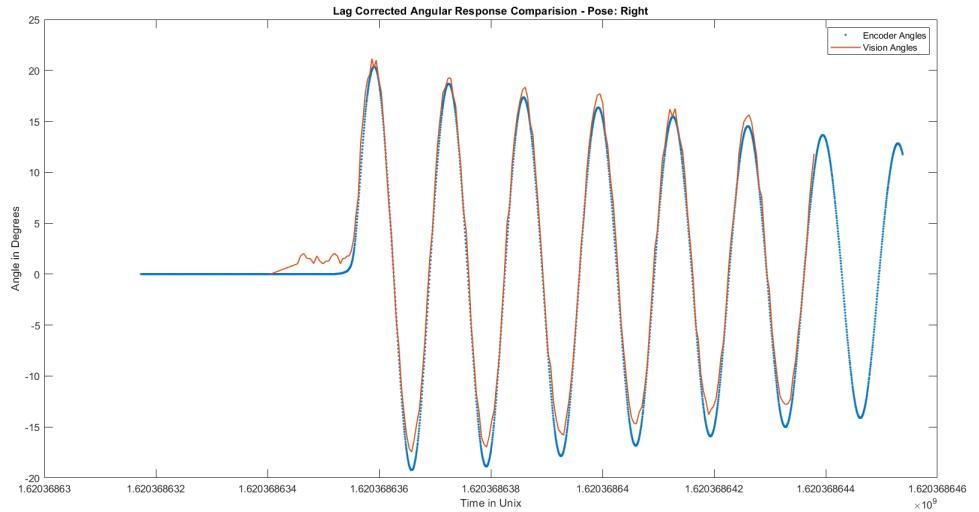


Figure G.17: Comparison between encoder and visual feedback - compensated for Lag

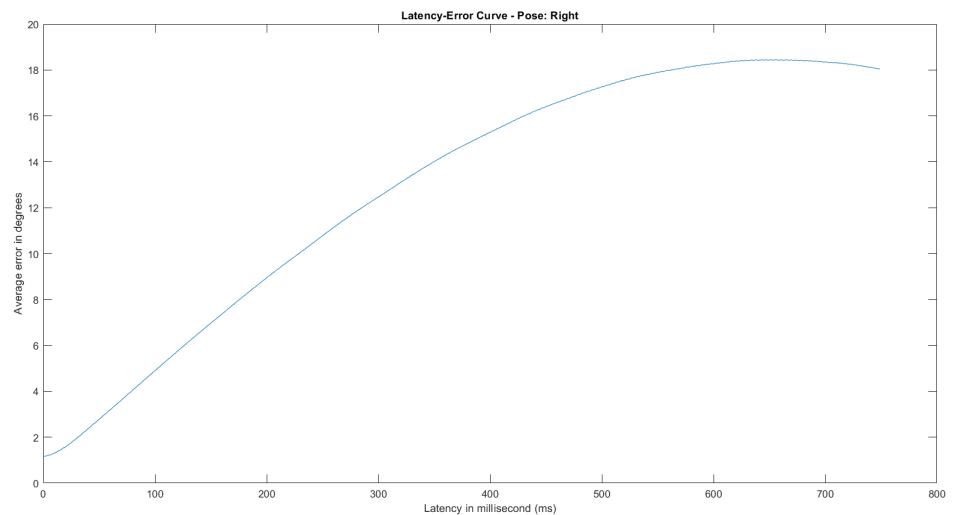


Figure G.18: Plot of error versus latency

APPENDIX H

BILL OF MATERIALS

The bill of materials incurred while constructing this project is shown in table H.1.

Table H.1: Bill of Materials

Item	Cost in PKR
5m timing belt	900
2× 8mm pulleys	1600
2× Linear ball bearings	860
2× U-Clamps	60
10mm thick acrylic sheet	1100
2× Cylindrical bearings	180
2× Cylindrical bearing holders	600
2× Heat sink & thermal paste	150
2× SICK rotary encoder	5500
Pendulum (Aluminum pipe)	500
2× Limit switches	50
2× 2m stainless steel pipes	3500
2× Arduino Mega2560	3000
VNH5019 Motor Driver	5000
Pololu DC Gearmotor	4000
TOTAL BILL	27000

The total cost comes out to about PKR 27000 or USD 180 which is magnitudes cheaper than the platforms available commercially.

APPENDIX I

GANTT CHART

The Gantt chart in figure (I.1) shows the work distribution of the project, spanning from August 2020 to May 2021.

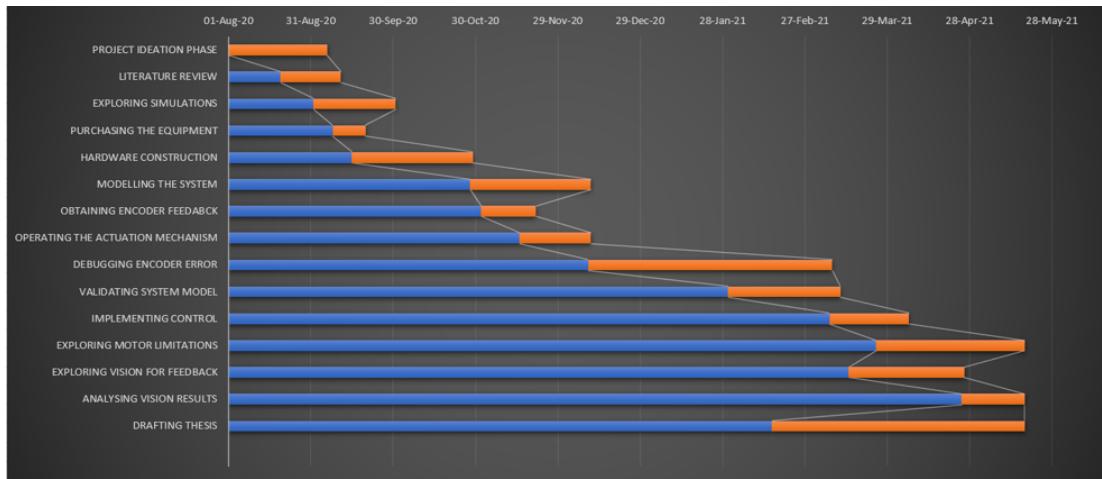


Figure I.1: Gantt Chart

APPENDIX J

USER MANUAL

J.1 Introduction

The objective of this user guide is to enable the reader to set up and utilize the cart-mounted inverted pendulum test-bench.

The inverted pendulum is a classical control problem which involves indirectly actuating a pendulum – which is essentially a rod that is free to move about a fixed point – in such a manner that the actuation leads to the pendulum attaining an upright inverted position with its center of mass directly above its pivot. This position is in contrast to its default orientation. Ideally, the pendulum is actively actuated or controlled to maintain this position despite the disturbances it may face. Fig. J.1 shows the setup of a cart-based inverted pendulum setup with a pendulum that is in the process of being actuated to its inverted state. The goal is to maintain $\theta = 180^\circ$ by controlling the force, F .

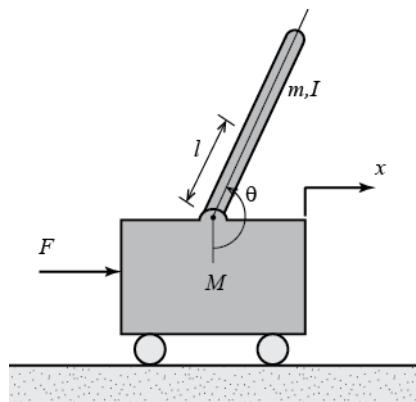


Figure J.1: An Inverted Pendulum Diagram [1]

J.2 System Specifications

Detailed system specifications can be found in chapter 4 in table 4.1.

J.3 Setting Up the Test-Bench

J.3.1 Checklist

Before getting started, you need to ensure that the platform is set up correctly for operation.

- Ensure that all the connections are in accordance with the circuit schematics given in appendix D. In particular, make sure that the:
 - Motor Driver is powered by a power supply no more than $24V/12A$ and no less than $5.5V$.
 - System Microcontrollers are powered by a power supply no more than $5V/1A$.
- Ensure that the belt is properly tensioned. To check, simply hold the belt and give it a sharp nudge in either direction. If no slippage occurs, the belt is properly tensioned. If you need to tension the belt, kindly refer to subsection J.3.2.
- Ensure that the cart is coupled tightly to the belt. You may check this by holding the cart between your thumb and forefinger and nudging it sideways slightly. If the cart does not budge, it is coupled tightly to the belt.

This check is only valid if the timing belt is properly tensioned. If the belt is not tensioned, you will have to hold the belt on one side of the cart with one hand, and nudge the cart in the opposite direction with the other hand. Repeat this procedure for both sides of the cart. If no slippage occurs, the belt and cart are properly coupled. If you need to couple the belt to the cart, kindly refer to subsection J.3.3.

- Ensure that the blocks on either end of the platform are coupled to a desk or some other solid surface tightly using the C-Clamps. Fig. J.2 shows a C-Clamp in action on the test-bench.

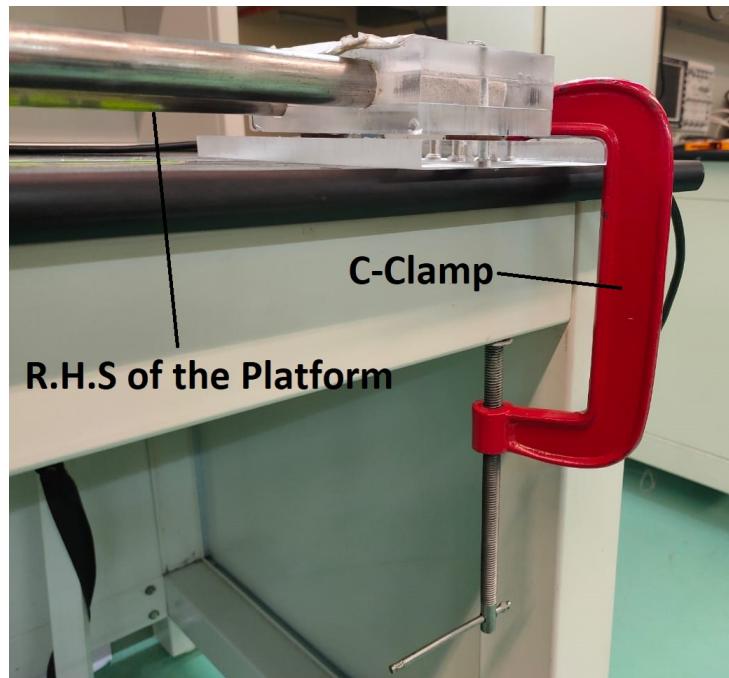


Figure J.2: Close-Up of the C-Clamp coupling the R.H.S Block to Desk

- Ensure that the track length between the limit switches is of the required magnitude. If not, adjust according to instructions in subsection J.3.4.

J.3.2 Tensioning the Timing Belt

Tensioning the timing belt is best achievable with two users. Consider the free pulley end of the test-bench. This is the right hand side of the platform. Check that the C-Clamp coupling the L.H.S block to the desk is tightened. Then, loosen the C-Clamp coupling the R.H.S block at the far-end to the desk. One user should hold both the pipes forming the tracks firmly, while the other user pulls the resting block to the right with one hand, and once it can not be extended anymore, tightens the C-Clamp, coupling the block to its new position on the desk. This ensures that the timing belt is properly tensioned. The process can also be carried out by only one user if required, but two users are recommended for ease.

If you need to slacken the belt, merely loosen the C-Clamps at the right most edge of the platform and nudge it slightly. The belt will loosen as required.

J.3.3 Coupling Cart to Timing Belt

The cart is coupled to the timing belt using two 3D printed clamps at the bottom of the backside of the cart. The clamps can be seen in Fig. 2.8. Two users are required for this procedure as well. Ensure that both C-Clamps are either loose or removed altogether from the platform. One user will hold the pipes and slightly twist them to raise the back side of the platform off the desk. The second user will then loosen the clamps located under the cart using a Phillips head screwdriver or equivalent. The second user may then stretch the belt to the required length before screwing it tightly. Once the cart is properly coupled to the timing belt, the users will have to carry out the process outlined in subsection J.3.2 to tension the belt before the platform can be used. Fig. J.3 shows the location on the underside of the cart where the timing belt is clamped to the cart.

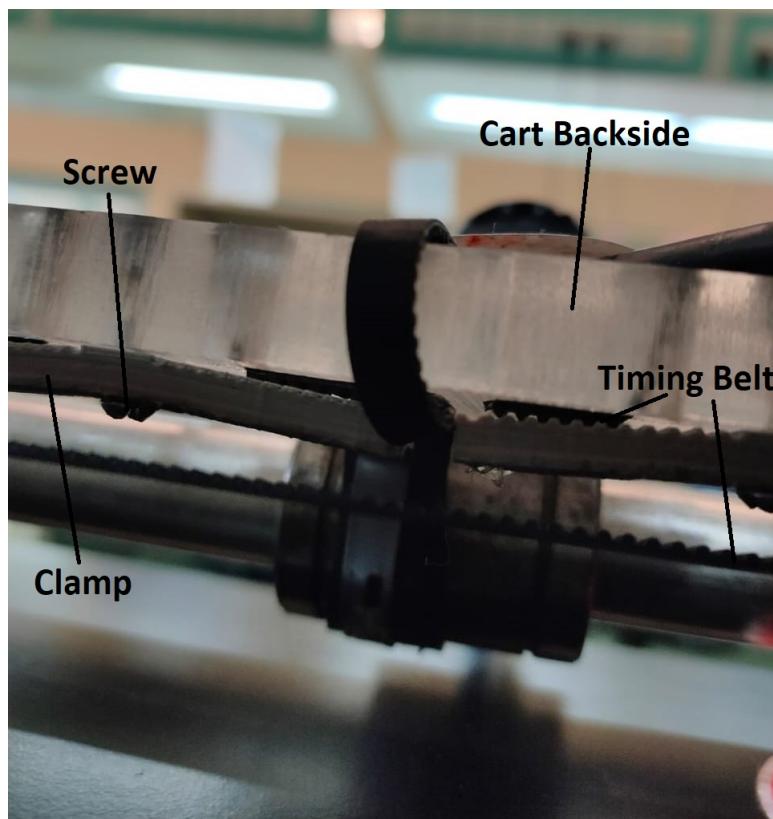


Figure J.3: Close-Up of the backside of the Cart

J.3.4 Varying the Track Length

The track length can easily be varied up to a maximum magnitude of $1.81m$ by moving the limit switches along the pipes. You can measure the distance of the track using a tape measure. You will need to use a slotted screw driver to loosen the hose clamp coupling the limit switch to the pipe. Then slide the limit switch to the desired distance. You should then tighten the hose clamp. Verify that the distance is correct once again before proceeding with the experiment. The limit switch, hose clamp and their connection to the pipe can be observed in Fig. J.4.

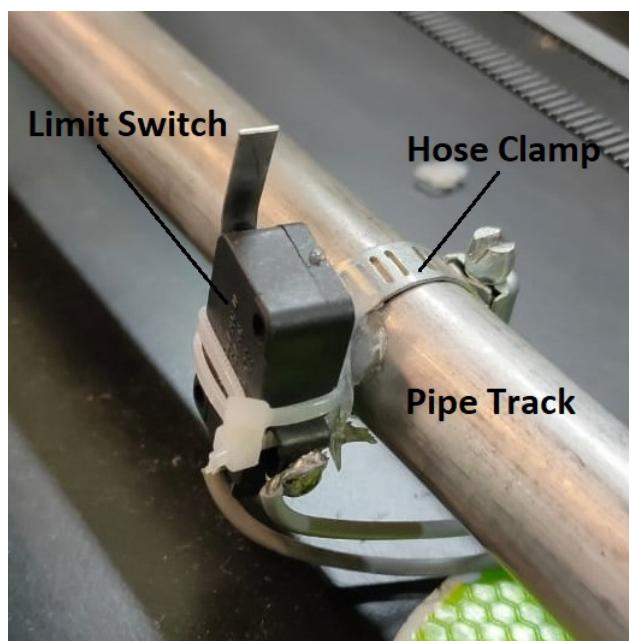


Figure J.4: Close-Up of the Limit Switch and Hose Clamp

J.4 Connecting Microcontrollers to your PC

There are two microcontrollers located on the test-bench labelled as DAQ Arduino and Controller Arduino respectively. You will need to connect these to your PC using a type A/B USB cable in order to program either of them. Once you have connected an Arduino to your PC, you will need to select its port on the Arduino IDE as follows:

Tools → Port → COMX

If you are unsure as to which port is your microcontroller is connected to, this information can be found in the control panel for your PC as follows:

Control Panel → Device Manager → Ports

It is advised that if you want to connect both Arduinos to the PC simultaneously, you should do that sequentially and note down the port of the first Arduino before you connect the second one. This is important because if you connect both Arduinos to the PC, you will know the two ports to which the Arduinos are connected but be unaware that which Arduino is connected to which port. This may lead to erroneous behaviour if you accidentally upload a code to the wrong Arduino, and is hence discouraged.

J.5 Programming the Microcontrollers

The codes with which both microcontrollers are programmed can be found in appendix F.2 and F.1.

If you are testing out a new control algorithm, you will primarily need to edit the control parameters at the beginning of the code (currently K_p , K_d , K_i), the expression in the function *getSpeed()* which maps control effort to motor speed, and the condition in *loop()* which determines the range of angles for which control is active.

J.6 Testing the Control Algorithm

After your algorithm has been uploaded, you will need to grab the pendulum and gently rotate it towards the inverted position until the controller takes over and the motor starts to move on its own. Thereafter, if your control algorithm is adequate, the system should reach a stabilized state about the inverted position.

J.7 Frequently Asked Questions

Q1. Why is there a grinding noise when the system is operational and the motor is rotating?

Turn off the system immediately. Check if the cart has run into a limit switch. Otherwise, check if the belt is tensioned as explained in subsections J.3.1 and

J.3.2. The grinding noise is most likely either due to the cart running into limit switches at either end of the track, or slippage between the motor and the timing belt.

Q2. Why is the cart wobbling side to side as it moves along the tracks?

This problem can occur if the linear bearings connecting the cart to the pipes are loosely coupled. The easy solution is just to check that the four screws connecting the two U-Clamps holding the two linear bearings to the cart are tightened properly. Fig. J.5 shows one linear bearing coupled to the underside of the cart using a U-Clamp.

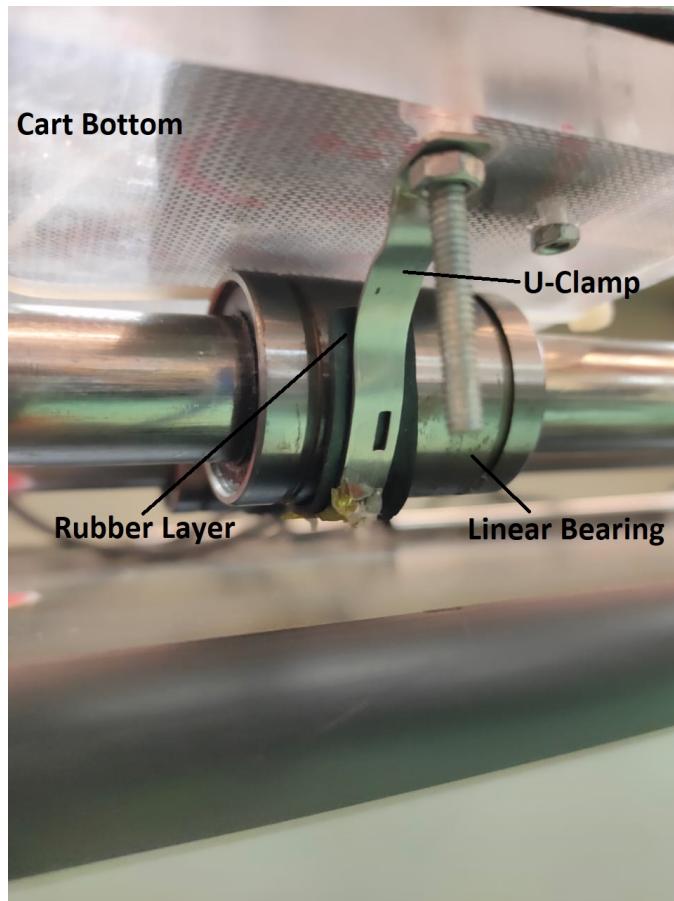


Figure J.5: Close-Up of the Linear Bearing and U-Clamp located under the Cart

Q3. Why is the motor not starting when the system is switched on?

There are several possible culprits. The most likely issue is a software error whereby the motor speed is initialized as zero. Make sure you debug the code before proceeding to hardware solutions.

To ensure that system hardware is working properly, first of all, disconnect the motor connections from the circuit and connect to an isolated power supply. Operating the motor in this fashion will confirm that the motor is indeed operational. One possibility for the error is that the motor connections to the motor driver have grown lose, so make sure to tighten them and check if the system works. Similarly, check that the limit switch connections to the circuit board are tight as well. If all of these requirements are met and the system still doesn't work, you will need to individually check all the connections between the motor driver and the controller Arduino to ensure that no connection has been lost. You will find it helpful to refer to the circuit schematic in appendix D for this purpose.

Q4. How do I fix a lose timing belt?

Refer to subsection J.3.2.

Q5. How do I replace the motor attached to the platform?

This is a simple process. First of all, make sure that the C-Clamps clamping the platform onto the desk are lose. This will slacken the belt. You may need to physically pull the belt slightly in order to slacken it. Then gently grasp the belt right next to the motor, and pull gently until it pops off the pulley attached to the motor. Use a hex head screwdriver to loosen and remove the pulley from the motor shaft. Use a torx head screw driver to disconnect the motor from its bracket. Place the new motor on to the bracket and screw it on tightly. Mount the pulley on the motor shaft again and tighten it. Then gently pull the timing belt back onto the pulley until it snaps on and fits the pulley. Follow the steps in subsection J.3.2 to tension the timing belt. Then nudge the motor and the belt right next to it in order to ensure that there is no slack anywhere in the locomotion mechanism. Fig. J.6 shows how the motor is coupled to the pulley and the timing belt.

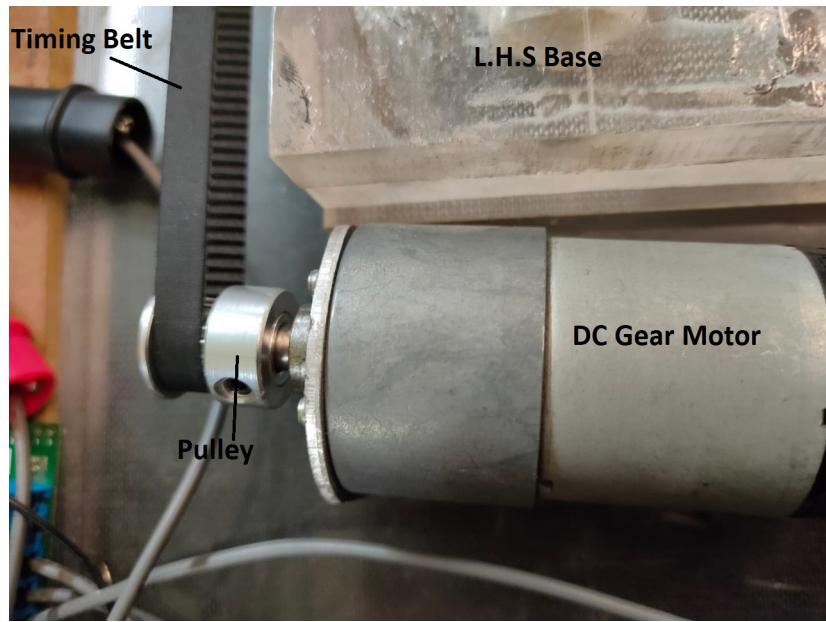


Figure J.6: Close-Up of the Motor connected to a Pulley and Timing Belt

Q5. How do I change the pendulum connected to the cart?

Note that currently the platform only supports 8mm thick rods as pendulums. To remove the existing pendulum, you merely need to use a Phillips head screw driver to unscrew the pendulum from the coupler that attaches it to the shaft mounted in the rotary encoder. The new rod can then be inserted into the couple and the screw tightened again. Make sure to give it a gently pull to ensure there is no slippage between the new pendulum and the coupler. Fig. J.7 shows how the pendulum is mounted upon the encoder via a coupler and screw connection.

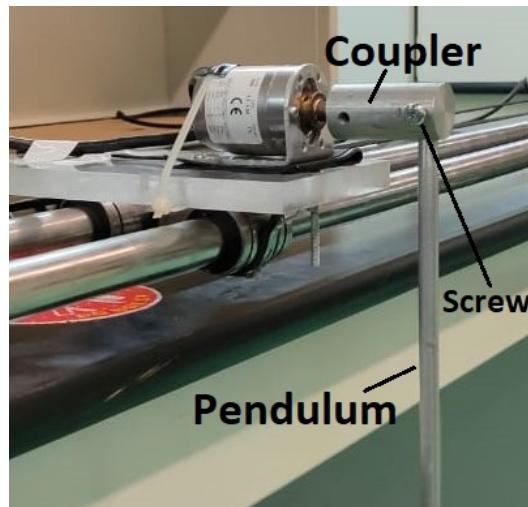


Figure J.7: Close-Up of the Pendulum coupled to Encoder

APPENDIX K

VIDEOS

K.1 Overview of System Hardware

The video showcases the system "big picture" as well as individual system components in order to provide the viewer of a better idea of how the system components come together. The video can be accessed at <https://youtu.be/DwfV664JOYo>.

K.2 Open Loop System Response

The open loop system response was recorded after multiple attempts to control the system robustly had failed. It was hypothesized that the inability to control the pendulum was not on account of sensor latency or algorithmic error, but motor limitation. It was judged that if the open loop test would display the pendulum falling in one direction only despite the motor moving to oppose it, that meant that the motor was insufficient for the purposes of this project. The video can be accessed at <https://youtu.be/-jagtOygaDo>.

K.3 Controlling the System

This video showcases the results of attempting to control the system. It can be observed that the system is able to withstand a small disturbance. However, a large disturbance of greater than 0.5 degree prevent it from regaining the equilibrium point. This limitation exists due to motor constraint which is unable to provide the control effort required to stabilize the system. The video can be accessed at https://youtu.be/sOku2s0o_Pc.

K.4 Visual Feedback from the System

The visual feedback was implemented for the platform using a ZED 2 camera. The computer vision algorithm uses the OpenCV library to track the pendulum bob and cart

center in order to calculate the pendulum angular displacement. The video demo can be accessed at <https://www.youtube.com/watch?v=rKELbMM6-yk>.

REFERENCES

- [1] MATLAB, “Control tutorials for matlab & simulink: Inverted pendulum: System modeling,” 2017, Available: <https://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=SystemModeling> [Accessed 3-12-2020].
- [2] O. Boubaker and R. Iriarte, *The Inverted Pendulum in Control Theory and Robotics: From Theory to New Innovations*. Oct. 2017, ISBN: 978-1-78561-320-3.
- [3] U. Yaman and Y. Silik, “Control of rotary inverted pendulum by using on–off type of cold gas thrusters,” *Actuators*, vol. 9, p. 95, Sep. 2020.
- [4] J. Apkarian, H. Lacheray, and P. Martin, “Student workbook: Linear inverted pendulum experiment for MATLAB/simulink users,” 2020, Available: shorturl.at/mKXZ7 [Accessed: 26-11-2020].
- [5] G. Belascuen and N. Aguilar, “Design, modeling and control of a reaction wheel balanced inverted pendulum,” pp. 1–9, 2018.
- [6] Quanser, “Rotary Inverted Pendulum,” Available :<https://www.quanser.com/products/rotary-inverted-pendulum/> [Accessed 3-12-2020].
- [7] Stang, “Thesis - the inverted pendulum - a design project,” May 2005.
- [8] Quanser, “Quanser Linear Inverted Pendulum Information,” Available: shorturl.at/mKXZ7 [Accessed: 26-11-2020].
- [9] Leybold, “Leybolds Product Information,” Available: <https://www.leybold-shop.com/technology/electrical-engineering/control-engineering-and-automation/applied-control-technology/technical-processes-with-large-scale-models/inverted-pendulum/ve6-3-5-11.html> [Accessed: 26-11-2020].
- [10] G. Lee and S. Jung, “Design and control of an inverted pendulum system for intelligent mechatronics system control education,” Jul. 2008.
- [11] W. White, J. Wagner, B. Blankenau, Z. Wang, and V. Salazar, “Design, build, and test of an autonomous inverted pendulum cart,” *Proceedings of the American Control Conference*, pp. 5893–5898, Jun. 2013.
- [12] P. Bakarac, M. Kaluz, and Ľ. Čirka, “Design and development of a low-cost inverted pendulum for control education,” pp. 398–403, Jun. 2017.
- [13] K. Mouratis, J. Fasoulas, and M. Sfakiotakis, “Development and control of a low-cost cart-pendulum educational platform,” Sep. 2018.

- [14] R. Tariq, S. Ali, and S. Rasheed, “Undergraduate thesis - design and implementation of control system for inverted pendulum,” 2010.
- [15] L. Wenzel, N. Vazquez, D. Nair, and R. Jamal, “Computer vision based inverted pendulum,” *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, vol. 3, pp. 1319–1323, Feb. 2000.
- [16] N. S. Nise, *Control Systems Engineering*, 3rd. USA: John Wiley & Sons, Inc., 2000, ISBN: 0471366013.
- [17] P. R. Electronics, “Pololu 37D Motor Datasheet,” Available: <https://www.pololu.com/file/0J1706/pololu-37d-metal-gearmotors.pdf> [Accessed 3-12-2020].
- [18] A. Kharola and P. Patil, “PID control of two-stage inverted pendulum,” pp. 1–4, Dec. 2016.
- [19] V. Pathak and S. Paliwal, “Analysis and control of inverted pendulum system using PID controller,” *International Journal of Engineering Research and Applications*, vol. 07, pp. 01–04, May 2017.
- [20] S. Jadlovska and J. Sarnovsky, “Classical double inverted pendulum – a complex overview of a system,” Jan. 2012.
- [21] T. Dabretau and A. Dareini, “Control of double inverted pendulum first approach,” 2015.
- [22] L. Dai, Y. Xia, M. Fu, and M. S. Mahmoud, *Advances in Discrete Time Systems*. InTechOpen, 2012.
- [23] M. Magaña and F. Holzapfel, “Fuzzy-logic control of an inverted pendulum with vision feedback,” *Education, IEEE Transactions on*, vol. 41, pp. 165–170, Jun. 1998.
- [24] M. M. Hasan, C. Saha, M. Rahman, M. Rabiu, M. R. I. Sarker, and S. Aditya, “Balancing of an inverted pendulum using pd controller,” *The Dhaka University Journal of Science*, vol. 1, pp. 1–6, Jun. 2012.
- [25] A. Johnsson and T. Rahlen, “Inverted pendulum stability regarding bandwidth and center of mass,” 2020.
- [26] K. Mouratis, J. Fasoulas, and M. Sfakiotakis, “Development and control of a low-cost cart-pendulum educational platform,” Sep. 2018.
- [27] M. Owais, A. Ul-Haque, H. Rahim, S. Aftab, and A. Jalal, “Control design and implementation of an inverted pendulum on a cart,” pp. 1–6, Dec. 2019.
- [28] K. Lawson, “Incremental quadrature encoder drift,” 2010.

- [29] F. Briz, J. A. Cancelas, and A. Diez, “Speed measurement using rotary encoders for high performance AC drives,” vol. 1, 538–542 vol.1, 1994.
- [30] H. Wang, A. Chamroo, C. Vasseur, and V. Koncar, “Hybrid control for vision based cart-inverted pendulum system,” *Proceedings of the American Control Conference*, pp. 3845–3850, Jul. 2008.
- [31] D. Honegger, H. Oleynikova, and M. Pollefeys, “Real-time and low latency embedded computer vision hardware based on a combination of fpga and mobile cpu,” *IEEE International Conference on Intelligent Robots and Systems*, Sep. 2014.
- [32] J. Lues, “Low latency vision-based control for robotics,” M.S. thesis, Massey University, 2018.
- [33] M. Haris, J. Memon, and M. Farhan, “Image processing techniques for fast and accurate estimation of pose of a double pendulum,” May 2020.
- [34] C. Kousheek, “Inverted pendulum: Control theory and dynamics, 2019,” Available: <https://www.instructables.com/Inverted-Pendulum-Control-Theory-and-Dynamics/> [Accessed 4-12-2020].
- [35] Madeinchina.com, “Solid Shaft Incremental Rotary Encoder,” Available: <https://yuncync.en.made-in-china.com/product/FBenNxZlZgrG/China-R141-O-360es-6mm-Solid-Shaft-Incremental-Rotary-Encoder.html> [Accessed 2-12-2020].
- [36] S. S. Intelligence, “DBS36 Core/DBS50 core: The multi-fit incremental encoder,” Available: <https://www.sick.com/ag/en/encoders/incremental-encoders/dbs36-core/dbs36e-bbal02500/p/p337953> [Accessed 25-11-2020].
- [37] Majju, “Limit-switch,” Available: <https://www.majju.com/product/lever-limit-switch/> [Accessed 3-12-2020].
- [38] Atmel, “Arduino Atmega 2560 Datasheet,” Available: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf [Accessed 2-12-2020].
- [39] G. I. M. P. Information, “GPS-2303/3303/4251/4303 Multi-Output D.C. Power Supply Datasheet,” Available: <https://www.gwinstek.com/en-global/products/downloadSeriesSpec/1200> [Accessed 3-12-2020].
- [40] P. R. Electronics, “Pololu Dual VNH5019 Motor Driver Shield for Arduino,” Available: https://cdn.sick.com/media/pdf/3/53/953/dataSheet_DBS36E-BBAL02500_1065070_en.pdf [Accessed 1-12-2020].

- [41] Y.-W. Tu and M.-T. Ho, “Design and implementation of robust visual servoing control of an inverted pendulum with an fpga-based image co-processor,” *Mechatronics*, vol. 21, pp. 1170–1182, Oct. 2011.
- [42] K. Mahmud, “Design and analysis of the control of an inverted pendulum system by matlab,” *2013 IEEE Global High Tech Congress on Electronics, GHTCE 2013*, pp. 207–211, Nov. 2013.
- [43] M. Stuflesser and M. Brandner, “Vision-based control of an inverted pendulum using cascaded particle filters,” *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, pp. 2097–2102, Jun. 2008.
- [44] S. A. R. Kashif, “Inverted pendulum model – differential equations,” Available: <https://youtu.be/OeMV0N8sU1Y> [Accessed 8-5-2021].
- [45] E. Eitel, “Basics of rotary encoder: Overview and new technologies,” Available: <https://www.machinedesign.com/automation-iiot/sensors/article/21831757/basics-of-rotary-encoders-overview-and-new-technologies> [Accessed 15-5-2021].
- [46] R. B. R. Staff, “Differences between encoder resolution, accuracy, and precision,” Oct. 2019, Available: <https://www.roboticsbusinessreview.com/news/differences-between-encoder-resolution-accuracy-and-precision/> [Accessed 15-5-2021].
- [47] K. S. University, “Perspective projection model,” Available: <http://faculty.salina.k-state.edu/tim/mVision/ImageFormation/projection.html> [Accessed 15-5-2021].
- [48] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. 2017, pp. 319–340, ISBN: 978-3-319-54412-0.
- [49] S. Olgierd, L. Gauthier, and D. Marek, “Multiview video: Acquisition, processing, compression, and virtual view rendering,” *Academic Press Library in Signal Processing*, vol. 6, 2018.
- [50] D. Pierre and L. Julien, “An exact formula for calculating inverse radial lens distortions,” *Sensors (Basel)*, vol. 9, Jun. 2016.
- [51] S. Labs, “Camera calibration,” Available: <https://www.stereolabs.com/docs/video/camera-calibration/> [Accessed 15-5-2021].
- [52] Google, “Googl Tech Direct Drive Inverted Pendulum,” Available: http://www.googltech.com/pro_view-65.html [Accessed 5-12-2020].

VITA

7.5 Sarwan Shah

Sarwan Shah is an electrical engineering major at Habib University. The courses that have added the most value to his undergraduate experience are Principles of Feedback and Control Systems, Instrumentation, Computer Vision, and Microcontrollers & Interfacing. Alongside his capstone project, he has also spent the last year and a half working on an exciting project related to water flow-meters and funded by the HEC. His hobbies include playing the piano and the guitar, and hanging out with his precious kitten 'Bano'.

7.6 Abdullah Siddiqui

Abdullah is completing his electrical engineering undergraduate degree from Habib University alongside a minor in physics. His favorite courses from his undergraduate studies have been Object Oriented Programming, Microcontrollers & Interfacing, Introduction to Robotics, Computer Vision, and Mobile Robotics. Interest in these courses is also what motivated him to work on the inverted pendulum as a control problem. Post-graduation, he currently plans to pivot to the software engineering domain and find work in the industry. In his spare time, Abdullah likes to binge-read books, particularly of the fantasy genre.

7.7 Adil Ali Khan

Adil Ali Khan is completing his undergraduate degree in electrical engineering from Habib University. Adil recently returned from an enriching experience at the University of Wisconsin-Stout, USA for a semester abroad through the Global UGRAD program. His interests lie in the domain of robotics and control, as well as the power industry. Adil also likes to study history in his free time, and play chess as a hobby.

