Inverted Pendulum Balancing using PID Controller

A Design Lab Project Report

Submitted by:

Aravind Nakka
(220102064)

Ojas Nune
(200108034)

Under the guidance of

Dr. Chayan Bhawal

Department of Electronics and Electrical Engineering
Indian Institute of Technology Guwahati
Guwahati - 781039, Assam

Contents

Chapter 1

Introduction

1.1 Problem

An inverted pendulum is a fascinating mechanical system (as shown in Fig. 1.1) where the center of mass is positioned above its pivot point. Unlike a regular pendulum that hangs downward and is naturally stable, the inverted pendulum is **inherently unstable**. it would fall over without assistance.

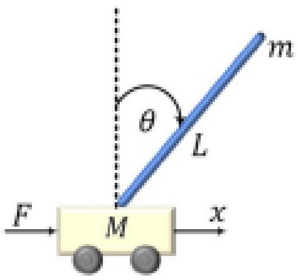

Figure 1.1: Simple Inverted Pendulum Balancing System

- To keep it upright, feedback control systems are used.

Now in this systems we have to:

- Monitor the pendulum's tilt angle continuously.
- Adjust the pivot position horizontally when the pendulum begins to lean.
- Maintain the balance by dynamically moving the base (or cart).

This setup is also referred to as a Broom-balancer, similar to the act of balancing a broomstick on one's palm.

A more advanced real-world analogy is:

- Missile or rocket attitude control during the launch stage.
- A booster rocket, like a broomstick, must be stabilized using thrust vectoring.

In this project, we simplify the problem by assuming:

- The pendulum's motion is constrained to a 2D plane.
- The cart moves only along a horizontal line.

1.2 Objective

The goal of this project is to design and construct an inverted pendulum balancing system that achieves stable and reliable performance.

The entire design can be broken down into two major areas:

1. Mechanical System Design

- Design the physical structure of the system.
- Select and design components like the cart, pendulum, and mounting platform.
- Identify suitable sensors to measure:
  - Linear displacement and velocity of the cart.
  - Angular displacement and velocity of the pendulum.

2. Control System Design

- Analyze the feedback control system.
- Build a mathematical model of the dynamic system.
- Derive the differential equations of motion.
- Linearize the model near the upright operating point.
- Evaluate the system's characteristics (stability, response time, etc.).
- Design a controller using:
  - State feedback control methods.
  - Open Loop Ziegler Nichols Method.
  - Closed Loop Ziegler Nichols Method.

Chapter 2

Design of the System

2.1 Basic Components of the System

Feedback control systems are closed-loop systems that generally consist of:

1. **The Plant:** The system to be controlled.
2. **Measurements:** Devices used to obtain information about the plant and send it back to the controller.
3. **The Controller:** The central processing unit of the control system, which compares the measured values of the plant with the desired values and adjusts the inputs to maintain the desired output.

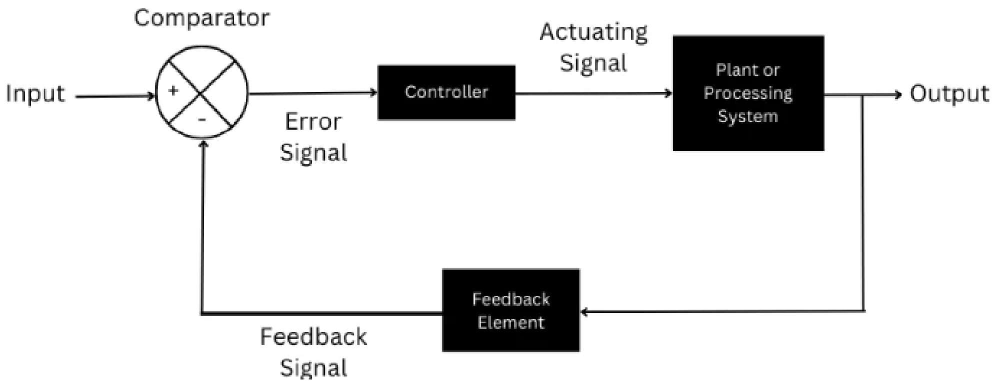

Figure 2.1: Block Diagram of Closed-loop Control System

The block diagram in Fig. 2.1 illustrates the closed-loop control concept. For the Inverted Pendulum Balancing system, the essential components are:

1. **Inverted Pendulum and Base:** The pendulum and the base on which the pivot is mounted.
2. **Driving System:** The actuating unit that controls the motion of the carriage.
3. **Sensors:** The measurement components that provide feedback data.

4. **Controller:** The intelligent part of the system that processes the feedback and sends actuation signals to the driving system.

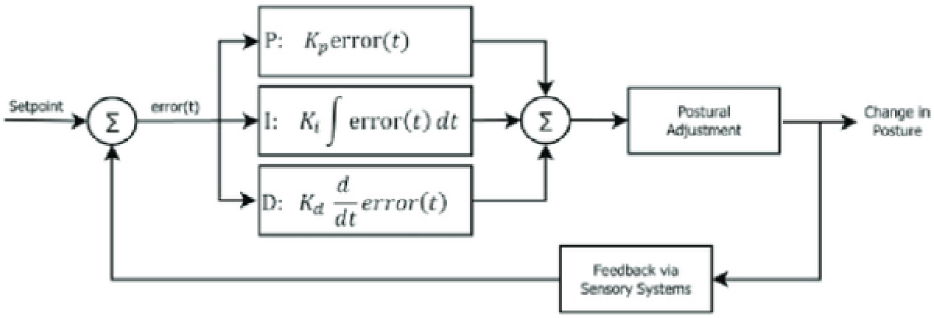

Figure 2.2: Components of Inverted Pendulum Balancing System

2.2 Decided Model

To choose the design model to build up for analysis and experiment, there are many factors to be considered, such as the available materials, the available components, the difficulty of making the parts, the complication of the mechanical structures and the cost of the instruments and materials, etc. After considering all these factors, we decided to proceed with the model as shown in Fig. 2.3.

2.2.1 Cart and Pendulum

As shown in Fig. 2.3, the cart (yellow colored) is mounted on a linear slide rail, which is fixed onto a wooden platform. This setup provides a stable base for the entire system. The cart is designed to move along the rail by applying torque through a pinion gear, which is connected to the shaft of a DC motor. To minimize friction and allow smooth movement, bearing balls and lubricants are used between the cart and the rail.

On top of the cart, a rectangular transparent acrylic sheet is mounted. This sheet serves as a platform for attaching various system components. Two L-shaped clamps are fixed to the acrylic sheet. One clamp is used to hold the optical rotary encoder (black colored), while the other is used to secure the DC motor (red colored).

The pendulum rod is connected to the rotary encoder via a white-colored clamp. The rotary encoder plays a crucial role in measuring the angular position of the pendulum, which is essential for the feedback control system.

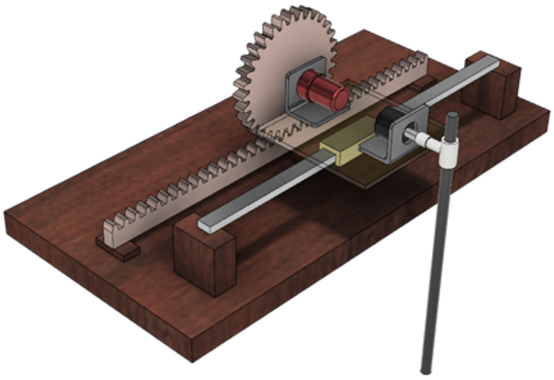

Figure 2.3: Orthographic View of the Model

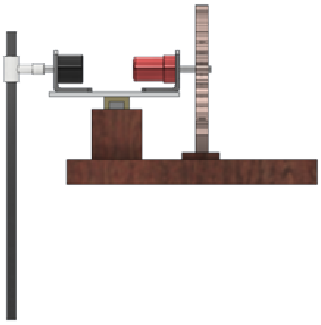

Fig. 2.3.1: Right Side View

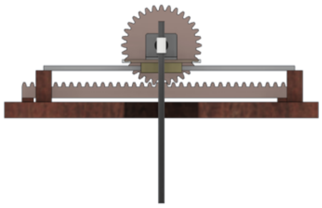

Fig. 2.3.2: Front View

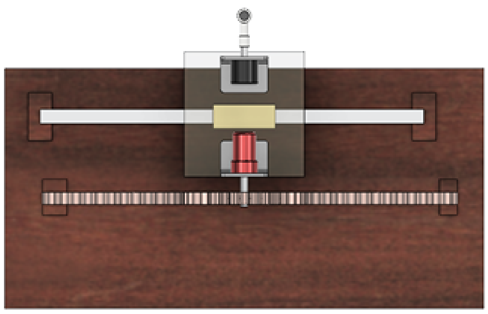

Figure 2.4: Top View

2.2.2 DC Motor

In the system, the DC motor is fixed on an L-clamp, and a 3D-printed pinion gear is attached to the shaft of the motor. The pinion gear runs on the rack gear that is attached to the wooden platform.

The DC motor selected for this purpose is **RHINO HEAVY DUTY PLANETARY GEARED MOTOR**. The specifications of the DC motor used in the design are as follows:

- Operating voltage = 12 V
- Motor Speed at Output Shaft = 103 RPM
- Stall Torque = 75 Kgfcm
- Rated Torque = 25 Kgfcm
- Gear Ratio = 1:174
- Weight = 300 g

Figure 2.5: Rhino 100RPM motor

SENSORS

An incremental optical rotary encoder has been used to measure the angular position of the pendulum. Hall effect magnetic encoder has been used to measure the angular position of the motor shaft and eventually the linear position of the cart.

Specifications of Rotary Optical Encoder:

- Model No. – 3806-OPTI-600-AB-OC
- Encoder Type – Incremental

Technical Details:

- Operating Voltage (V_{DC}) – (5 - 24)
- Current Consumption (mA) – $\leq$ 40
- Maximum Speed – 5000 RPM
- Pulse Per Revolution (PPR) – 600
- Counts Per Revolution (CPR) – 2400

Figure 2.6: Incremental Optical Rotary Encoder

Specification of the Hall Effect Magnetic Encoder:

- Model No. – SCX35O
- Input Supply voltage (V) – (4.5-18)
- Supply Current(A)-8
- Output Current-0.1

Figure 2.7: Hall Effect Magnetic Encoder

The angular velocity of the pendulum and linear velocity of the cart has beeb calculated from these measured quantities.

Chapter 3

Control System Analysis and Design

3.1 Dynamic Equations

To analyze the dynamic system of the Inverted Pendulum Balancing system, we first need to determine the equations of motion that govern the system. The system is shown in Fig. 3.1, which consists of a cart and an inverted pendulum attached to it. Here

- M is the mass of the cart;
- m is the mass of the pendulum;
- l is the half-length of the pendulum;
- f is the force applied to the cart to keep the pendulum upright;
- x is the position of the cart;
- θ is the angular position of the pendulum.

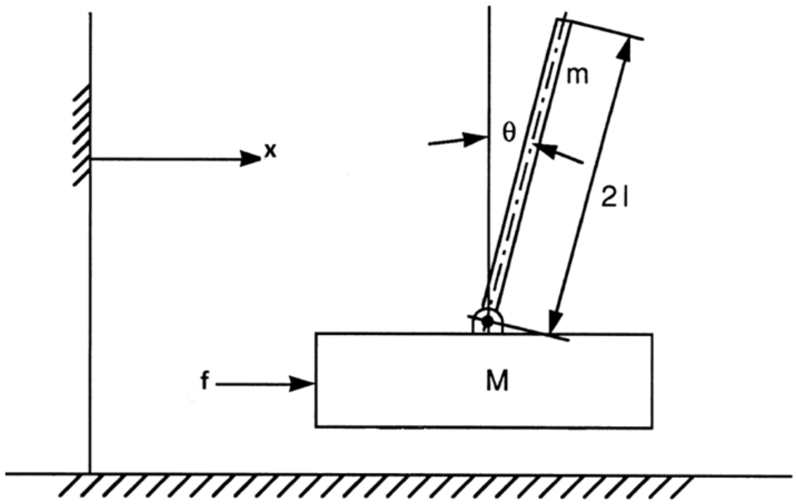

Figure 3.1: Inverted pendulum with slider

The free-body diagram of the system can be drawn as Fig. 3.2 and Fig. 3.3

- H_f and V_f are the forces exerted by the cart on the pendulum.
- b_2 is the viscous friction coefficient for rotary motion of the pendulum.
- b_1 is the viscous friction coefficient for linear motion of the cart.
- g is the gravitational acceleration.
- I is the moment of inertia of the pendulum with respect to the center of gravity.

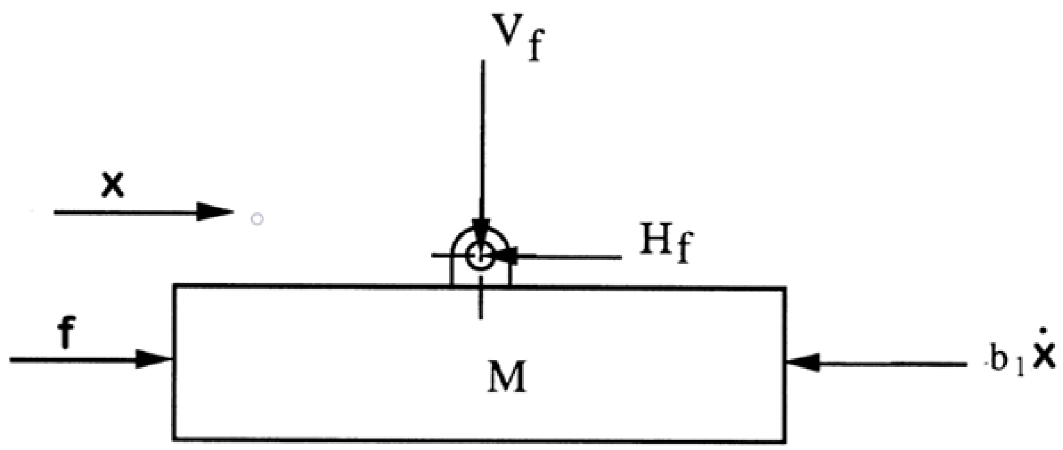

Figure 3.2: Free body diagram of cart (slider)

By applying Newton's second law to the horizontal motion of the cart, and to the horizontal, vertical and rotary motions of the pendulum, we have the following equations:

$$f - H_f - b_1\dot{x} = M\ddot{x} \tag{3.1}$$

$$V_f - mg = m\frac{d^2(l\cos\theta)}{dt^2} \tag{3.2}$$

$$H_f = m\frac{d^2(x + l\sin\theta)}{dt^2} \tag{3.3}$$

$$I\ddot{\theta} = (V_f\sin\theta - H_f\cos\theta)l - b_2\dot{\theta} \tag{3.4}$$

where, $\quad I = \frac{1}{3}ml^2$

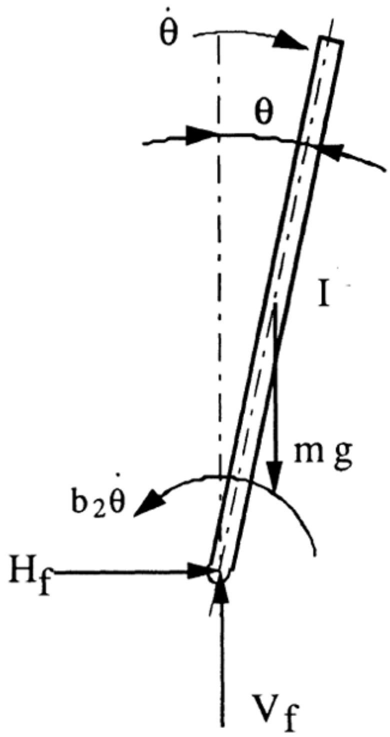

Figure 3.3: Free body diagram of pendulum

Neglecting the friction coefficients b_1 and b_2 (as they are minimal due to lubrication), and substituting V_f and H_f into eqs. (3.1) and (3.4), we get:

$$I\ddot{\theta} = l(mgsin\theta - m\ddot{x}cos\theta - ml\ddot{\theta}) \tag{3.5}$$

$$f = (M+m)\ddot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2\sin\theta \tag{3.6}$$

Obviously, these differential equations are nonlinear, they can not be solved except by numerical methods, so further simplification is needed. Since the object of controlling the system is to keep the pendulum upright, it is reasonable to assume that θ and $\dot{\theta}$ will remain close to zero. With this assumption, the equations can be linearized by retaining only those terms that are linear in θ and $\dot{\theta}$ and neglecting higher order terms because they will be insignificantly small when θ and $\dot{\theta}$ are close to zero. So after simplifications, we get the following equations from eq. 3.5 and 3.6.

$$I\ddot{\theta} = lmg\theta - lm\ddot{x} - ml^2\ddot{\theta} \tag{3.7}$$

$$f = (M+m)\ddot{x} + ml\ddot{\theta} \tag{3.8}$$

3.2 DC Motor Characteristics

To calculate the voltage required to exert a certain force on the slider, the characteristics of the DC motor are modeled as shown below.

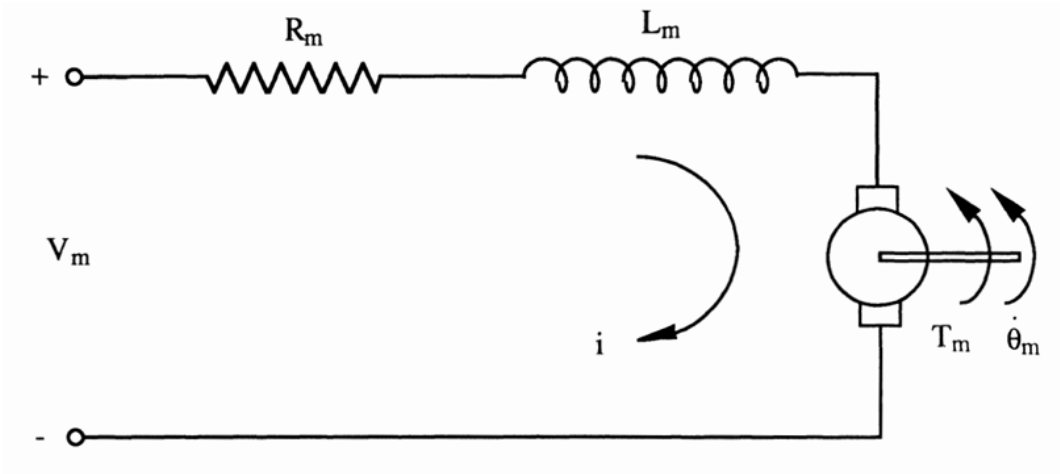

Figure 3.4: Circuit of DC Motor

In the circuit, parameters are:

- V_m : Input armature voltage (V)
- i : Armature current (A)
- R_m : Armature resistance (Ω)
- L_m : Armature inductance (H)
- T_m : Output torque (Nm)
- θ_m : Angular position of motor shaft (rad)

Motor parameters:

- Torque constant, $K_T = 0.0459$ N·m/A
- Back emf constant, $K_b = 0.23$ V·s/rad
- $R = 2\Omega$
- $L = 2$ mH
- Moment of inertia of motor shaft, $I_m = 0.05$ kg·m^2

$$T_m = K_T i - b\dot{\theta}_m \tag{3.9}$$

Applying KVL on the motor circuit:

$$V_m = L_m \frac{di}{dt} + iR_m + K_e\dot{\theta}_m \tag{3.10}$$

Dividing by R_m:

$$\frac{V_m}{R_m} = \frac{L_m}{R_m}\frac{di}{dt} + i + \frac{K_e\dot{\theta}_m}{R_m} \tag{3.11}$$

Assuming L_m is small, $\frac{L_m}{R_m}\frac{di}{dt} \approx 0$, so we get:

$$T_m = \frac{K_T V_m}{R_m} - \frac{K_T K_e}{R_m}\dot{\theta}_m - b\dot{\theta}_m \tag{3.12}$$

Equating torque to force applied:

$$T_m - fr = I_m\ddot{\theta}_m \tag{3.13}$$

where I_m is moment of inertia of motor shaft.

Chapter 4

Laplace Transforms for Equations

4.1 Transfer Function of Our System

- V_m : Input armature voltage (V)
- i : Armature current (A)
- R_m : Armature resistance (Ω)
- L_m : Armature inductance (H)
- T_m : Output torque (Nm)
- θ_m : Angular position of motor shaft (rad)
- K_T : Torque constant (m/A)
- K_b : Back emf constant (V·s/rad)

$$V_m = L_m \frac{di}{dt} + iR_m + k_b \frac{d\theta}{dt} \tag{4.1}$$

$$V(s) = L_m s I(s) + R_m I(s) + K_b s\Theta(s) \tag{4.2}$$

$$T_m(s) = K_T I(s) - bs\Theta(s) \quad \text{(Neglecting } b_1\text{)} \tag{4.3}$$

$$T_m(s) = Js^2\Theta(s) + b_2 s\Theta(s) \quad \text{(Neglecting } b_2\text{)} \tag{4.4}$$

From (3.5) and (3.6),

$$I(s) = \frac{Js^2\Theta(s)}{K_T} \tag{4.5}$$

Substitute into (1):

$$V(s) = \frac{(L_m s + R_m)Js^2\Theta(s)}{K_T} + K_b s\Theta(s) \tag{4.6}$$

$$\frac{\omega(s)}{V(s)} = \frac{K_T}{K_T K_b + R_m Js + L_m Js^2} \tag{4.7}$$

Substituting values

- $K_T = 0.0459$ Nm/A
- $K_b = 0.23$ Vs/rad
- $R_m = 2\Omega$
- $L_m = 2$ mH
- $J = 04077\text{Kgm}^2$

We get-

$$\Rightarrow \frac{\omega(s)}{V(s)} = \frac{386.05}{246.76 + 55.56s + 2.54s^2} \tag{4.8}$$

4.2 PID Controller Design

- The PID controllers prioritize the pendulum angular position error while giving less importance to the cart position.
- The tuning of the PID parameters is performed using the Zeigler-Nichols technique.
- Initially, the PID controller for the pendulum position is tuned by setting the cart position PID parameters to zero.
- Once the pendulum position is maintained at the upright position,tuning of both control loop is performed.

The Ziegler-Nichols open-loop tuning method, also known as the process reaction method, involves performing an open-loop step test on a process and analyzing the resulting response to determine controller parameters.

The steps include:

1) performing a step test

2) extracting process reaction curve parameters (dead time, time constant, and ultimate value), and

Controller Type	K_p	T_i	T_d
P	T/L	∞	0
PI	$0.9T/L$	$L/0.3$	0
PID	$1.2T/L$	$2L$	$0.5L$

$$K_i = K_p/T_i$$

$$K_d = T_d K_p$$

$$Controller: K_p + \frac{K_i}{s} + sK_d = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

Figure 4.1: Ziegler Nichols open loop table

3)using these parameters in Ziegler-Nichols equations to calculate controller gain, integral time, and derivative time for a P, PI, or PID controller

Parameter	Rise time	Overshoot	Steady-state error	Stability
K_p	decrease	increase	decrease	degrade
K_I	decrease	increase	eliminate	degrade
K_D	minor change	decrease	no effect	improve if "small"

Figure 4.2: PID parameters increasing effect

Chapter 5

Matlab Simulations

5.1 Open Loop Ziegler Nichols Method

The step response of the system is:-

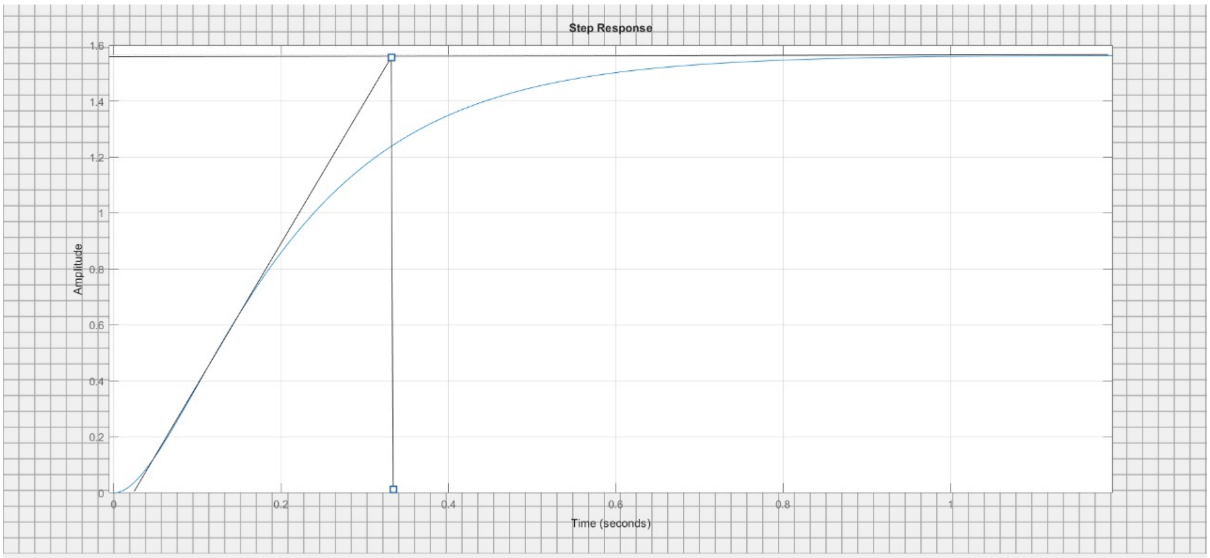

Figure 5.1: s-curve

$$\Rightarrow \frac{\omega(s)}{V(s)} = \frac{386.05}{246.76 + 55.56S + 2.54S^2} \tag{5.1}$$

From the graph we get that

- $K = 0.025$
- $L = 0.3$
- $T = 1.55$

Using the above table we have calculated these values.

- $K_p = 14.4$

- $K_i = 288$
- $K_d = 0.72$

By selecting the appropriate PID controller parameters based on the system response characteristics and desired performance, the PID controllers have been tuned to effectively control the system.

And we found the final output values as

- $K_p = 14.4$
- $K_i = 150$
- $K_d = 0.4319$

Finally, the model is simulated, and the results of the PID controller are shown in Figure 5.2. The simulation demonstrates the behavior of pendulum position as the control input, which is the acceleration, is applied. It can be observed that pendulum positions converge to their respective reference values.

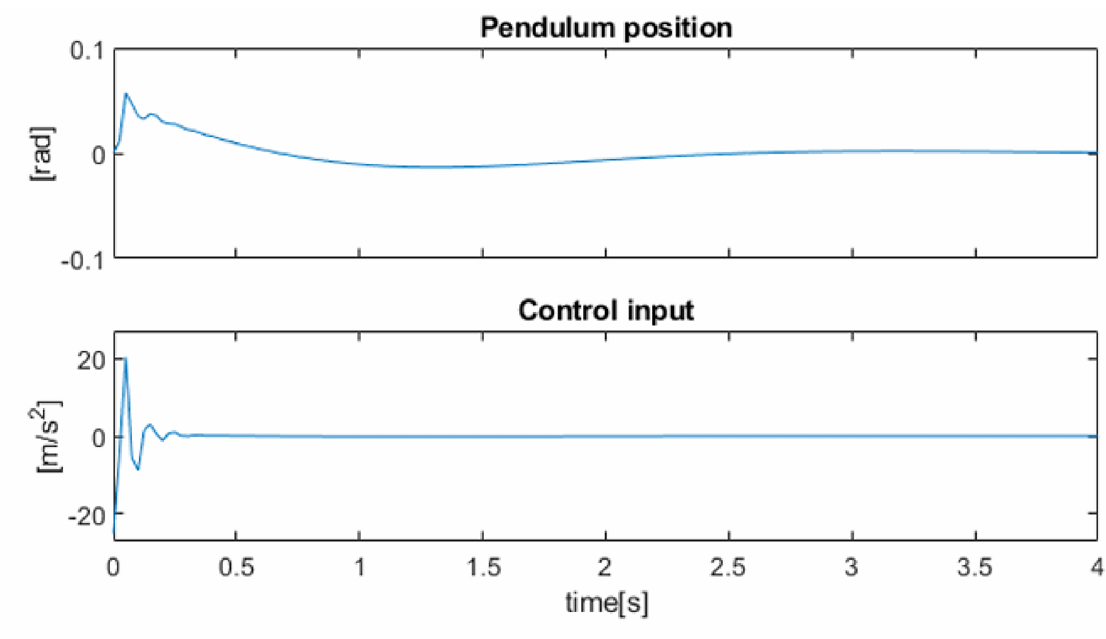

Figure 5.2: Pendulum Balancing with PID Controller

5.2 Closed Loop Ziegler Nichols Method

Closed loop Ziegler Nichols method cannot be applied on $\frac{\omega(s)}{V(s)}$, because the root locus doesn't cross the imaginary axis. Therefore, we consider the output as $\theta(s)$.

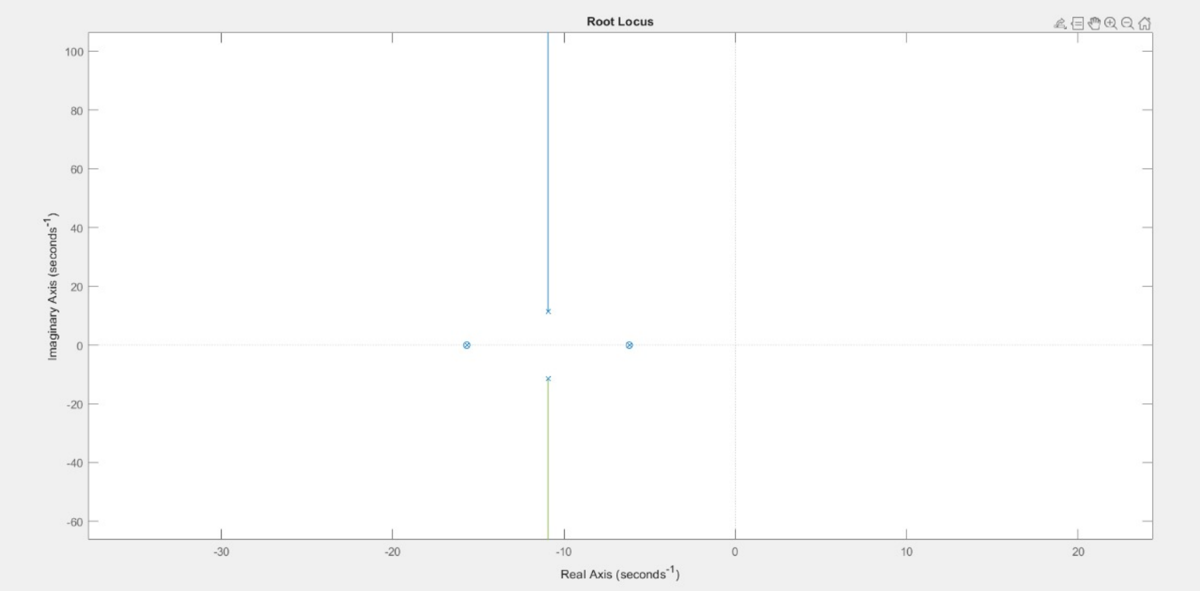

Figure 5.3: Root Locus with output as $\omega(s)$

$$\frac{\theta(s)}{V(s)} = \frac{K_T}{K_T K_b s + R_m J s^2 + L_m J s^3} \tag{5.2}$$

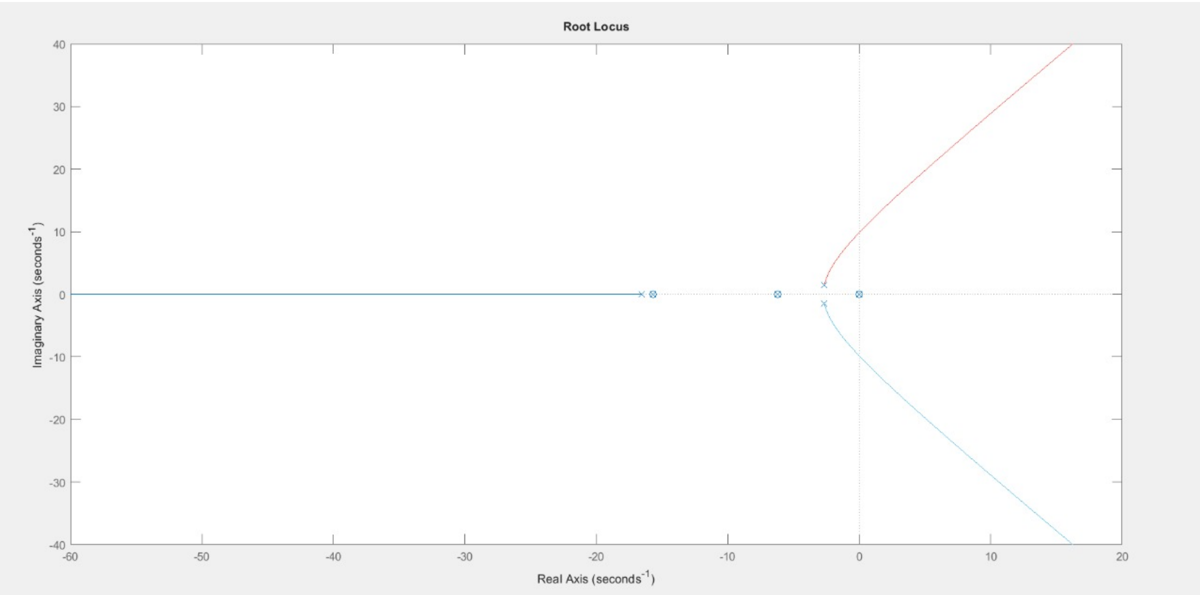

Figure 5.4: Root Locus with output as $\theta(s)$

Initial Conditions

- $x = 0.00$ m
- $\dot{x} = 0.00$ m/s
- $\theta = 0.05$ rad
- $\dot{\theta} = 0.1$ rad/s

5.3 Arduino IDE Code

```
#include <PID_v1.h>
#include <Encoder.h>

// Mechanical System Properties
// max linear travel ~8200 Encoder counts
// full pendulum rotation =7200 Encoder counts

// Variable Definitions
// Pin Definitions
byte pin_pwm = 6;
byte pin_left = 4;
byte pin_right = 5;
byte pin_rot_enc1 = 2;
byte pin_rot_enc2 = 3;

// Speeds
byte setup_drive_speed = 105;
byte setup_zero_speed = 60;

// Linear and angle Limits for Controller
int limit_lin_min = 800;
int limit_lin_max = 7400;
double limit_angle = 200;  // Working region: -200 to +200

// Deadband Variables
byte output_deadband = 0;
byte rot_deadband = 0;

// Miscellaneous Variables
int rot_offset = 0;
double Err_angle;
double lastErr_angle = 0;
double timer;
double rot_speed;
double angle_array[5];
double timer_array[5];
double angle;

// Encoder Setup
Encoder rotEnc(pin_rot_enc1, pin_rot_enc2);

// PID Setup
double Output, rot_Setpoint;
double rot_kp = 14.4, rot_ki =150, rot_kd = 0.4319;
PID rot_Controller(&angle, &Output, &rot_Setpoint, rot_kp, rot_ki, rot_kd, DIRECT);
```

```
void setup() {
  Serial.begin(115200);
  rotEnc.write(-3600);  // Set when pendulum is hanging straight down
  delay(500);

  rot_Controller.SetMode(AUTOMATIC);
  rot_Controller.SetOutputLimits(-255, 255);
  rot_Controller.SetSampleTime(3);
  delay(300);
}

void loop() {
  // Shift arrays
  for (int i = 4; i > 0; i--) {
    angle_array[i] = angle_array[i-1];
    timer_array[i] = timer_array[i-1];
  }

  // Read present values
  timer = millis();
  timer_array[0] = timer;
  angle = rotEnc.read();
  angle_array[0] = angle;

  // Compute rotational speed
  rot_speed = (angle_array[0] - angle_array[4]) /
              (timer_array[0] - timer_array[4]);

  // Main Controller
  if (true) {  // No linear position limits
    if (angle < limit_angle && angle > -limit_angle) {
      rot_Setpoint += rot_offset;
      rot_Controller.SetTunings(rot_kp, rot_ki, rot_kd);
      rot_Controller.Compute();

      if (Output >= 0) {
        Output = map(abs(Output), 0, 255, setup_zero_speed, 255);
        go_right(Output);
      } else {
        Output = map(abs(Output), 0, 255, setup_zero_speed, 255);
        go_left(Output);
        Output = -Output;
      }
    } else {
      go_stop();
      Output = 0;
    }
  } else {
```

```
    go_stop();
    delay(200);
  }

  Serial.print("NA,");
  Serial.print(angle);
  Serial.print(",");
  Serial.println(Output);
}

// Procedure Definitions
void go_left(int velocity) {
  digitalWrite(pin_right, LOW);
  digitalWrite(pin_left, HIGH);
  analogWrite(pin_pwm, velocity);
}

void go_right(int velocity) {
  digitalWrite(pin_right, HIGH);
  digitalWrite(pin_left, LOW);
  analogWrite(pin_pwm, velocity);
}

void go_stop() {
  analogWrite(pin_pwm, 0);
}
```

5.4 Response from the Experiment

The experimental system balanced the pendulum for a few seconds before the cart hit the rail's edge. The behavior aligned with simulation predictions.

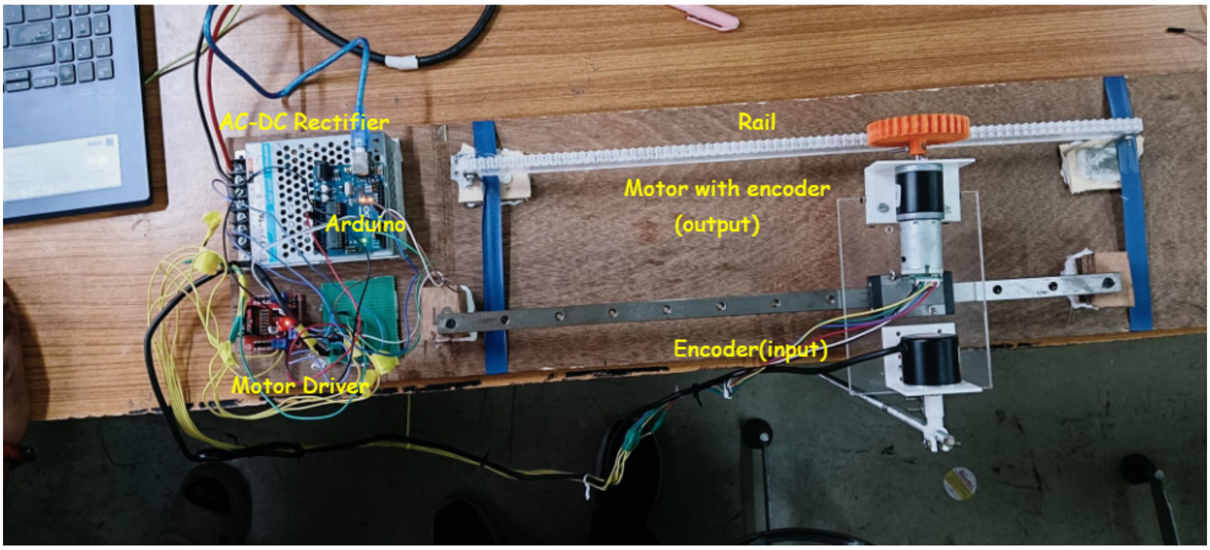

Figure 5.5: Our experimental setup

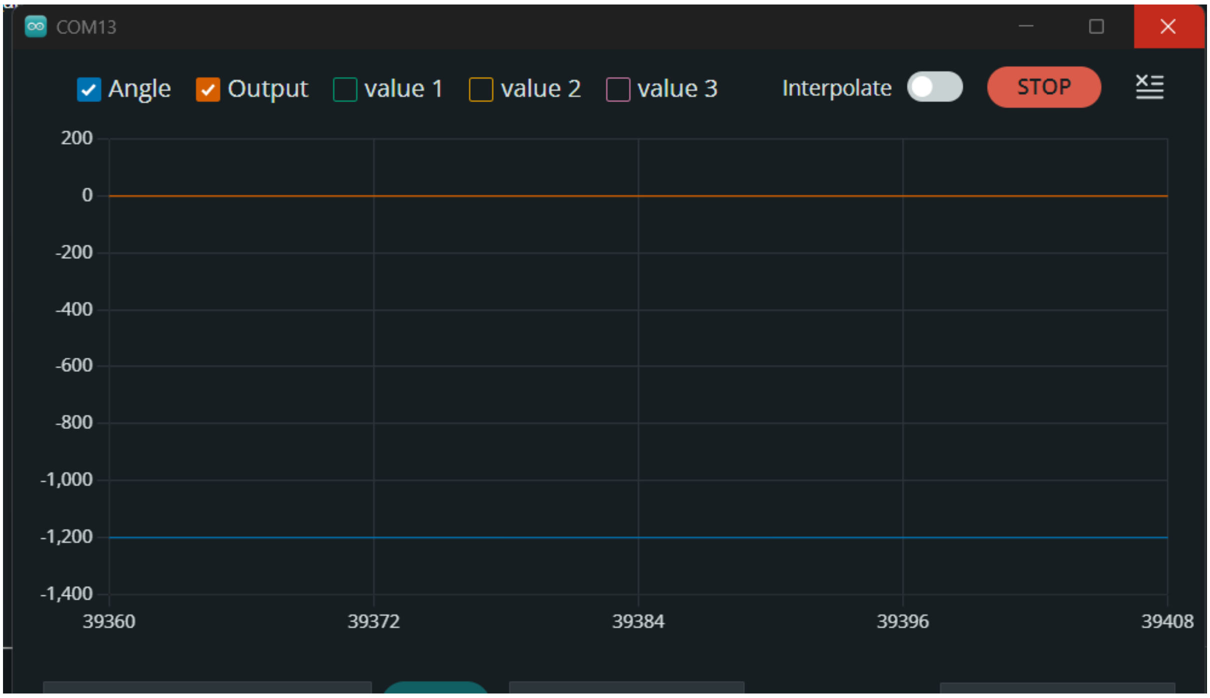

Figure 5.6: Initial state of stable equilibrium

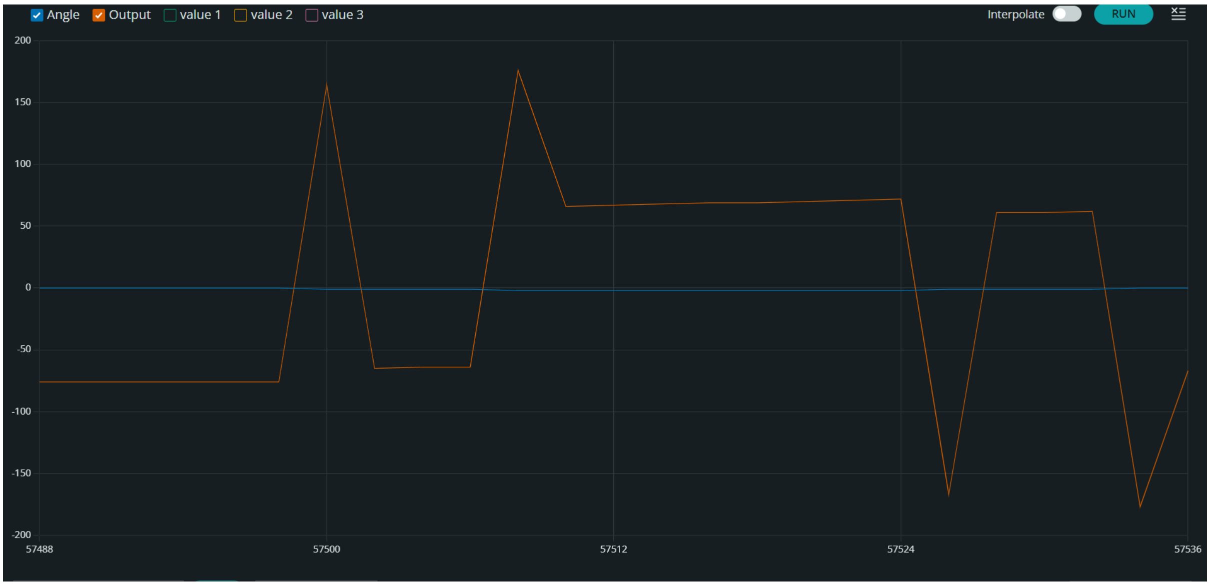

Figure 5.7: Transition state

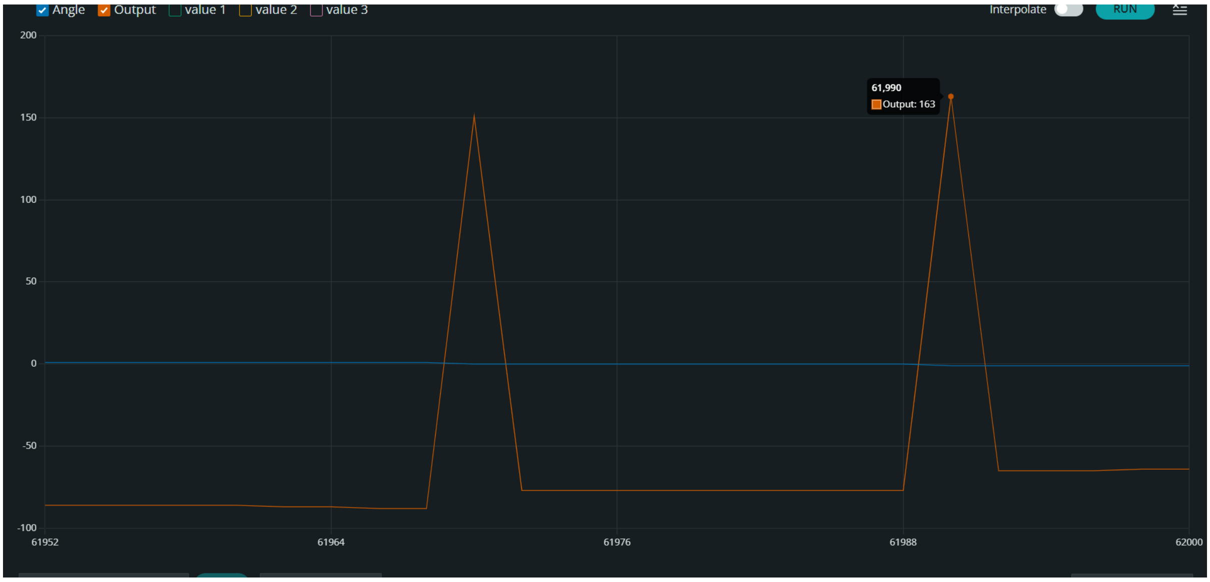

Figure 5.8: Steady state plot

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The system successfully demonstrated balancing capability. However, hardware limitations such as rail length affected the duration of balance. For the rail limitations we have provided support the the system at the end of the rails then the system stands upright for quite good amount of time.

6.2 Future Work

Future improvements include the use of optical interrupters, each at both ends of the rail. So we can measure the distance of the cart from the end of the rail. We would also be able to use two PID controllers on the system instead of one. Using which we would be able to balance the system more accurately. In addition, we plan to improve the system by using a better motor driver or coolant, as the motor driver is getting heated fast and it is affecting the stability of the system after some time.

6.3 Video link

`https://drive.google.com/drive/folders/1PebZWGDltm78Wg5cxvXAEFTq3Fm6KGSo`
Video Link