Admittance Control for 2D Plotter

A Project Report Submitted by

Team Members:

Arashdeep Singh M23IRM003

Chaitanya Patil M23IRM004

Mayank Kumar M23IRM007

Muskan Suman M23IRM008

in partial fulfillment of the requirements for the course project of

Introduction To Medical Robotics

Indian Institute of Technology Jodhpur

Inter-disciplinary Research Division (IDRD)

April, 2025

1 Introduction

This project implements admittance control on a 2D plotter to achieve smooth, compliant motion in response to external forces. Initially, a joystick was used to control motor movement across the X and Y axes. Later, Force Sensing Resistors (FSRs) replaced the joystick to detect applied forces and drive motion accordingly. The system dynamics were modeled using Lagrangian mechanics, and a PID controller was designed to align the real system's behavior with the desired virtual dynamics. This work demonstrates basic principles of force-based control, highlighting its potential for applications requiring safe and adaptive interaction, such as in medical robotics.

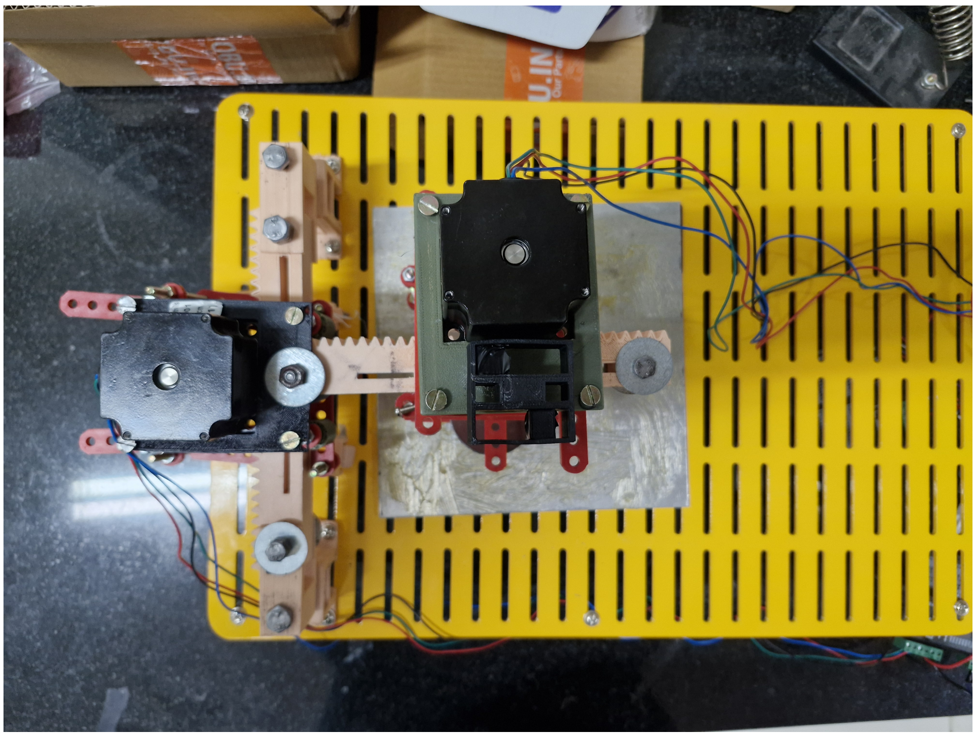


Figure 1.1: Top View of Assembly

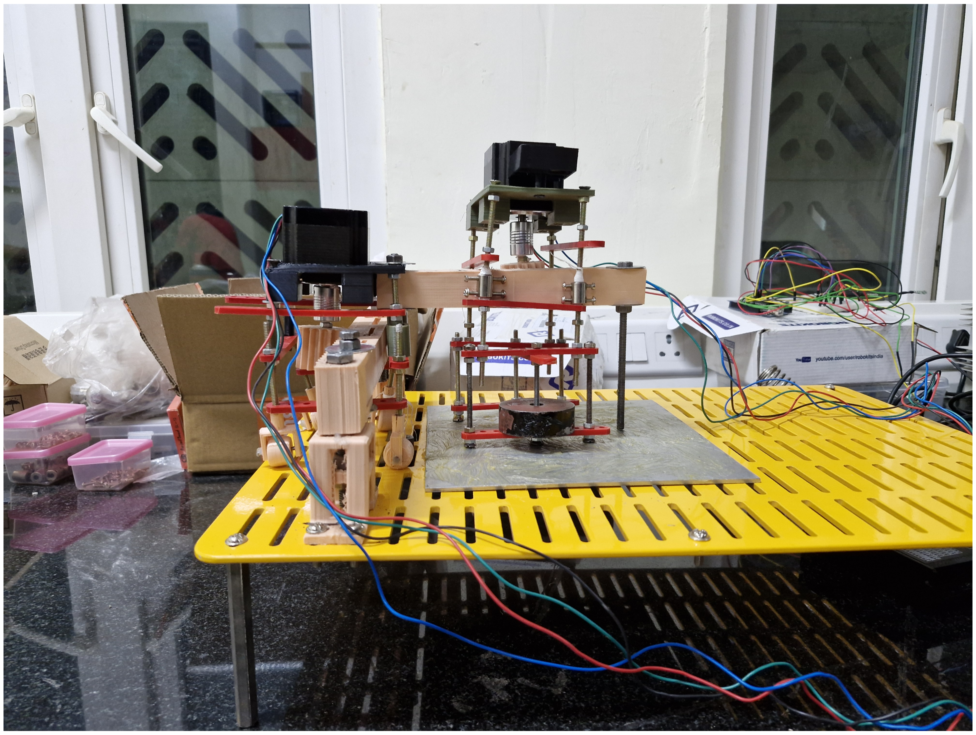


Figure 1.2: Side View of Assembly

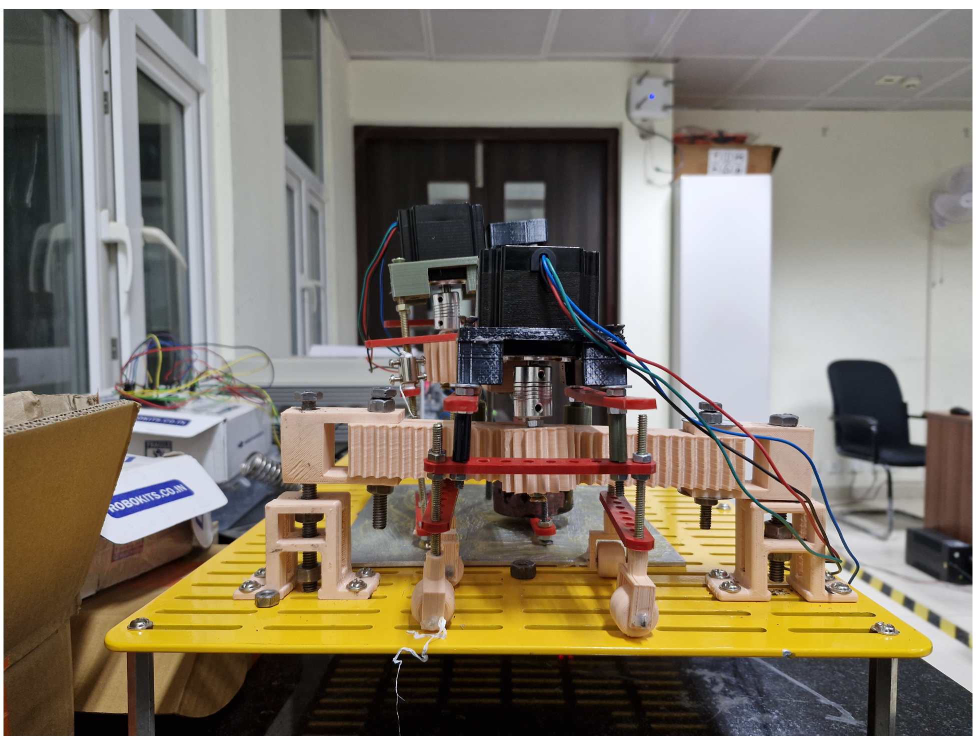


Figure 1.3: Back View of Assembly

1.1 Experimental Setup Description

The experimental setup for the 2D plotter designed to implement admittance control is shown in figurers. The platform facilitates planar motion along the X and Z axes and is constructed using a combination of 3D-printed components, wooden frames, and metallic rods. The structure is mounted on a perforated yellow acrylic base, providing both stability and flexibility for modular adjustments.

In our specific implementation, the 2D plotter serves as the robotic system whose end effector motion is governed by the admittance control strategy. The force sensor attached to the end effector measures external forces, which are processed through the virtual dynamics model to compute the reference velocity. The PID controller then determines the necessary control forces to enforce this velocity in the real system. By leveraging both the virtual model for reference generation and the real model for execution, we achieve a compliant and adaptive system capable of handling external force variations effectively.

To effectively implement admittance control, we employ a 2D prismatic system in a horizontal setup, offering multiple advantages. Operating in a horizontal plane minimizes the effects of gravitational potential energy, simplifying system modeling and ensuring force measurements remain unaffected by gravity. Unlike robotic arms with rotational joints, which require complex transformations, the prismatic system moves strictly in the X-Y plane, reducing system complexity and making control implementation more straightforward. The absence of rotational degrees of freedom results in linear and predictable motion, making the system more stable and well-suited for precision control applications. Additionally, prismatic motion directly maps applied forces to linear accelerations, enhancing control accuracy and force interpretation, which is particularly beneficial in force-based interactions. The primary components of the setup include:

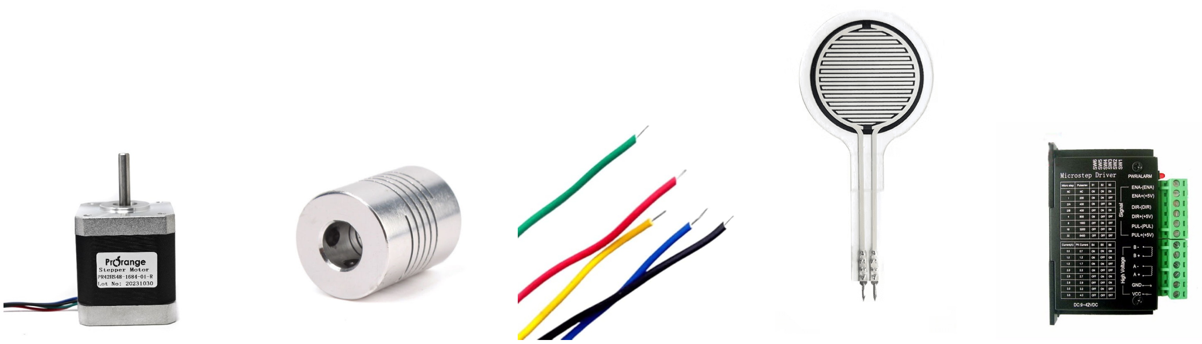
- **Stepper Motors:** Two stepper motors control the motion along the X and Z directions. These motors are mounted on customized supports at different heights to enable independent axis actuation.
- **Force Sensing Resistors (FSRs):** FSRs are integrated at the end-effector to detect externally applied forces, enabling force-based motion control.
- **Couplings and Linkages:** Shaft couplings connect the motors to the moving frame, ensuring efficient torque transfer and mechanical compliance.
- **Electronics and Wiring:** Multiple color-coded wires connect the motors, sensors, and the microcontroller unit. These wires are organized to avoid interference with the moving parts.
- **Control System:** An Arduino microcontroller and motor drivers are employed to process the sensor inputs and implement the admittance control strategy.
- **Rack and Pinon**: Rack and Pinion system with pressure angle of 20 degree, module 2mm is printed on 3D printer with ABS as material.

The system is designed to operate in a horizontal plane, minimizing gravitational effects on force measurements and simplifying dynamic modeling. External forces applied on the end-effector are sensed by the FSRs, and the motion is regulated using a virtual mass-spring-damper model through admittance control.

Overall, the setup is optimized for simplicity, modularity, and functionality, providing an effective platform for studying compliant motion control strategies in robotic systems.

2 Equipment and Consumables Required

S. NO.	Product	Quantity
1	Stepper Motor + Driver	2
2	Force Sensor	4
3	Multicolored Wires	1 Bundle
4	Shaft Coupling	2
5	Joystick	1



(a) Stepper Motor (b) Coupling (c) Wires (d) Force Sensor (e) Driver

Figure 2.1: Assembly components of joystick setup

Figure 2.2: 2D Joystick

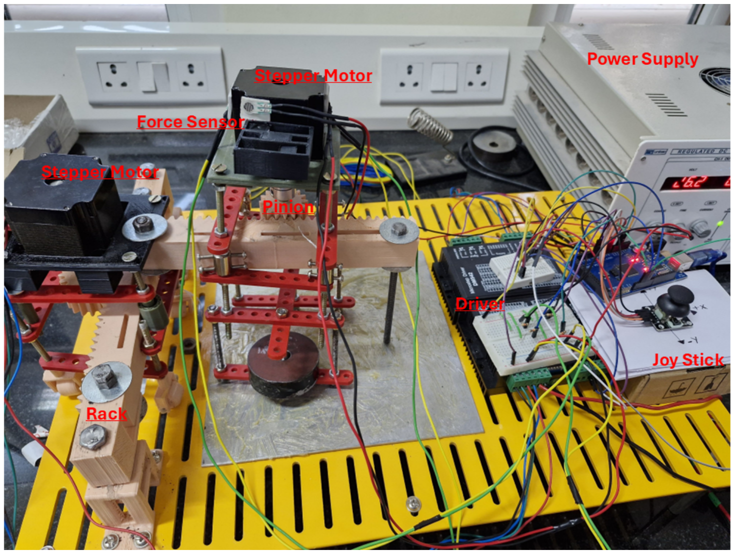


Figure 2.3: Conceptual Labeled Diagram

3 Implementation

In this project we have done implementation in three phases. In first phase we have motor control using joystick. Then in second phase will inlcude force sensor into the setup will use force values for motor control. In final phase will study system dynamics and will try to implement torque control for motor.

3.1 Motor Control Based on Joystick Movement

A dual-axis analog joystick provides two independent control signals via built-in potentiometers. Each potentiometer is linked to one axis (horizontal X or vertical Y) of the joystick thumbstick. When the stick is centered, both potentiometers output a mid-scale voltage (around half of the supply voltage, e.g., 2.5 V on a 5 V system). Moving the joystick changes the resistance of the potentiometers, causing the output voltages to increase or decrease linearly with the deflection. These voltages are typically fed into a microcontroller's analog-to-digital converter (ADC) channels (commonly labeled VRx and VRy).

The microcontroller continuously samples these two analog values to determine the joystick's position along each axis.For example, on a 10-bit ADC (0–1023 range), the neutral position reads approximately 512, pushing the stick right/up increases that reading toward 1023, and pushing left/down decreases it toward 0. This analog readout scheme allows precise and proportional control of the connected devices. The system is configured so that the joystick's horizontal (right/left) motion drives Motor 1 (the X-axis motor) and its vertical (up/down) motion drives Motor 2 (the Z-axis motor). The resulting mapping of joystick deflection to motor action is:

- **Horizontal Movement (Right/Left) – Controls Motor 1 (X-axis movement)**:
  - Joystick Right (X+): The X-axis voltage rises above mid-scale. The MCU drives Motor 1 clockwise, moving the plotter carriage along the +X direction.

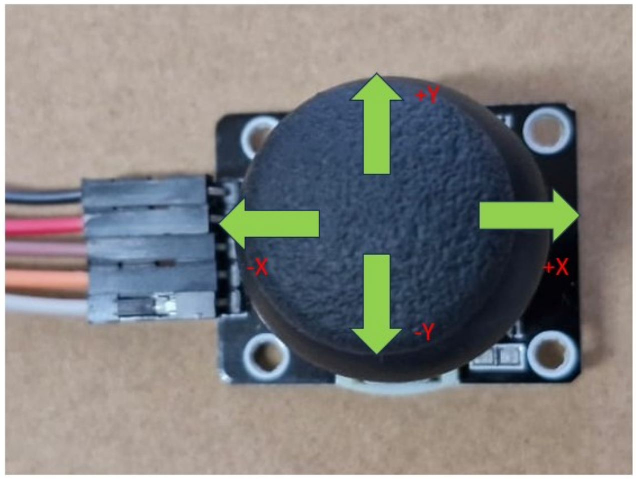


Figure 3.1: Motor control using joystick

  - Joystick Left (X–): The X-axis voltage falls below mid-scale. The MCU drives Motor 1 counterclockwise, moving the carriage along the -X direction.

- **Vertical Movement (Up/Down) – Controls Motor 2 (Z-axis movement)**:
  - Joystick Up (Y+): The Y-axis voltage rises above mid-scale. The MCU drives Motor 2 clockwise, moving the carriage along the +Z direction (upward).
  - Joystick Down (Y–): The Y-axis voltage falls below mid-scale. The MCU drives Motor 2 counterclockwise, moving the carriage along the -Z direction (downward).

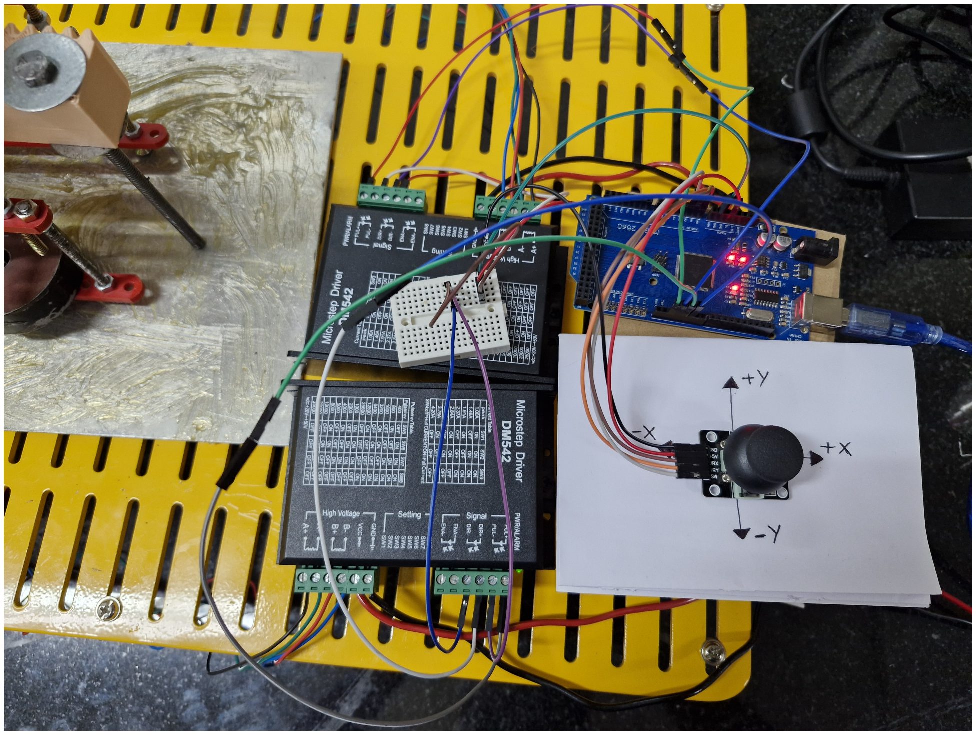


Figure 3.2: Assembly of joystick setup

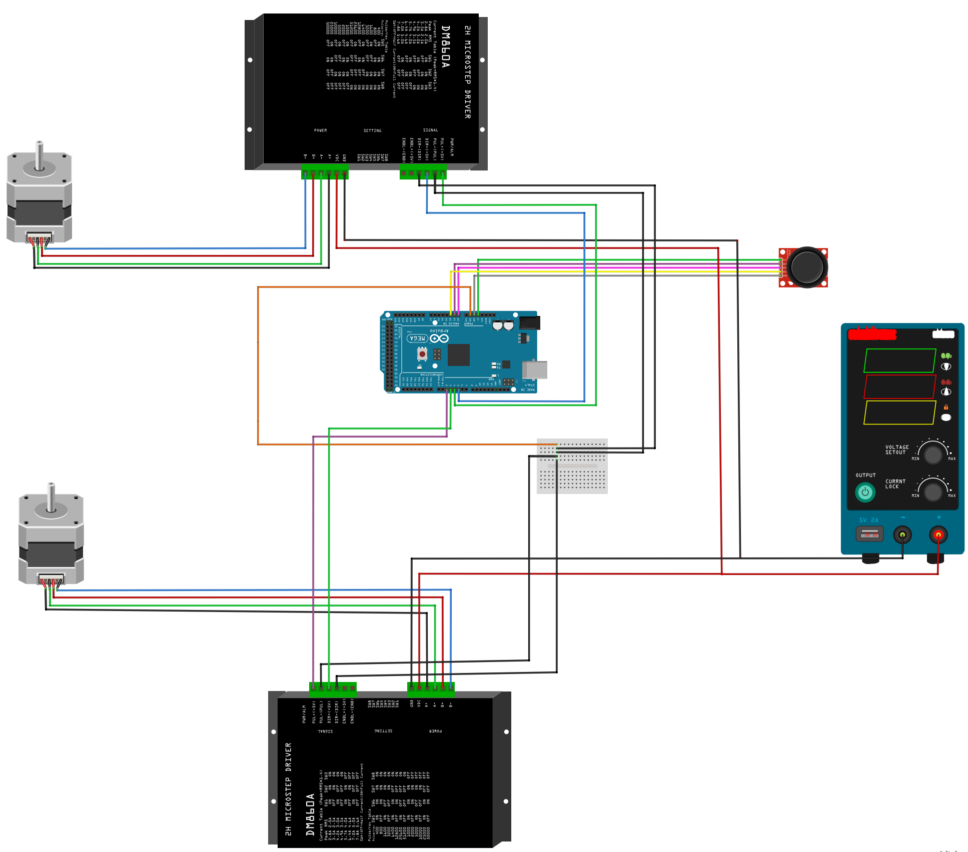


Figure 3.3: Circuit Diagram of Joy Stick Control

3.2 FSR-Based Force and Pressure Measurement for Motion Detection

In Phase 2 of the project, we integrate **Force Sensing Resistors (FSRs)** to measure applied forces and detect directional input for motion control. A total of **four FSR sensors** are used to sense forces in four directions — **right, left, up, and down** — replacing joystick input with force-based control.

Each FSR acts as a variable resistor, with high resistance ($>10\,\mathrm{M\Omega}$) when no force is applied, and decreasing resistance with applied force. It is used in a voltage divider with a fixed resistor, and the resulting voltage, which varies with force, is read by the Arduino's ADC. The ADC values range from 0 to 870 (out of 1023), and a force of $7\,\mathrm{N}$ corresponds to the maximum ADC reading.

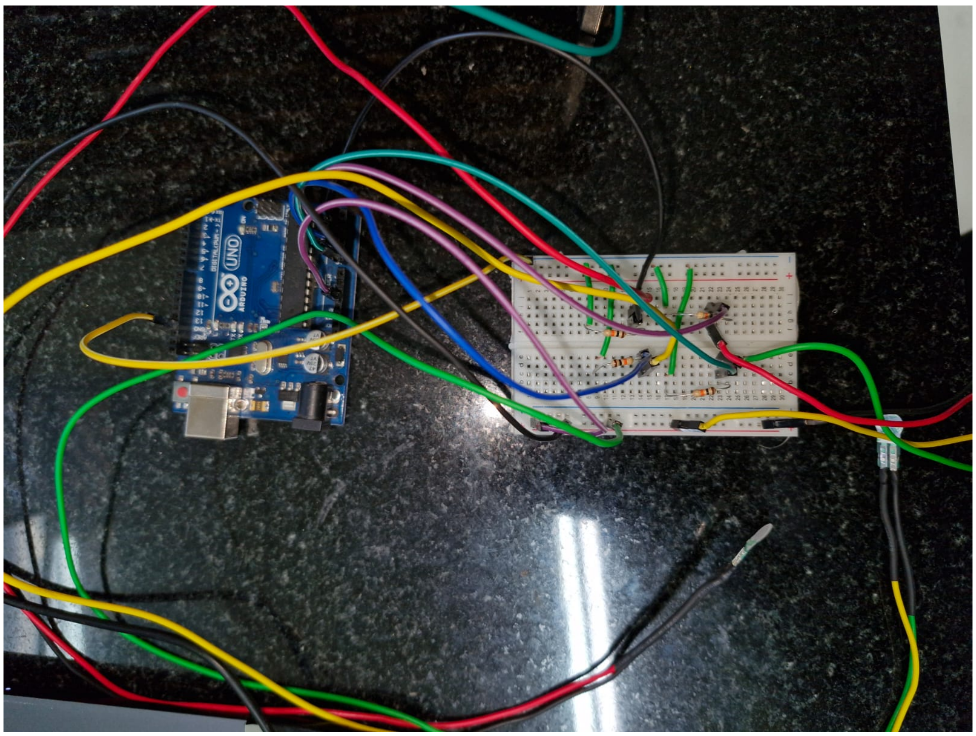


Figure 3.4: Force Sensing Resistors (FSRs)

3.2.1 Working Principle

- **Force Sensing Resistor (FSR):** An FSR is a sensor whose resistance decreases when a force or pressure is applied. Here Circuit for single Force Sensing Resistors

  The relationship between force and resistance is **inversely proportional**.

- **Basic Operation:**
  - As the applied force increases, the resistance of the FSR decreases.
  - A **voltage divider circuit** is used to convert the changing resistance into a measurable voltage signal.
  - This voltage is read by an Arduino microcontroller to estimate the applied force and pressure.

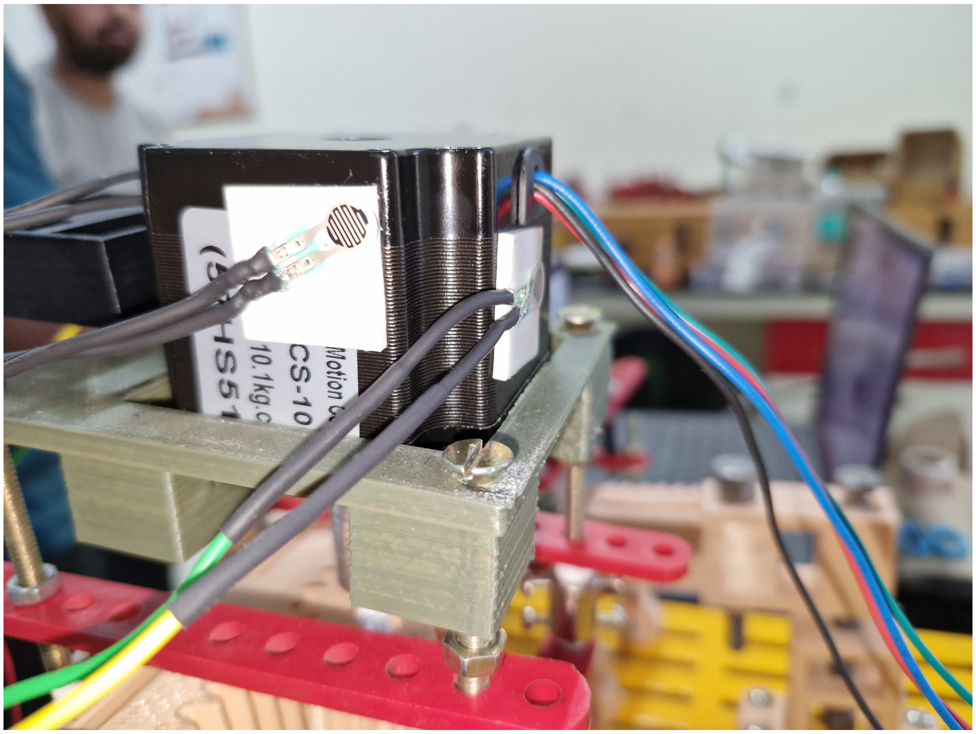


Figure 3.5: Force Sensor on End Effector

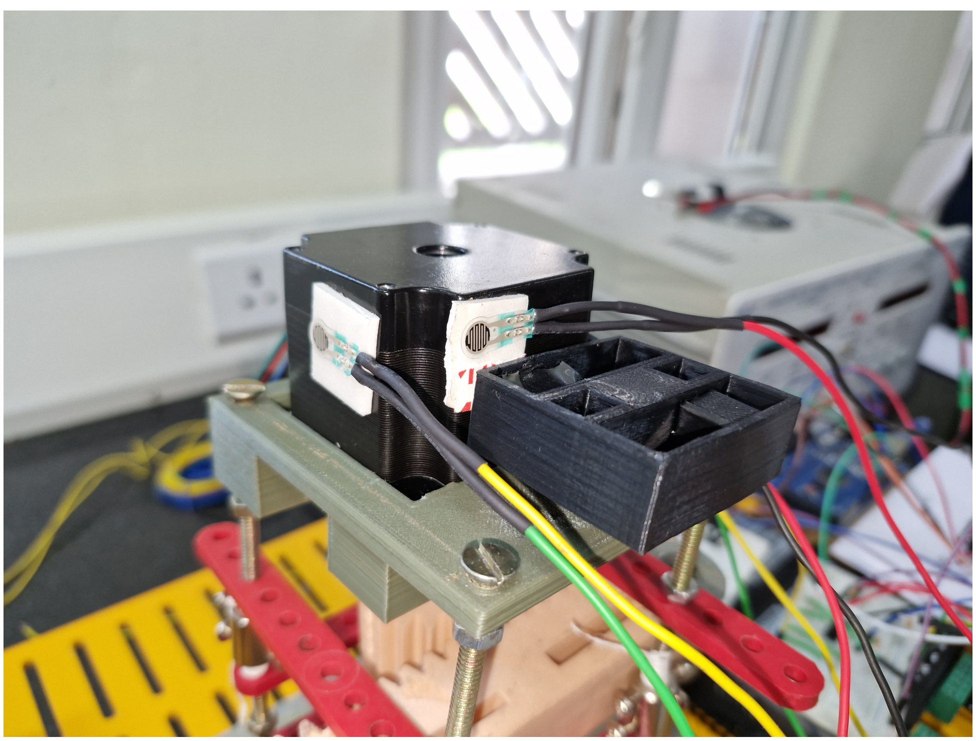


Figure 3.6: Force Sensor on End Effector

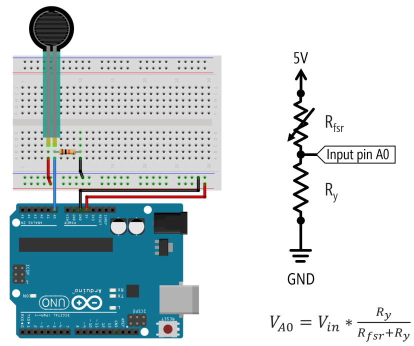


Figure 3.7: Force Sensing Resistors CIRCUIT

3.2.2 Circuit Configuration

Voltage Divider Circuit:

Each FSR is connected in series with a known fixed resistor (R_{fixed}) and powered by a constant voltage source (V_{CC}). The output voltage (V_{out}) across the FSR is given by:

$$V_{\text{out}} = V_{\text{CC}} \times \frac{R_{\text{FSR}}}{R_{\text{fixed}} + R_{\text{FSR}}} \tag{1}$$

Rearranging, the FSR resistance (R_{FSR}) is calculated by:

$$R_{\text{FSR}} = R_{\text{fixed}} \times \left(\frac{V_{\text{CC}} - V_{\text{out}}}{V_{\text{out}}} \right) \tag{2}$$

Analog-to-Voltage Conversion:

The Arduino's `analogRead()` function reads a value between 0 and 1023, corresponding to 0V to 5V. Voltage is calculated as:

$$V_{\text{out}} = \text{analogRead}(A0) \times \frac{V_{\text{CC}}}{1023} \tag{3}$$

3.2.3 Force and Pressure Calculation

Force Estimation:

The applied force (in grams) is determined using the hyperbolic relationship:

$$\text{Force (grams)} = \frac{K}{R_{\text{FSR}}} \tag{4}$$

where K is a calibration constant determined experimentally.

Force Conversion:

The force in grams is converted to Newtons:

$$\text{Force (N)} = \text{Force (grams)} \times 0.00980665 \tag{5}$$

Pressure Calculation:

Pressure is calculated as:

$$\text{Pressure (Pa)} = \frac{\text{Force (N)}}{\text{Area (m}^2\text{)}} \tag{6}$$

where the sensor's effective contact area is known.

3.2.4 Calibration Process

A calibration process is essential to determine the constant K accurately:

- Known weights are applied sequentially.
- The value of K is adjusted until the computed force values match the real force values.

3.2.5 Data Output

- The Arduino displays real-time measured values — analog reading, voltage, resistance, force, and pressure — on the **Serial Monitor**.
- Measurements are updated every **200 milliseconds**.

FSR-to-Motor Mapping

Each of the four FSR sensors corresponds to a direction of motion:

The resulting mapping of FSR activation to motor action is:

- **Horizontal Force (Right/Left) – Controls Motor 1 (X-axis movement)**:
  - Right FSR pressed: Motor 1 moves clockwise, moving the plotter carriage along the +X direction.
  - Left FSR pressed: Motor 1 moves counterclockwise, moving the carriage along the -X direction.
- **Vertical Force (Up/Down) – Controls Motor 2 (Z-axis movement)**:
  - Up FSR pressed: Motor 2 moves clockwise, moving the carriage along the +Z direction (upward).
  - Down FSR pressed: Motor 2 moves counterclockwise, moving the carriage along the -Z direction (downward).

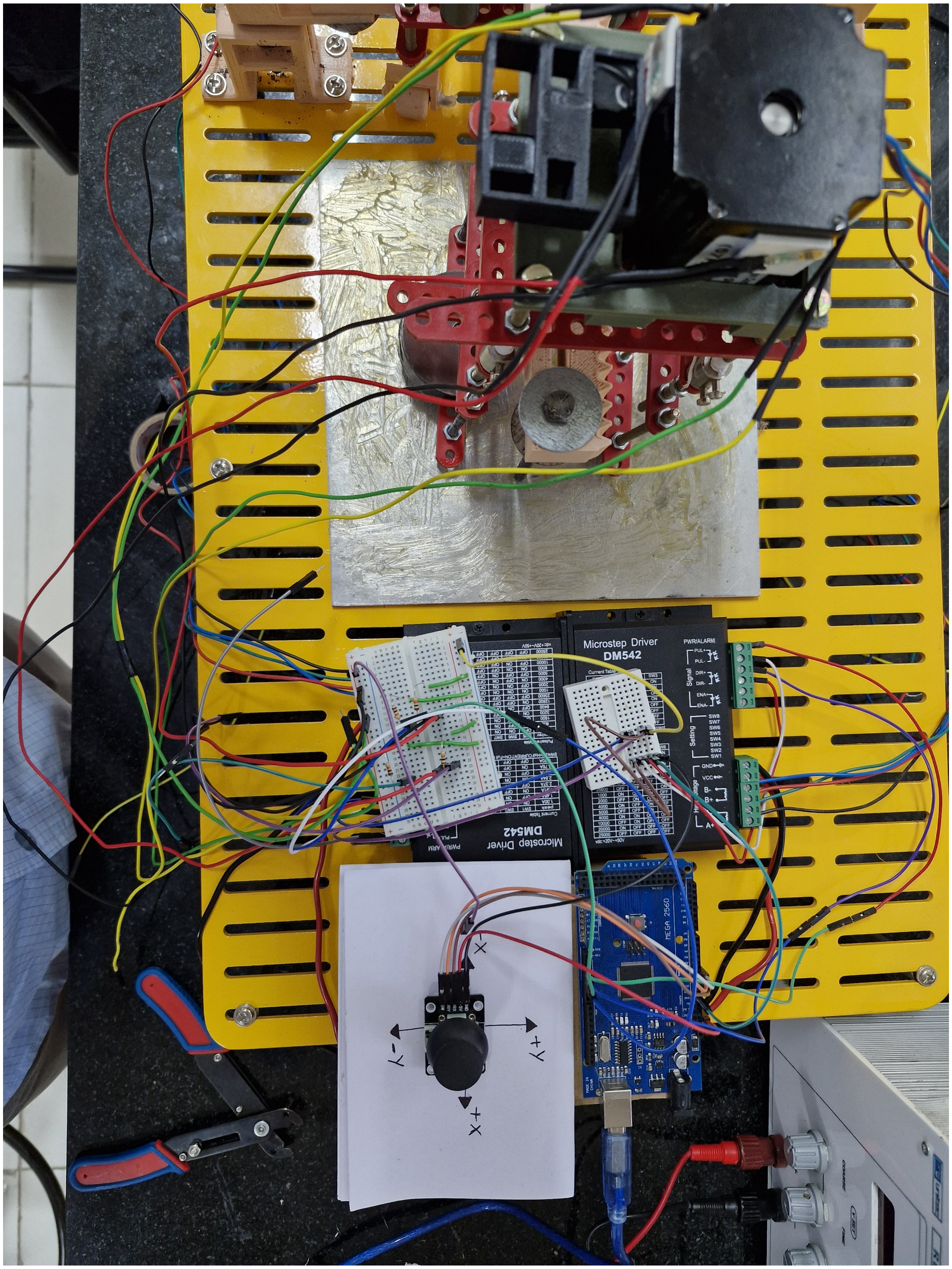


Figure 3.8: Assembly of joystick with force sensor setup

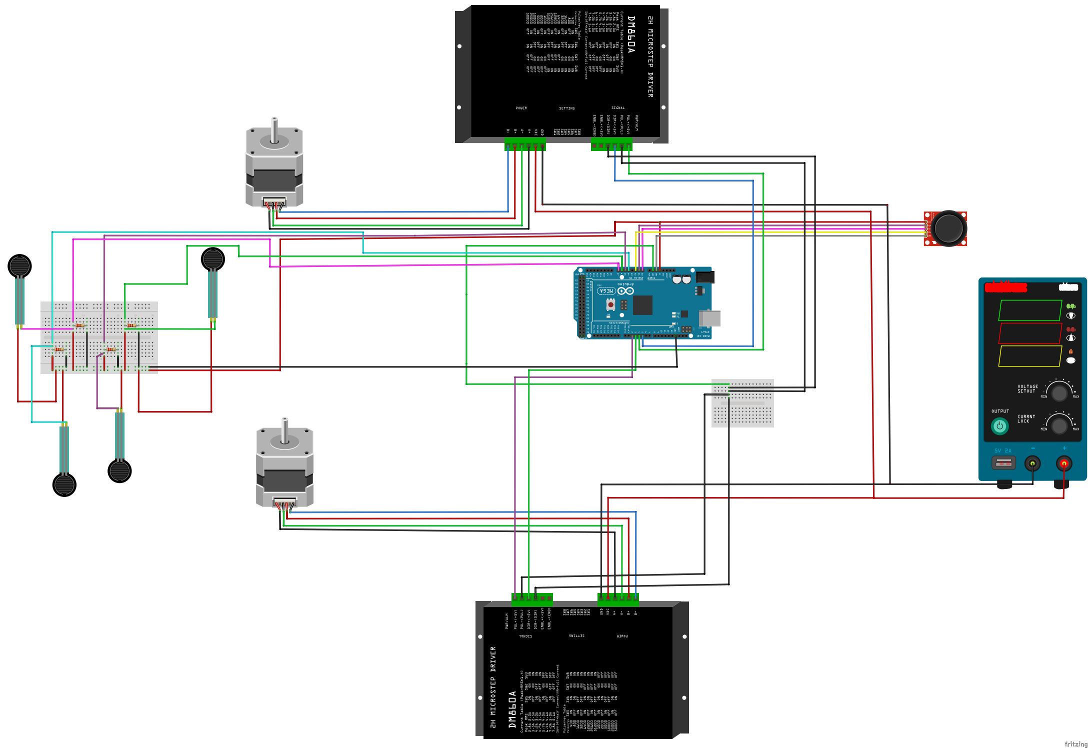


Figure 3.9: Circuit Diagram of Joy Stick and Force Control

4 Lagrangian Dynamics for the 2D Plotter

4.1 Introduction

This document presents the derivation of the Lagrange dynamics for a 2D horizontal plotter. The plotter typically has one axis that moves the entire setup, while the second axis motion affects only its own final position and motor mounting.

4.2 Coordinates

- x: The horizontal displacement of the setup, representing the absolute position in the horizontal plane.
- y: The relative position in the x direction.

4.3 Kinetic Energy

4.3.1 Translational Kinetic Energy

$$T_x = \frac{1}{2} M_x \dot{x}^2 \tag{7}$$

$$T_y = \frac{1}{2} M_y \dot{y}^2 \tag{8}$$

where M_x and M_y are the effective masses moving in the x and y directions, respectively.

4.3.2 Rotational Kinetic Energy

$$T_{\theta_1} = \frac{1}{2} I_1 \dot{\theta}_1^2 \tag{9}$$

$$T_{\theta_2} = \frac{1}{2} I_2 \dot{\theta}_2^2 \tag{10}$$

where I_1 and I_2 are the moments of inertia for the rotational components.

4.3.3 Total Kinetic Energy

$$T = \frac{1}{2}(M_x + I_1)\dot{x}^2 + \frac{1}{2}(M_y + I_2)\dot{y}^2 \tag{11}$$

4.4 Potential Energy

Assuming the mechanical design is stiff and the gravitational component is constant, the potential energy is negligible:

$$V = 0 \tag{12}$$

4.5 Lagrangian Mechanics

The Lagrangian L is given by:

$$L = T - V = \frac{1}{2}(M_x + I_1)\dot{x}^2 + \frac{1}{2}(M_y + I_2)\dot{y}^2 \tag{13}$$

4.6 Equations of Motion

Using the Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0 \tag{14}$$

we derive the equations of motion for the system.

4.6.1 For x-axis

$$(M_x + I_1)\ddot{x} = F_{\text{ext},x} \tag{15}$$

4.6.2 For y-axis

$$(M_y + I_2)\ddot{y} = F_{\text{ext},y} \tag{16}$$

4.7 Admittance Control

Admittance control aims to impose a desired dynamic behavior on the system in response to external forces. This involves defining virtual dynamics and actual dynamics.

4.7.1 Virtual Dynamics

Virtual dynamics represent the desired behavior of the system. The virtual model is defined by:

$$M_v\ddot{x}_v + B_v\dot{x}_v + K_v x_v = F_{\text{ext}} \tag{17}$$

where M_v, B_v, and K_v are the virtual mass, damping, and stiffness, respectively, and x_v is the virtual position.

For Different values of dynamics coeffecients following behaviour of system dynamics is observed.

B	1
K	0.1
M	0.8

Table 4.1: Coefficents For Virtual Dynamics

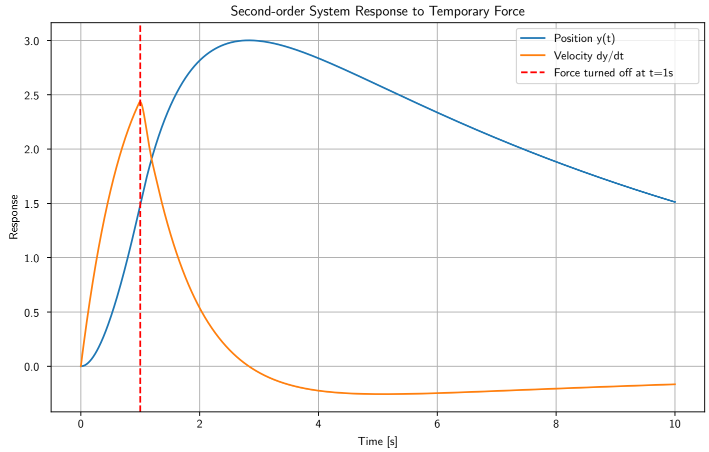


Figure 4.1: System Behaviour

B	1
K	0.1
M	0.1

Table 4.2: Coefficents For Virtual Dynamics

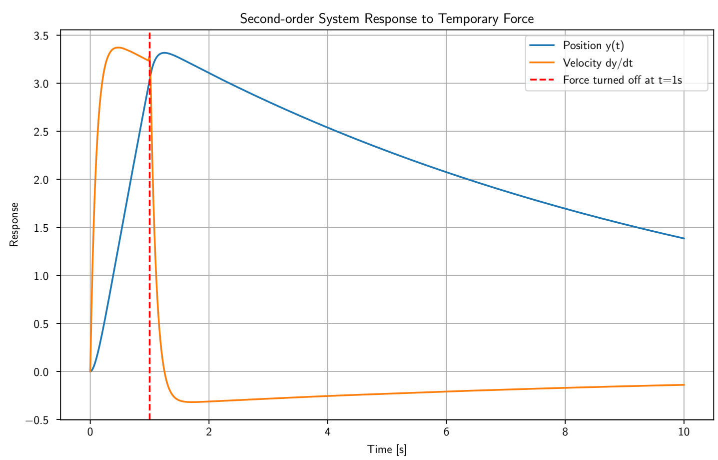


Figure 4.2: System Behaviour

B	2
K	0.1
M	0.1

Table 4.3: Coefficents For Virtual Dynamics

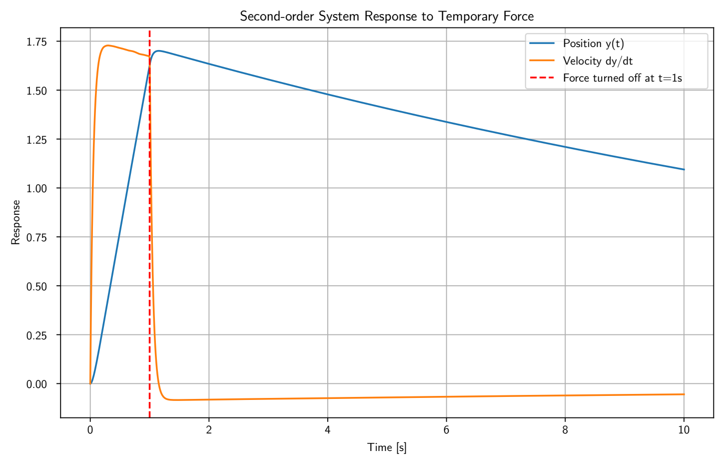


Figure 4.3: System Behaviour

B	2
K	0.8
M	0.1

Table 4.4: Coefficents For Virtual Dynamics

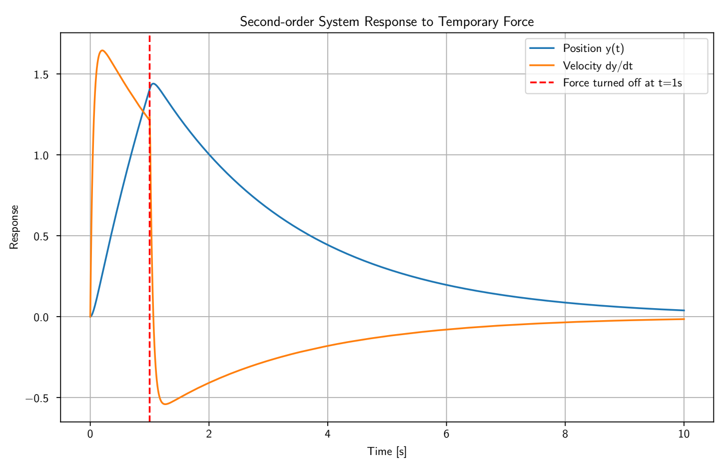


Figure 4.4: System Behaviour

4.8 Conclusion

The derived equations of motion describe the dynamics of the 2D horizontal plotter, accounting for both translational and rotational kinetic energies. The potential energy is considered negligible due to the stiffness of the mechanical design. Admittance control logic, incorporating virtual and actual dynamics, ensures that the system behaves as desired in response to external forces.

5 Working

In this project, we tried to implement admittance control on a 2D plotter setup, where the end effector is equipped with a force sensor to measure the applied external force. The primary objective is to regulate the system's response based on external force inputs, ensuring compliant motion while maintaining precise control over the end effector. The system consists of both virtual and real mathematical models, with a PID controller responsible for computing the control input to drive the real system.

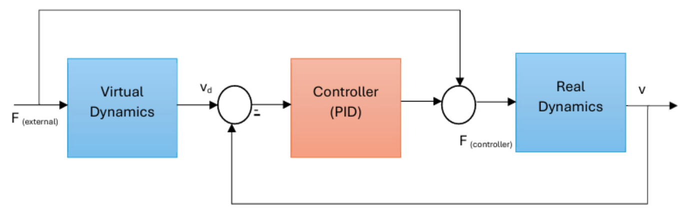


Figure 5.1: Basic admittance control diagram.

The overall structure of the admittance control system is illustrated in Fig. 1, which represents a basic stand-alone admittance control framework for an uncoupled robotic system. The control architecture consists of the following key components:

1. **Virtual Dynamics:** The external force applied by the user or environment is first processed by the virtual dynamics block, which determines the desired velocity reference v_d. This step ensures that the system behaves according to a predefined dynamic model, typically defined by mass-damping relationships, allowing the system to respond smoothly to applied forces.

2. **Error Computation:** The desired velocity is compared with the actual velocity v of the system to compute the velocity tracking error e. This error serves as the input to the controller, ensuring that the system continuously adjusts to match the desired motion profile.

3. **PID Controller:** A PID controller processes the velocity error e and generates a control force F_{control} to enforce the desired motion on the physical system. The PID controller optimizes this force to minimize deviations from the expected trajectory by adjusting the proportional, integral, and derivative terms based on system feedback.

4. **Robot Dynamics:** The real system receives the control force F_{control} from the actuator while also being influenced by the externally applied force F_{ext}. The interaction of these forces determines the actual velocity v of the system, which is then fed back into the error computation loop to maintain precise control.

5. **Feedback Loop and System Stability:** The feedback mechanism ensures that the system continuously refines its control inputs, allowing it to adapt to changes in applied forces while maintaining smooth motion. The admittance control approach enables the robot to dynamically adjust to variations in external forces, making it highly suitable for medical applications that require controlled and adaptive interactions with the environment.

6 Conclusion

In this project, we aimed to implement admittance control on a 2D plotter setup by modeling system dynamics and designing a control strategy based on external force measurements. Although we successfully developed the experimental setup, calibrated the force sensors, and designed a basic control loop using a virtual mass-spring-damper model, we were ultimately unable to implement full admittance control. The main limitation was our inability to achieve direct torque control of the stepper motors, which restricted us from realizing the intended compliant behavior based on external forces. Despite this, the project provided valuable insights into force-based motion control, sensor integration, and system modeling, laying a strong foundation for future work involving torque-controlled actuators.