3 D.O.F ROBOTIC ARM

A Project Report

Submitted to the APJ Abdul Kalam Technological University

in partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Electrical & Electronics Engineering

by

ADIL MOHAMMED(CHN21EE003)

AFSAL MUHAMMED(CHN21EE007)

VISHESH V RAJ(CHN21EE051)

ALAN BIJU VARGHESE(CHN21EE010)

DEPARTMENT OF ELECTRICAL ENGINEERING

COLLEGE OF ENGINEERING CHENGANNUR

KERALA

April 2025

DECLARATION

We hereby declare that the project report **3 D.O.F ROBOTIC ARM**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of **Shri. Biju Kumar K** This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

ADIL MOHAMMED
AFSAL MUHAMMED
VISHESH V RAJ
ALAN BIJU VARGHESE

Chengannur
10-03-2025

ABSTRACT

In the dynamic landscape of contemporary technological advancement, robotics stands as a prime example of innovation aimed at meeting the evolving needs of society. The project aims to design, model, and prototype a "3 Degrees of Freedom (DOF) robotic arm with a movable base", intended for various industrial and research applications requiring flexibility and adaptability. What sets this design apart from traditional robotic arms is the inclusion of a movable base. This mobile base provides the arm with enhanced agility, allowing it to cover a wider working area and reach previously inaccessible positions. This mobility is achieved through a custom-designed mobile platform that incorporates wheels or tracks, coupled with an efficient drive system for movement control. The primary goals of this project include the integration of hardware and software components to control the arm's movement, ensure stability during operation, and maintain precise positioning. The design process includes selecting suitable actuators, sensors, and controllers to maintain the robustness of the system in a dynamic environment. Additionally, a key consideration is the development of an intuitive user interface for easy control and programming, allowing operators to specify complex tasks for the robotic arm to execute autonomously or via manual guidance. Moreover, the potential applications of this robotic arm are vast, ranging from industrial automation in manufacturing to autonomous systems in research labs, and even in service robots for dynamic environments like hospitals or warehouses. The successful implementation of this 3 DOF robotic arm with a movable base could represent a significant advancement in the development of multifunctional robotic platforms that can adapt to different environments and tasks with ease

ACKNOWLEDGEMENT

We take this opportunity to express our deepest sense of gratitude and sincere thanks to everyone who helped us to complete this work successfully. We express our sincere thanks to **Dr. RAJU MANUEL** , Head of Department, Electrical and Electronics Engineering, College of Engineering Chengannur for providing us with all the necessary facilities and support. We would like to express our sincere gratitude to the **DR. RAJEEVAN A K**, department of Electrical and Electronics Engineering, College of Engineering Chengannur for the support and co-operation. We would like to place on record our sincere gratitude to our project guide **Shri. Biju Kumar K**, Assistant Professor, Electrical and Electronics Engineering, College of Engineering Chengannur for the guidance and mentorship throughout this work. Finally we thank our family, and friends who contributed to the successful fulfillment of this project work.

ADIL MOHAMMED
AFSAL MUHAMMED
VISHESH V RAJ
ALAN BIJU VARGHESE

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 OVERVIEW

This project focuses on the development of a 3 Degrees of Freedom (DOF) robotic arm with a movable base, designed to enhance flexibility and efficiency in robotic systems. By combining precise manipulation with mobility, this system can navigate a larger workspace and perform complex tasks that traditional fixed-base robots cannot. The project addresses key challenges in current control systems and aims to improve adaptability and operational efficiency

1.2 GENERAL BACKGROUND

The rapid advancement of robotics and automation technologies has led to the development of more versatile and adaptable robotic systems.This project aims to design and prototype such a robotic arm, combining hardware and software to create a functional system that can perform complex manipulation tasks while navigating within its workspace. At the heart of the control system is a Raspberry Pi 4, a powerful and compact single-board computer. The Raspberry Pi serves as the main controller for the robotic arm, handling the decision-making, data processing, and coordination of various subsystems. With its advanced processing capabilities, the Raspberry Pi enables real-time control and integration with other components, making it ideal for robotics applications. The movable base of the robot, powered by motors controlled by the Arduino, is designed to extend the robotic arm's reach within a defined workspace, allowing it to navigate and reposition itself as needed. The Raspberry Pi Camera is integrated into the system for visual feedback, allowing the robotic arm to interact with its environment through object detection and recognition. The camera aids in tasks such as object tracking, picking, and precise placement by providing a real-time video feed that the system can analyze to make informed decisions. This visual input enhances the arm's functionality and helps in automating tasks that require high accuracy, such as assembly or sorting operations.

Together, the combination of Raspberry Pi, Arduino, and the camera system creates a highly functional, adaptable robotic platform that can handle a variety of tasks. The integration of these components into a cohesive system aims to improve the arm's performance, expand its range of capabilities, and allow it to function effectively in both industrial and research settings

1.3 PROBLEM IDENTIFICATION

In developing a 3-DOF robotic arm with a movable base, several challenges may arise, particularly in areas of control, coordination, and stability. The complexity of synchronizing the arm's movements with a mobile base requires advanced control algorithms to ensure smooth operation. Balancing the weight distribution between the arm and the base is critical, as an unbalanced system could affect performance and precision. The power supply must be managed efficiently, as both the robotic arm and the movable base require substantial energy, which could limit battery life. Additionally, achieving high accuracy in positioning is challenging due to the dynamic nature of the movable base, making calibration and real-time feedback crucial. To address these issues, sensor fusion, precise kinematic modeling, and advanced power management techniques can be used to optimize the system's overall performance. Moreover, the cost and complexity of integrating multiple actuators, sensors, and control systems must be carefully considered to ensure the project's success.

1. Design: To Create a Robust Frame That Provides Three Degrees of Freedom

2. Develop the Control Systems and Implement a Control System Using a Microcontroller

3. Test and Calibrate: Ensure Precise Movements, and Arm Orientation

4. 1. Cost: To Make an Affordable Robotic Arm That Can Move Precisely and Efficiently.

Chapter 2

Literature Review

The paper by Bianchi et al. (2012) – Advanced Motion Control for Robotic Systems: Bianchi et al. (2012)[4] explored advanced motion control techniques for robotic arms, focusing on improving the precision and responsiveness of robotic systems through enhanced algorithms. Their work introduced sophisticated control strategies, including adaptive control, which dynamically adjusts the robot's behavior based on real-time feedback. This technique helps ensure that even under varying loads or environmental changes, the robot's movement remains stable and accurate. For a 3-DOF robotic arm with a movable base, such adaptive control methods are essential to account for the added variability and instability that comes with a mobile base. The ability to adapt to different conditions ensures that the robotic arm can maintain high accuracy and smooth functionality in complex, real-world environments.

Dufour and Bossens (2010) – Precision and Control in 3-DOF Robotic Arms: Dufour and Bossens (2010) explored the precision and accuracy required for robotic arms, particularly focusing on systems with three degrees of freedom (3-DOF). They discussed the use of inverse kinematics for controlling the arm's movement. By calculating the joint angles required to position the end effector at a desired location in space, the study highlighted the importance of accurate mathematical modeling and control algorithms to ensure the robot's precision. Their research demonstrated how the implementation of these algorithms is crucial for achieving accurate and repeatable movements in robotic arms used in industrial and medical applications. This work is foundational for understanding how robotic arms can operate precisely while maintaining optimal positioning in 3D space.

The paper authored by Aimn Mohamed Ahmed Ghiet focuses on the topic of controlling a robot arm using Arduino technology . I t details the design, implementation, and testing of a system that utilizes Arduino microcontrollers to control the movements and functions of a robotic arm. It covers aspects such as the hardware setup, software programming, and communication protocols employed to achieve the desired control functionalities. Additionally, it discusses the applications and potential benefits of such a system in various fields, including education, research, and industrial automation. This work contributes to the expanding domain of robotics by demonstrating a practical and accessible approach to controlling robotic arms using Arduino technology

The paper by Sharath Surati, Shaunak Hedaoo, Tushar Rotti, Vaibhav Ahuja, and Nishigandha Patel (2021) **[3]** focuses on the comprehensive review of pick and place robotic arm technologies and methodologies. Their work has included analyzing various kinematic configurations, end-effector designs, and control strategies for effective object manipulation. Surati et al.'s research has been geared toward evaluating the performance of different arm designs under various operational conditions, allowing for informed selection of appropriate mechanisms based on specific application requirements. Their work in categorizing and comparing robotic arm implementations highlights the importance of tailored design approaches for specific manipulation tasks.

The research conducted by Aimn Mohamed Ahmed Ghiet (2017) **[4]** explores the practical implementation of robotic arm control using Arduino microcontrollers, demonstrating the viability of cost-effective control solutions for precision manipulation. Ghiet's work addresses the challenges of implementing complex kinematic calculations on resource-constrained platforms and provides optimized approaches for servo motor control and position feedback. The research emphasizes the importance of efficient code implementation and control loop optimization to achieve responsive arm movements despite the limitations of microcontroller-based systems.

Chapter 3

Requirement Analysis

3.1 INTRODUCTION

This chapter provides a Requirement Analysis of Societal Needs, Target Customers, and Market Feasibility/Survival

3.2 Societal Needs

The development of a 3-DOF robotic arm with a movable base addresses several societal needs by advancing automation and robotics in various industries. As industries increasingly seek to improve efficiency, precision, and safety, robotic arms with enhanced mobility and flexibility can significantly reduce human labor in repetitive, hazardous, or high-risk tasks. For example, in manufacturing, assembly, or packaging lines, such robotic arms can improve production speed while ensuring consistent quality. Furthermore, in medical fields, mobile robotic arms can assist in surgeries or rehabilitation, offering greater precision and flexibility in handling delicate tasks. Additionally, the affordability and versatility of such a robotic arm can make advanced automation accessible to smaller businesses, fostering innovation and competitiveness in emerging markets. This project also opens up opportunities for education, providing an accessible platform for learning and experimenting with robotic technology. Ultimately, the 3- DOF robotic arm with a movable base can improve productivity, safety, and quality of life in numerous sectors, contributing to the broader advancement of automation technology in society.

3.3 Target Customers

3.3.1 Small and Medium-Sized Manufacturing Companies

Need for Automation: These companies often seek affordable solutions to automate tasks such as assembly, packaging, and material handling. Efficiency and Productivity: A robotic arm with precise control and flexibility allows these companies to improve productivity, reduce manual labor, and maintain consistent quality.

3.3.2 Research Organizations

Prototyping and Testing: Research labs working on robotics, AI, and automation can use the robotic arm as a testbed to prototype and develop new technologies. Experimentation with Control Systems: The robotic arm allows researchers to experiment with advanced control systems, kinematics, and sensor integration, making it a valuable asset for developing new techniques.

3.3.3 Healthcare Industry

Assisting in Medical Procedures: In medical fields like surgery or rehabilitation, robotic arms offer precision and flexibility in handling delicate tasks, such as minimally invasive surgery or physical therapy. Accessible for Hospitals and Clinics: The relatively affordable price of the robotic arm makes it a viable option for hospitals or clinics looking to integrate automation and robotics i nto patient care without huge costs.

3.4 Market Feasibility/Survival

The market feasibility of the 3-DOF robotic arm with a movable base is strong due to its affordability, versatility, and broad potential applications across various industries. With the growing trend of automation in sectors like manufacturing, healthcare, and education, the demand for cost-effective robotic solutions is increasing. Small and medium-sized businesses, in particular, are seeking affordable automation options to improve efficiency, reduce labor costs, and maintain quality. The robotic arm's unique feature of a movable base adds flexibility, allowing it to tackle tasks that traditional robotic arms cannot, making it an attractive solution for diverse tasks.

Educational institutions and research organizations also represent key target markets, as this robotic arm provides a hands-on, affordable learning tool for students and researchers. Additionally, its potential for growth in emerging markets, along with the continuous expansion of the global robotics market, ensures a promising future. By meeting the needs of cost-sensitive customers and offering a scalable, innovative design, the project has strong market survival potential, positioning itself for long-term success.

Chapter 4

Hardware Components

This chapter describes the key hardware components used in the project, including their specifications, functions, and applications. Each section contains a detailed description and a corresponding image of the component.

4.1 Lithium-Ion Battery (3.7V)

A 3.7V lithium-ion battery is a rechargeable power source widely used in portable electronics, robotics, IoT devices, and embedded systems due to its high energy density, long lifespan, and lightweight nature. Unlike traditional lead-acid or nickel-based batteries, lithium-ion batteries offer a higher charge capacity, lower self-discharge rate, and longer cycle life, making them ideal for modern electronic applications.

These batteries typically operate at a nominal 3.7V, but when fully charged, they can reach 4.2V, and they discharge down to around 3.0V before requiring a recharge. The capacity of the battery, measured in milliampere-hours (mAh), determines how long it can supply power to a device, with capacities varying from 500mAh to over 5000mAh, depending on the application.

Figure 4.1: Lithium-Ion Battery (3.7V)

4.2 Arduino Uno

The Arduino Uno is a widely popular microcontroller development board based on the ATmega328P microcontroller. It is specifically designed for electronics prototyping, automation, and embedded systems applications, offering a user-friendly and open-source platform for both beginners and experienced engineers.

The board features 14 digital input/output pins, including 6 PWM-enabled pins, as well as 6 analog input pins, which allow it to interface with a wide range of sensors, actuators, and external hardware components.

Figure 4.2: Arduino Uno Microcontroller Board

4.3 USB Camera

A USB camera is a compact and versatile digital imaging device that connects to a computer, microcontroller, or single-board computer via a USB interface. These cameras are commonly used for capturing images, video streaming, computer vision, surveillance, and artificial intelligence-based applications.

USB cameras play a crucial role in computer vision applications, particularly in areas like facial recognition, object detection, motion tracking, and automated surveillance.

Figure 4.3: USB Camera for Computer Vision Applications

4.4 MG996R Servo Motor

The MG996R is a powerful high-torque metal-geared servo motor designed for applications that require strong and precise rotational movement. It operates within a voltage range of 4.8V to 7.2V, producing a torque of 9.4 kg/cm at 4.8V and up to 11 kg/cm at 6V, making it significantly stronger than standard servos.

Figure 4.4: MG996R High-Torque Servo Motor

4.5 SG90 Servo Motor

The SG90 is a lightweight and compact plastic-geared servo motor that is widely used in applications requiring small, precise movements. Unlike the MG996R, which is built for high torque applications, the SG90 provides a modest torque of around 1.2 kg/cm at 4.8V and 1.6 kg/cm at 6V.

Figure 4.5: SG90 Servo Motor for Small-Scale Applications

4.6 Raspberry Pi 4

The Raspberry Pi 4 is a powerful single-board computer (SBC) designed for applications that require computational power beyond that of a traditional microcontroller. It is equipped with a quad-core Cortex-A72 processor running at 1.5GHz and comes in 2GB, 4GB, and 8GB RAM configurations.

The board features dual 4K micro HDMI outputs, USB 3.0 ports, Gigabit Ethernet, Wi-Fi 802.11ac, and Bluetooth 5.0, making it highly versatile for both consumer and industrial applications.

Figure 4.6: Raspberry Pi 4 Single-Board Computer

4.7 12V DC Motor Details

The **12V DC motor** used in this project operates on a voltage rating of 12V DC with a current consumption of approximately **1.5A to 2A**, depending on the load. It delivers a torque of around **1.2 to 1.5 kg.cm**, making it suitable for driving the **movable base** of the robotic arm. The motor has a speed range of approximately **100 to 300 RPM**, which is ideal for smooth and controlled movements. It features a **6mm shaft diameter** and comes with an integrated **gearbox** to increase torque while reducing speed, ensuring precise and stable motion. The motor is bidirectional, meaning its rotation direction can be reversed by switching the polarity of the power supply. It operates efficiently within a temperature range of **-10°C to 50°C**.

In the robotic system, **four 12V DC motors** are employed to power the base, enabling multi-directional movement, including **forward, backward, left, and right**. The motors are controlled by an **L298 motor driver**, which receives **PWM signals** from the microcontroller (Arduino/Raspberry Pi) to regulate both the speed and direction. The motor's performance can be fine-tuned by adjusting the PWM signal, allowing for smooth and accurate navigation.

4.8 12V DC Motor Details

The **12V DC motor** used in this project operates on a voltage rating of 12V DC with a current consumption of approximately **1.5A to 2A**, depending on the load. It delivers a torque of around **1.2 to 1.5 kg.cm**, making it suitable for driving the **movable base** of the robotic arm. The motor has a speed range of approximately **100 to 300 RPM**, which is ideal for smooth and controlled movements. It features a **6mm shaft diameter** and comes with an integrated **gearbox** to increase torque while reducing speed, ensuring precise and stable motion. The motor is bidirectional, meaning its rotation direction can be reversed by switching the polarity of the power supply. It operates efficiently within a temperature range of **-10°C to 50°C**.

Figure 4.7: 12V DC Motor used in the project

In the robotic system, **four 12V DC motors** are employed to power the base, enabling multi-directional movement, including **forward, backward, left, and right**. The motors are controlled by an **L298 motor driver**, which receives **PWM signals** from the microcontroller (Arduino/Raspberry Pi) to regulate both the speed and direction. The motor's performance can be fine-tuned by adjusting the PWM signal, allowing for smooth and accurate navigation.

Chapter 5

Design

5.1 Design Specifications

- Motor Driver: The driver required to run the motor is a 25W, 2A L298N motor driver.

- Servo Motor:

  - The servo used is a lightweight SG90 servo with an operating voltage of around 4.8V–6V.
  - Torque:
    - 1 kg/cm = 0.098 N/m
    - 2 kg/cm = 2 × 0.98 = 0.96 N-m
  - Since the maximum weight is around 1.5 kg, the servo should be able to handle the load and weight.

5.2 Battery Specification

5.2.1 Base Power Requirements

- Two 12V DC motors need 2A current each:

$$2 \times 2A = 4A$$

- A 6V DC servo requires a current of 1A:

$$1 \times 1A = 1A$$

- An Arduino Uno requires around 1A current:

$$1 \times 1A = 1A$$

Total current:

$$4A + 1A + 1A = 6A$$

For a runtime of 1 hour:

$$\text{Battery capacity} = 6A \times 1h = 6Ah$$

Adding a buffer of 40%:

$$6Ah \times 1.4 = 8.4Ah$$

Thus, a battery of **10,000 mAh** is selected for the base.

5.2.2 Arm Power Requirements

- **SG90 Servo:**
  - Operates at 250mAh at heavy loads.
  - For 3 servos:

$$3 \times 250mAh = 750mAh$$

  - Adding a buffer of 30%:

$$0.75Ah \times 1.3 = 0.975Ah \approx 1000mAh$$

- **MG995 Servo:**
  - Battery usage at high load: 1A

- For a runtime of 1 hour:

$$1A \times 1h = 1Ah$$

- Adding a buffer of 40%:

$$1Ah \times 1.4 = 1.4Ah \approx 1400mAh$$

5.3 Detailed Design

This chapter presents the detailed design of the robotic arm, covering key aspects such as design specifications, circuit design, simulation, and overall design work. The design specifications define the mechanical and electrical requirements, ensuring optimal functionality. The circuit design outlines the electrical connections, including motor drivers, sensors, and the microcontroller. Simulation is performed to analyze the system's performance before physical implementation. Finally, the design work integrates all components, ensuring smooth operation and efficiency.

5.4 Mechanical Plan of the Robotic Arm

The mechanical plan of the robotic arm depends on a robot controller with capabilities similar to a human arm. The automated arm framework typically comprises:

- **Links:** Representing the ”bones” of the arm.
- **Joints:** Connecting the links to form an open kinematic chain.
- **Actuators:** Driving the movement of the joints.
- **Sensors and Controllers:** Providing feedback and control.

One end of the kinematic chain is attached to the robot base, while the other end is equipped with a tool (hand, gripper, or end-effector) comparable to a human hand to perform tasks and interact with the environment.

There are two types of joints used:

- **Prismatic joints:** Allow linear motion.
- **Rotational joints:** Allow angular motion.

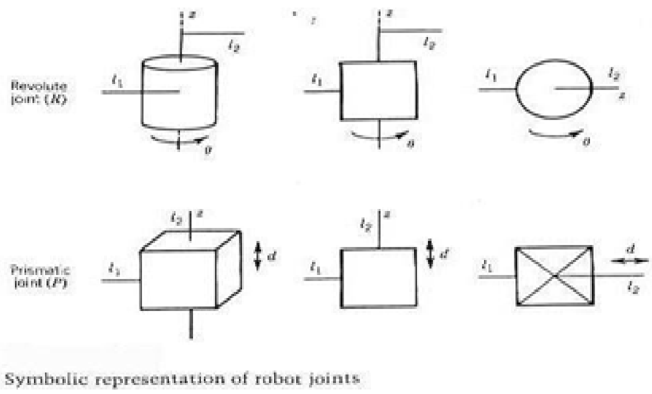

Figure 5.1: Two types of joints in the robotic arm.

5.5 Kinematic Chain and Degrees of Freedom

The links of the robotic arm are connected by joints allowing rotational motion, forming a kinematic chain. The mechanical arm is designed with four degrees of freedom (DOF), as it is sufficient for most operations while balancing complexity and cost.

In a mechanical system, the number of DOF is determined by the number of independent joint variables.

5.5.1 Number of Degrees for Specific DOF

The robotic arm uses four servo motors, with each motor representing one DOF. The motors operate at 6V, and the rotation ranges for each DOF are shown in Table 5.1.

DOF	**Motor Type**	**Range of Rotation**
1	Base rotation	0° to 180°
2	Shoulder movement	0° to 135°
3	Elbow movement	0° to 120°
4	Gripper movement	0° to 90°

Table 5.1: Number of Degrees for Specific DOF

5.6 AutoCAD of Parts

The base of the robotic arm is designed to be slightly wider at the bottom, slanted at 20 degrees to the top, and reinforced with ribs for added stiffness. The part is designed with a tetragonal shape to evenly distribute the load, accommodating the weight of the object being lifted, servo motors, and links.

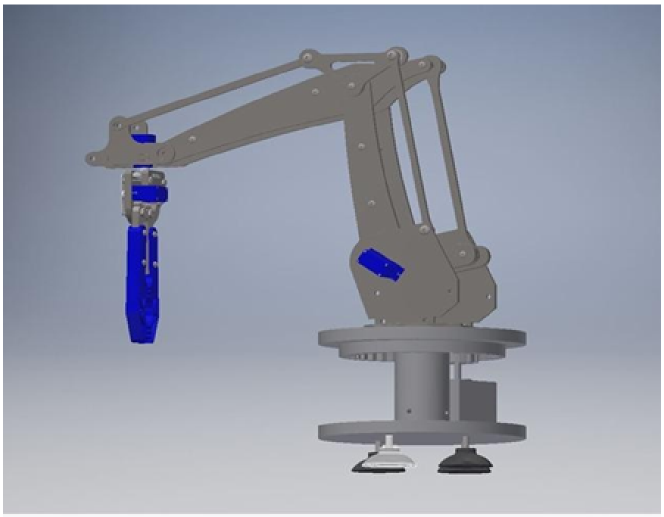

Figure 5.2: AutoCAD design of the robotic arm.

The base appendage is intended for rotational motion, while the motor at the base handles the up-down movement. The overhanging section contains the wiring, which is activated by the base servo motor. The next appendage, shown below, features a broad design with space for the motor.

The lower part connects to the previous appendage using a linking wire, housing the motor for the gripper. The gripper consists of:

- **Lower claw:** Fixed at a specific angle.
- **Upper hook:** Moves with the motor.

The croissant-shaped hook design enhances the gripping performance.

5.7 Robotic Base

The four-wheel controlled robotic rover is designed with a 120-degree V-type wheel arrangement, providing:

- Enhanced stability, maneuverability, and traction.
- Independent control of each wheel using a microcontroller (Arduino or Raspberry Pi).

The V-shaped wheel configuration:

- Improves handling of rough terrain and sharp turns.

- Balances weight distribution, reducing mechanical stress and enhancing traction.

The rover uses:

- **Motor drivers:** (L298N or DRV8833) to regulate power.
- **Power supply:** A 12V lithium-ion or lead-acid battery pack.

5.8 Circuit Diagram

The circuit diagram of the proposed model is divided into two sections: the robotic base and the arm.

5.8.1 Circuit Diagram of the Base

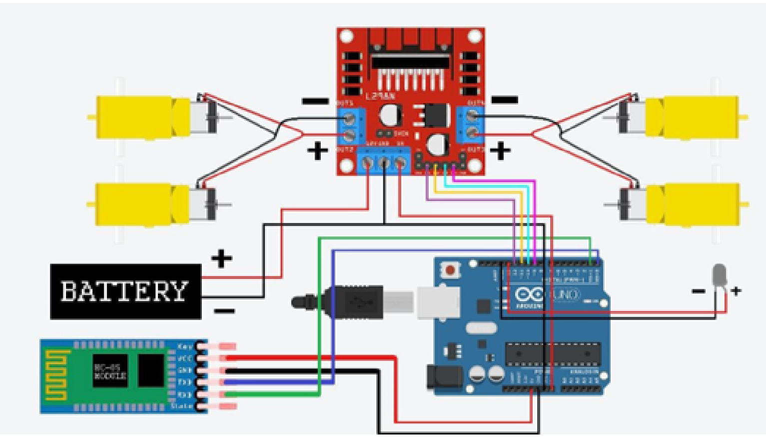

Figure 5.3: Circuit diagram of the base.

Main Components and Their Functions

- **Arduino Uno:** The brain of the system, processing Bluetooth commands and controlling motor movement.
- **HC-05 Bluetooth Module:** Enables wireless communication with the Arduino.
- **L298N Motor Driver:** Controls speed and direction of the four DC motors.
- **Battery Pack:** Powers the motors and Arduino.

5.8.2 Circuit Diagram of the Arm

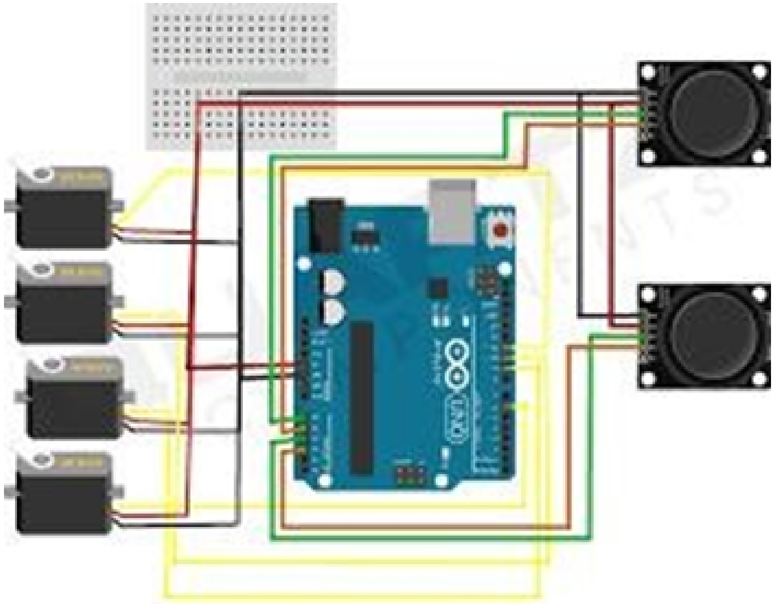

Figure 5.4: Circuit diagram of the robotic arm.

Components Used

- **Arduino Uno:** Controls the servos based on joystick input.
- **Joystick Modules (2x):** Each provides analog voltage signals for X and Y axes.
- **Servo Motors (4x):** Controlled by the Arduino's PWM signals.
- **Breadboard:** Used for power distribution.
- **Connecting Wires:** Link the components together.

Chapter 6

Simulation

6.1 Simulation of Robotic Arm and Base

This chapter presents the separate simulations of the robotic base and robotic arm, demonstrating their electronic configurations and movement controls.

6.1.1 Robotic Arm Simulation

Figure 6.1 illustrates the simulation of the robotic arm, which uses a Raspberry Pi Pico microcontroller connected to four servo motors. Each servo motor has three connections:

- **Power:** Typically 5V.
- **Ground:** Shared across all servos for a common electrical reference.
- **Signal wire:** Connected to the Pico's GPIO pins for independent movement control.

The Pico generates the control signals for each servo through specific GPIO pins, allowing precise movement control. This setup is widely used in robotic applications for simultaneously controlling multiple servos with synchronized motion.

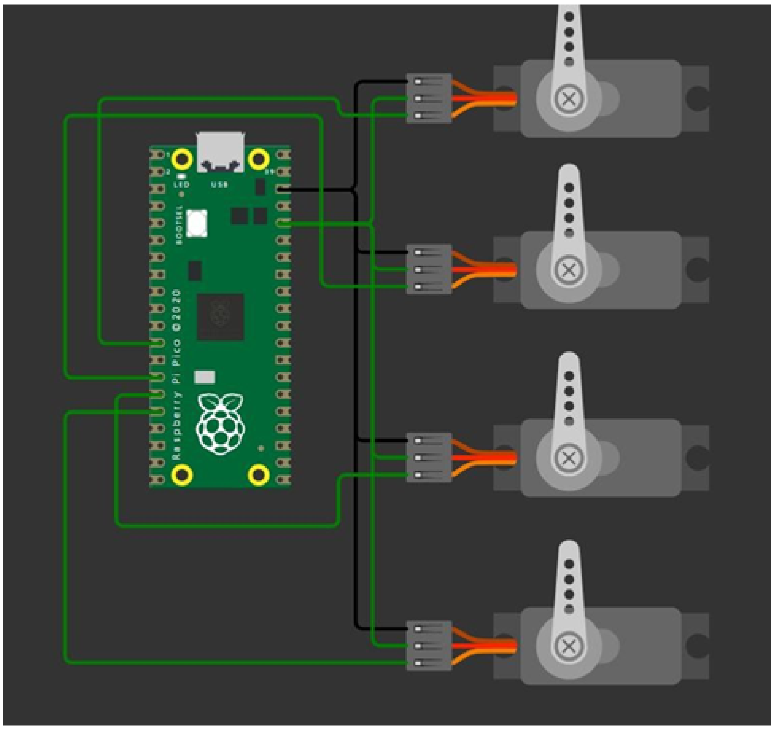

Figure 6.1: Robotic Arm Simulation using Raspberry Pi Pico and Servo Motors.

6.1.2 Robotic Base Simulation

Figure 6.2 displays the simulation of the robotic base. The circuit consists of an Arduino Uno microcontroller controlling two DC motors through an L298N motor driver module.

The L298N motor driver receives control signals from the Arduino through its input pins:

- **IN1** and **IN2**: Control the direction of Motor 1.
- **IN3** and **IN4**: Control the direction of Motor 2.

The motors are powered by a 12V supply (V1), while the logic of the L298N module is powered by a 5V supply (V2). A potentiometer (V3) is included, potentially for controlling the motor speed. This setup is commonly used in robotic projects that require dual motor control, providing precise and smooth movement capabilities.

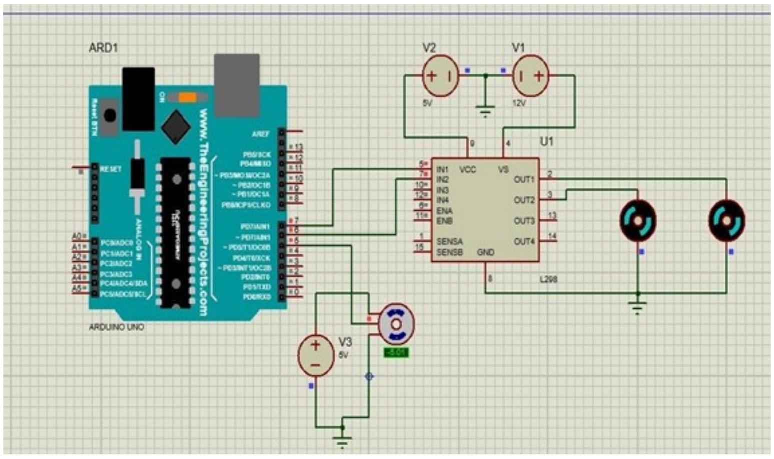

Figure 6.2: Robotic Base Simulation using Arduino Uno and L298N Motor Driver.

Chapter 7

Frame Design and Implementation

7.1 Introduction

Designing machine components demands expertise in Mathematics, Engineering Mechanics, Strength of Materials, Theory of Machines, and Engineering Drawing. Machines integrate linkages, gears, belts, and other mechanisms to achieve specific tasks efficiently.

When designing a robot, the initial steps involve determining its dimensions and workspace layout to meet specific requirements. Following this, each actuator's specifications are chosen. The arm connects to a heavy base for stability, especially when grasping objects. While stepper and gear motors are considered, servo motors are preferred for physical movement due to their ability to return to their initial position. Servos are programmed via signals from the microcontroller to meet the robot's needs.

7.2 Mechanical Design of the Arm

Below are the templates illustrating the parts necessary for assembling the robot arm. Each part is meticulously machined from acrylic with precision, allowing for a slight modular property, facilitating easy stacking or fitting with other parts.

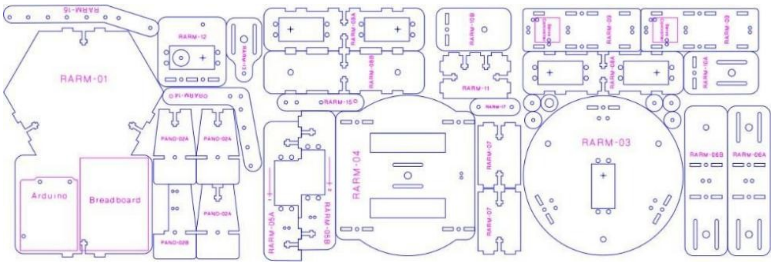

Figure 7.1: Templates of the Robot Arm

First, we assemble the base of the robot by assembling the base plate assembly and attaching the servo motor underneath, securing it in place with screws, as depicted in the image below. Then, we proceed with the assembly of the robot arm base, attaching a servo motor on one side and securing it firmly with screws.

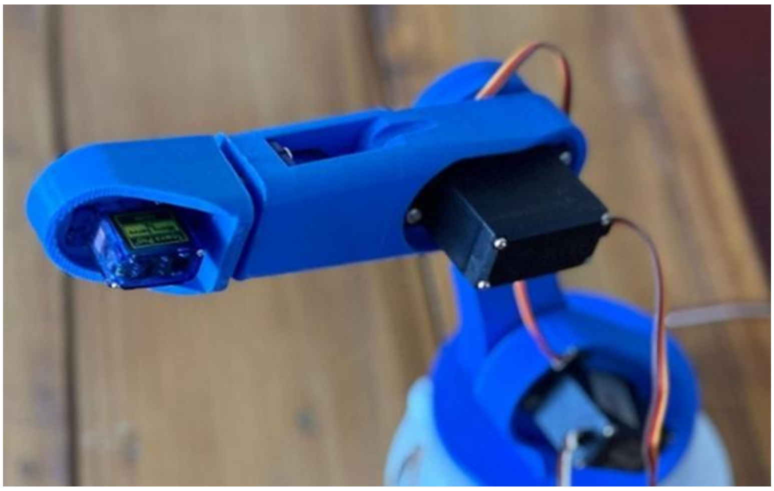

Figure 7.2: Servo motor attached to base assembly

Next, we attach two brackets and firmly fasten them on either side to ensure stability. The screws play a crucial role in enhancing stability and strength, particularly in the joints and overall frame of the robot. To attach the servo motor, we first remove the attached horns, position the servo motor as required, screw the machined parts together, and then reattach the servo horns using rivets to the respective bracket.

Figure 7.3: Screws attached to the servo motor

Using nuts and washers, we tighten the assembled robot arm, ensuring that it is snug but not overly tight to allow the servo motors to move freely within a limited range. Tightening the screws excessively can create tension in the servos, leading to their malfunction.

The purpose of the brackets is to enable swiveling of the top and bottom components along the axis of the servo motor's shaft. Once the first bracket is attached to the base, the next step is to attach the servo motor. Similarly, we remove the horns, position the servo motor, secure the parts together, and reattach the servo horns to the bracket using rivets.

Figure 7.4: Servos attached to the arm

In the arm assembly, only one horn of each of the two servo motors is attached to screws, allowing them to move back and forth with slight looseness to facilitate movement.

The gripper (end effector) of the arm employs a single actuator and operates on a fundamental physical gear principle. When the mini servo activates, it rotates the attached gear, thereby expanding or contracting the gripper accordingly. Once the arm is assembled, the gripper is affixed to the arm's end, and a mini servo is connected to it using the same method.

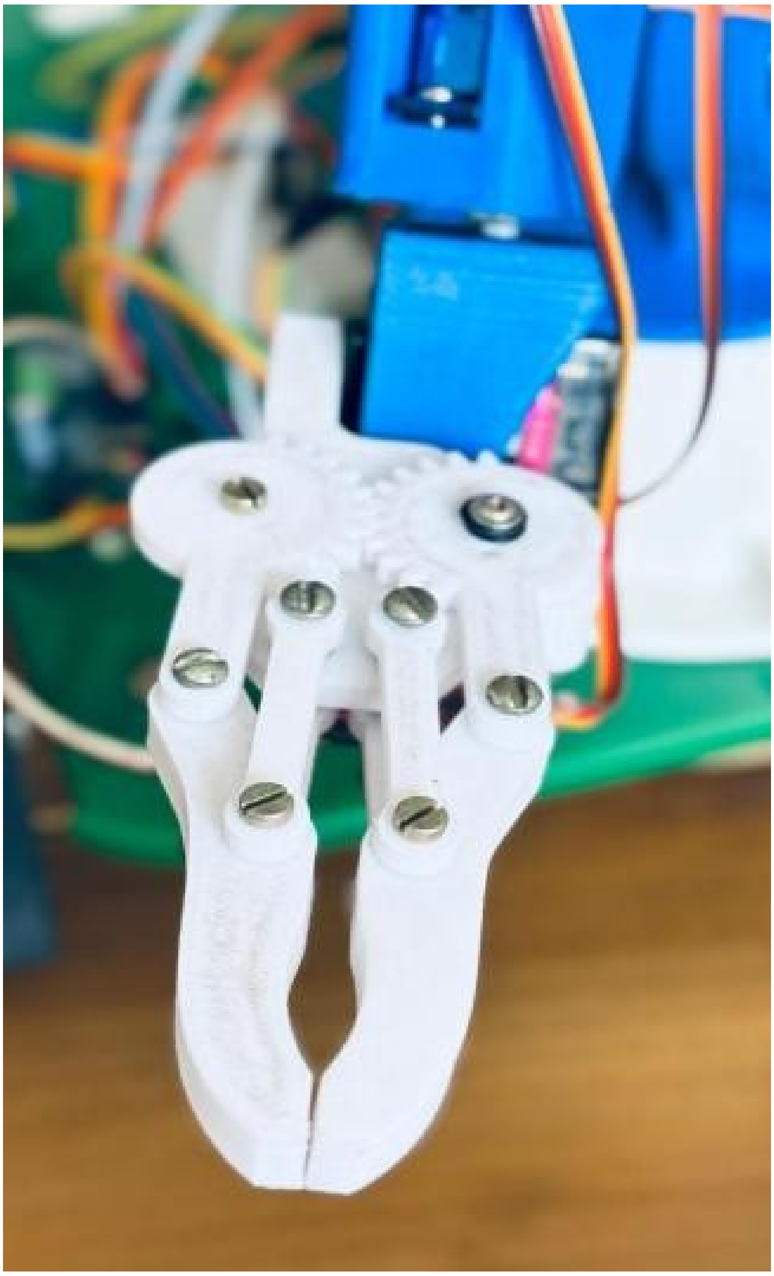

Figure 7.5: Gripper attached to the robot

The robot arm's circuit houses the microcontroller, a buck converter, an LED indicator for power supply status, and an electrolytic capacitor. The capacitor helps mitigate voltage fluctuations resulting from the servo motors' sudden current spikes, ensuring the system operates consistently and reliably.

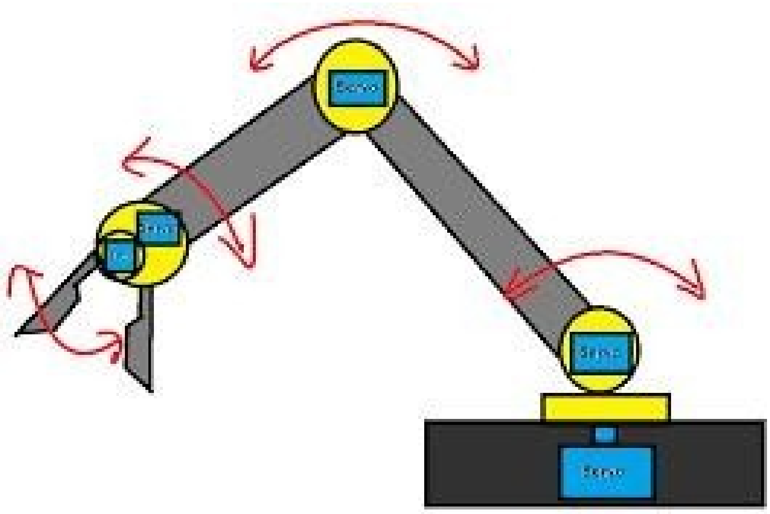

Figure 7.6: Movement of the arm

To prevent the robot arm from tipping over due to the powerful movements of the servo motors, a heavy block is securely attached below the breadboard. This block adds stability to the structure and prevents imbalance.

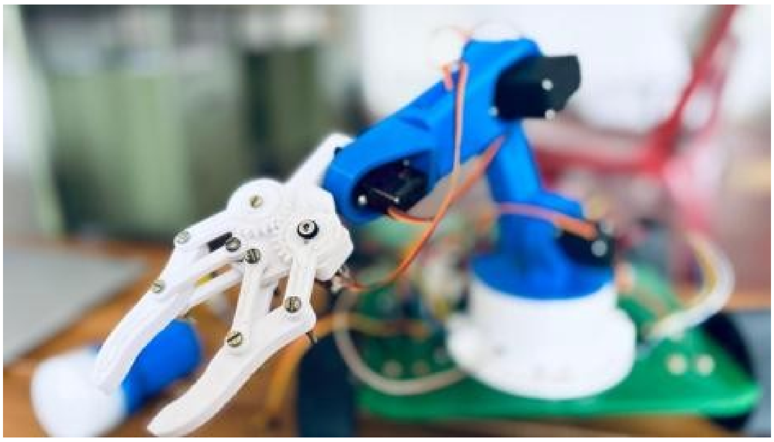

Figure 7.7: Construction of the robot arm

7.3 Frame Design of the Base

The frame is constructed with a triangular configuration, ensuring balanced weight distribution and smooth movement. Each corner of the base houses a servo-driven wheel or actuator, allowing precise control over direction and positioning. This design is ideal for robotic applications, exploration vehicles, or automated platforms, offering enhanced mobility in various terrains. The 120-degree structure provides a strong, adaptable foundation for future modifications and additional functionalities.

7.4 Wheel Controlling Setup

The wheel control setup is designed to enable smooth and precise movement of the robot. The figure below shows the setup for wheel controlling. Each servo-driven wheel is independently controlled, allowing for omnidirectional movement.

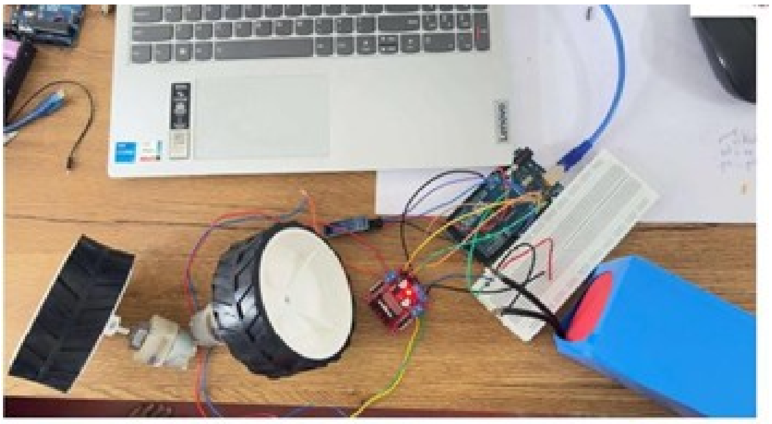

Figure 7.8: Wheel controlling setup

The base is powered by an Arduino/Raspberry Pi, with a Bluetooth, Wi-Fi, or RF module enabling wireless communication. Motor drivers control the wheels' movement, allowing for smooth navigation. The mobile app interface provides an easy-to-use control system, making it ideal for robotics, automation, and remote surveillance applications. This base showcases the integration of embedded systems, wireless communication, and motor control for a smart and interactive vehicle.

Chapter 8

Software Implementation

8.1 Base Movement Control Code

The base movement control code is responsible for receiving serial commands and controlling the movement of the robotic platform using four DC motors.

8.1.1 Code Overview

The code begins by defining the motor and LED pins:

```
#define led1 = 13
#define in1 = 12
#define in2 = 11
#define in3 = 10
#define in4 = 9
```

The `setup()` function initializes the serial communication and configures the motor and LED pins as outputs:

```
Serial.begin(9600);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
pinMode(12, OUTPUT);
pinMode(13, OUTPUT);
```

The `loop()` function continuously checks for serial data. When a character is received, it is compared with predefined commands to determine the movement:

- `'F'`: Moves the base forward by rotating motors in the appropriate direction.
- `'B'`: Moves the base backward.
- `'R'`: Rotates the base to the right.
- `'L'`: Rotates the base to the left.
- `'C'`: Moves the base diagonally (forward-right).
- `'A'`: Moves the base diagonally (forward-left).
- `'O'`: Turns the LED on.
- `'s'`: Turns the LED off.
- `'S'`: Stops all motors.

The corresponding motor pins are set to `HIGH` or `LOW` for directional control. The LED is toggled using the `digitalWrite()` function.

—

8.2 Robotic Arm Control Code

The robotic arm code uses an ESP8266 Wi-Fi module to receive commands via a web interface and control the arm's motors.

8.2.1 Wi-Fi Configuration

The code includes the ESP8266 libraries and defines the Wi-Fi credentials:

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>


const char* ssid = "BELLARI 4G";
const char* password = "sangeethindeWifi";
```

The server listens for connections on port 80:

```
ESP8266WebServer server(80);
```

In the `setup()` function, the ESP8266 connects to the specified Wi-Fi network and displays the local IP address:

```
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("Connected!");
Serial.println(WiFi.localIP());
```

—

8.3 Motor Control

The motor pins are configured as outputs:

```
#define IN1 D1
#define IN2 D2
#define IN3 D3
#define IN4 D4
#define ENA D5
#define ENB D6

pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
pinMode(ENA, OUTPUT);
pinMode(ENB, OUTPUT);
```

The functions to control the motor movements include:

- `moveForward()`: Rotates motors in the forward direction.

- `moveBackward()`: Rotates motors backward.
- `moveLeft()`: Rotates the robot left.
- `moveRight()`: Rotates the robot right.
- `handleStop()`: Stops all motors by setting all pins to `LOW`.

Each function uses `digitalWrite()` to control the motor direction and `analogWrite()` to control the speed:

```
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(ENA, 1023);
analogWrite(ENB, 1023);
```

—

8.4 Web Server Handling

The server handles web requests and routes them to the corresponding movement functions:

```
server.on("/", handleRoot);
server.on("/forward", moveForward);
server.on("/backward", moveBackward);
server.on("/left", moveLeft);
server.on("/right", moveRight);
server.on("/stop", handleStop);
```

In the `loop()` function, the server continuously handles client requests:

```
server.handleClient();
```

—

8.5 Conclusion

The software implementation effectively handles the robot's movements through both serial communication and Wi-Fi-based commands. The base platform uses basic serial commands for motor control, while the robotic arm leverages an ESP8266 Wi-Fi module and a web server to receive and execute movement commands. This modular software design ensures flexibility and remote control capabilities for the robotic system.

Chapter 9

Cost Estimation

9.1 Introduction

This chapter provides a detailed cost estimation of the components and materials used for the development and implementation of the **3-DOF robotic arm with a movable base**. The total cost is calculated based on the individual prices of the hardware components, ensuring transparency and accuracy in the budget assessment.

9.2 Component Costs

The table below lists the components used in the project along with their quantity, unit price, and total price.

Component	Quantity	Unit Price ()	Total Price ()
12V DC Motor	4	200	800
L298 Motor Driver	1	120	120
3.7V Battery	4	60	240
Arduino Uno	1	200	200
Camera	1	500	500
FTDI USB to Serial	1	150	150
MG996R Servo Motor	3	300	1200
SG90 Servo Motor	2	150	300
Raspberry Pi	1	5500	5500
128 GB SD Card	1	700	700
Robotic Arm	1	2000	2000
Chassis	1	1000	1000
Tyres	4	200	800
Power Supply	1	2000	2000
Raspberry Pi Cooler	1	500	500
PCA Multiple Servo Board	1	300	300

Table 9.1: Component costs for the 3-DOF robotic arm project.

9.3 Total Cost

The total cost of the **3-DOF robotic arm with a movable base** project is calculated by summing the individual component prices:

$$\text{Total Cost} = 800+120+240+200+2000+150+1200+300+5500+700+2000+1000+800+2000+500+300$$

$$\text{Total Cost} = 15,860$$

9.4 Conclusion

The estimated cost for building the **3-DOF robotic arm with a movable base** totals 15,860. This cost covers all necessary hardware components, ensuring smooth functionality, stable performance, and reliable operations. The budget estimation highlights

Chapter 10

Conclusion

10.1 Introduction

This chapter delves into the intricacies of the challenges faced during the design and execution phases of this project. It explores the nuances of each problem encountered and the corresponding resolutions implemented. Additionally, it outlines potential avenues for further development and expansion of the project in the future.

10.2 Conclusion

The project involving the design and implementation of the 3-DOF robotic arm with a camera and movable base successfully demonstrates the seamless integration of mechanical design, electronics, and software control. The system achieved precision manipulation and real-time monitoring through coordinated movements and object detection.

The following key points summarize the outcomes:

- **Successful Integration:** The project effectively combines mechanical design, electronics, and software programming to achieve synchronized and efficient operation.

- **Versatile Applications:** This prototype shows potential for various applications, including military, medical, social, educational, and industrial domains. It can assist individuals with special needs by providing autonomous manipulation capabilities.

- **Seamless Software Integration:** The project showcases smooth communication between the Processing language (C++) and Arduino programming (C) through serial port communication, ensuring real-time data transmission and control.

During the design and implementation phases of the project, several challenges and limitations were encountered:

- **Servo Motor Accuracy:** The servomotors utilized in this project demonstrate exceptional responsiveness. However, the accuracy of movement depends on the motor resolution, control algorithms, and mechanical stability. Small errors in joint movement can lead to positioning inaccuracies, especially in delicate tasks.

- **Camera Limitations:** The onboard camera's resolution, frame rate, and processing capabilities may restrict real-time image recognition and object detection efficiency. Additionally, external lighting conditions can significantly impact image clarity.

- **Processing Delays in Camera Feedback:** If the camera processing speed is slow or the system lacks sufficient computational power, visual feedback delays may occur. This can affect real-time operations and reduce automation efficiency.

- **Stability of the Movable Base:** The movable base must maintain stability to prevent excessive vibrations or tipping, especially when the arm extends or lifts objects. Operating on uneven surfaces can hinder movement precision and lead to instability.

Chapter 11

FUTURE SCOPE

11.1 Introduction

The integration of a 3-degree-of-freedom (DOF) robotic arm with a movable base offers significant potential for automation, flexibility, and efficiency. This report explores the future applications, technological advancements, and possible areas of improvement for such systems across various industries.

11.2 Industrial Applications

The combination of a mobile base and a robotic arm enhances automation capabilities in industrial settings. Flexible Material Handling the robot can autonomously navigate warehouses and factories to handle and transport objects. It can sort, pack, and perform quality inspections efficiently. Assembly Line Automation the arm can move between different stations to carry out repetitive tasks with high precision. Increases flexibility and productivity compared to fixed robotic arms.

11.3 Research and Development

The 3-DOF robotic arm with mobility serves as a platform for advancing robotics and artificial intelligence research. Robotic Mapping and Navigation (SLAM), When equipped with cameras, the robot can map unknown environments. Beneficial for autonomous navigation and exploration research. Artificial Intelligence and Machine Learning can be used to develop AI-based motion planning, obstacle avoidance, and adaptive control algorithms.Enhances autonomous decision-making capabilities.

11.4 Autonomous Mobile Manipulation

Combining mobility with object manipulation makes the robot suitable for complex tasks in dynamic environments. Warehouse and Logistics Automates transportation and material handling in warehouses. Improves efficiency in inventory management and order fulfillment. Agriculture is Used for precision farming tasks such as fruit picking, plant inspection, or pesticide spraying. Enhances productivity in agricultural operations.

11.5 Defense and Surveillance

The robotic system can be deployed in defense and emergency operations. Search and Rescue Missions Equipped with cameras and sensors, the robot can explore disaster zones to locate survivors. Can manipulate debris or deliver emergency supplies. Military Applications Suitable for surveillance, payload transport, and bomb disposal operations. Reduces human risk in hazardous areas.

11.6 Conclusion

The 3-DOF robotic arm with a movable base has immense potential across industries, including manufacturing, logistics, healthcare, defense, and research. With the integration of AI, IoT, and autonomous navigation, such systems can become increasingly versatile and efficient. The future of these robotic systems lies in enhanced mobility, adaptability, and intelligence, making them invaluable for automation and real-world applications.

Chapter 12

BIBLIOGRAPHY

1. B. Zhao, Q. Song, and W. Liu (2024) Power characterization of isolated bidirectional dual-active-bridge DC-DC converter with dual-phase-shift control.

2. Aimn Mohamed Ahmed Ghiet, (2017) Robot Arm Control with Arduino. Dept. of Mechanical and Aeronautical Engg. Turkish Aeronautical Association University.

3. L. Huang (2019) Multi-Fuzzy control-based energy management strategy of battery/super-capacitor hybrid energy system of electric vehicles.

4. Gonzalez et al. (2019) examined control strategies for hybrid robotic systems.

5. Stephan Waldherr; Roseli Romero Sebastian Thrun, A Gesture Based Interface for Human-Robot Interaction., Autonomous Robots, September 2000, Volume 9, Issue 2, pp 151-173

Chapter A

Appendix: Source Code

A.1 Base Movement Control Code

```
/* VERSION - 1.3 */
#define led1 = 13
#define in1 = 12
#define in2 = 11
#define in3 = 10
#define in4 = 9

void setup() {
    Serial.begin(9600);
    pinMode(9, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(11, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(13, OUTPUT);
}

void loop() {
    if (Serial.available() > 0) {
        char inputvalue = char(Serial.read());
```

```
if (inputvalue == 'F') {
    digitalWrite(12, HIGH);
    digitalWrite(11, LOW);
    digitalWrite(10, HIGH);
    digitalWrite(9, LOW);
}

else if (inputvalue == 'B') {
    digitalWrite(12, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(10, LOW);
    digitalWrite(9, HIGH);
}

else if (inputvalue == 'R') {
    digitalWrite(12, LOW);
    digitalWrite(11, LOW);
    digitalWrite(10, HIGH);
    digitalWrite(9, LOW);
}

else if (inputvalue == 'L') {
    digitalWrite(12, HIGH);
    digitalWrite(11, LOW);
    digitalWrite(10, LOW);
    digitalWrite(9, LOW);
}

else if (inputvalue == 'C') {
    digitalWrite(12, LOW);
    digitalWrite(11, HIGH);
```

```
            digitalWrite(10, HIGH);
            digitalWrite(9, LOW);
        }

        else if (inputvalue == 'A') {
            digitalWrite(12, HIGH);
            digitalWrite(11, LOW);
            digitalWrite(10, LOW);
            digitalWrite(9, HIGH);
        }

        else if (inputvalue == 'O') {
            digitalWrite(13, HIGH);
        }

        else if (inputvalue == 's') {
            digitalWrite(13, LOW);
        }

        else if (inputvalue == 'S') {
            digitalWrite(12, LOW);
            digitalWrite(11, LOW);
            digitalWrite(10, LOW);
            digitalWrite(9, LOW);
        }
    }
}
```

—

A.2 Robotic Arm Control Code

```
#include <ESP8266WiFi.h>
```

```
#include <ESP8266WebServer.h>

// WiFi credentials
const char* ssid = "BELLARI 4G";
const char* password = "sangeethindeWifi";

// Web server on port 80
ESP8266WebServer server(80);

// Motor pins
#define IN1 D1
#define IN2 D2
#define IN3 D3
#define IN4 D4
#define ENA D5
#define ENB D6

void setup() {
    Serial.begin(115200);
    WiFi.begin(ssid, password);

    // Wait for WiFi connection
    Serial.print("Connecting to WiFi");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("Connected!");

    // Show IP address
    Serial.println(WiFi.localIP());
```

```
    // Setup motor pins
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
    pinMode(ENA, OUTPUT);
    pinMode(ENB, OUTPUT);

    // Default motor state
    stopMotors();

    // Handle web requests
    server.on("/", handleRoot);
    server.on("/forward", moveForward);
    server.on("/backward", moveBackward);
    server.on("/left", moveLeft);
    server.on("/right", moveRight);
    server.on("/stop", handleStop);

    // Start server
    server.begin();
    Serial.println("Server started");
}

void loop() {
    server.handleClient(); // Listen for web clients
}

// ---------- Motor Movement Functions ----------

void moveForward() {
    Serial.println("Moving Forward");
```

```
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENA, 1023);
    analogWrite(ENB, 1023);
    handleRoot();
}

void moveBackward() {
    Serial.println("Moving Backward");
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    analogWrite(ENA, 1023);
    analogWrite(ENB, 1023);
    handleRoot();
}

void moveLeft() {
    Serial.println("Turning Left");
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENA, 1023);
    analogWrite(ENB, 1023);
    handleRoot();
}

void moveRight() {
```

```
    Serial.println("Turning Right");
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    analogWrite(ENA, 1023);
    analogWrite(ENB, 1023);
    handleRoot();
}

void handleStop() {
    Serial.println("Stopping");
    stopMotors();
    handleRoot();
}

void stopMotors() {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
}
```

—

A.3 Conclusion

This appendix contains the complete source code used to control the robotic platform and arm. The base code handles serial communication for basic movement, while the robotic arm code uses an ESP8266 Wi-Fi module to receive and execute web-based commands.