Design and Development of a Telechir Robotic Arm with Adaptive Grip Control

A PROJECT REPORT

Submitted by

Austin Jeremiah J	**311121106012**
Clatson J	**311121106016**
Geo Joseph	**311121106018**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

LOYOLA - ICAM COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
CHENNAI - 600 034

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2025

ANNA UNIVERSITY:CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**DESIGN AND DEVELOPMENT OF A TELECHIR ROBOTIC ARM WITH ADAPTIVE GRIP CONTROL** " is the bonafide work of **AUSTIN JEREMIAH J (311121106012), CLATSON J (311121106016), GEO JOSEPH (311121106018)** who carried out the project work under my supervision.

SIGNATURE

Ms. Jenifer Suriya L.J.

HEAD OF THE DEPARTMENT

Assistant Professor,

Department of Electronics and Communication Engineering

Loyola-ICAM College of Engineering and Technology
Loyola Campus, Nungambakkam
Chennai - 600 034.

SIGNATURE

Mr. Robert Rajkumar S.

SUPERVISOR

Assistant Professor,

Department of Electronics and Communication Engineering

Loyola-ICAM College of Engineering and Technology
Loyola Campus, Nungambakkam
Chennai - 600 034.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

Handling delicate objects in industries requires precise grip control to prevent damage while ensuring secure handling. This project addresses this challenge by integrating machine vision and real-time sensor feedback to dynamically estimate and adjust grip force.

The master-slave design consists of a glove (master) equipped with force resistance sensors to detect finger movements. These readings are processed by a microcontroller and transmitted wirelessly to the slave unit using Wi-Fi communication. The robotic arm (slave) is 3D printed and driven by servo motors, with PID control for precise actuation.

Object detection is employed to identify objects, and the model predicts the required grip force. The model is enhanced dynamically using the TensorFlow Data API and transfer learning, improving prediction accuracy over time.

This project presents a practical approach to human-like robotic grip control by integrating machine learning, real-time data processing, and adaptive feedback, enhancing performance in industrial and assistive robotic applications.

ACKNOWLEDGEMENT

We express our deepest gratitude to the Lord Almighty for His benevolent blessings and guidance, which have enabled the successful completion of our project.

We extend our heartfelt thanks to our parents and friends for their unwavering support and encouragement throughout this journey, which has been a source of immense strength and motivation.

We sincerely thank **Rev. Dr. S. Sebastian SJ,** Director, **Dr. Antony Michael Raj L,** Principal, for their invaluable guidance and support.

Our heartfelt gratitude goes to **Ms. Jenifer Suriya L J**, Assistant Professor and Head, Department of Electronics and Communication Engineering, for her consistent guidance and support throughout this project and during the semester.We are profoundly grateful to our guide and mentor, **Dr. Anitha Juliette A**, Associate Professor and project coordinator , for her invaluable insights, mentorship, and constant encouragement, which were instrumental in the successful completion of this project.

We extend our special thanks to **Mr. Robert Rajkumar S**, Assistant Professor and project guide, for his immense efforts, guidance, and engagement that greatly facilitated the completion of our project.

We also express our gratitude to **Mr. Jackson Irudhayam S**, Assistant Professor, Department of Mechanical Engineering for providing access to the Fab Lab, which was crucial for the execution of this project.

Finally, we thank all the faculty members of our department and the lab assistants for their support, contributions, and encouragement, which played a vital role in the successful completion of this endeavor.

TABLE OF CONTENT

I ABSTRACT iii

II LIST OF FIGURES vii

1 INTRODUCTION

1.1 Overview of Telechir Robotics **1**

1.2 Challenges in Adaptive Grip Control **2**

2 LITERATURE REVIEW

2.1 Review of Existing Systems **3**

2.2 Research Gaps **4**

3 METHODOLOGY

3.1 System Overview **5**

3.2 Design of Master [Glove] **6**

3.2.1 Hardware Components

3.2.2 Data Transmission

3.2.3 Data Analysis

3.3 Design of Slave [Robotic Arm] **8**

3.3.1 Hardware Components

3.3.2 Control Strategy

3.3.3 Error Correction

3.4 Object Detection and Grip Force Calculation **10**

3.4.1 Key Libraries Used

3.4.2 Model Selection and Preparation

3.5 Grip Detection via Computer Vision **11**

3.6 Error Correction & Dataset Updates **12**

4 TESTING AND EVALUATION 13

5 CONCLUSION & FUTURE SCOPE 18

REFERENCES 19

APPENDICES 20

LIST OF FIGURES

FIGURE NUMBER	TITLE	PAGE NUMBER
3.1.1	Block Diagram	5
3.1.2	Flowchart	5
3.2.1.1	Force Resistance Sensor	6
3.2.1.1	ESP32 8266	7
3.3.1.1	Servo Motor	8
3.3.1.2	Raspberry Pi 5	9
4.1	Glove / Master	13
4.2	Robotic arm / Slave	14
4.3	Object Identification with Grip Allocation	15
4.4	a) Robotic Arm holding Mobile Phone b) Robotic Arm holding cello tape	15
4.5	Complete System Integration	16

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF TELECHIR ROBOTICS

Robotics has emerged as a transformative technology, reshaping various industries by enhancing automation, precision, and efficiency. Among the specialized branches of robotics, *telechir robotics*—derived from “tele” (remote) and “cheir” (hand in Greek)—plays a crucial role in enabling remote manipulation of objects using robotic systems. These systems are designed to mimic the complex and nuanced movements of the human hand and arm, providing users with the ability to perform tasks at a distance with high precision.

A telechir robotic arm essentially functions as a remotely controlled extension of the human arm, capturing motion and replicating it through actuators and joints. This capability is particularly valuable in environments where direct human intervention is either unsafe, impractical, or impossible—such as hazardous industrial zones, biomedical laboratories, or space exploration missions.

The primary objective of this project is to design and develop a telechir robotic arm capable of performing complex tasks with human-like dexterity. A key feature of the proposed system is its adaptive grip control mechanism. Unlike traditional robotic arms with fixed or manually programmed grip force, this system will dynamically estimate and adjust the grip strength based on the object's properties, including size, texture, and fragility. This intelligent adaptation ensures the safe and effective handling of diverse objects, ranging from rigid components to delicate and breakable items.

1.2 CHALLENGES IN ADAPTIVE GRIP CONTROL

While the mechanical replication of human arm movements has advanced significantly, achieving human-like adaptability in gripping remains a critical and unsolved challenge in robotics. Human hands possess an inherent ability to judge the necessary amount of force required to hold or manipulate objects based on experience, tactile feedback, and visual cues. Recreating this functionality in a robotic system requires the integration of advanced sensors, control algorithms, and machine learning techniques.

One of the foremost challenges lies in grip force estimation. Applying too much force can lead to the breakage of delicate items such as glassware or medical tools, while too little force may cause the object to slip, resulting in operational inefficiency or potential damage. This delicate balance is difficult to maintain in dynamic environments, especially when dealing with unfamiliar or irregularly shaped objects.

In this project, these challenges are addressed through a combination of machine vision and real-time force feedback. The robotic arm will use computer vision to identify and classify objects, which in turn informs the system of the appropriate grip force required. Additionally, Force Sensitive Resistors (FSRs) embedded in the robotic hand will provide real-time feedback on the applied force, allowing continuous adjustment and correction via control algorithms such as PID controllers and reinforcement learning models. This ensures a stable, responsive, and intelligent gripping mechanism capable of adapting to new situations and object types.

CHAPTER 2

LITERATURE REVIEW

2.1 REVIEW OF EXISTING SYSTEMS

Existing systems for robotic arms typically rely on predefined grip forces or limited sensory feedback, which makes them unsuitable for handling delicate objects. Research has shown that incorporating machine vision and adaptive control enhances grip precision and object handling efficiency. Notable approaches include:

- Andrew Belford et al. (2020) used miniaturized strain sensors to provide a sense of touch in humanoid robots, improving grip sensitivity. [1]
- Sebastian Mick et al. (2022) developed a 3D-printed human-like robotic arm as a testbed for human-robot interaction strategies. [2]
- Ze-sheng Ding et al. (2022) introduced a benchmark dataset for chemical laboratory apparatus detection, improving monitoring and safety in robotic applications. [3]
- Richard Josiah C. Tan Ai et al. (2022) designed and optimized an adaptive robotic gripper using generative design and finite element analysis, achieving a 79% mass reduction while maintaining grip strength. [4]
- Sainul Islam Ansary et al. (2023) developed an adaptive robotic gripper capable of grasping objects with varied shapes using movable pulleys and tendon wires, validated on common household objects. [5]
- Muhammad Aqib et al. (2023) introduced a multifunctional underactuated two-fingered adaptive gripper with a double parallelogram mechanism, enabling multiple grasping techniques and stable handling. [6]
- Jaime Hernandez et al. (2023) conducted a comprehensive review of robotic arm grippers, comparing actuation mechanisms, degrees of freedom, and grasping capabilities. [7]

2.2 RESEARCH GAPS

Although recent developments in telechir robotic systems have improved remote manipulation and control accuracy, several technical limitations still exist that hinder their broader application and performance in dynamic environments. These gaps highlight areas where further research and development are necessary to enhance adaptability, learning capability, and operational precision.

Despite these advancements, key challenges remain:

- Lack of real-time adaptive grip force adjustment based on object type and weight.
- Insufficient use of reinforcement learning to minimize slippage or object damage.
- Limited dataset updates and transfer learning mechanisms for adapting to new objects without full retraining.

CHAPTER 3

METHODOLOGY

3.1 SYSTEM OVERVIEW

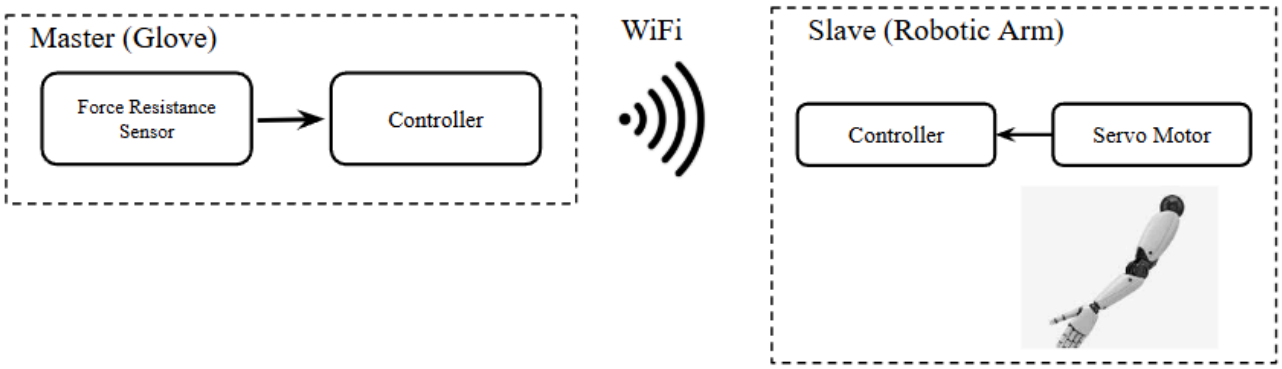

Figure. 3.1.1 Block Diagram

The block diagram illustrates the working of a telechir robotic arm using a master-slave architecture. The master unit consists of a wearable glove embedded with force resistance sensors, which detect the user's hand movements. These sensor readings are processed by a controller and transmitted wirelessly via Wi-Fi.On the slave side, the controller receives the data and drives the servo motors of the robotic arm accordingly, replicating the user's hand gestures in real time. This setup enables accurate and responsive remote control of the robotic arm.

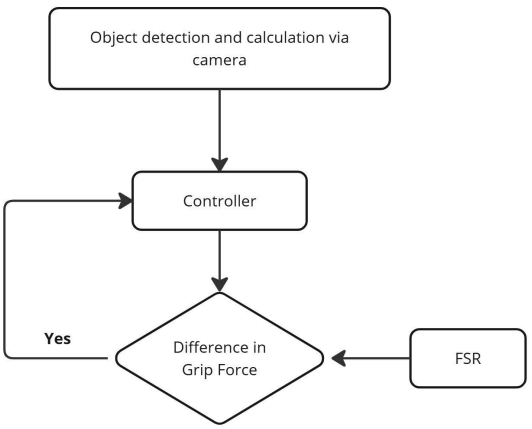

Figure. 3.1.2 Flowchart

The flowchart further illustrates the adaptive grip control process of the robotic arm. The system starts with object detection and grip force calculation using a camera. The data is sent to the controller, which compares the calculated grip force with the actual force measured by the FSR. If a discrepancy is detected, the controller adjusts the grip force in real-time to ensure a secure and precise grip, mimicking human-like handling. This approach ensures accurate object grasping, preventing excessive force that could damage delicate objects.

3.2 DESIGN OF MASTER (GLOVE)

3.2.1 Hardware Components

The master unit consists of a wearable glove equipped with input components designed to detect finger movements. These movements are captured through appropriate sensors or mechanical mechanisms, and the data is processed by an ESP8266 or ESP32 microcontroller. The microcontroller converts the sensor inputs into corresponding angular position values and transmits the data wirelessly using the MQTT protocol for real-time communication with the slave unit.

a) **Force Sensitive Resistors :** Attached to the glove to measure finger pressure.

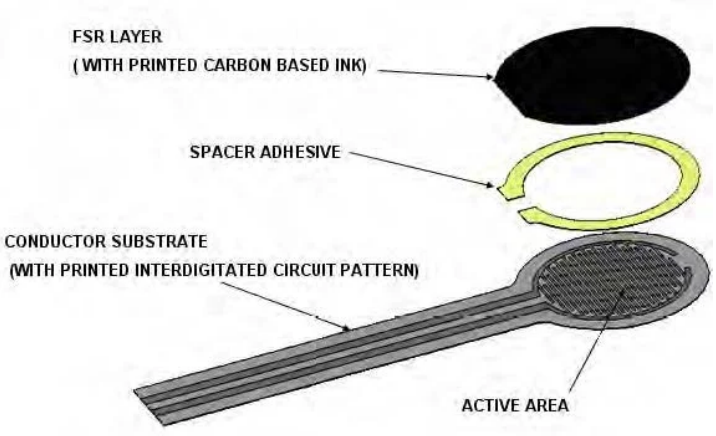

Figure. 3.2.1.1 Force SensitiveSensor

b) **ESP8266/ESP32 Module:** Used for data acquisition and wireless transmission.

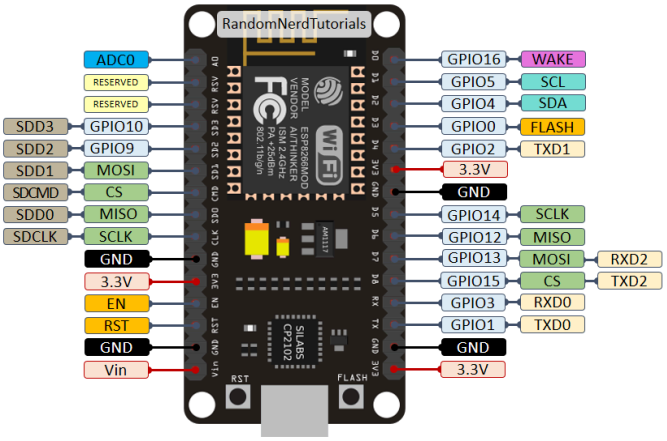

Figure. 3.2.1.1 ESP32 8266

c) **Voltage Divider Circuit:** Converts sensor resistance changes into readable voltage values.

d) **Power Source:** A rechargeable lithium-ion battery supplies power to the system for portability.

3.2.2 Data Transmission

The system uses an ESP module to detect finger movements and convert them into control signals. These signals are transmitted wirelessly via MQTT, enabling seamless and real-time communication between the master glove and the robotic arm.

- The ESP module reads sensor values corresponding to finger movements and maps them to respective angular positions.
- Using Wi-Fi connectivity, the ESP transmits the processed data wirelessly via the MQTT protocol.
- The MQTT broker facilitates efficient, real-time data transfer between the master glove and the slave robotic arm, ensuring low-latency communication and reliable remote operation.

3.2.3 Data Analysis

The real-time data transmitted via MQTT can be visualized and analyzed to understand hand motion and control accuracy. This enables gesture-based control for teleoperation, assistive devices, and rehabilitation, making the system reliable and scalable for various applications in robotics and automation.

- The real-time data transmitted via MQTT can be visualized and analyzed using custom dashboards or local applications, offering insights into hand motion and control precision.
- The captured data enables gesture-based applications and supports responsive robotic control in teleoperation scenarios.
- With further processing, gestures can be classified for applications such as remote robotic manipulation, assistive devices, and rehabilitation support.

This approach ensures a reliable and scalable system for real-time motion tracking, suitable for integration into domains like robotic control, assistive technology, and industrial automation.

3.3 DESIGN OF SLAVE (ROBOTIC ARM)

3.3.1 Hardware Components

The slave unit consists of a 3D-printed robotic arm modeled after the InMoov prosthetic hand. The arm is designed to replicate human-like movements and grip control using servo motors and real-time feedback.

a) **MG90S Servo Motors:** Provide high torque and precise control for finger movements.

Figure. 3.3.1.1 Servo Motor

b) **Raspberry Pi 5:** Handles data processing, motor control, and communication with the master unit.

c) **Power Supply:** A dedicated power source ensures stable operation of the servos and controllers.

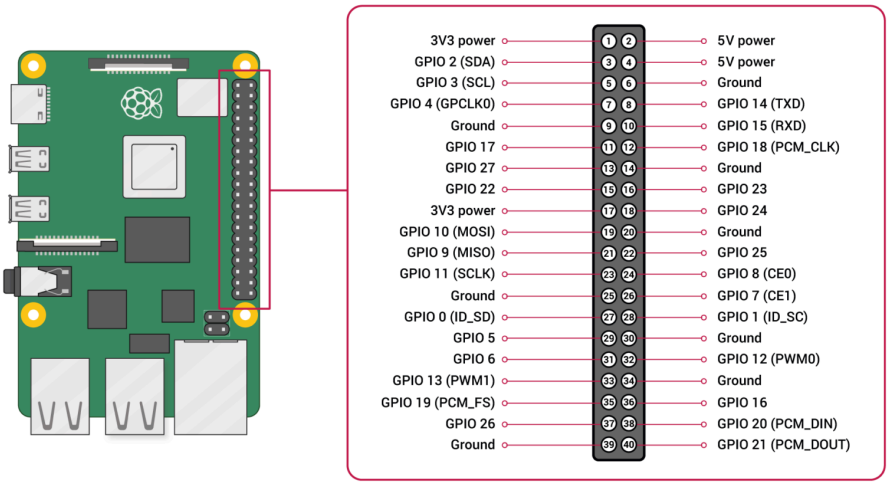

Figure. 3.3.1.2 Raspberry Pi 5

3.3.2 Control Strategy

The control strategy uses a PID controller to fine-tune finger positions and grip force based on real-time feedback. Sensor data from the cloud is processed by the Raspberry Pi to generate accurate motor commands for responsive control.

- **Proportional-Integral-Derivative (PID) Control:** Used to adjust the finger position and grip force based on real-time feedback.
- **Real-Time Processing:** The Raspberry Pi receives sensor data from the cloud and translates it into motor commands.

3.3.3 Error Correction

A Python-based PID controller adjusts the PID constants to reduce overshoot and stabilize grip force. This adaptive control minimizes slippage and damage, allowing the robotic arm to handle various objects with human-like precision across multiple domains.

- PID constants (P, I, D) are adjusted using a Python-based PID controller to minimize overshoot and ensure stable grip force.
- Adaptive control minimizes object slippage and prevents damage during handling.

This approach ensures that the robotic arm can handle objects of different sizes, shapes, and weights with human-like precision, making it suitable for applications in manufacturing, healthcare, and service robotics.

3.4 OBJECT DETECTION AND GRIP FORCE CALCULATION

This module integrates object detection using a progression of YOLO models with a calibrated grip force calculation for a robotic arm. In the early stages of the project, we experimented with YOLOv3—a pre-trained model known for its robustness but limited by lower accuracy and slower inference speeds. Based on early tests, we transitioned to YOLOv8 for its enhanced real-time detection performance and higher precision.

3.4.1 Key Libraries Used

To implement computer vision-based object detection and grip control, several key libraries were integrated. OpenCV handled real-time video capture, image preprocessing, and drawing bounding boxes for object visualization. NumPy (np) supported efficient numerical operations, especially for managing image arrays and integrating sensor data for precise control. Roboflow was used to label, annotate, and export datasets, significantly simplifying the model training pipeline for object detection tasks.

3.4.2 Model Selection and Preparation

The project initially used YOLOv3 for object detection, but it lacked the accuracy and speed required for real-time robotic control. To overcome these limitations, YOLOv8 was adopted, offering enhanced precision, faster processing, and greater efficiency. A custom dataset of 15 manually labeled images was created using Roboflow, capturing various object orientations and robotic arm features. This enabled YOLOv8 to reliably detect both the robotic arm and surrounding objects, forming a strong base for visual-based interaction and precise grip control.

3.5 GRIP DETECTION VIA COMPUTER VISION

3.5.1 Robotic Arm Detection

The system uses a pre-trained object detection model to identify the robotic arm in real-time from the video feed. By analyzing each frame, the software locates the arm based on its unique visual features. This detection is critical for initiating the grip mechanism and ensuring the arm interacts only with intended objects.

3.5.2 Interaction Verification

Once the robotic arm is detected, the system performs interaction verification by comparing the arm's contour with the bounding box of the target object. If an overlap is found, it confirms that physical contact has occurred between the arm and the object. This verification step ensures that the grip is applied only when the arm is properly aligned.

3.5.3 Controlled Grip Application

After verifying contact, the system applies a predefined static grip force to the object. This force is carefully calibrated for different object types to prevent damage or slippage. By automating this step, the system ensures consistent and safe handling across various scenarios, maintaining precision in delicate operations.

3.6 ERROR CORRECTION AND DATASET UPDATES

3.6.1 Using Pose Estimation for Checking Grip Stability

To improve the system's robustness, pose estimation is integrated to track the position of the robotic fingers and the object during manipulation. Advanced models like MediaPipe Hands or OpenPose are used to detect finger joint positions and ensure they align properly with the object's surface. This helps maintain grip stability even during motion.

3.6.2 Approach to Use Pose Estimation

The grip verification process begins with detecting the object’s position using object detection models such as YOLOv8 or Faster R-CNN. The system tracks the object’s bounding box across multiple frames to detect any shifting or slippage. Simultaneously, the robotic arm’s fingers are monitored using pose estimation or, when visibility is limited, via OpenCV contour detection. This dual detection strategy ensures precise grip monitoring.

3.6.3 Check Stability and Update Dataset

If the system detects instability, such as unexpected shifts in the object’s centroid or excessive pressure from the robotic fingers, it dynamically adjusts the grip force. This real-time feedback loop not only improves grip control but also helps update the model by logging interaction data. Over time, the dataset is refined with these inputs, enabling incremental improvements through transfer learning. This approach ensures continuous optimization of the robotic arm’s performance in adapting to new objects and conditions.

CHAPTER 4

TESTING AND EVALUATION

The complete testing environment comprises an integrated system featuring multiple hardware and software components working in coordination. The master controller (ESP32) receives tactile input from force-sensitive resistors , while the slave unit (a robotic arm controlled by Raspberry Pi 5) manages real-time object detection and manipulation.

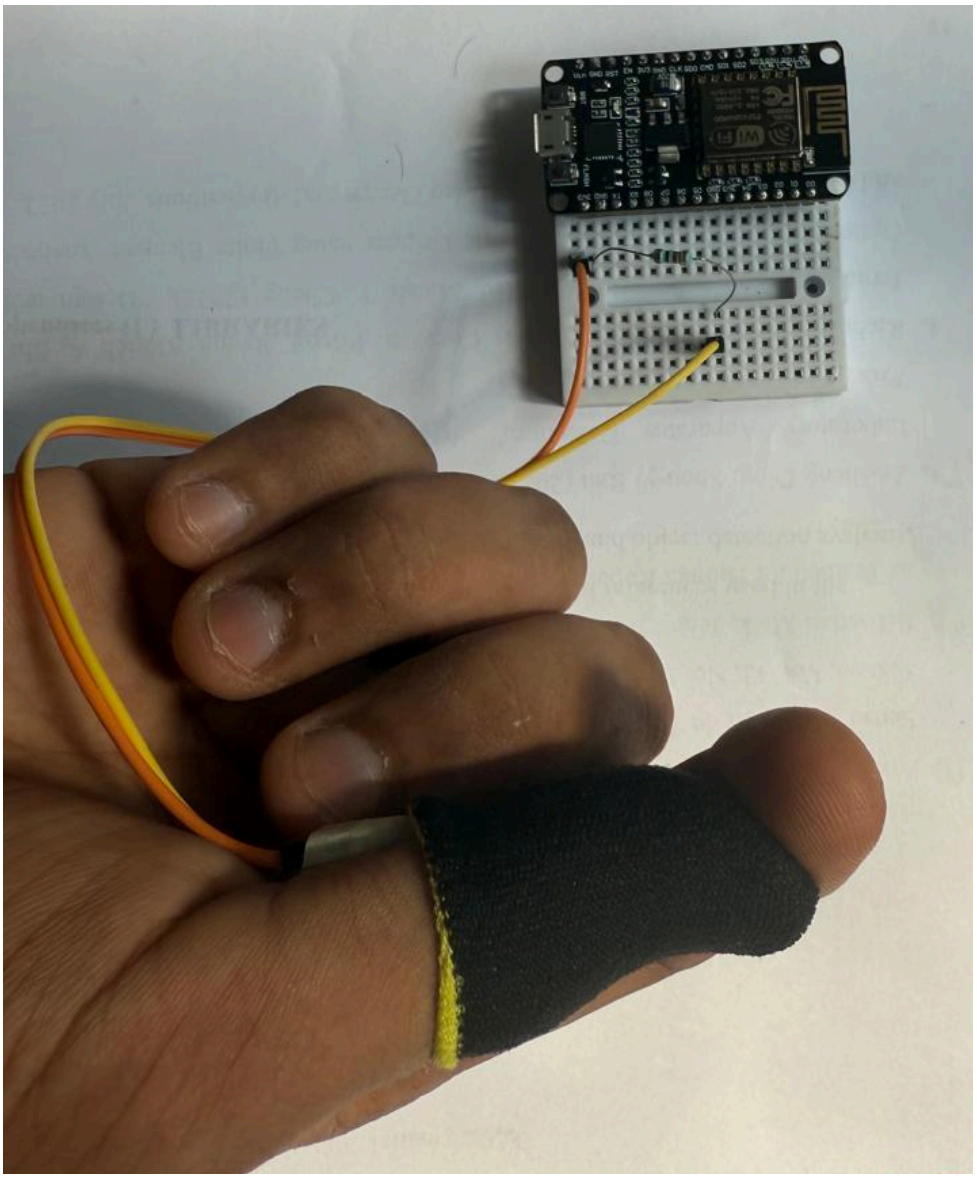

Figure 4.1 Glove / Master

The ESP32 microcontroller efficiently interprets analog input from the FSR sensors using threshold-based logic. When force readings surpass a set threshold of 150, the ESP32 transmits a binary signal “1” via MQTT; otherwise, it sends “0”. This approach simplifies the interaction state to binary logic, allowing fast and reliable communication with the slave system (as seen in Image 1, which shows the ESP32 wiring diagram and signal flow).

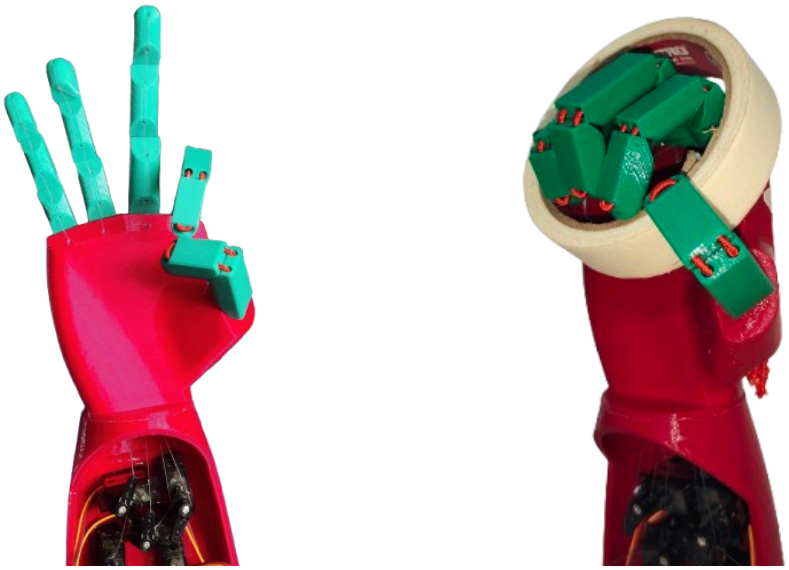

Figure 4.2 Robotic arm / Slave

The Raspberry Pi handles the motion control system, visual processing, and grip force classification. Upon receiving the interaction signal from the ESP32, the Pi initiates object detection and maps each identified item to a corresponding grip force level between 1 and 4. The hardware communication interfaces and controller board connections are illustrated in figure 4.2, highlighting the system’s reliability in managing real-time motor actions.

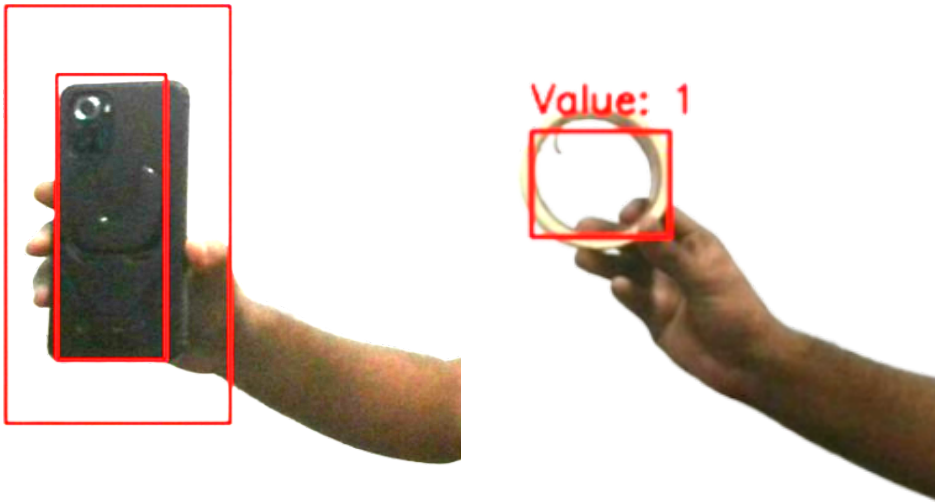

Figure 4.3 Object Identification with Grip Allocation

Pre-Interaction Phase

Before any physical contact, the system's vision module detects and classifies objects within the workspace. As demonstrated in figure 4.3, the system overlays red bounding boxes on detected objects, with a label indicating "object=1, interaction=0." This confirms successful object identification and confirms that no interaction has occurred yet.

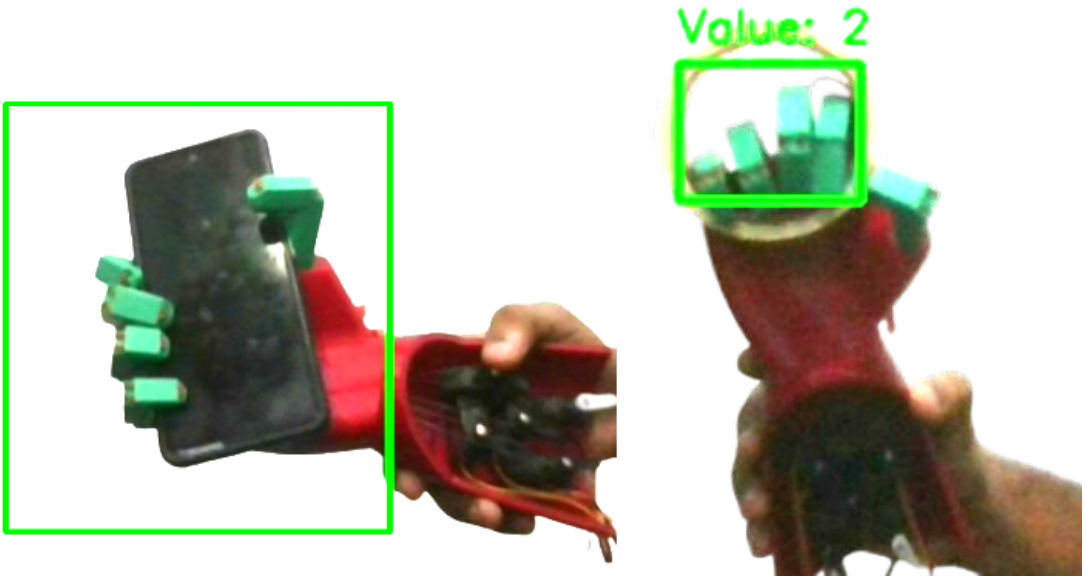

Figure 4.4 a) Robotic Arm holding Mobile Phone
b) Robotic Arm holding cello tape

Active Interaction Phase

Once the robotic arm makes contact (when FSR value equals 1), the system updates the bounding box color to green and overlays the applied grip force level, as shown in figure 4.4. This transition marks the start of the interaction state and verifies both physical contact and the object's grip force classification, even when multiple items are within the frame.

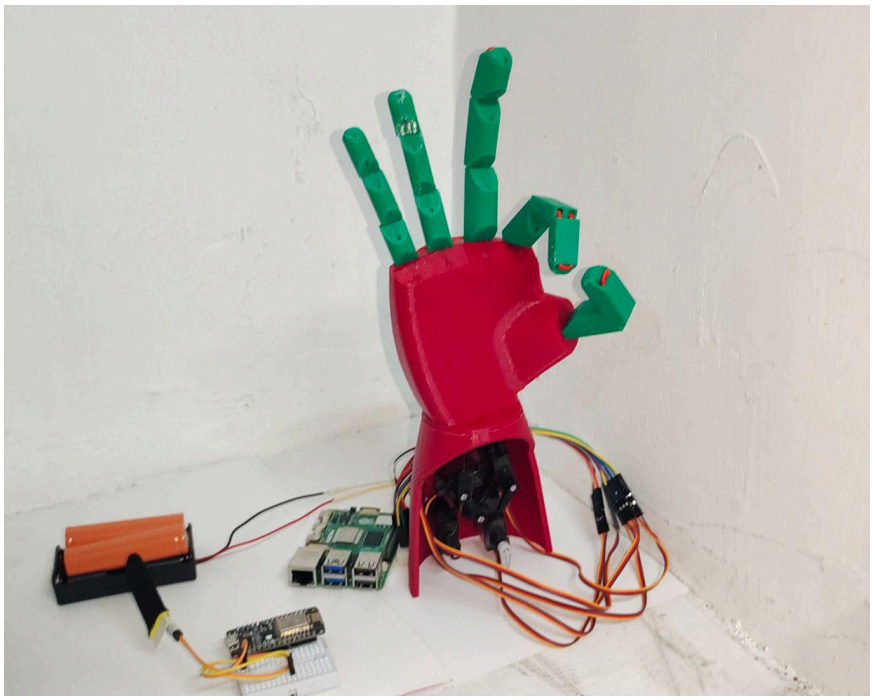

Figure 4.5 Complete System Integration

The comprehensive experimental setup, presented in figure 4.5, displays the synchronized arrangement of the robotic arm, ESP32, FSR glove, Raspberry Pi, and the camera module. All subsystems are mounted on a unified testing platform. Data flows from the FSR-equipped glove to the ESP32, which relays interaction states to the Raspberry Pi. The Raspberry Pi not only controls the robotic arm through PID algorithms but also utilizes YOLOv8-based object detection to determine object class and assign an appropriate grip force (1–4 scale). The system's performance is optimized within the physical limits of the arm's degrees of freedom and the vision system's sensitivity to ambient lighting.

Performance Metrics

- **Grip Success Rate:** High accuracy in gripping a variety of objects using the four-level classification model.
- **Object Interaction Detection:** Consistent and precise tracking of which object is actively manipulated.
- **Adaptation Capability:** Effective grip force modulation based on object type and visual feedback.
- **System Limitations:** Sensitivity to lighting variations and physical constraints of the robotic arm's joint range affect performance in edge cases.

These evaluations confirm the effectiveness of integrating tactile threshold detection with visual classification for real-time adaptive grip control. Despite some limitations, the system exhibits robust, coordinated behavior suitable for advanced telechiric robotic applications.

CHAPTER 5

CONCLUSION & FUTURE SCOPE

This project harnesses cutting-edge technologies such as machine vision, real-time data processing, and dynamic control mechanisms to ensure highly precise handling of objects, accommodating varying sizes and fragilities. The integration of PID control algorithms and the implementation of YOLOv3 for object detection have significantly enhanced the system's efficiency and reliability, positioning it as a powerful solution for industrial applications.

Moreover, this project paves the way for exciting avenues of future research and development. One potential direction is the incorporation of advanced reinforcement learning algorithms, which could further improve grip control precision and reduce error rates. Expanding the system’s capabilities to manage more complex tasks—such as handling delicate medical instruments or performing intricate surgical procedures—would greatly increase its versatility and applicability. Future work may also focus on energy-efficient design and reducing latency in real-time processing to optimize the system for broader use across various industries.

REFERENCE

[1] Andrew Belford, Mohsen, "Using Miniaturized Strain Sensor to Provide Sense of Touch in Humanoid Robot," *Micro- and Nano Electromechanical System*, Vol. 42, No. 1, pp. 421-425, 2022.

[2] Sebastian Mick, Jenny Benoius, "3D Printed Human-Like Robotic Arm as Testbed for Human Robot Control Strategies," *Front Neurorobot*, Vol. 27, pp. 81–94.2022

[3] Ze-sheng Ding, Shou-yi Ran, "A New Benchmark Dataset for Chemical Laboratory Apparatus Detection," *Artificial Intelligence and Big Data Processing*, pp. 1231-1236.2022

[4] Richard Josiah C. Tan Ai, Marcus Corso S. Pilapil, Ryann Aldrich A. Shi, Jeruel Lawrence D. Badugas, Sted Micah T. Cheng , "Design and Optimization of an Adaptive Robotic Gripper using Finite Element Analysis and Generative Design," *Computer-Aided Design and Applications*, 2022.

[5] Sainul Islam Ansary, Sankha Deb, Alok Kanti Deb, "Design and Development of an Adaptive Robotic Gripper," *Journal of Intelligent & Robotic Systems*, 2023.

[6] Muhammad Aqib, Abid Imran, Khurram Khan, Muhammad Arsalan, et al., "Design and Implementation of Shape-Adaptive and Multifunctional Robotic Gripper," *Journal of Field Robotics*, October 2023.

[7] Jaime Hernandez, Md. Samiul Haque Sunny, Javier Sanjuan, Mohammad Rahman, "Current Designs of Robotic Arm Grippers: A Comprehensive Systematic Review," *Robotics*, January 2023.

Appendix 1

LIBRARIES

The following libraries were utilized within the program code for the MQTT communication and computer vision system:

paho.mqtt.client : This Python library enables MQTT protocol-based communication, allowing seamless interaction between the ESP8266 sensor node and the computer-based monitoring system.
(https://pypi.org/project/paho-mqtt)

gpiozero.AngularServo : A class from the gpiozero library to control servo motors by setting angle values; simplifies GPIO usage.
(https://gpiozero.readthedocs.io/en/stable/api_output.html#angularservo)

Time : Provides time-related functionalities including delays, timestamps, and FPS (frames per second) calculations.
(https://docs.python.org/3/library/time.html)

cv2 (OpenCV) : OpenCV is used for real-time video processing, object detection using YOLO, color filtering (HSV), and drawing bounding boxes.
(https://yolov8.org/yolov8-documentation/)

numpy : Supports efficient numerical and matrix operations, essential for processing object detection data.
(https://numpy.org/devdocs/)