ASHESI UNIVERSITY

IMPROVED TELEOPERATION OF DUAL ROBOTIC MANIPULATORS USING VISION BASED CONTROL

CAPSTONE PROJECT

B.Sc. Computer Engineering

Opanin Akuffo

2021

ASHESI UNIVERSITY

IMPROVED TELEOPERATION OF DUAL ROBOTIC MANIPULATORS USING VISION BASED CONTROL

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Engineering.

Opanin Akuffo

2021

DECLARATION

I hereby declare that this capstone is the result of my original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

Candidate's Name:

Opanin Kwadwo Akuffo..

Date:

3 October 2020...

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University College.

Supervisor's Signature:

....

Supervisor's Name:

Dr. Stephen K Armah...

Date:

14/05/2021

....

Acknowledgements

To my supervisor, Dr. Stephen K. Armah whose encouragement and academic advice helped me undertake this project.

Abstract

The advancement of manual control of dual robotic manipulators has considerably lagged behind the progress of the single robotic manipulator. The widespread solutions for dual robotic manipulators consist mainly of joysticks and wearable technology, which are often expensive and complex to operate or uncomfortable to use. In order to bridge the divide between single and dual manipulators and provide alternatives to currently existing solutions, this study proposes an improved way of teleoperating dual robotic manipulators through vision-based control. By leveraging the Leap Motion Controller as the visual input sensor, the FRDM KL25Z board as the microcontroller unit, and Bluetooth as the transmission media, a simple real-time solution for visually controlling two 4-DOF robotic manipulators was designed, prototyped, and tested. Experimental results showed that the proposed solution is accurate, relatively cheaper, easy to use and understand, and more than capable of competing with the currently existing solutions.

Table of Content

List of Tables

List of Figures

List of Abbreviations

Abbreviations	Meaning
3D	Three-Dimensional
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
DOF	Degrees of Freedom
FRDM	Freedom
GPIO	General Purpose Input/Output
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HCI	Human Computer Interaction
I2C	Inter-Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IR	Infrared
ISO	International Organization for Standardization
LED	Light Emitting Diode
MCU	Microcontroller Unit
PWM	Pulse Width Modulated
RC	Remote Controlled
SDK	Software Development Kit
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
VR	Virtual Reality
WIFI	Wireless Fidelity

Chapter 1: Introduction

This chapter introduces the study by discussing the background, problem definition, objectives, proposed solution, and scope of this project.

1.1 Background

The rapid advancement in technology is paving the way for robust yet accurate and straightforward solutions in robotics. In the past, the go-to solution for manual robotic arm control was the joystick or a box with many complicated knobs and buttons. These controllers, such as the Morolipi V.2, are time-consuming to use, require expert operators, and the study of complex reference manuals before desired control could be achieved [1]. Presently, with the rise in bleeding-edge technologies, new forms of HCI and gesture control devices have made their way into mainstream consumer markets. Examples include wearable controllers such as Thalmic Labs Myo Armband and optical-based controllers such as Microsoft's Kinect and Ultraleap's Leap Motion controller. By leveraging these new HCI devices and technologies, some progress has been made in the field of robotic control. For example, G. Du et al., Pititeeraphab et al., and Kristoff et al. have developed interfaces that allow for control of a single robot arm by tracking an operator's arm movements using optical sensors like the Kinect [2] and the Leap Motion controller [3] or through electromyographic sensors like the MYO Armband [4]. The strides made in applying novel technologies towards controlling a single robotic arm have been commendable; however, the same is not the case for controlling dual robotic arms.

Since the deployment of the first robotic arm by George Devol in 1962, the integration of robotic arms within industries such as manufacturing and food preparation has increased exponentially. Aside from reducing production and operation costs, robot integration provides solutions to complex, repetitive tasks with greater accuracy and precision than humans. Most mainstream robotic arm integrations are executed using automatic control. However, with the rise of robot integration in exploration (deep sea and space), medicine, military, and nuclear research, the need for accurate yet straightforward manual controllers has increased. Throughout time, teleoperation or remote control, as it is commonly known, has been represented through many diverse shapes and forms, from the age-old joystick controllers to the more advanced electromyographic controllers. Teleoperation is a significant area within robotics and paves the way towards endless possibilities. Despite the promise and considerable advancement within the field of teleoperation, it still has its shortcomings.

1.2 Problem Definition

Manual control of dual robotic arms is vital in nuclear decommissioning, deep sea and space exploration, remote surgeries, and bomb disposal. Currently, most teleoperated dual robotic manipulators are controlled either using very complex controllers or wearable technology. Wearable controllers such as sensor sleeves, data gloves, and exoskeletons hinder the operator's movements and comfortability, which sometimes leads to unintended external control [5]. It also costs more to develop for dual-arm control, using twice as much material and sensors needed for single-arm control. Additionally, hand, the more mainstream yet complex dual robotic arm remote controllers such as that of the ASTACO-SoRa Dual arm Robot [6] require expert operators. Even with that, some manipulator manoeuvres are difficult and time-consuming to execute. Picture an African surgeon who is fresh out of medical school and has a burning desire for robotics and its applications in surgeries; this surgeon would be disappointed to find out that (s)he may never have the chance to execute a remote surgery because the only commercially available solution is a $2 million robot (da Vinci Surgical System). Not only is the robot very expensive, but it requires extensive training to use due to its very complex remote console and use of proprietary software. There ought to be cheaper, more innovative, and more straightforward solutions that all can access. Many creative and simple solutions exist for single-arm robotic control, yet dual-arm robotic systems seem stuck behind a cloud of complexity. In this light, it is worth investigating other ways of naturally and seamlessly controlling dual-arm robotic systems.

1.3 Objectives

The main objective of this project is to improve the teleoperation of dual 4-DOF robotic manipulators through contactless visual-based control, requiring little to no operator skill or training before use.

Specific objectives:

- Track both hand and finger movements of the operator.
- Convert tracked position data into position data which the robotic manipulator can process.
- Perform signal processing and relay data to the MCU in real-time.
- Use a microcontroller to drive servo motors on both robot manipulators accordingly.
- Render a virtual 3D representation of the operator's hands in real-time to show motion.

1.4 Proposed Solution

This project proposes a solution that leverages the Leap Motion controller to provide contactless, real-time, and vision-based teleoperation of two 4-DOF robot manipulators. This solution would be of high importance in the field of teleoperations and would provide increased awareness of the depth and potential that vision-based control holds. Compared to similar optical sensors such as the Kinect or Orbbec Persee, which are more expensive and used to track the entire body, the Leap Motion controller was the natural choice for the optical sensor in this project because of its specific focus and accuracy of arm and finger tracking.

1.5 Scope

This project focuses on how effective control of dual robotic arms can be achieved using contactless visual-based solutions. It also presents a case for IoT through remote sensing and telecommunication. The project also investigates different mapping techniques currently being used by visual-based controllers for single-arm robotic manipulators and their effectiveness in dual robotic arm control. However, the scope can be extended to include collision detection and avoidance and improved operator telepresence by providing complete VR immersion through a VR headset and attaching extra 3D depth sensors on the manipulators.

Chapter 2: Literature Review and Related Work

This chapter explores the works other researchers have done relating to the visual control of single and dual robotic manipulators and the challenges and advances made by the researchers in their work. This chapter aims to gain insight into what ideas can be leveraged for this report's advancement and shows how it differs from other papers within the same field of study.

2.1 A Markerless Human-Robot Interface Using Particle Filter and Kalman Filter for Dual Robots

In this journal article [7], the authors document the teleoperation of the end effector of dual robotic manipulators based on markerless tracking technology. Their method involved using a Leap Motion Controller alongside a Kalman and Particle Filter to accurately screw a bolt into a nut by mimicking the hand movements of a novice operator. The authors used the filters to predict the position of the operator's hand and reduce errors and noise originating from the sensor and other devices. Their findings were further bolstered by comparing their results to other markerless tracking methods performing a similar task [2][8]. The article is a well-researched paper with numerous defined and understandable graphs and detailed results and explanations. This source shows that the control of dual robotic manipulators using the Leap Motion controller is possible.

2.2 Analysis of the Accuracy and Robustness of the Leap Motion Controller

This journal article [9] reports on the accuracy, robustness, and repeatability of the Leap Motion controller using a single robot manipulator and a pen as the reference point for accuracy. The authors ran the tests for repeatability and accuracy following ISO 9283, the standard for performance characteristics tests of industrial robot manipulators. They performed a thorough evaluation with 5000 measurements taken for each test case. Despite the Leap

Motion controller not achieving its quoted theoretical accuracy of 0.01 mm, it averaged 0.7 mm under natural conditions. That is more accurate than other object tracking controllers within its price range. The paper [9] concludes with the authors citing other use cases for the Leap Motion controller. Being the first published study on the Leap Motion controller, this paper [9] serves as the primary reference concerning the accuracy of the Leap Motion controller for all other academic work utilizing the controller. This source shows that the Leap Motion controller is worthy of use in teleoperating environments requiring intricate movements.

2.3 Improved Teleoperation of an Industrial Robot Arm System Using Leap Motion and MYO Armband

In this paper, the authors [10] propose a better way to teleoperate an industrial robot arm in real-time using the Leap Motion controller in conjunction with the MYO Armband. The Leap Motion controller is used to track and detect the operator's arm movements in real-time, while the MYO Armband is used to detect abnormal electric potential signals in the operator's forearm caused by unexpected arm movements (physiological tremors). The authors [10] improve teleoperation by introducing a new signal preprocessing technique that works simultaneously with the Leap Motion controller's tracking to temporarily pause the robot's end-effector when a tremor in the operator's forearm occurs. This source is a genuinely innovative paper that harnesses the power of two standalone teleoperation technologies into one well-rounded solution.

2.4 Intuitive and Adaptive Robotic Arm Manipulation using the Leap Motion Controller

In this paper, the authors [11] propose a new intuitive and adaptive teleoperation approach focused on Ambient Assisted Living and essential activities of daily living to satisfy and ease the needs of elderly and disabled individuals. This novel method involved the manipulation of a 6-DOF Jaco robotic arm using the Leap Motion controller. The algorithm developed by the authors [11] reduces random errors derived from continuous hand tracking using the Leap Motion controller by constantly adapting to the operator's hand tremor. This source is a well-written paper which depicts the various way in which teleoperation can be of use aside from the industrial setting. It also shows the breadth to which teleoperation is helpful in multiple industries and the robustness of the Leap Motion controller for visual-based control.

2.5 Development of a Neural Network-Based Control System for the DLR-HIT II Robot Hand Using Leap Motion

In this journal article, the authors [12] propose a new way of teleoperating a robot hand using a neural network-based adaptive controller and the Leap Motion sensor. The paper [12] focuses more on the intrinsic control of the fingers of a humanoid arm and uses a DLR-HIT II robot hand as a result. Additionally, inverse kinematics is used to transform the Cartesian positioning data from the Leap Motion sensor into joint angles and fed directly to the robotic hand via User Datagram Protocol. The authors' [12] stable neural network-based controller, as proven by the Lyapunov stability theorem, is used to compensate for dynamic uncertainties from contactless tracking of the hand. This paper is fascinating because it includes machine learning to improve the teleoperation process. It was a challenge to understand the paper as most of the concepts used were high-level, and it seems the paper skewed towards a more advanced audience. Nonetheless, it is a well-detailed report that provides perspective on the limitless possibilities and depth when implementing visual-based teleoperating techniques.

Chapter 3: Design

This chapter discusses the design, planning, and methodology used in creating the proposed solution. The user and system requirements, communication interface selection, system design architectures, and design components are expanded and elaborated.

3.1 Requirements Specifications

This study has two main requirements, user requirements and system requirements. Adequately evaluating both requirements involves carrying out extensive research using numerous peer-reviewed papers, some of which are highlighted in the literature review. In addition to papers, requirements were also determined using interviews with various engineers (electrical, controls, and communication), individuals interested in robotics, and individuals with any experience involving remote control of a physical device such as an RC vehicle.

3.1.1 User Requirements

Throughout this study, the user refers to the operator controlling the dual robotic manipulators with his/her hands. From research conducted and interviews, the user requirements of the overall system must be:

- Easy to understand
- Easy to use
- Safe
- Comfortable
- Accurate
- Responsive

3.1.2 System Requirements

Two main subsystems or interfaces must be developed to create a single system capable of interfacing and enabling the visual control of dual robotic manipulators: the operator's subsystem and the robotic manipulators' subsystem. Both subsystems contain hardware and software components determined by a requirements specification. The system requirements are as follows:

- The system must be capable of tracking both the hand and finger movements
- The system must be low-cost
- The system must accurately track the user’s hand and finger movements
- The system must perform some signal processing and manipulation
- The system must be secure
- The system must be responsive
- The system must be able to transmit the operator's tracked data wirelessly to control the robotic manipulators.
- The system must be capable of converting the operator's tracked data into positional data which can be used to control the robotic manipulators
- The system must be capable of operating regardless of which operator's hand(s) are in its field of view (left, right, or both)
- The system must be able to control both manipulators using a single MCU

3.2 Communication Interface Selection

Table 3.1 shows a Pugh Matrix used to select the mode of communication to be used between the operator’s subsystem and the robotic manipulators subsystem:

- Concept A: Communication interface using Bluetooth via the HC-05 Bluetooth module.
- Concept B: Communication interface using Wi-Fi via the ESP8266 module.
- Concept C: Communication interface using GPRS via the SIM800L GSM GPRS module.

Table 3.1: Pugh Matrix for Communication Interface Selection

Criteria	A (Bluetooth)	B (WIFI)	C (GPRS)
Ease of Setup	1	-1	-1
Cost	1	0	-1
Reliability	1	-1	0
Power Consumption	1	0	-1
Range	-1	0	1
Transmission Speed	0	1	-1
Net Score	3	-1	-3
Rank	1	2	3

On the Pugh Matrix, a score of 1 indicates that concept is favourable under the criteria, a 0 indicates neutrality, meaning the concept is neither favourable or unfavourable, and a -1 indicates the concept is unfavourable under the criteria. It is also imperative to note that the main aim of this study is to provide proof of feasibility on the effectiveness of visual-based control on dual robotic manipulators. Thus, the Pugh Matrix is skewed towards ease of setup and cost. In a production environment where the system ought to be versatile, a communication method with a long-range and high transmission speed would hold more weight than the other criteria.

3.3 System Design Architecture

In this report, various design architectures describe and detail the proposed solution. Three design architectures are to visually depict the proposed solution: a high-level diagram, an activity diagram, and a schematic design.

3.3.1 High-Level Design

The high-level diagram in Figure 3.1 shows the essential design components required to build the overall system and gives an overview of how the system works.

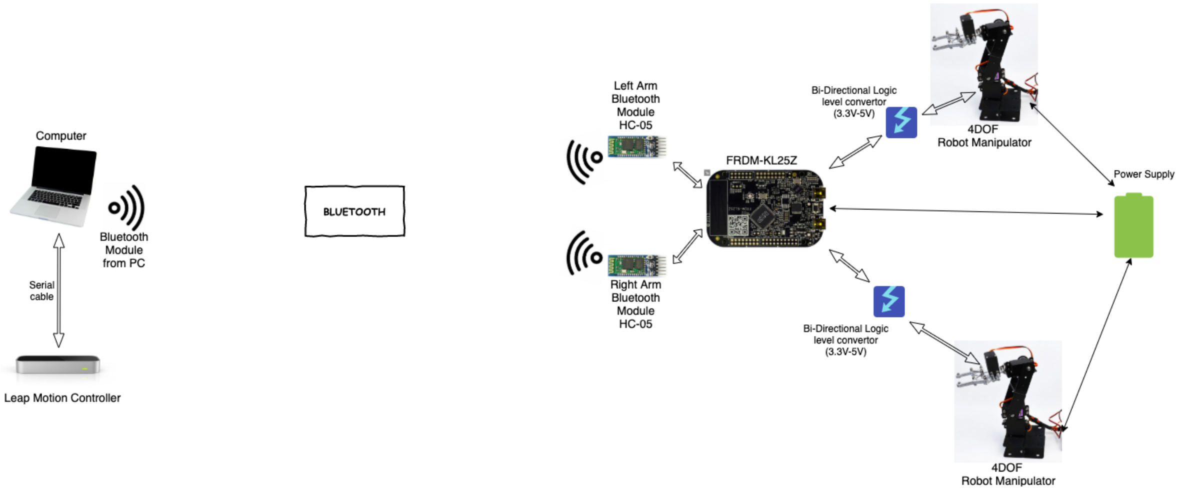

Figure 3.1: High-Level Design of the System

3.3.2 Schematic Diagram

The schematic diagram in Figure 3.2 shows the wiring and connections of the electrical components required to build the robotic manipulators' subsystem and how they relate to each other.

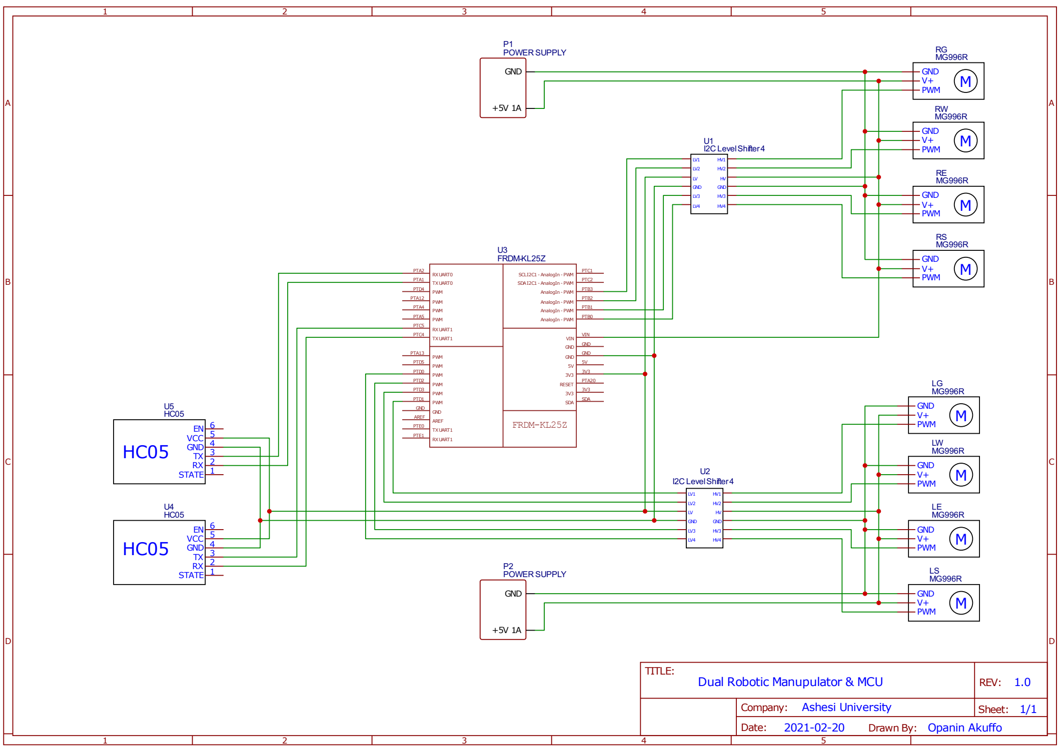

Figure 3.2: Schematic Diagram of the Robotic Manipulators Subsystem

3.3.3 Activity Diagram

The activity diagram in Figure 3.3 gives the persona journey of the operator and the steps needed for he/her to interact with the system. Additionally, the activity diagram shows the tasks running behind the scenes in both subsystems and how they interact and merge into forming a single system as a whole.

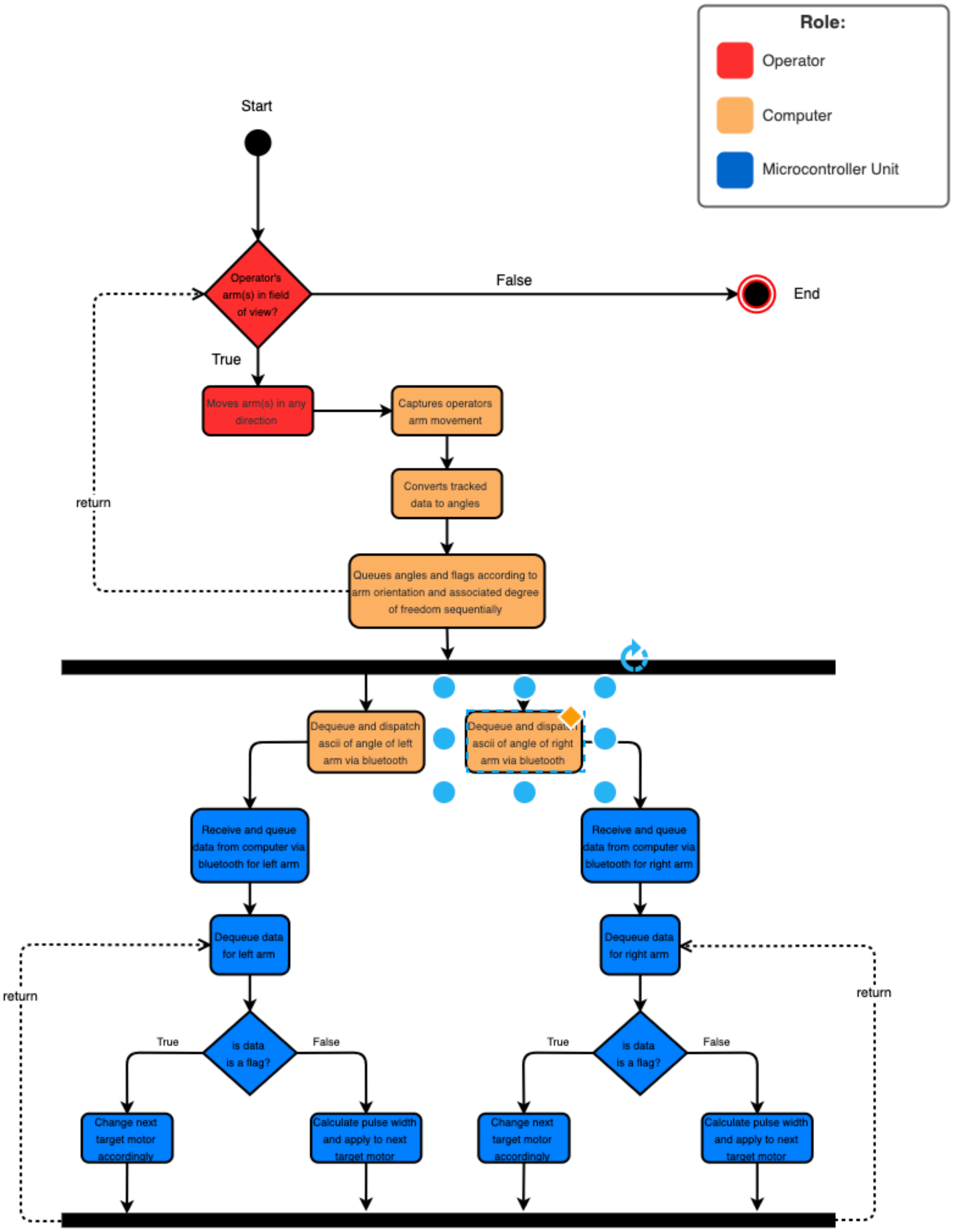

Figure 3.3: Activity Diagram of the System

3.4 Design Components

This section provides background information on each major hardware component used in designing the system. It also details the reasons for choosing each of these components.

3.4.1 Leap Motion Controller

Figure 3.4: Leap Motion Controller Exploded View

The Leap Motion Controller is a low-cost optical hand tracking device consisting of two monochromatic infrared stereo cameras and three infrared LEDs. With a field view of 150 degrees, the two IR cameras monitor whole hand movements, whiles the three IR LEDs monitor the joints on each hand within the device's field of view [Venna & Patel]. With the help of the user's central processing unit, the Leap Motion software can run complex mathematical algorithms, enabling it to track all ten fingers with sub-millimeter accuracy up to 0.7mm per axis at approximately 300 frames per second [9]. Though robust and very accurate, the Leap Motion does have its shortcomings. Since it uses IR light technology that does not shine through objects, the device cannot detect and track accurately when there is an overlap of the hands within its field of view. Regardless, the device works on the three major operating systems (Windows, Mac & Linux), and its API can be accessed using major programming languages such as Python, Java, C++, C#, JavaScript, Unreal, etc.

3.4.1.1 Mapping Leap Motion Positional Data to Real World 3D Coordinates

The Leap Motion Controller provides tracking data of both hands simultaneously through data objects called Frames. Each frame contains a list of objects which give information about each entity detected by the controller. This study uses only the **hand** object in each frame to gather information required for control. Inside the hand object, the following attributes are accessed and combined:

- is right: This indicates whether the hand in the frame is the right hand or not.
- confidence: This provides a confidence score on how accurate data within the frame is. This study uses only data with a confidence score above 80%.
- palm_velocity: This indicates the palm(s) velocity within the frame in the x, y, and z direction. In order to minimize haphazard and erroneous movements, this study uses only data where the hand in the frame was moving less than 7cm/s.
- grab_strength: This provides information as to the degree of how close or open the palm(s) is. This study uses this data to control the gripper of the manipulators.
- palm_position: This provides the displacement of the center of the palm(s) in the frame to the Leap Motion controller's origin in the x, y, and z direction. This study uses this data in the x, y, and z directions to control the wrist, elbow, and shoulder of the manipulators, respectively.

Though the leap motion is excellent at providing tracked positional data of a hand(s) within its field of view, the data it provides is relative to the center position of the motion controller.

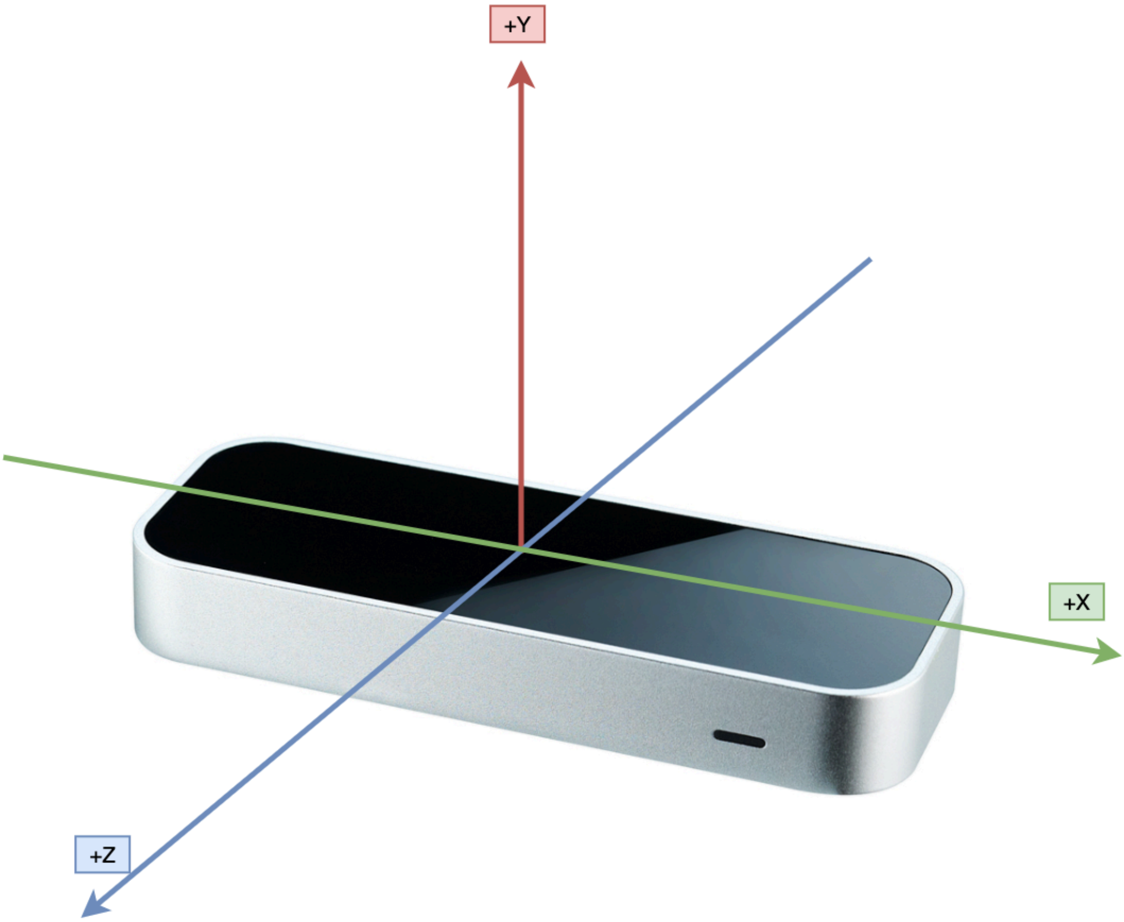

Figure 3.5: Leap Motion Controller Reference Axis

Thus, before any data from the motion controller can be used and understood by the MCU controlling the servo motors, it needs to be converted. To aid with the conversion, a linear equation is used to change the Leap Motion displacement data to real-world 3D coordinate angles between 0 to 180 degrees. The mapping equation used is shown in Eq.1.

$$ANGLE_{VALUE} = \frac{LEAP_{VALUE} - LEAP_{MIN}}{LEAP_{RANGE}} \times ANGLE_{RANGE} + ANGLE_{MIN} \quad (1)$$

$$LEAP_{RANGE} = LEAP_{MAX} - LEAP_{MIN} \quad (2)$$

$$ANGLE_{RANGE} = ANGLE_{MAX} - ANGLE_{MIN} \quad (3)$$

where:

$ANGLE_{MIN}$ and $ANGLE_{MAX}$ are 0 and 180 degrees, respectively

$LEAP_{MIN}$ and $LEAP_{MAX}$ are arbitrarily set in each direction during program initialization.

3.4.2 FRDM-KL25Z Development Board

Figure 3.6: FRDM-KL25Z Development Board

The Freedom KL25Z is a low-cost, low-power development platform built on the Arm Cortex-M0+ processor from the Kinetis MCU family. Prototyping and implementing designs can be done easily and quickly with the KL25Z board because of its multitude of features: a 48 MHz processor, 128 kB flash storage, numerous GPIO pins, PWM pins, I2C pins, timers, and a default flash programming interface. In this study, the FRDM-KL25Z board was chosen as the preferred MCU for controlling both robotic arms over the cheaper and more widely used ATmega328p because of the number of PWM pins available on the FRDMKL25Z board. The ATmega328p provides only 6 PWM outputs, and to control two 4-DOF robotic arms with one board, 8 PWM outputs would be needed. The FRDM board offers up to 10 PWM outputs, making it ideal for this study and further works.

3.4.3 Servo Motors

Figure 3.7: MG996R Servo Motor

The MG996R servo is a lightweight, low-cost, and high torque metal gear dual ball bearing servo with a maximum stalling torque of 10 kg/cm. This servo motor rotates approximately from 0 to 180 degrees and back when a PWM wave with a duty cycle between 1 – 2 milliseconds, and a frequency of 50 Hz is applied to its signal pin. In this study, eight servo motors were used in conjunction with two robotic arm chassis to mimic the range of motion of two human hands, each with four degrees of freedom: the shoulder, elbow, wrist, and palm, as shown in Figure 3.8.

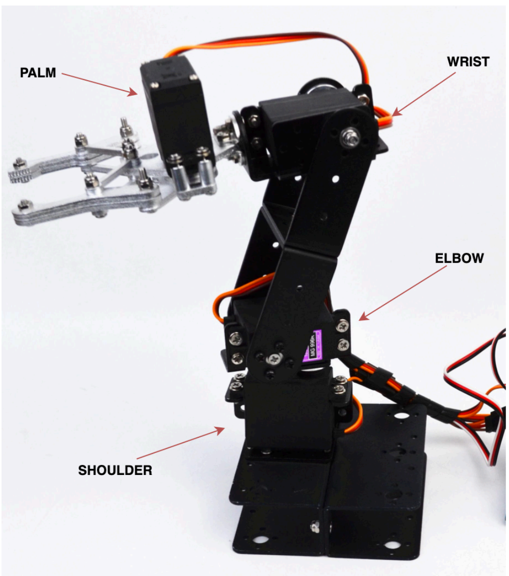

Figure 3.8: Human Hand VS Manipulator Degree of Freedom

3.4.4 Bluetooth Modules

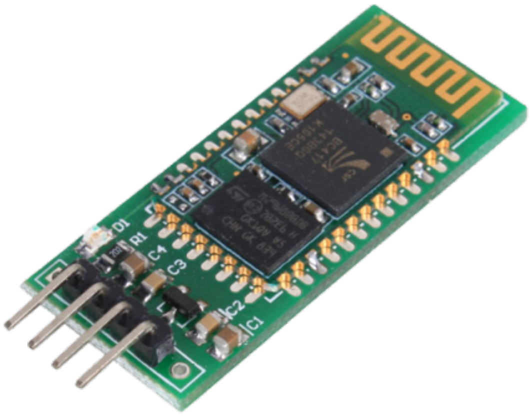

Figure 3.9: HC-05 Bluetooth Module

The HC-05 is a serial Bluetooth module used for full and half-duplex communication and data transfer. This module operates using the IEEE 802.15.1 protocol and offers data transfers at multiple baud rates (9600, 115200, etc.). The module can also operate in master, slave, or master/slave modes. The module offers Frequency Hopping Spread Spectrum as a layer of security and has a maximum range of 10 meters. This study uses two HC-05 Bluetooth modules to receive tracked positional data from the operator's computer via UART for control of both robotic manipulators.

3.4.5 Bi-Directional Logic Level Shifters

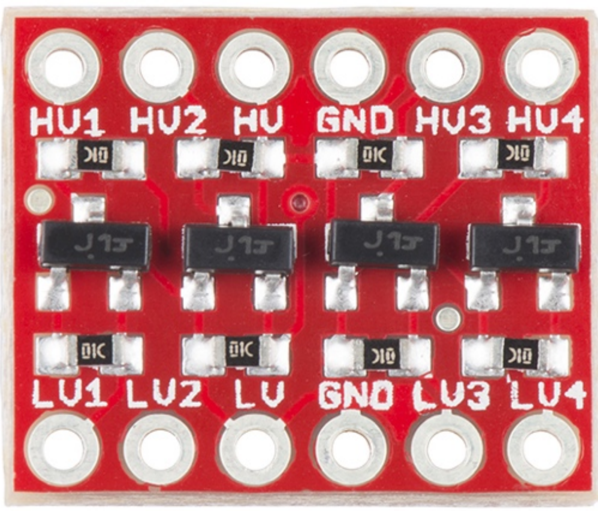

Figure 3.10: BSS138 4 Channel Level Shifter

The BSS138 Bi-Directional Logic Level converter is a low-cost electronic device capable of safely stepping down 5V to 3.3V and stepping up 3.3V to 5V. This device is easy to use and is inbuilt with four channels, making it capable of converting the voltage across four pins simultaneously. This study uses two BSS138 converters to step up 3.3V PWM outputs from the FRDM-KL25Z board to 5V required to operate the servo motors.

3.5 Bill of Materials

The Bill of Materials as shown in Table 3.2 provides detailed information about the procurement of all the electrical components used in designing the system.

Table 3.2: Bill of Materials

BOM Level	Part Number	Part Name	Description	Cost	Quantity	Unit	Procurement Type	Manufacturer	Supplier	Supplier Part/ID Number
0	FRDM-KL25Z	FRDM-KL25Z	KL1x, KL2x, mbed-Enabled Development Freedom Kinetis ARM® Cortex®-M0+ MCU 32-Bit Embedded Evaluation Board	$15.00	1	Each	OTS	NXP	Digi-Key	FRDM-KL25Z-ND
1	SNAM5300	4DOF RC Robot Arm	Metal materials, hard and durable	$30.00	2	Each	OTS	Small Hammer	Small Hammer	SNAM5300
2	MG996R	Servo Motor	Positional Rotation DC Motor Servomotor Series MG996R, RC (Hobby) 4.8 ~ 6VDC	$ 4.35	8	Each	OTS	Terasic Inc	Geek Electronics	GE0000082
3	HC-05	HC-05 Bluetooth Module	Bluetooth V2.0+EDR. 2.4GHz radio transceiver and baseband with master or slave config.	$ 2.66	2	Each	OTS	N/A	AliExpress	32806048234
4	BSS138	BSS138 4 Channel Logic Level Shifter	Converts 3.3V to 5V	$ 2.95	2	Each	OTS	SparkFun	SparkFun	BOB-12009
5	90-0002	Leap Motion Controller	Optical hand tracking module for hand tracking with a 150° × 120° field of view, 60-cm interactive zone, 27 distinct hand elements, and 120-Hz refresh rate.	$89.95	1	Each	OTS	UltraLeap	Adafruit	2106
6	N/A	Male to Male Jumper Wires	Multiple jumpers connected next to one another on a 2.54mm female header. Bundle of 40 cables	$ 1.56	1	Each	OTS	N/A	Oku Electronics	3369
7	N/A	Female to Female Jumper Wires	Multiple jumpers connected next to one another on a 2.54mm female header. Bundle of 40 cables	$ 1.56	1	Each	OTS	N/A	Oku Electronics	3365
8	N/A	12V Power Adapter	12V, 2A power adapter	$ 2.59	2	Each	OTS	N/A	Oku Electronics	1043
9	N/A	Bread Board Power Supply	The is a breadboard power supply module that provides dual 5 V and 3.3 V power rails and has a multi-purpose female USB socke	$ 1.73	2	Each	OTS	N/A	Oku Electronics	2646
10	N/A	Solderless Breadboard 840 Tie-Points	This breadboard consists of one Terminal Strip with 640 Tie-points and 2 Distribution Strips with 200 Tie-points.	$ 1.56	2	Each	OTS	N/A	Oku Electronics	2430

Chapter 4: Implementation

This chapter discusses the implementation of the hardware and software components of the system as a whole.

4.1 Hardware Setup

For building the circuitry of the robotic manipulators, Figure 3.2 details the schematic used. Two solderless breadboards and jumper cables were used to enable rapid prototyping. Additionally, for efficient power management, each robot arm was given its dedicated power supply. Each servo was powered using 5V - 1A adapters via a breadboard power supply. The Leap Motion Controller is connected to a computer via a USB serial cable on the operator's end. Both the operator's and the robotic manipulators' subsystems were connected wirelessly using Bluetooth. Figure 4.1 shows the hardware setup for the robotic manipulator on the left and that of the operator on the right.

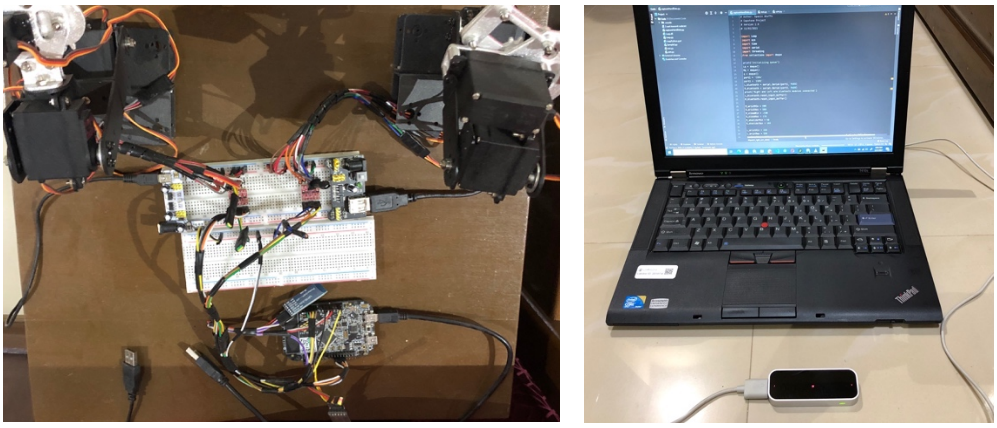

Figure 4.1: Hardware Component Setup

4.2 System Software Component

To create a single system capable of interfacing and remotely connecting the Leap Motion Controller, the FRDMKL25Z microcontroller, and the dual robotic manipulators, software for the two main subsystems have to be developed: the operator's subsystem software component and the robotic manipulators' subsystem software component.

4.2.1 Initialization & Protocols followed by Communication Interface

As indicated by the Pugh Matrix in Table 3.1, Bluetooth is used as the primary communication interface and data transfer mode between both subsystems. The Bluetooth transceiver in the operator's computer serves as the transmitter, and two HC-05 modules connected to the MCU serve as the receivers. Since two HC-05 modules are used, the communication interface for the whole system is made up of two separate communication lines: one for the left manipulator and the other for the right manipulator. The communication protocol observed by both communication interfaces is as follows: a baud rate of 9600, a payload size of 8 bits, and no parity bit.

4.2.2 Operator's Subsystem Software Component

Data is gathered, processed, and transmitted using Python programming language on the operator's end of the system. The Python API provided by the Leap Motion SDK is used to access the data generated by the Leap Motion controller through a Listener class and its associated functions. Other Python packages, modules, and collections are used as followed:

- Deque: A data structure from the Python collections library used to store processed data that is yet to be transmitted.
- Sys: Used to listen and track processes running indefinitely until the operator executes a keyboard interrupt.
- Serial: Used to open and close serial ports and connections on the computer and transmit and receive data serially via Bluetooth.
- Time: Used to add necessary delays before data transmission. This avoids flooding and loss of data during transmission.
- Threading: Used to execute the three main tasks on the operator's subsystem simultaneously. These tasks are:
    1. Listening for and processing data from the motion controller.

2. Transmitting data required to control the left robotic manipulator.
3. Transmitting data required to control the right robotic manipulator.

Since serial communication is being utilized and to ensure that the data transmitted is received accurately by each servo motor, eight flags are used. Each flag is queued right before its corresponding positional data is queued. The flags, numbered from 181 to 188, uniquely represent each degree of freedom on each manipulator. The purpose of these flags is to enable the MCU on the receiving end to correctly identify which servo motor the subsequent positional data belongs to. Additionally, because a maximum of 8 bits is transmitted at a go, all payload, flags, and corresponding angles are converted to their corresponding ASCII value before transmission. This encrypts the data being transmitted and serves as a layer of security in case of any external interception.

4.2.3 Robotic Manipulators Software Component

On the robotic manipulators' subsystem, data is received, processed, and sent to the appropriate servo motors by the MCU for actuation. Processing is done by converting the data from ASCII back to its corresponding integer value and run through some logic sequences. Since no operating system is installed on the MCU and instructions are run on bare metal, another linear equation similar to that in Eq. 1 is used to convert the angle value into a PWM value understood by the servo motor. This PWM value is applied to the appropriate servo motor and subsequently moves the manipulator towards the operator's desired position. The program on the MCU runs using a scheduler consisting of two functions: one to monitor tasks for the right robotic manipulator and the other for the left manipulator. Lastly, two finite state machines are used to handle flagging and inputting data to the servos, and interrupts are used to manage data reception to the two UART connections.

Chapter 5: Results and Discussion

This chapter discusses the various testing and experimentation undergone by the system as well as the accuracy and latency of the system.

5.1 System Testing

A simple control test was developed and carried out to test whether the system satisfies its requirements, objectives, and purpose. The test involved timing how fast different operators can execute a simple pick and place task and comparing their times. Regardless of their proficiency, each operator was giving a minute to free roam with the system and get a feel of how vision-based control works. The pick and place task required the operator to use the dual robotic manipulators to pick up two objects, positioned opposite each other, and place them in a common area. The task involved moving the manipulators in all their axis of motion, thereby evaluating the system's functional requirements. By using operators who had never been in contact with the system, the user requirements of the system could also be assessed. Each operator was given the same task to execute. Figure 5.1 shows the test setup, and Table 5.1 shows the test results of the 15 operators who participated.

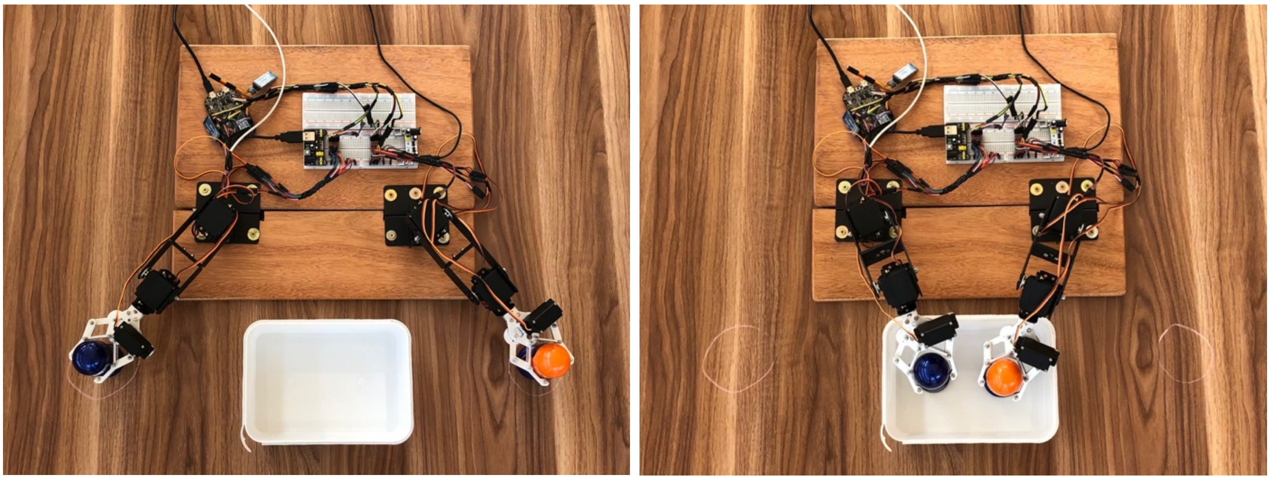

Figure 5.1: Pick and Place Test Setup

Table 5.1: Pick and Place Test Results

Operator Order	Duration (seconds)	Proficiency	Number of Tries	Completed Task
1	22	Expert	1	Yes
3	35	Intermediate	1	Yes
8	49	Beginner	1	Yes
4	52	Intermediate	2	Yes
14	72	Beginner	2	Yes
11	77	Beginner	2	Yes
12	82	Beginner	2	Yes
5	95	Beginner	3	Yes
7	102	Beginner	3	Yes
15	119	Beginner	3	Yes
2	124	Beginner	4	Yes
9	144	Beginner	4	Yes
13	137	Beginner	5	Yes
6	N/A	Beginner	> 5	No
10	N/A	Beginner	> 5	No

Explanation of evaluation criteria used in Table 5.1:

- Proficiency: This describes the operator's skill level before undergoing the test. An expert operator is a user who thoroughly understands how the system works and has used the system many times before the test. A beginner operator is any user who does not understand how the system works and has never used the system before the test. An intermediate operator is a user who has some understanding of how the system works and has used the system between 1-4 times before the test.
- Number of tries: This is the number of times the user took to complete the task. Each time an operator drops any object or gets stuck, the number of tries is incremented, and the operator restarts the task. If the operator does not complete the task within five attempts, they are marked as unable to complete.
- Duration: This is the total time, in seconds, the operator used to complete the task from the first try till the task is completed.

With an 87% test completion rate, it is safe to say that the system does indeed satisfy its requirements and objectives.

5.2 Accuracy

To evaluate the system's accuracy, position data (leap motion data and their associated angles) captured and transmitted from the operator's subsystem are compared to the actual positions (angles) moved by the robotic manipulators. The actual position of the robotic manipulators is generated via shoulder, wrist, and elbow servo motors whose motion is determined by the position of the operator's hand in the X, Y, and Z reference axis of the Leap Motion Controller [refer to Figure 4.2] respectively. With the aid of a digital protractor and a pre-set reference, the angle moved by the servo motor is measured. Measurements are taking every 5-degree interval as determined by the operator's software and range between 0 to 180 degrees. This process is repeated for the other five servo motors, and the data is plotted and compared using a scatter plot. Each plot shows the distance between the operator's hand and the Leap Motion controller within a particular reference axis, the required output angle as determined by the operator's software, and the actual angle moved by the servo motor. The maximum angular deviation from the true value is determined and is used to calculate the maximum inaccuracy within the system.

After performing this experiment 5 times for each servo motor and taking the average for the angles moved by the servo motor, the results are shown from Figure 5.1 to Figure 5.6. The results show the whole system has an inaccuracy of 1.67% (maximum of 3 degrees deviation in any position), with all servo motors having this maximum inaccuracy, except the left manipulator's elbow servo having an inaccuracy of 1.11%.

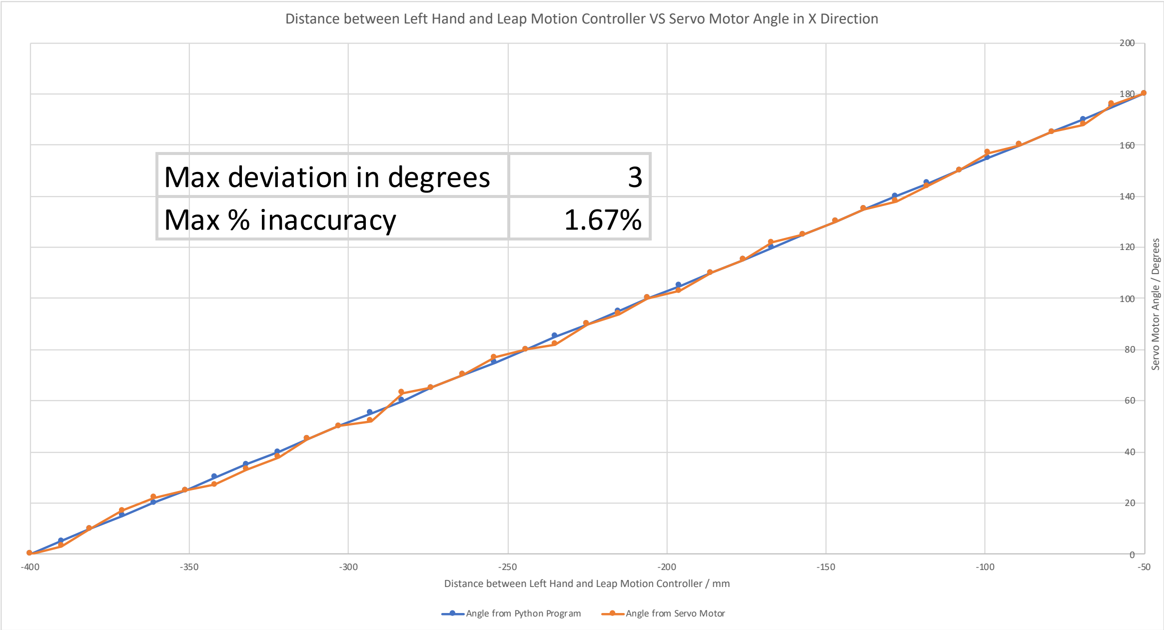

Figure 5.2: Accuracy Testing of Left Shoulder Servo

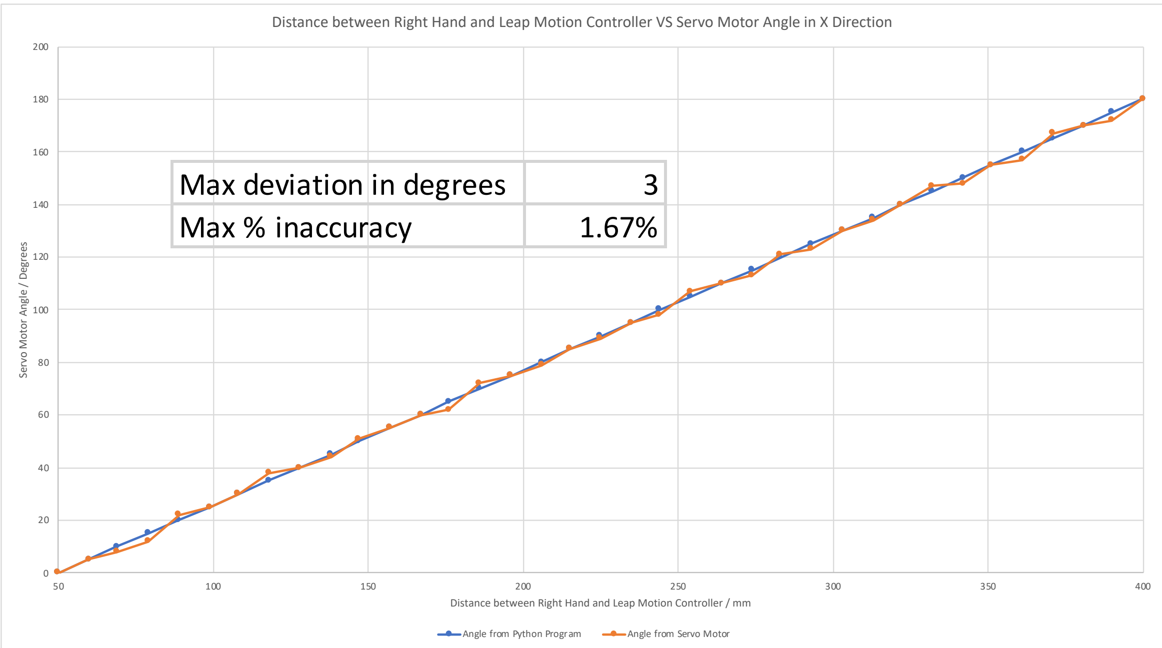

Figure 5.3: Accuracy Testing of Right Shoulder Servo

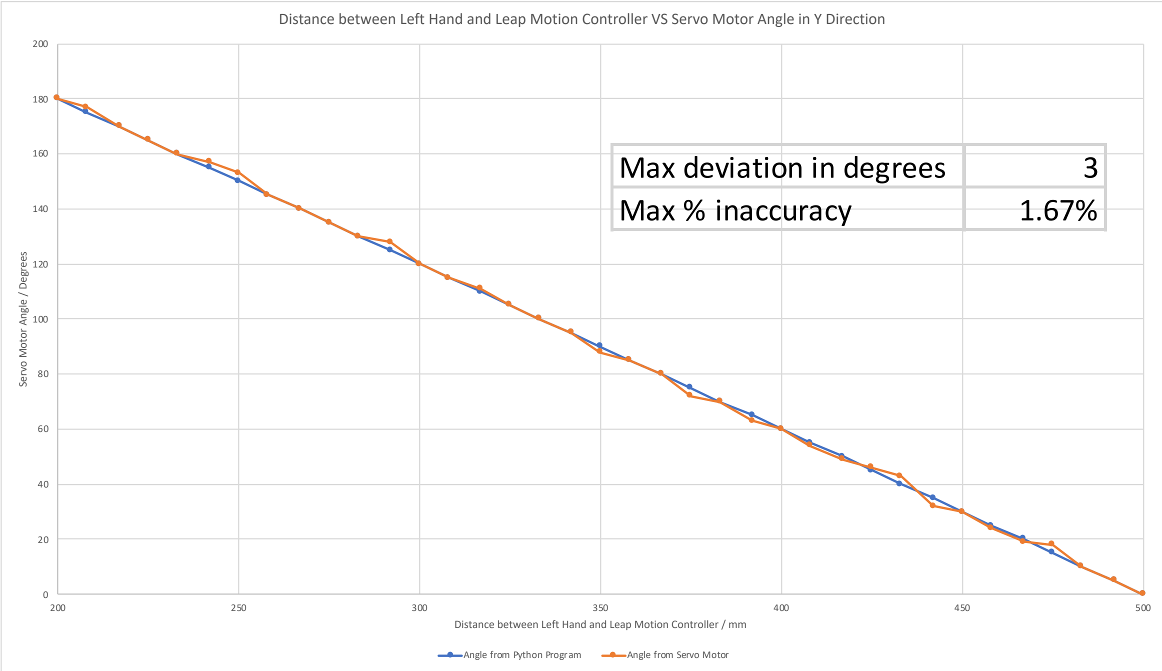

Figure 5.4: Accuracy Testing of Left Wrist Servo

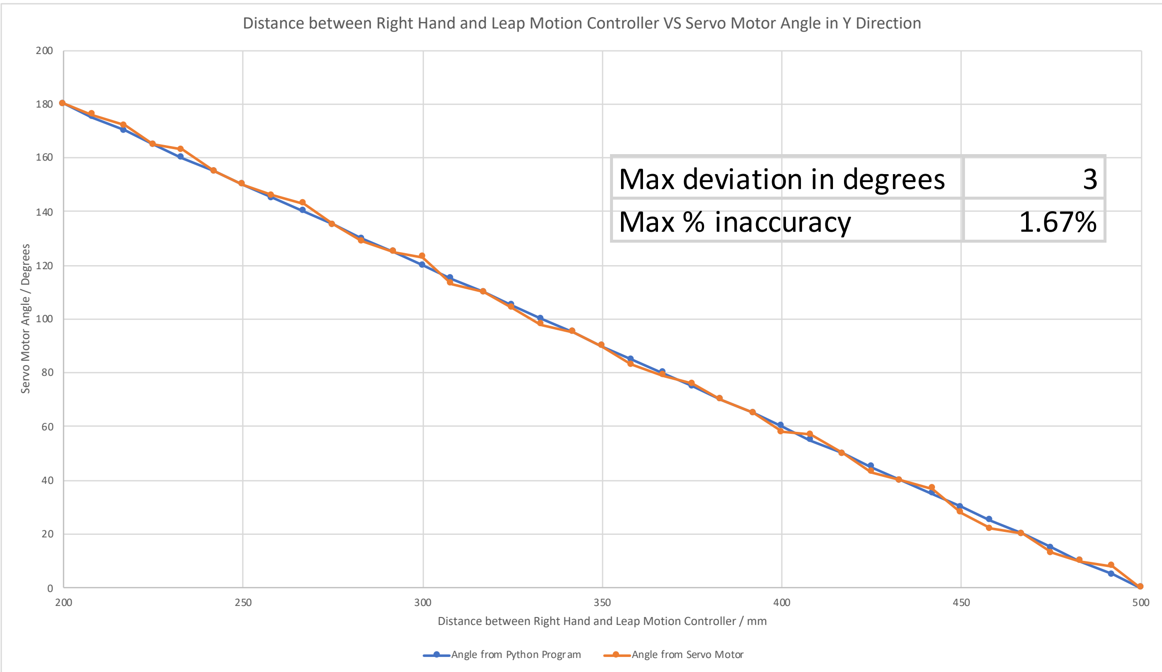

Figure 5.5: Accuracy Testing of Right Wrist Servo

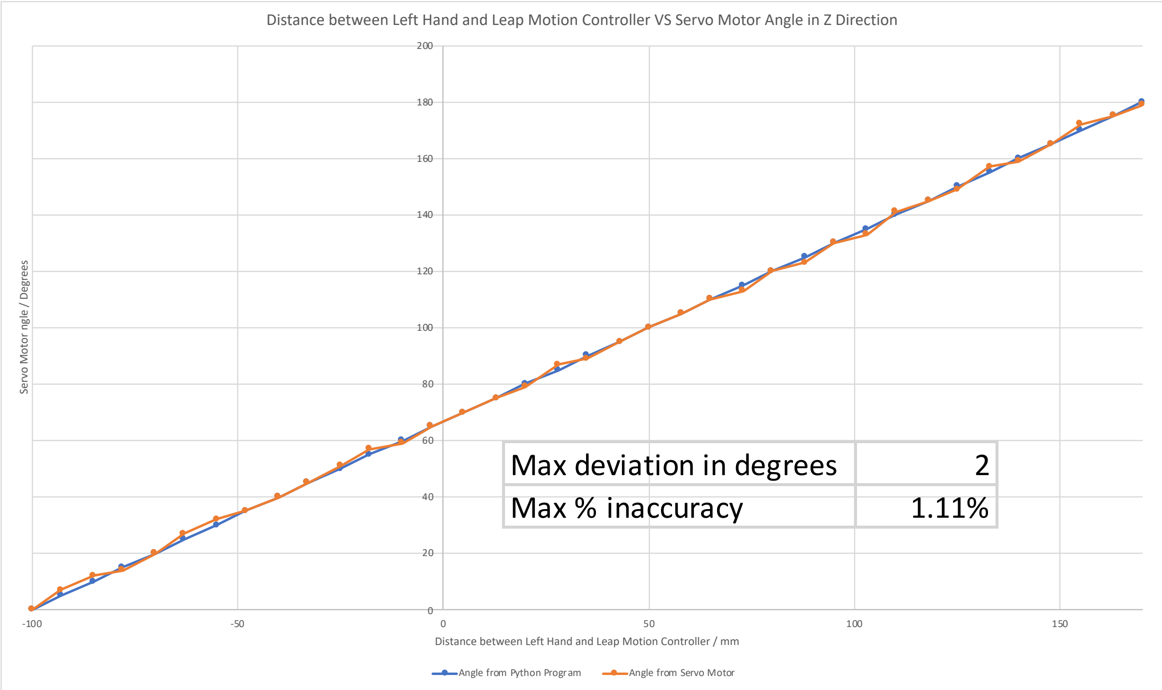

Figure 5.6: Accuracy Testing of Left Elbow Servo

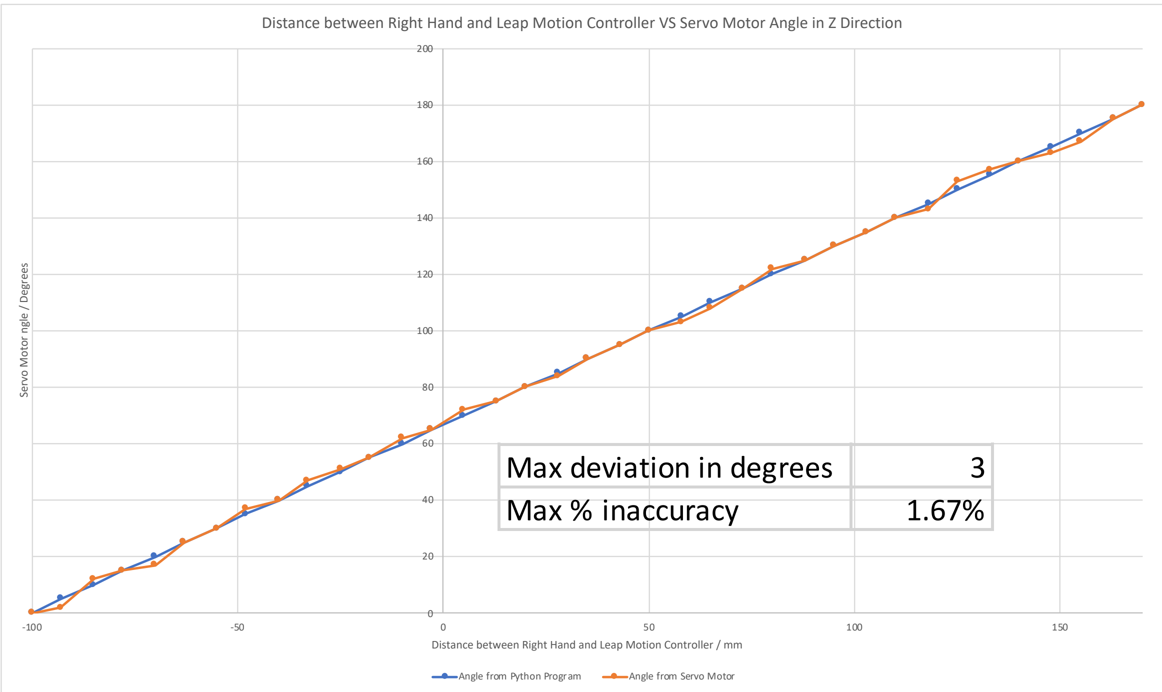

Figure 5.7: Accuracy Testing of Right Wrist Servo

5.3 Latency

As far as latency is concerned, the system is quite responsive in relaying and mimicking the operator's arm movements. During implementation, it became apparent that the setup on the robotic manipulators' subsystem, consisting of the two HC-05 modules and the FRDM board, was incapable of receiving multiple data instantaneously without data loss during transmission. Thus, a delay of 800ms was imposed after each data transfer at the operator's subsystem to ensure accurate data transfer at all times between both subsystems. As a result, the average latency of the complete system is 915ms: this consists of the total time taken from the point where the operator completes a particular gesture (opens palm) to the point where the servo motors execute the said gesture. Within this period, data is captured by the Leap Motion controller, converted into real-world coordinate data, transmitted to the robotic manipulators' subsystem, converted into PWM signals, and actuated by the servo motors.

Chapter 6: Conclusions and Future Works

This chapter presents final remarks, the system limitations, and future works to enhance and improve the project beyond expectations.

6.1 Conclusion

This study proposes an improved way of teleoperating dual robotic manipulators by using vision-based control. The study detailed how two 4-DOF robotic manipulators can mimic a human operator's arm movements accurately and precisely through the use of the Leap Motion serving as the visual sensor, the FRDM-KL25Z acting as the main microcontroller unit, and Bluetooth as the primary communication medium. Additionally, safe teleoperation is ensured by avoiding sharp, unintended movements by the operator using data gathered from the Leap Motion. Results and testing show that this system is not only a one size fits all solution, but it is relatively easy to use and understand compared to alternative modes of control.

6.2 Limitations & Further Works

No system is perfect, and as such, this system has some shortcomings that need improvement. Ideally, a system like this should operate in near real-time with as little latency as possible and should also be used across large distances. However, the current system has an average latency and range of 915 ms and 10 m, respectively. The system also lacks adequate feedback for the operator regarding the environmental conditions around the robotic manipulators. Lastly, when the operator switches position quickly, the motion of the manipulators can be overly jagged and haphazard.

In light of the limitations stated above, in future work, it is imperative to test with different communication protocols capable of providing latencies below 100 ms and used over very long distances. Remote sensing capabilities, such as environmental sensing and object detection, must also be implemented and attached to the manipulators to make the system smarter and keep the operator well informed. Lastly, filters can be developed to reduce noise originating from the sensor and eliminate any jagged motion during operation.

References

[1] R. Mardiyanto, M. F. R. Utomo, D. Purwanto and H. Suryoatmojo, "Development of hand gesture recognition sensor based on accelerometer and gyroscope for controlling arm of underwater remotely operated robot," 2017 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, 2017, pp. 329-333, doi: 10.1109/ISITIA.2017.8124104.

[2] G. Du and P. Zhang, “Markerless human–robot interface for dual robot manipulators using Kinect sensor,” Robotics and Computer-Integrated Manufacturing, vol. 30, no. 2, pp. 150–159, Apr. 2014, doi: 10.1016/j.rcim.2013.09.003.

[3] Y. Pititeeraphab, P. Choitkunnan, N. Thongpance, K. Kullathum and C. Pintavirooj, "Robot-arm control system using LEAP motion controller," 2016 International Conference on Biomedical Engineering (BME-HUST), Hanoi, 2016, pp. 109-112, doi: 10.1109/BME-HUST.2016.7782091.

[4] R. Kristof, V. Ciupe, C. Moldovan, I. Maniu, M. Banda and A. Stoian, "Arduino mobile robot with Myo Armband gesture control," 2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 2019, pp. 000294-000297, doi: 10.1109/SACI46893.2019.9111627.

[5] L. Heisnam and B. Suthar, "20 DOF robotic hand for tele-operation: — Design, simulation, control and accuracy test with leap motion," 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA), Kollam, 2016, pp. 1-5, doi: 10.1109/RAHA.2016.7931886.

[6] S. Okada, K. Hirano, R. Kobayashi, and Y. Kometani, “Development and Application of Robotics for Decommissioning of Fukushima Daiichi Nuclear Power Plant,” vol. 64, no. 4, 2020, Accessed: Oct. 06, 2020. [Online]. Available: http://www.hitachi.com/rev/archive/2020/r2020_04/04d02/index.html.

[7] G. Du and P. Zhang, "A Markerless Human–Robot Interface Using Particle Filter and Kalman Filter for Dual Robots," in IEEE Transactions on Industrial Electronics, vol. 62, no. 4, pp. 2257-2264, April 2015, doi: 10.1109/TIE.2014.2362095.

[8] J. Kofman, S. Verma and X. H. Wu, "Robot-manipulator teleoperation by markerless vision-based hand-arm tracking," Int. J. Optomechatron., vol. 1, no. 3, pp. 331-357, 2007.

[9] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, “Analysis of the Accuracy and Robustness of the Leap Motion Controller,” in Sensors (Basel), vol. 13, no. 5, pp. 6380–6393, May 2013, doi: 10.3390/s130506380.

[10] Y. Fan, C. Yang and X. Wu, "Improved Teleoperation of an Industrial Robot Arm System Using Leap Motion and MYO Armband," 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 2019, pp. 1670-1675, doi: 10.1109/ROBIO49542.2019.8961758.

[11] D. Bassily, C. Georgoulas, J. Guettler, T. Linner and T. Bock, "Intuitive and Adaptive Robotic Arm Manipulation using the Leap Motion Controller," ISR/Robotik 2014; 41st International Symposium on Robotics, Munich, Germany, 2014, pp. 1-7.

[12] C. Li, A. Fahmy and J. Sienz, "Development of a Neural Network-Based Control System for the DLR-HIT II Robot Hand Using Leap Motion," in IEEE Access, vol. 7, pp. 136914-136923, 2019, doi: 10.1109/ACCESS.2019.2942648.