EASTERN INTERNATIONAL UNIVERSITY

SCHOOL OF ENGINEERING

Project D Report

A Real-time AI- and Computer Vision-based Methods for Controlling Direct Current Motors

By Võ Phạm Mai Uyên – 1831100044

June 2023

ABSTRACT

Hand poses gesture recognition has attracted significant attention in robotics because it enables intuitive and seamless interaction and control between humans and robots. By recognizing and interpreting hand gestures, robots can comprehend and promptly respond to real-time human commands, allowing for efficient and effective human-robot communication. This technology holds great potential for enhancing robotic systems' usability and user experience, paving the way for various applications in various domains. However, the previous methods are mostly affected by noise, changing environmental lighting conditions, unstable, and inflexibility. Especially, applications of the wheelchair system are designed to meet the needs of people with limited mobility in body movements that require stability and precision from the hand gesture system. In this research of the project, we propound an approach to improve the performance of the existing above limitation and results in lower sensitivity to illumination changes, minimizing the number of cumbersome connections to the maximum extent possible by employing computer vision technique using a webcam. In order to validate the proposed method, we design a system that includes a hand gesture based on extracting features from the Mediapipe package; an STM32F407VG microcontroller applies the motion signals derived from hand gestures to regulate the motion of the DC servo motor, and PID controller is applied to allow precise speed management. A database system is also employed to record the motion history of the motor based on the speed and direction of movements. As a result, hand gesture recognition controls the movements of the DC servo motor, ensuring its stable performance across diverse lighting conditions while exhibiting rapid and precise responsiveness.

Keywords: Hand pose gestures, Mediapipe package, PID controller, MySQL database system

ACKNOWLEDGEMENT

To begin with, I would like to express our sincere thanks to our family, who provided financial support and valuable time for our research. In addition, I would like to extend my gratitude to the EIU FABLAB for providing the best equipment and environment for me to carry out this project. Furthermore, I would like to thank the members of the engineering department, especially the Automatic Control Engineering guys, for accompanying and sharing with us.

I respectfully thank my supervisor, Dr Nguyen Tan Nhu, who has always supported me with valuable advice, experiments, and guidance for this project.

Lastly, we acknowledge the examination and supervision efforts of the Committee in evaluating my project report, program, and hardware.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

PID	Proportional Integral Derivative Controller

PWM	Pulse Width Modulation
UART	Universal Asynchronous Receiver Transmitter
PCB	Printed Circuit Board
GPIO	General-Purpose Input/Output
DC	Direct Current
AI	Artificial Intelligence
SQL	Structured Query Language
ML	Machine Learning

Chapter 1. Introduction

1.1Overview

Hand poses gestures encompass distinct hand and finger configurations representing specific symbols, signs, or actions, finding applications in domains like human-computer interaction, sign language recognition, virtual reality, and gaming[1]. Examples of commonly recognized hand pose gestures include thumps up, peace signs, fists, high five, OK signs, victory signs, and pointing, each serving to convey particular meanings or commands within diverse contexts[2]. The utilization of hand gestures for controlling the movement of motor applications has seen a significant rise and becoming increasingly prevalent, especially in wheelchair applications systems which provide a more accessible and inclusive control solution for individuals with limited mobility or physical disabilities[3].

There are several ways to control the movement of electric motors based on hand gestures. For instance, using Wireless Hand Gesture Recognition for an Automatic Fan Speed Control System: Rule-Based Approach [4]. This method used a gyroscope, Bluetooth modules, and Arduino microcontrollers to develop a wireless hand gesture-controlled fan to alter the rotation speed of the motor. However, the limitation of this method is that it makes users feel confused and hindered as they have to wear gloves and use many cables. For adding control electric motor, more specifically like a wheelchair, can be controlled by speech[5] or even by identifying the form and position of lips[6]. However, both these systems are prone to unwanted movements. At the same time, the seat is being operated during a conversation, and it requires the wheelchair should be put in a stable environment without noise. Another way is using hand gestures to control Real-Time Robotic arms[7].

This project employs an Image processing technique using a web camera to detect the vital features of the hand: and fingers by computational geometry calculation, enabling real-time interaction between hand gestures and the robot. Moreover, image processing is applied to recognize the hand of the user, a skin detection is implemented to discard other objects in the scene to determine and control the direction of the mobile robot[8]. Nevertheless, when similar objects are in the scene, the system tends to be unstable, so it is necessary to develop a robust algorithm for the deception of the object of interest, and two of these methods could be noise and depend on the angle of contraction and light conditions.

1.2 Objective

From the methods mentioned above, we have discovered a new method that is faster and more accurate for controlling the motor by utilizing machine learning. In recent years, Artificial intelligence (AI), including Machine Learning(ML), has made impressive advancements as essential tools for smartly examining such data and creating relevant practical applications[9][10]. For example, there are some applications of Machine Learning, such as computer vision, speech recognition, and language processing[11][12][13]. We apply ML in our project using the Midiapipe package[14] to detect hand gestures and control mobile robots.

MediaPipe is a framework for building pipelines to perform inference over arbitrary sensory data. With MediaPipe, a perception pipeline can be made as a graph of modular components, including model inference, media processing algorithms, and data transformations[14]. We utilize the hand detection package from MediaPipe[15][16] to control mobile robots. The primary goal of this project is to design and implement a hand gesture system that facilitates the control motion of an electric motor. Furthermore, the system includes integrating a user interface for real-time monitoring of the system's status and recording its movements.

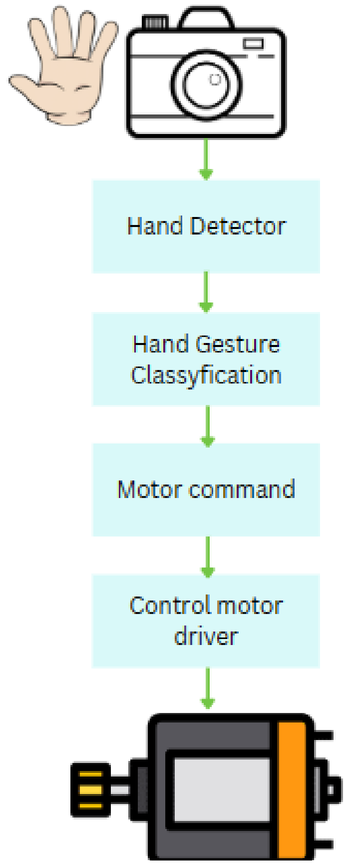


Figure 1. 1 The flow chart of the proposed method of controlling DC motor using hand gestures

Chapter 2. Related Works

Hand gesture recognition has gained growing popularity across diverse domains to enhance application quality, encompassing areas such as gaming or robotics…, so accurately recognizing gestures is still challenging in its development. Many hand gesture systems have been introduced to provide high accuracy of hand gestures, as listed in Table 2.1. This survey summarizes the performance of the existing hand gesture methods. In general, hand gestures system is often classified into three main groups: data glove, based on traditional image processing techniques for computing shapes, and deep learning techniques for learning features.

Glove-based approaches typically involve using inertial measurement unit (IMU) sensors affixed to the upper region of the hand. These sensors, comprising accelerometers, gyroscopes, and occasionally magnetometers, calculate precise angles using specialized algorithms. The existing glove-based method achieves real-time processing. However, their performance is not good enough over time due to the drift and noise characteristics of the sensor; in addition, their real-life usage is limited due to the need to wear cumbersome devices when interacting with the system[26]. Recently, several approaches have attempted to improve the accuracy and stability of the data glove, and it aids in mitigating the system's complexity and unwieldiness. The traditional image processing-based use webcam or depth camera to capture the information in the frames; the purpose of this method is to determine the shapes of hand gestures via a series of techniques such as data acquisition[27], skin detection[27], and hand segmentation[28][29]. However, the processing time of deep learning-based methods is slower than previous methods due to the optimization process and the handling of numerous parameters in the mathematical model after the model has been trained, so they require a powerful device to perform inference models. Therefore, it is necessary to introduce a more effective method that can achieve high accuracy and real-time processing under different lighting conditions and better handle multiple gestures simultaneously.

Table 2. 1 Survey on existing hand-posed gestures

Year	**Authors**	**Method**	**Discussion**
Hand Gestures Based on Data Glove			
2016	Pushpendra[17]	ADXL335 accelerometer for measuring the angles.	Wheelchair system with low cost and high accuracy while operating with capable of carrying a load up to 110kg
2017	Hasan et al. [18]	An accelerometer is employed to detect angles.	Rapid response, limited flexibility, cumbersome and sensor-induced noise, and low user experience

2020	Alvin et al.[4]	The generation of coordinates through the utilization of gyroscope axes	Cumbersome, Rapid response, not flexible, noise by sensors, and low user experience
Hand Gestures Based on Traditional Image Processing			
2007	R. Posada-Gómez et al.[8]	Geometric recognition via traditional image processing to recognize an object	Tends to be unstable with a similar object High flexibility and efficiency
2012	Ahmad et al.	Color filtering, Blob Size Filter, Center of Gravity.	The properties of center gravity control the direction of the robot. This method limits robot operating space.
2014	Ganesh et al.[7]	Computational geometry calculations to determine the position of individual fingers	High accuracy and low computation cost but influenced by ambient light
2015	Sunjie et al.[19]	Hand gestures based on the Leap Motion method.	Use a depth camera to measure the distance between hand and camera, open loop error is 4.65%, and the system has high precision.
2017	Alexandru et al.[20]	Adaptive binarization algorithm based on image processing techniques	Low processing time of 0.0295s (34 fps), the algorithm demonstrates strong performance under varying lighting conditions
2017	Weiguo et al.[21]	Skin Colour Detection, Binary Quantization, Horizontal/Vertical Projection Histogram, Discrete Fourier Transform, Classifier.	The method could recognize 24 static American Language hand signs, and with speed at 60 frames/s with a recognition accuracy of 93.3%
2017	Xiang et al.	Skin recognition technique, Color Segmentation, Depth segmentation	The results show great performance with an accuracy of approximately 99%, tested under different conditions.

2018	Yassine et al.[22]	Background subtraction, Skin color detection, Area calculation, and Centroid computation.	The implementation of a low-cost hand gesture-controlled wheelchair demonstrated favorable outcomes when evaluated on actual patients
2019	Shining et al. [23]	Ostsu adaptive threshold segmentation, Mathematical Morphology, Linear Gaussian Filter	Work well with a single background, and dynamic gestures reach more than 80%, fast recognition speed.
Hand Gestures Based on DeepLearning			
2020	Mihai et al.[24]	CNN, Feature Extraction	CNN gives a more stable response when showing multiple fingers, achieving 96% accuracy
2017	Tanu et al.[25]	Euclidean Distance, Principal Component Analysis (PCA), K-significant Eigenvectors	Using PCA for feature extraction, which handles noise(variation of brightness or color information)

Chapter 3. Design System and Methodology

A DC motor control system (Figure 3.1) has been developed to realize this project. The system was designed using a PC to implement a hand gestures-based Mediapipe package combined with inner product space calculations.

The motor parameters are stored in a database system, and a user interface is implemented using PyQt5 [26]; the motors' control and other devices were implemented using an

STM32F407 [27]. The hardware components, comprising the Control Box and Motor, are specifically designed to establish computer communication using the UART protocol [28]. Upon receiving commands from the computer, the STM32 controller generates signals to govern the movement of the motor.

The control software is created using Qt Designer [29] and relies on the pyqt5[26] libraries, which provide extensive design capabilities. This software implements nearly all connection stages, visualization, and control functionalities. The specific information regarding the components presented blows.

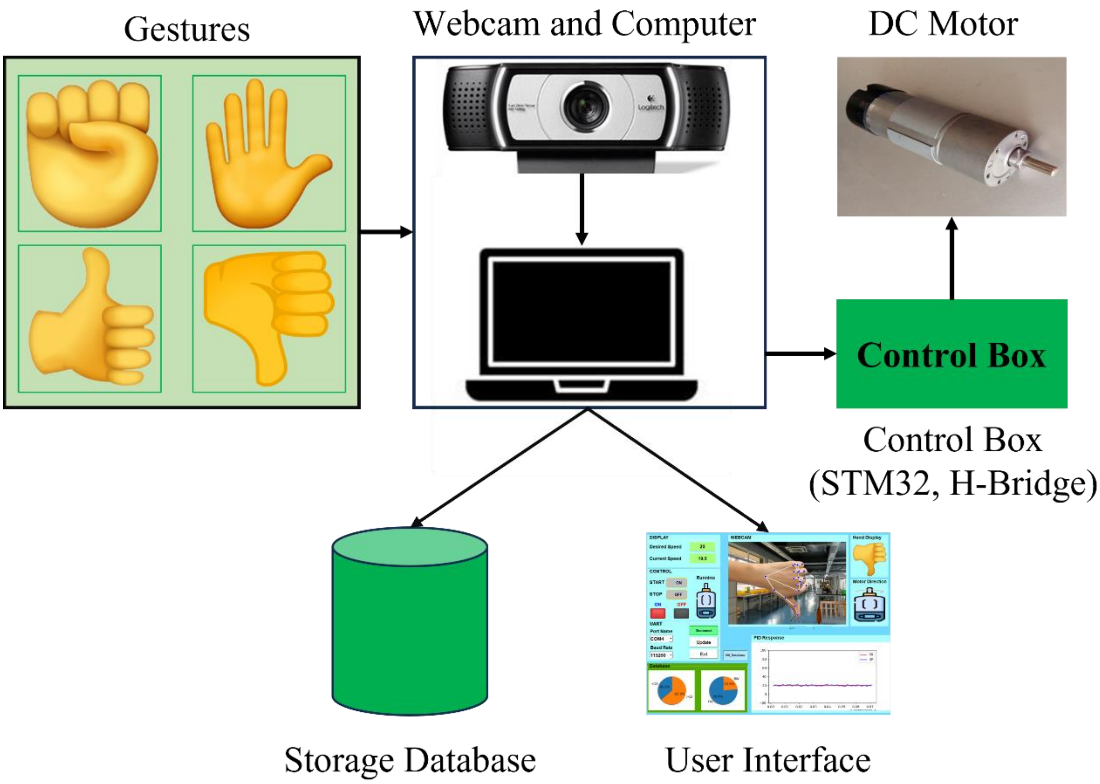


Figure 3. 1 Overall procedural flow of controlling motor based on hand gestures

3.1 Hardware

3.1.1 STM32F407

The project report focuses on the STM32F405xx and STM32F407xx family, built upon the powerful Arm® Cortex®-M4 32-bit RISC core, capable of operating at speeds up to 168 MHz. Notably, the Cortex-M4 core includes a single-precision Floating Point Unit (FPU) that supports all Arm single-precision data-processing instructions and data types. Moreover, it incorporates a comprehensive range of Digital Signal Processing (DSP) instructions and a memory protection unit (MPU) to enhance application security.

Regarding memory, the STM32F405xx and STM32F407xx family integrates high-speed embedded memories, featuring Flash memory with capacities of up to 1 Mbyte and up to

192 Kbytes of SRAM. Additionally, there are up to 4 Kbytes of backup SRAM available. The family also provides an extensive array of enhanced I/Os and peripherals connected to two APB buses, three AHB buses, and a 32-bit multi-AHB bus matrix.

Across all devices within this family, three 12-bit ADCs, two DACs, and a low-power RTC are offered. Furthermore, there are twelve general-purpose 16-bit timers, two PWM timers specifically designed for motor control, and two general-purpose 32-bit timers. A notable feature included is a true random number generator (RNG). Additionally, these devices come equipped with standard and advanced communication interfaces[27].

In this study, STM32 is utilized to generate peripherals such as input-output signals, interrupt timer, external interrupt, pulse width modulation, I2C, UART

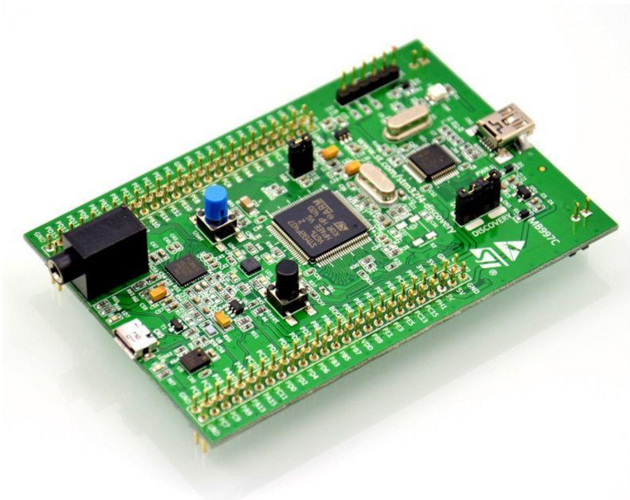


Figure 3. 2 Kit STM32F407VG

3.1.2 DC Servo Motor and H Bridge

a. DC Servo Motor

This project used a DC Gear Motor to control the gate to simulate how the gate is opened. By applying PID, the motor will be controlled with encoder pulses which can decide the opening range for the gate.

DC Servo Geared Motor engine is integrated with two AB encoders to read and control the exact position and rotation of the engine in applications that need high accuracy: PID control, self-self-robot onion,...

DC Servo Geared Motor engine has a metal structure for durability and high stability used in robotic models, cars, and boats;... the geared box of the engine is abundant. The ratio of transmission makes it easier for you to choose between traction and speed (the greater the traction, the slower the speed, and vice versa); the engine uses high-quality raw materials (pure copper core, steel leaves 407, Strong magnetic magnets, ...) for superior strength and durability than low prices on the market today (using aluminum core, magnetic magnet[30].

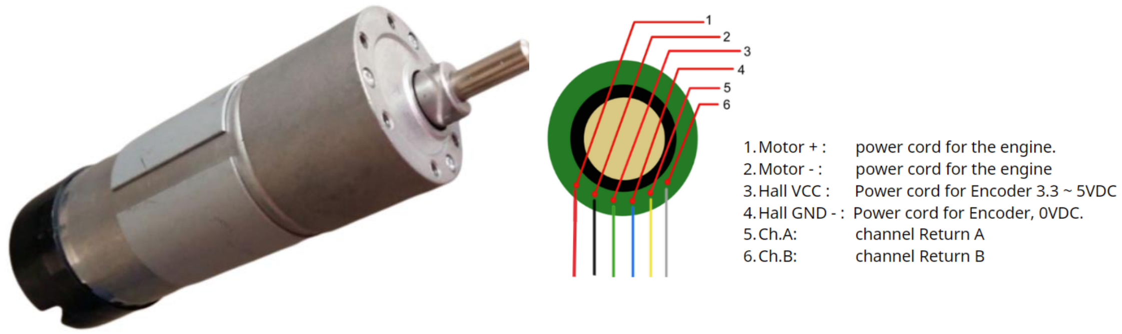


Figure 3. 3 DC Servo Motor and Encoder

b. The H bridge:

The H-Bridge is a crucial electronic circuit utilized in power systems, primarily enabling the control of voltage polarity applied to a load. Its significance becomes evident in various applications, including the operation of DC motors in both forward and backward directions and its extensive use in robotics.

In this project, an H bridge was added to control the DC servo motor at different speeds.

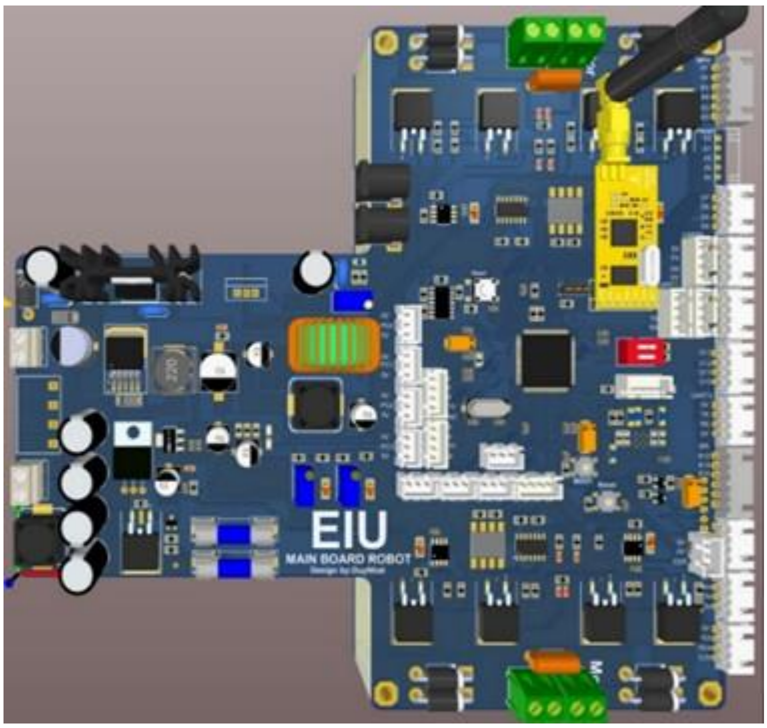


Figure 3. 4 3D PCB of H Bridge Circuit

3.1.3 LCD

The LCD 1602 module is controlled through a microcontroller or other compatible interfacing devices, enabling the display of alphanumeric characters, symbols, and basic graphical elements. This functionality presents customized information such as text messages, numbers, and status indicators.

Integrating the LCD 1602 module into the project makes it possible to provide clear and concise visual feedback, enhancing the overall user experience. Its application can be found in various electronic devices, ranging from digital meters to embedded systems, where the simplicity and versatility of the module are advantageous.

Including the LCD 1602 module in the project facilitates effective information communication and showcases the project's commitment to incorporating user-friendly features and enhancing accessibility[31].

Figure 3. 5 Module LCD 1602

Feature[31]:

- Type: Character
- Display format: 16 x 2 characters
- Built-in controller: ST 7066 (or equivalent)
- Duty cycle: 1/16
- 5 x 8 dots, including cursor
- + 5 V power supply
- LED can be driven by pin 1, pin 2, or A and K
- N.V. optional for + 3 V power supply
- Optional: Smaller character size (2.95 mm x 4.35 mm)

3.1.4 Camera

In this project, detection is significant, so the camera cannot leak. I used the C930 HD Pro Webcam to support me in collecting the frame and then send to the main processor.

The extended 90° diagonal field of view displays individual users and their workspace in a perfect visual frame while two integrated omnidirectional mics capture audio clearly up to one meter away. 4X digital zoom at 1080p and H. 264 video compression ensure smooth video with the highest level of detail. C930e is certified for Microsoft Teams® and Skype™ for Business, and works with other popular applications including BlueJeans, Cisco Webex™, Fuze, Google Meet™, GoToMeeting®, Lifesize Cloud, Microsoft DirectShow, Pexip, RingCentral Video, Vidyo, and Zoom®[32].

Figure 3. 6 Logictech C930 HD Pro Webcam

3.2 Software

3.2.1 Hand Detection

As mentioned in the introduction objective, I utilized Hand Detection Package from the Mediapipe framework to detect the hand of the user via camera.

The MediaPipe Hand Landmarker task enables users to identify the landmarks of hands in an image. The hand landmark model bundle precisely detects the positions of 21 hand-keypoint coordinates within the identified hand areas.

The hand landmark model bundle contains a palm detection model and a hand landmarks detection model. The Palm detection model locates hands within the input image, and the hand landmarks detection model identifies specific hand landmarks on the cropped hand image defined by the palm detection model[15].

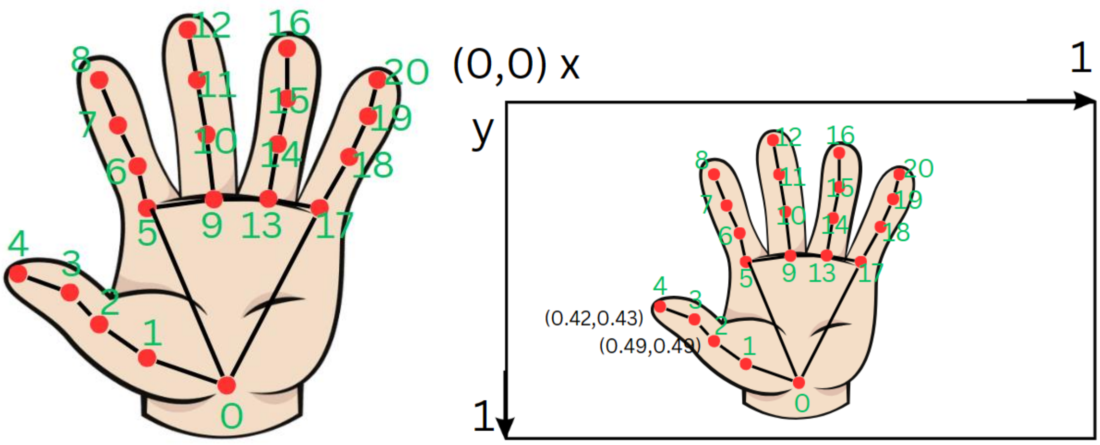
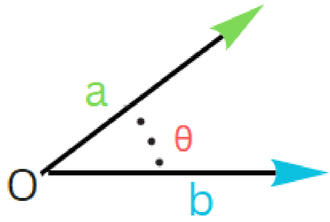


Figure 3. 7 21 Hand Landmarks

After receiving 21 key points from the Mediapipe package, four hand poses have been classified: Open-palm, Closed-fist, Thumb-up, and Thumb-down. These are calculated based on the angle between two vector methods with the below function to categorize these four gestures.



$$cos\theta = \frac{a.b}{|a||b|}$$

The equation is applied in the project to calculate the angles through landmarks of hand and after classified gestures. The below table takes note of the name of each angle calculated.

Table 3. 1 angle between vectors

First point of vector a (Hand landmark number)	Origin of two vectors (Hand landmark number)	First point of vector b (Hand landmark number)	Name of angle
3	2	0	A
6	5	0	B
7	6	5	C
10	9	0	D
11	10	9	E
14	13	0	F
15	14	13	G
18	17	0	H
19	18	17	I

a.) Open-palm

All five fingers must be straightened when the hand is in an open-palm pose.
Firstly, I find the A angle with the coordinates (x, y, z) of keypoint 3, 2, and 0 are:
point 0 [6.30657911e-01 6.47334218e-01 5.61352010e-07]
point 2 [0.48846927 0.49735931 -0.05927572]
point 3 [0.43717837 0.42051688 -0.07917768].

a = [x3 – x2, y3 – y2, z3 – z2] = [-0.0512909, -0.07684243, -0.01990196]

b = [x0 – x2, y0 – y2, z0 – z2] = [0.14218864, 0.14997491, 0.05927628]
a.b = -0.0512909 * 0.14218864 + -0.07684243 * 0.14997491 + -0.01990196 * 0.05927628 = -0.0199971

$$|a|.|b| = \sqrt{(-0.0512909)^2 + (-0.07684243)^2 + (-0.01990196)^2}.$$

$$\sqrt{(0.14218864)^2 + (0.14997491)^2 + (0.05927628)^2} = 0.020318767$$

$$\theta = acos\frac{-0.0199971}{0.020318767} = 2.96 => A = 169.79\ (degree)$$

Similarly, with the different angles.

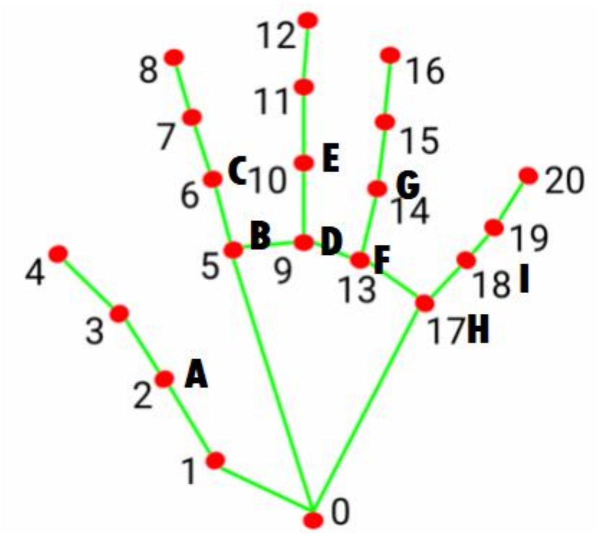


Figure 3. 8 Hand landmarks and angles

After calculating and doing the experiment, we recorded the angle range between multiple vectors in the table below.

Table 3. 2 Result angle of open-palm gesture

Angles	Range of degrees
A	160 - 180
B	160 - 180
C	160 - 180
D	160 - 180
E	160 - 180
F	160 - 180
G	160 - 180
H	160 - 180
I	160 - 180

b.) Closed-fist

In the closed-fist hand pose, the finger is not straightening and has different angles from the open-palm gesture.

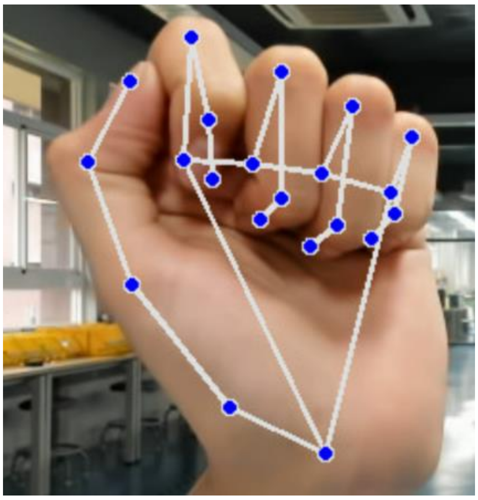


Figure 3. 9 Closed-fist gesture

A similar process as the open-palm calculation is applied in this part and recorded in the degree range in the table below.

Table 3. 3 Result angle of closed-fist gesture

Name of angle	Range of degree
A	90 – 130
B	30 – 145
C	40 – 170
D	30 – 140
E	40 – 170
F	25 – 130
G	45 – 155
H	20 – 130
I	50 - 160

c.) Thumb-up and Thumb-down

From part b.) above, I found that thumb-up, thumb-down, and closed-fist gestures are similar in the degree of vectors. However they are one significant distinction in the thumb finger.

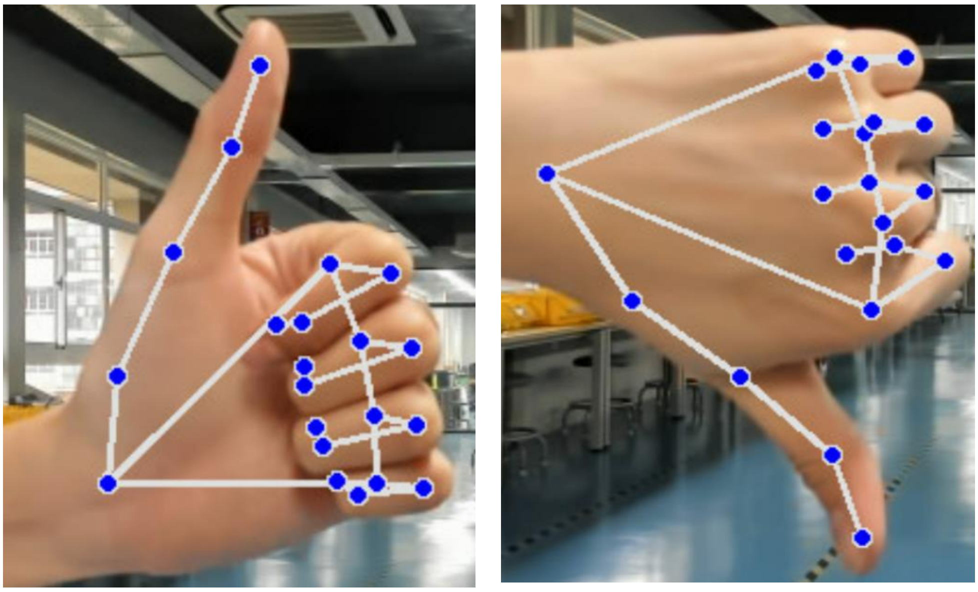


Figure 3. 10 Thumb-up and thumb-down hand-posed

The degree of A angle in this gesture ranges from 160 to 180 degrees, and the other angle from B to I are similar to the closed-fist part.
In contrast, thumb-up and thumb-down gestures exhibit remarkable similarities, necessitating a novel approach to differentiate between these highly similar gestures. Extensive research on these hand poses has revealed a distinguishing factor: the orientation of the thumb finger.
Firstly, I create a vector from the point 0 hand landmark perpendicular to the ground with the coordinate of y equal 0, x, and z coordinates are the x and z coordinates of point 0 key points. Then calculate, the degree of two vectors is applied.

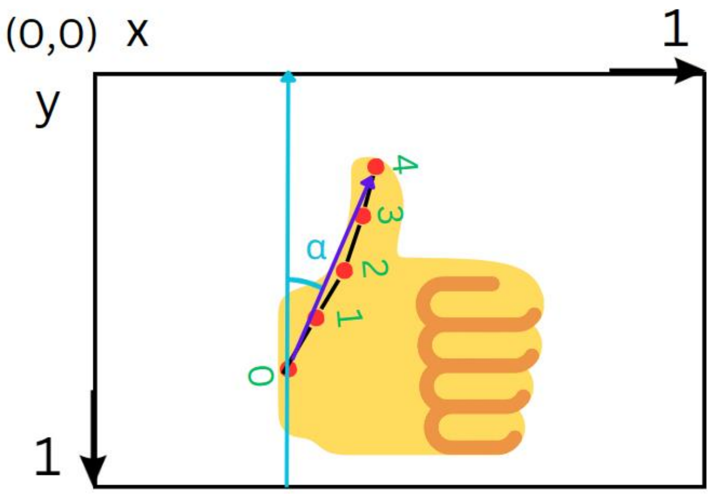


Figure 3. 11 Thumb-up gesture with alpha angle

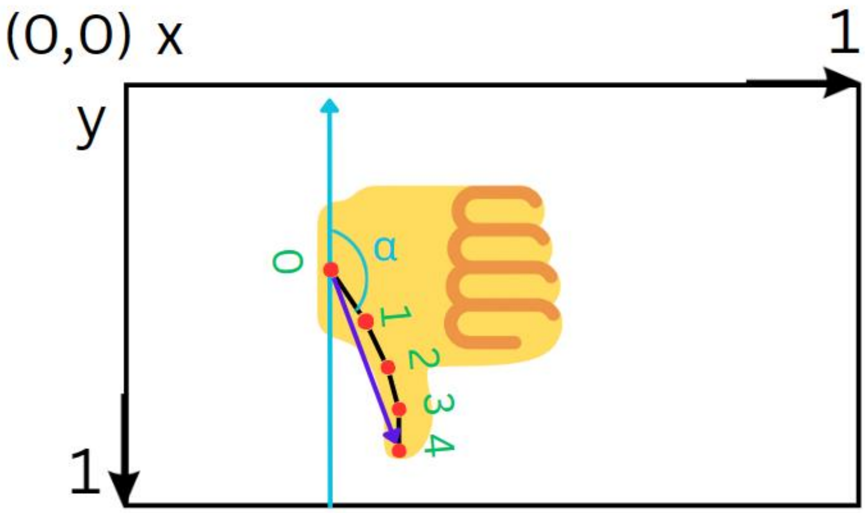


Figure 3. 12 Thumb-down gesture with alpha angle

From two Fig 3.10 and Fig 3.11, the alpha angle (α) of the two hand poses differs. In the thumb-up gesture, the α is in the range from 0 to 90 degrees, while the α in the thumb-down gesture is in the range of 90 to 180 degrees.

I recorded the experimental and research results in the table below.

Table 3. 4 Result angle of thumb-up and thumb-down gesture

Name of angle	Range of degree
A	160 – 180
B	50 - 140
C	50 - 180
D	30 – 160
E	20 – 150
F	30 – 160
G	30 – 140
H	20 – 170
I	5 - 150
α angle of thumb-up	0 - 90
α angle of thumb-down	90 - 180

3.2.2 Control Motor

Kit STMF407VG controls DC servo motor by emitting pulses, and this pulse is sent to the H bridge and then controls the motor. However, the expected speed (setpoint) and the real speed are not always the same, and there is a difference I call an error. I applied the PID Controller method in this project to reduce error and improve system stability.

What is PID Controller?

Proportional-integral-derivative (PID) control is undoubtedly the most widely used control strategy today. Over 90% of control loops are estimated to employ PID control, often with the derivative gain set to zero (PI control). Over the last half-century, much academic and industrial effort has focused on improving PID control, primarily in tuning rules, identification schemes, and adaptation techniques. It is appropriate to consider the state of the art in PID control and new developments in this control approach[33].

PID. controller is used to:

- Minimize the settling error
- Decrease settling time and overshoot
- Reduce oscillating

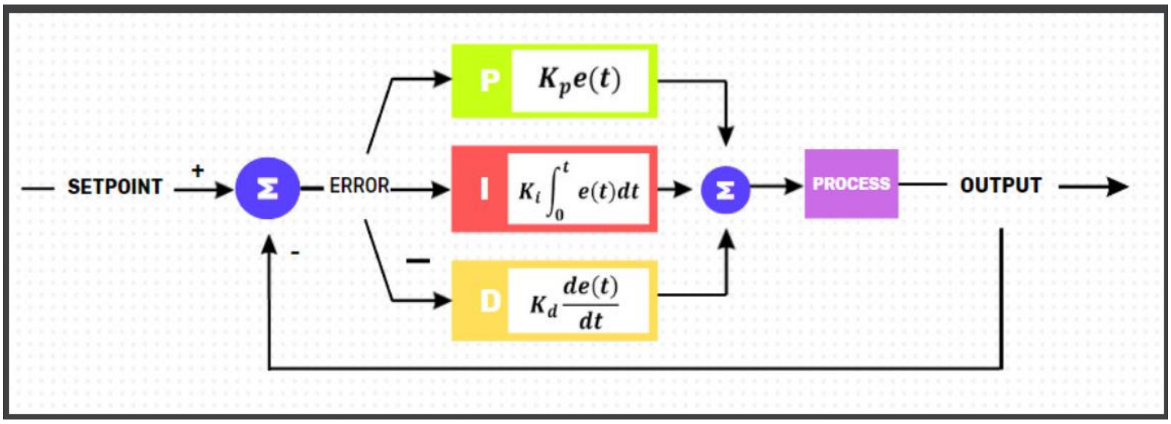


Figure 3. 13 PID Controller

Where:

- **P**: proportional to the current value of the set-point (SP) − PV error.
- **I** : accounts for past SP − PV error values and integrates them over time to produce the **I** term.
- **D** : is the best estimate of the future trend of the SP − PV error based on its the current rate of change.

This project aims for the motor to speed exactly like the speed sent from Kit STM32F407VG The error here is the subtraction of the set point and speed return. The output is the angle we will give the motor the right speed.

Table 3. 5 Value of Kp, Ki, and Kd in the control system

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	Decrease	No effect in theory	Improve if K_d small

How to choose the value of Kp, Ki, and Kd?

The process of choosing these values is called tuning. In broad terms, there are three approaches to determining the optimal combination of these settings: manual tuning, tuning heuristics, and automated methods.

Our DC motor has 8640 pulses per round and can run a maximum of 50 rounds per minute (rpm). The duty cycle kit STM32F407VG emits from 0 to 100 percent, similar to 0 to 24V in the H bridge. After reading the number of pulses from the encoder sensor, I convert this pulse to velocity in an rpm unit to calculate the error following the below equation:

$$speed = ((pulses / 8640.0) * (1/0.01) * 60)$$

Where:

speed: velocity of the motor

pulses: number of pulses read from the encoder

So, the error will be equal: setpoint – feedback speed

In our main code, the duty cycle is the output calculated by the PID controller

3.2.3 SQL

A database is a structured collection of related data stored and managed using specialized systems and software. It allows for the efficient storage, retrieval, updating, and management of information. A database consists of tables, data items, relationships between data items, and rules that define how the data is organized and accessed. Databases are widely used in various fields, including enterprise management, information systems, storage repositories, and web and mobile applications

I use MySQL Workbench to create a database to save and access the database.

Figure 3. 14 MySQL Workbench Application

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and more.

MySQL Workbench enables a DBA, developer, or data architect to design, model, generate, and manage databases visually. It includes everything a data modeler needs to create complex ER models, forward and reverse engineering. It also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort.

MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. The SQL Editor provides color syntax highlighting, auto-complete, reuse of SQL snippets, and execution history of SQL. The Database Connections Panel enables developers to easily manage standard database connections, including MySQL Fabric. The Object Browser provides instant access to database schema and objects.

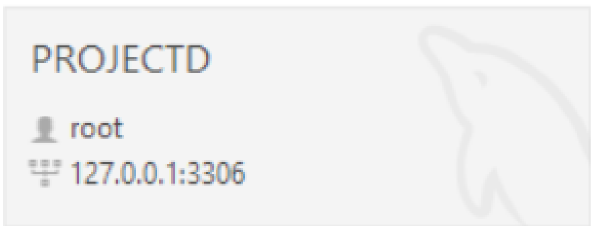


Figure 3. 15 Database of the project using MySQL Workbench

The main codes are shown below to insert data into the database via the MySQL Workbench application.

```
"""
INSERT INTO {} (speed, dir)
VALUES (%s, %s)
""".format(self.table_name)
```

The main codes are shown below to get data from the database via the MySQL Workbench [34] application.

```
f"SELECT  speed, dir FROM {self.table_name}"
```

speed	dir	timestamp
27	1	2023-06-22 17:11:39
28	1	2023-06-22 17:11:42
29	1	2023-06-22 17:11:45
30	-1	2023-06-22 17:11:48
31	1	2023-06-22 17:11:51
32	1	2023-06-22 17:11:54
33	1	2023-06-22 17:11:57
34	1	2023-06-22 17:12:00
35	-1	2023-06-22 17:12:03
36	-1	2023-06-22 17:12:06
37	-1	2023-06-22 17:12:09
38	-1	2023-06-22 17:12:12
39	-1	2023-06-22 17:12:15
40	-1	2023-06-22 17:12:18

Figure 3. 16 Database of the project, which recorded speed, direction, and time

3.2.4 Application and programming language

a.) Python encryption language

Figure 3. 17 Python programming language

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development and for use as a scripting or glue language to connect existing components. Python's simple, easy-to-learn syntax emphasizes readability and reduces program maintenance costs. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

Often, programmers fall in love with Python because of its increased productivity. Since there is no compilation step, the edit-test-debug cycle is high-speed. Debugging Python programs is straightforward: a bug or wrong input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. The interpreter prints a stack trace when the program does not catch the exception. A source-level debugger allows inspection of local and global variables, evaluating arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, the quickest way to debug a program is often to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

b.) C encryption language

C is a general-purpose programming language created by Dennis Ritchie at the Bell Laboratories in 1972.

Figure 3. 18 C programming language

It is a very popular language, despite being old. C is strongly associated with UNIX, as it was developed to write the UNIX operating system.

c.) Visual Studio Code (VS Code)

As mentioned, I employ Python programming to detect hand and send data to kit STM32F407VG. VS Code is used to write and execute the main code.

Figure 3. 19 Visual Studio Code

Visual Studio Code is a free coding editor that helps you start coding quickly. Visual Studio Code supports many languages, including Python, Java, C++, JavaScript, and more.

d.) Keil μVision 5

Keil μVision 5 application is used to build C code and connect to kit STM32F407VG.

Figure 3. 20 Keil µVision 5

Keil-C 5 is an integrated development environment (IDE) and compiler for 8051 microcontrollers and ARM-based microcontroller families. It is developed by ARM (formerly Keil Software) and is a popular tool used for programming embedded applications on microcontrollers. Keil-C 5 provides a user-friendly graphical interface with a compiler, debugger, and other supporting tools for developing and testing embedded software on various microcontroller platforms.

The table below shows which GPIO is used in this project.

Table 3. 6 GPIO used in the project

Number	GPIO	Purpose
1	PB 6	Rx data (Receive data)

2	PB 7	Tx data (Transmit data)
3	PA 0	Read encoder channel A
4	PA 1	Read encoder channel B
5	PA 8	Emit PWM
6	PA 9	Change the direction of motor

e.) QT Designer

Qt Designer is the QT tool for designing and building graphical user interfaces (GUIs) with Qt Widgets. Widgets and forms created with Qt Designer integrate seamlessly with programmed code, using Qt's signals and slots mechanism so that you can easily assign behavior to graphical elements. All properties set in Qt Designer can be changed dynamically within the code.

Figure 3. 21 QT Designer

3.2.5 Flowchart

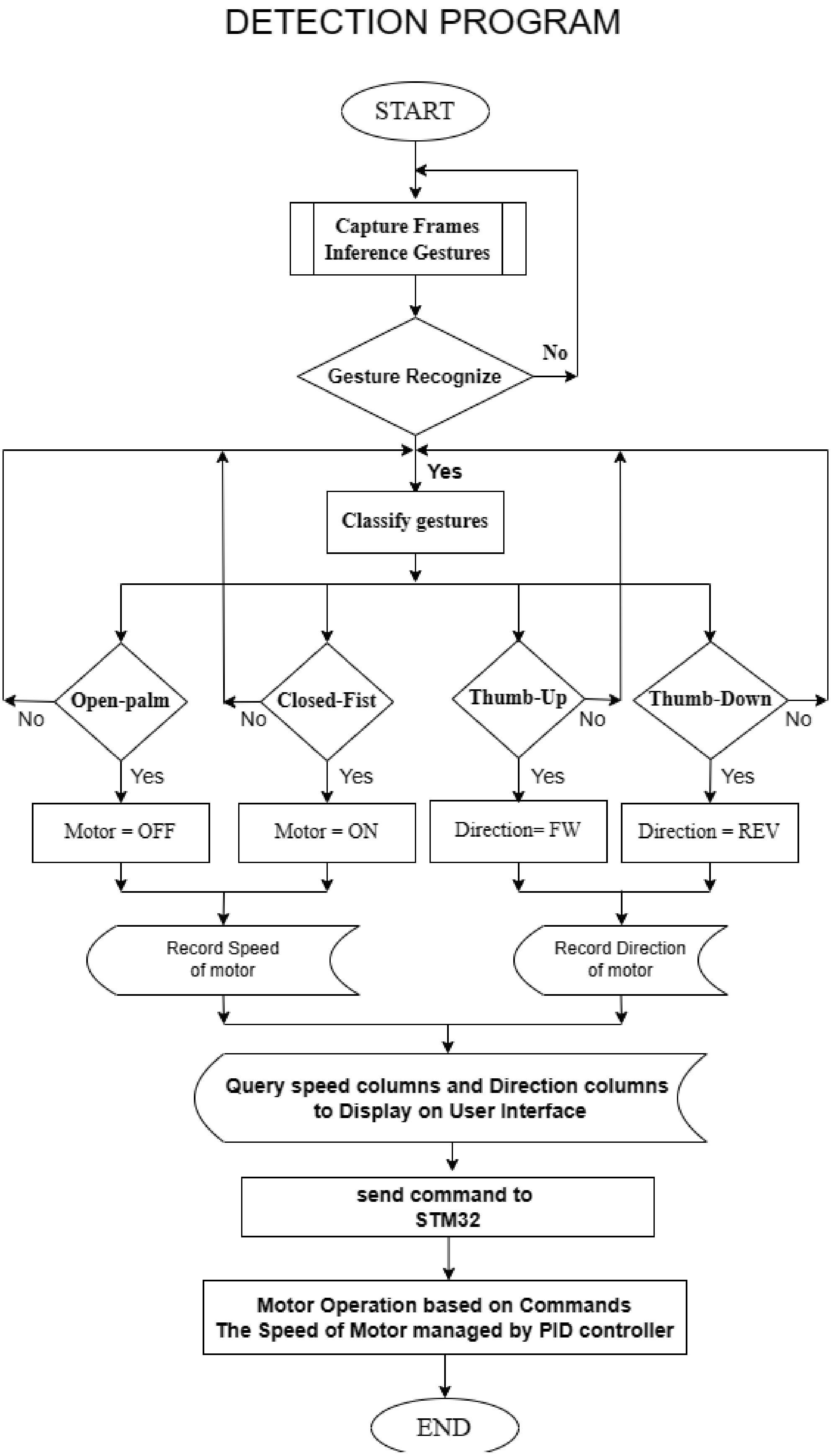


Figure 3. 22 Flowchart

From the flowchart, when the system is powered, the camera will take the frame, and the system detects the hand and then classify. Then PySerial sends motor and direction data to kit STM32F407VG, which is processed after transmission. The motor variable is used for the active motor, 0 is the stop motor, and 1 is the active motor. At the same time, the direction variable is used to set the direction of the motor if 1 is run forward when -1 is reverse. The speed of the motor and direction in which STM32F407VG send by UART is inserted into the database. The data will be get from the database and then shown in the user interface via two pie charts, which show the trend of direction and speed of the motor.

Chapter 4: Experimental Result

This project has aimed to make the DC motor operate according to several motions by the help of hand gestures, and this goal was achieved through the prototype

4.1 Hardware Configuration of Result

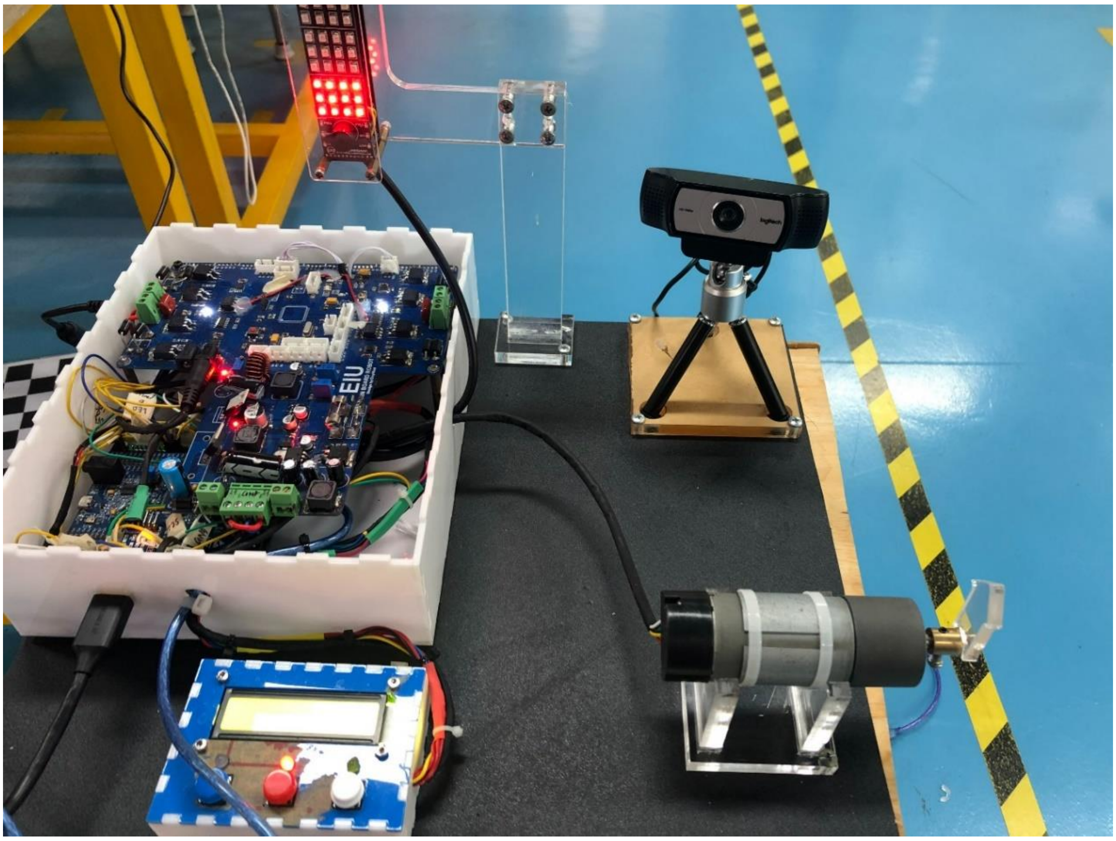


Figure 4. 1 Hardware result

Figure 4.5 shows that the hardware prototype was designed, including a control box, a DC motor, a webcam, and a control panel to control our system. This hardware was implemented at EIU FABLAB for research and learning purposes; most of the components work stably to meet the needs of the topic.

4.2 Gesture Recognization Result

Figure 4. 2 Result of hand-posed detection Open-palm

Figure 4. 3 Result of hand-posed detection Closed-fist

Figure 4. 4 Result of hand-posed detection Thumb-up

Figure 4. 5 Result of hand-posed detection Thumb-down

Figure 4.2 – 4.5 illustrates gestures like thumb-up, thumb-down, closed-fist, and open-palm for recognizing and classifying. The user interface also states the motor while running or stopping, PID response and database graph are designed for observation of the current

condition of the DC motor and managing the speed and direction of the motor, respectively. In addition, we could control the motor from the control panel and set desired rates from this interface.

Chapter 5: Conclusion and Perspectives

5.1 Evaluation

The hand gesture detection-based control DC servo motor was designed and built as fully functional and met the requirements I set out to achieve. Including detection of four hand poses, control DC motor, and create a database that can access.

What I learned from the project included technical skills like designing and building a detection-hand system and more personal skills like working effectively and managing a project from start to finish.

Finally, I can successfully apply knowledge in my major about automatic control (PID controller) and computer programming (database).

5.2 Drawback

PID coefficient still exists a little bit of trouble that leads our motor to run overshoot and make minor errors in angle.

The project needs to improve the detection of multiple hands; when two hands are in the frame, the system can not classify the reliability of which gesture.

In addition, the position of the user's hand is extremely important; the hand has to be put in front of the camera and perpendicular to the camera so that the program can detect the correctness of which gesture. With different users and the size and shape of the hand, the system needs to be improved to detect and classify better.

5.3 Future Work

Improve the PID controller's accuracy to ensure that the motor runs at truthfulness speed.

I have to apply two stages: stage one is to detect hand pose for extracting 21 landmarks using the Mediapie Hand Tracking package, and stage two is to calculate the angle-based inner product method and then classify hand-posed gestures. So, to improve and enhance the system's accuracy, the task in future work is to combine the two stages mentioned above into one stage; this will increase the system's accuracy, and then it could detect more hand shapes from different users.

Bibliography

[1] F. Wen, Z. Zhang, T. He, and C. Lee, "AI enabled sign language recognition and VR space bidirectional communication using triboelectric smart glove," *Nat. Commun.*, vol. 12, no. 1, p. 5378, 2021.

[2] L. Feng, Y. Du, S. Liu, L. Xu, J. Wu, and H. Qiao, "Hand Gesture Recognition with Leap Motion: Volume 1," 2019, pp. 46–54. doi: 10.1007/978-3-030-02686-8_4.

[3] X. Gao, L. Shi, and Q. Wang, "The design of robotic wheelchair control system based on hand gesture control for the disabled," in *2017 International Conference on Robotics and Automation Sciences (ICRAS)*, 2017, pp. 30–34.

[4] A. S. Alon and J. A. B. Susa, "Wireless Hand Gesture Recognition for an Automatic Fan Speed Control System: Rule-Based Approach," in *2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*, 2020, pp. 250–254. doi: 10.1109/CSPA48992.2020.9068687.

[5] G. Pacnik, K. Benkic, and B. Brecko, "Voice operated intelligent wheelchair - VOIC," in *Proceedings of the IEEE International Symposium on Industrial Electronics, 2005. ISIE 2005.*, 2005, pp. 1221–1226 vol. 3. doi: 10.1109/ISIE.2005.1529099.

[6] U. Canzler and K.-F. Kraiss, "Person-adaptive facial feature analysis for an advanced wheelchair user-interface," in *conference on Mechatronics \& Robotics*, 2004, pp. 871–876.

[7] G. C. B and C. R. B. V, "Real time robotic arm control using hand gestures," in *2014 International Conference on High Performance Computing and Applications (ICHPCA)*, 2014, pp. 1–3. doi: 10.1109/ICHPCA.2014.7045349.

[8] R. Posada-Gómez, L. Sanchez Medel, G. Alor-Hernández, A. Martinez Sibaja, A. Aguilar-Laserre, and L. Leija-Salas, "A Hands Gesture System Of Control For An Intelligent Wheelchair," 2007, pp. 68–71. doi: 10.1109/ICEEE.2007.4344975.

[9] O. Koteluk, A. Wartecki, S. Mazurek, I. Kołodziejczak, and A. Mackiewicz, "How Do Machines Learn? Artificial Intelligence as a New Era in Medicine," *J. Pers. Med.*, vol. 11, no. 1, 2021, doi: 10.3390/jpm11010032.

[10] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 3, p. 160, 2021, doi: 10.1007/s42979-021-00592-x.

[11] N. Cummins, A. Baird, and B. W. Schuller, "Speech analysis for health: Current state-of-the-art and the increasing impact of deep learning," *Methods*, vol. 151, pp. 41–54, 2018, doi: https://doi.org/10.1016/j.ymeth.2018.07.007.

[12] S. Hegde, S. Shetty, S. Rai, and T. Dodderi, "A Survey on Machine Learning Approaches for Automatic Detection of Voice Disorders," *J. Voice*, vol. 33, no. 6, pp. 947.e11-947.e33, 2019, doi: https://doi.org/10.1016/j.jvoice.2018.07.014.

[13] A. Le Glaz *et al.*, "Machine Learning and Natural Language Processing in Mental Health: Systematic Review," *J Med Internet Res*, vol. 23, no. 5, p. e15708, May 2021, doi: 10.2196/15708.

[14] C. Lugaresi *et al.*, "MediaPipe: A Framework for Building Perception Pipelines." 2019.

[15] A. Vakunov, C.-L. Chang, F. Zhang, G. Sung, M. Grundmann, and V. Bazarevsky, "MediaPipe Hands: On-device Real-time Hand Tracking," 2020.

[16] G. Sung *et al.*, “On-device Real-time Hand Gesture Recognition.” 2021.

[17] P. Jha and P. Khurana, “Hand gesture controlled wheelchair,” *IJCTA*, vol. 9, no. 41, pp. 243–249, 2016.

[18] H. Zaman, asif alam Joy, K. Akash, and S. Fayad, “A simple and effective way of controlling a robot by hand gesture,” 2017, pp. 330–333. doi: 10.1109/ICCONS.2017.8250737.

[19] S. Chen, H. Ma, C. Yang, and M. Fu, “Hand gesture based robot control system using leap motion,” in *Intelligent Robotics and Applications: 8th International Conference, ICIRA 2015, Portsmouth, UK, August 24-27, 2015, Proceedings, Part I 8*, 2015, pp. 581–591.

[20] A. Pasarica, C. Miron, D. Arotaritei, G. Andruseac, H. Costin, and C. Rotariu, “Remote control of a robotic platform based on hand gesture recognition,” in *2017 E-Health and Bioengineering Conference (EHB)*, 2017, pp. 643–646. doi: 10.1109/EHB.2017.7995506.

[21] W. Zhou, C. Lyu, X. Jiang, P. Li, H. Chen, and Y.-H. Liu, “Real-time implementation of vision-based unmarked static hand gesture recognition with neural networks based on FPGAs,” in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2017, pp. 1026–1031.

[22] Y. Rabhi, M. Mrabet, and F. Fnaiech, “Intelligent control wheelchair using a new visual joystick,” *J. Healthc. Eng.*, vol. 2018, 2018.

[23] S. Song, D. Yan, and Y. Xie, “Design of control system based on hand gesture recognition,” in *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, 2018, pp. 1–4.

[24] M.--B. Simion, D. Seli\csteanu, and D. Șendrescu, “Dc motor control using hand gestures,” in *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*, 2020, pp. 149–153.

[25] T. Srivastava, R. S. Singh, S. Kumar, and P. Chakraborty, “Feasibility of Principal Component Analysis in hand gesture recognition system,” *arXiv Prepr. arXiv1702.07371*, 2017.

[26] “No Title”, [Online]. Available: https://pypi.org/project/PyQt5/

[27] STMicroelectronics, “DS8626, STM32F407xx,” no. August, p. 203, 2020.

[28] “No Title”, [Online]. Available: https://www.bing.com/search?q=uart+protocol+stm32&qs=n&form=QBRE&sp=-1&ghc=1&lq=0&pq=uart+protocol+stm32&sc=10-19&sk=&cvid=F8852EA6021640889E51B2823537C2E3&ghsh=0&ghacc=0&ghpl=

[29] “No Title”, [Online]. Available: https://www.bing.com/search?q=qt+designer&aqs=edge.0.69i59j69i57j0l6j69i60.1288j0j1&FORM=ANAB01&PC=ACTS

[30] H. Maekawa, “Compact servo driver for torque control of DC-servo motor based on voltage control,” in *1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No.99TH8399)*, 1999, pp. 341–346. doi: 10.1109/AIM.1999.803189.

[31] M. Data and A. M. Ratings, “LCD-020N004L Vishay 20 x 4 Character LCD STANDARD VALUE UNIT ELECTRICAL CHARACTERISTICS CONDITION

UNIT LCD-020N004L," *Datasheet*, pp. 1–3, 2016.

[32] L. Cloud and R. Video, "C930E Business Webcam," 2021, [Online]. Available: www.logitech.com/C930e

[33] C. Knospe, "PID control," *IEEE Control Syst. Mag.*, vol. 26, no. 1, pp. 30–31, 2006.

[34] "No Title", [Online]. Available: https://www.mysql.com/products/workbench/