Punjab Engineering College (Deemed to be University)
Chandigarh

EXPLORE INNOVATE EXCEL

HAND GESTURE CONTROLLED ROBOTIC HAND

MINOR PROJECT

MENTOR: DR DIVYA DHAWAN

MADE by: Akshit Maheshwari(19105027)
Karan Naurd(19105038)
Garvit Banga(19105087)
Kshitij Sethi Jain(19105100)

Acknowledgement

We would like to thank Dr. Divya Dhawan, (Professor Electronics and communication department) who guided and helped us throughout the project. It was because of Divya Dhawan ma'am that we made this project successfully. We would also like to thank Punjab Engineering College(PEC) for giving us the opportunity to work on this project.

We sincerely would like to thank all the people that believed in us and made possible this period of professional and personal growth. First of all our parents who supported us during difficulties. Even if they do not have a clear idea of what we are actually doing, they were always excited to share with us satisfactions and great results

We would also want to thank Punjab Engineering College, and in particular, the Electronics and communication department, that offers an extremely high level of teaching and that brings willing people to a successful carrier.

We would also like to thank all the group members who did all the work through day and night to successfully complete the project even in the covid times.We hope that this wonderful friendship, in and outside the lab, will last forever.

Lastly we want to thank everyone that appreciated or criticised our work. The process of research need your support to go ahead and to get always new incitement.

DECLARATION

We hereby declare that the project work entitled "*Hand Gesture controlled robotic hand*" is an authentic record of our work carried out as a requirement Minor project in the 5th semester of our degree of B-Tech (Electronics and Communication Engineering), Punjab Engineering College under the guidance of Dr. Divya Dhawan.

Kshitij Sethi Jain(19105100)

Akshit Maheshwari(19105027)

Karan Naurd(19105038)

Garvit Banga(19105087)

Date: 20/12/2021

Certified that the above statement made by the students is correct to the best of our knowledge and belief.

Dr. Divya Dhawan
Electronics and Communication Engineering

Punjab Engineering College

(Deemed to be university)

Abstract

The present day world is all about various types of technology which has paved the way for multi-functional devices. The one application of technology is automation. Automation deals with technology, programs, robotics to achieve outcomes with minimal human input. Robotics include designing, construction, operation and use of robots. Most robots are used to do repetitive actions, work in hazardous environment and in medical fields too. This kind of robots can also be used in present pandemic situation.

The interest of this project is in the research and development of humanoid robots that has been steadily growing in recent years. The application of such robotic systems are diversified and insightful .A robot is a complex machine that involves the conjunction of many technologies working harmoniously together to provide to the final user, a nice interface to interact.

Our project is one such robotic system which was built in the form of a hand. The aim is to investigate how well the robotic hand could imitate the movements of a user and replicate those movements . The earlier work regarding this type of hand gesture controlled robotic arm requires the user to wear electronic gloves in order to control the hand. But this project is not limited to glove assistance. This hand can duplicate the gestures of the user by enacting in front of the camera.

Index

Topic **Page number**

CHAPER 1
INTRODUCTION

Overview

This chapter gives brief insight into various non-theoretical aspects of the project. This chapter will help to understand why we need this project and how will it serve the humanity or the human race for betterment. It will also give insights about what we are going to do in this project and what all things motivated us to do so.

1.1 Motivation

During covid times, covid testing booths were made where the medical personnel was required to be near the patient for testing. The risk for covid was still a problem although the medical staff took precautions. One of the substitutes for this situation was to implement gesture-controlled robotics. But most of the gesture-controlled robotics hands require the user to wear a sensing glove to operate. The hardware used to operate these gloves was high, this made the overall project uneconomical.

There was lot of health hazards to mine workers or the workers working in unsafe environments. So reduce the stress and health hazard of labour this project was created to cater the needs of labours.

In medical industry there was many problems where contagious denies could spread to the nurses serving the patients so there was a need of a Robot which would undertake these tasks and decrease human dependency in unsafe places.

With the advancement in the field of Machine learning a major portion of this cost could be reduced using image processing as a input to the robotic.

1.2 Need For Project

As a method of Human-Computer Interaction (HCI), hand gesture recognition can serve as a substitute to standard remote controls or keystrokes currently prescribed. It can do away with the need to learn complex control systems as it provides a natural intuitive system . Presently, there are two main types of HCIs that can interpret human hand gestures. The first is the Data Glove method, which has the user wear a glove of some description that requires the use of accelerometers, gyroscopes, power source for wireless gloves or a network of data and power lines for a wired glove. The second method is by using a form of *Computer Vision* to isolate the hand and track its movements; in some experiments a coloured glove is used to aid the tracking process. Both methods of hand gesture recognition have their arguments for and against.

With the *Data Glove* method, experiments have found that the trending accuracy tends to be higher than some computer vision methods, upwards of 95% gesture recognition due to their higher sampling rate of 100 samples per second. However, there are a few draw backs. First is the high cost of a data glove . Second is accelerometers drift, resulting in a continuous lowering of accuracy over extended periods of activity . Finally, data gloves tend to be cumbersome and uncomfortable when used or long periods at a time and not robust enough for outdoor use .

1.3 Objective

The main objective of this project was to replicate a human with grip strength by the help of machine learning and hardware. This project also aimed at reducing the cost of already developed robotic arms which used the concept of gloves to control the hand.

Chapter 2
Literature review

Overview

This chapter gives details about the various research papers which were read to complete this project. It also includes what were the conclusions and important aspects we learned by reading all the research papers and how did it help in completing the project.

2.1 Research Paper

Survey on Design and Development of competitive low cost Robot Arm With Four Degrees of Freedom by Ashraf Elfasahany[1]:
In this paper the representation of the design, development and implementation of robot arm is done, which has the ability to perform simple tasks, such as light material handling. The robotic arm is designed and made from acrylic

A survey on Arduino Controlled Robotic Arm by Ankur Bhargava[2]
In this paper a 5 Degree of Freedom (DOF) robotic arm have been developed. It is controlled by an Arduino Uno microcontroller which accepts input signals from a user by means of a set of potentiometers. The arm is made from four rotary joints, where rotary motion is provided by a servomotor. Each link has been first designed using Solid works Sheet Metal Working Toolbox and then fabricated using a 2mm thick Aluminium sheet. The servomotors and links thus produced assembled with fasteners produced the final shape of the arm. The Arduino has been programmed to provide rotation to each servo motor corresponding to the amount of rotation of the potentiometer shaft. A robot can be defined according to the nature of the relative movements between the links that constitute it.

Review on development of industrial robotic arm by Rahul Gautam[3]:
This selective operation robotic control method is need to be overcome the problem such as placing or picking object that at distant from the worker. The robotic arm has been developed successfully as the movement of the robot can be controls precisely. It is expensive to change the cable and therefore the designing to reduce the friction on table, is crucial to increase time between maintenance.

Ranjith Kumar Goud and B. Santosh Kumar[4]
They have invented a pick and drop robot. They wanted it to be used for diffusing a bomb remotely with safety. For the robotic arm, they used a pair of motors and another pair as the wheels of the robot for controlling the movement. Connectivity is established using Bluetooth. The micro-controller used is LPC2148. They had also attached wireless camera for remote surveillance. They have worked on this project mainly for industrial and military applications.

Arpit Sharma, Ritesh Verma, Saurabh Gupta, Sukhdeep Kaur Bhatia[5]
They have configured an android Smartphone which can control a robot via Bluetooth technology. The phone uses motion sensors and records the gestures sent via an android mobile phone. It also has an inbuilt accelerometer and Bluetooth module for controlling the movements of a robot.

2.2 Choosing The Microcontroller

Arduino is a versatile hardware which helps in many IEEE and electronics project. It can be used in many wide space project as well as automation.

Arduino UNO can connect upto 6 motors . But since this hand was big and needed big motors to carry the fingers with the help of string we needed a powerful motor. Arduino UNO was not able to deliver the load of powerful motors that too 10 motors. So to deliver the correct output and maintain the integrity of the project another microcontroller was chosen for this task. So finally the microcontroller perfect for this hand

gesture robotic arm was PCA9685.PCA9685 was available to deliver the right energy output and was easily able to move the joints and eventually the fingers.

Arduino UNO is also used alongside PCA9685 because it acted like a bridge between the the motors and the computer. That is it received data from PC and send it to other supporting modules. That is PCA9685 in our case.

2.3 Conclusion

After reading all the documents and research paper related to the project we inferred that Arduino UNO is a versatile hardware and a quite useful one but still it cannot be used in the project because of the limitations of the output power from Arduino. Along side this it could not connect 10 motors simultaneously. All the research and reading helped to understand the best hardware which could be used for the robotic arm.

CHAPTER 3

Hardware and software used

Overview

This chapter covers all the hardware and the software which has been used in the robotic hand. It explains why all these modules were used and what were the limitations of these hardware and software modules.

3.1 Software Used

3.1.1 Python -

Fig 3.1 python logo

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python was used to record and mimic the gestures of the human hand. The libraries used for the purpose are :

1) OpenCV -

Fig 3.2 open cv logo

OpenCV is a great tool for **image processing and performing computer vision tasks**. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. It supports multiple languages including python, java C++.

OpenCV's application areas include:

1. 2D and 3D feature toolkits
2. Egomotion estimation
3. Facial recognition system
4. Gesture recognition
5. Human-computer interaction (HCI)
6. Mobile robotics
7. Motion understanding
8. Object identification

OpenCV was used to capture and record the gestures and movements of the fingers of the users.

2) MediaPipe -

Fig 3.3 Media Pipe logo

MediaPipe is **a cross-platform library developed by Google** that provides amazing ready-to-use ML solutions for computer vision tasks.

Main feature of MediaPipe are:
1. Cutting edge ML models
2. Face Detection
3. Multi-hand Tracking
4. Hair Segmentation
5. Object Detection and Tracking
6. Objectron: 3D Object Detection and Tracking
7. AutoFlip: Automatic video cropping pipeline

MediaPipe was used to analyse all the movements and gestures which were recorded with the help of OpenCV. Hand was recognised and tracked using MediaPipe then the hand was divided in 21 coordinates which were used to provide the parameters required by Arduino for its functioning.

3.2 Hardware Used-

3.2.1Exo Skeleton

The exoskeletal of the model that is the palm and fingers were made using cardboard. The main benefits of cardboard was:

- It was easily available
- Best out of waste
- Cheap
- Durable
- Easy to mould

Fig 3.4 exoskeleton

3.2.2 Basic Construction

The prototype consists of 4 fingers with 3 joints each and a thumb with 2 joints.

Each finger was controlled using two threads . One thread was used to control upper two joints and the other thread for controlling lower joint. Using the combination movement of these two threads every possible orientation of finger was achieved.

3.2.3 Components used

1) Servo MG 996R

MG 996R is a 'Positional rotation servo motor' that is capable of rotating 180 degrees. It includes physical stops located in the gear mechanism to stop turning outside these limits to guard the rotation sensor. Position of the gear is controlled using a PWM wave of frequency 50Hz with a duty cycle of 1ms - 2ms. Fig 3.5 shows a servoMG 996R

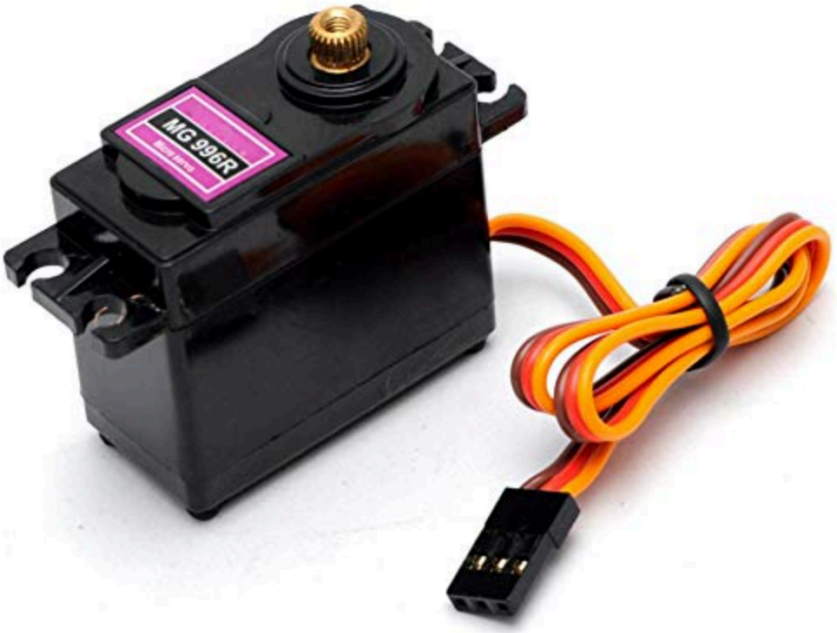


Fig 3.5 servo MG 996R

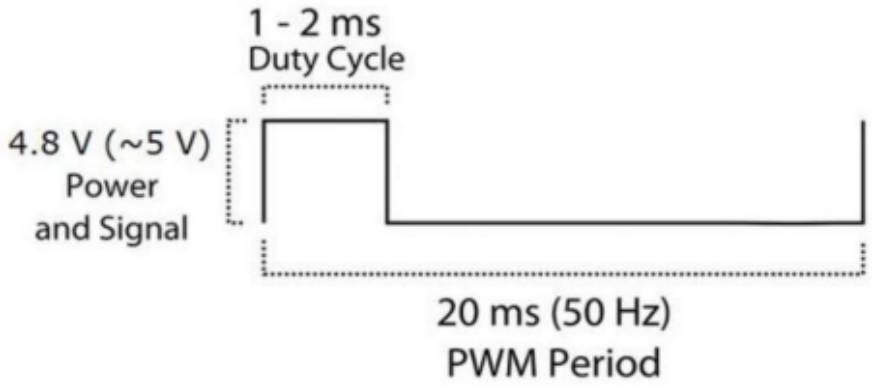


Fig 3.6 PWM wave

1ms duty cycle give 0 degree position and 2ms duty cycle gives 180 degree position. Position varies linearly with the duty cycle. Figure 3.6 shows a PWM wave.

More specfications of MG 996R Servo are as follows-

Weight – 55g

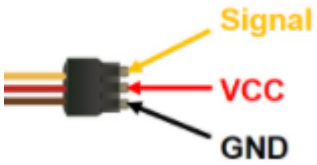
Dimension – 40.7 * 19.7 * 42.9 mm (approx)

Stall torque – 10kgf-cm (5V)

Opeating voltage – 4.8v - 7v

Operating speed – 0.15 s/60 degrees (5V)

-Pin Connections



Servo motors are connected with PCA 9685 using three pins . Pin with yellow wire recieves the PWM control signal , pin with red wire is the power supply pin and the last pin with brown wire is the ground pin.

2) PCA 9685

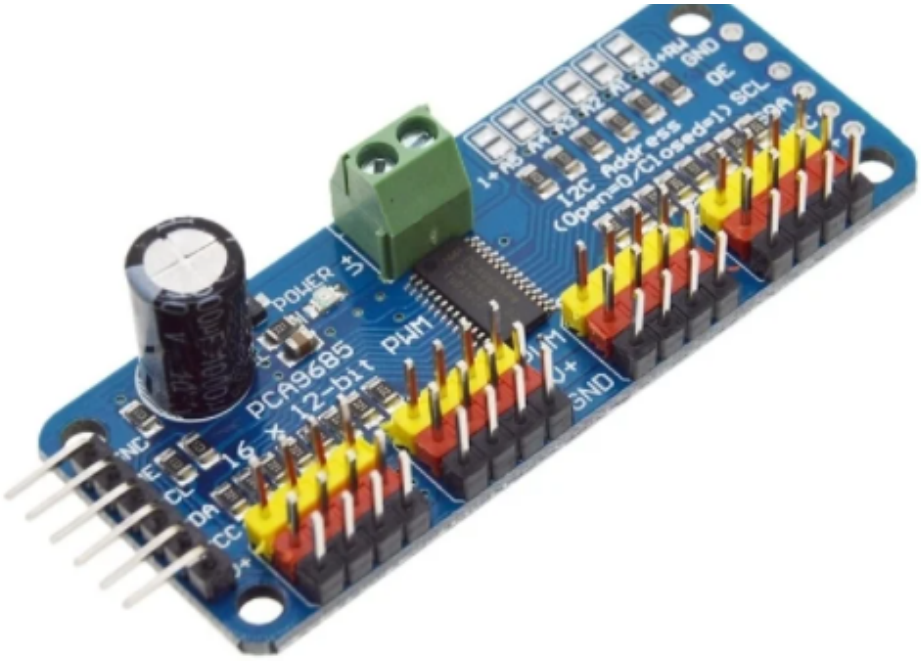


Fig 3.7 PCA 9685

PCA 9685 is an 16 channel , 12 bit PWM Fm + IC bus controller . PCA 9685 was choosen because of its capability to control upto 16 servo

motors at a time . It uses PWM wave to control the motion of servo motor.

Pin connections-

-VCC pin is connected with 5V pin of arduino for powering the module.

-GND pin is connected with GND pin of arduino.

-SDA pin is connected with A4 pin of arduino for data communication.

-SCL pin is connected with A5 pin of arduino for data communication.

-V+ pin is connected with power supply for powering the connected servo motors.

3) Arduino uno

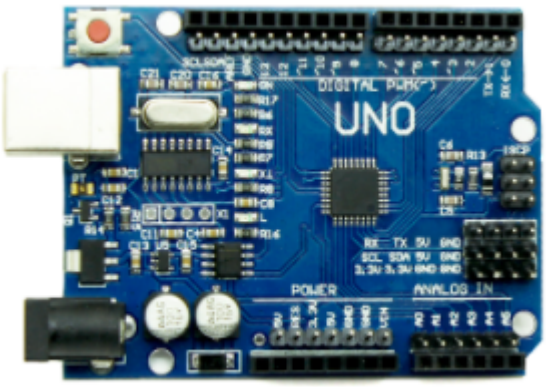


Fig 3.8 Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button.

It is used as an interface between the computer and PCA 9685 module. It was connected with computer using the USB port . Data is received from the computer serially at 9600 baud.

Pin connections

-5V pin is connected with Vcc of PCA 9685.

-GND is connected with GND of PVA 9685.

-To communicate with PCA 9685 analog pins A4 and A5 are connected with SDA and SCL of PCA 9685 respectively.

4) Power supply

In worst case each MG 996R servo requires upto 900mA current , which along with 9 other servo motors add upto 9A in worst case scernio . To met with such high current demand a 220V AC to 5V DC 10A down converter was used to power the servo's.

Positive terminal from power supply is connected with 'V+' pin of PCA 9685 and ground terminal from power supply is connected to 'Gnd' pin of PCA 9685.

3.2.3 PIN diagram

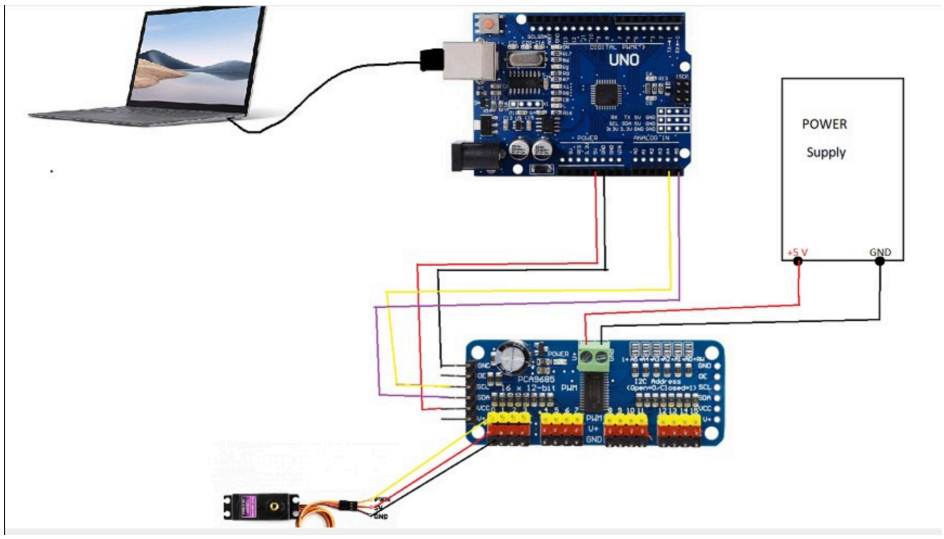


Fig 3.9

The image is captured and processed by the laptop, then the angles are measured and then sent to arduino through serial port the arduino converts the received data into equivalent Pam waves to control the motors accordingly. The arduino communicates with PCA9685 through analog pins A4 and A5 then the motors are controlled using PCA9685

Chapter 4
Working of The Project

Overview

Hand of the user is detected using OpenCV library then movements and gestures of the fingers are recorded. These recorded movements are analysed using MediaPipe library. MediaPipe plots 21 points on the hand.These points are saved and angles are calculated between them. The calculated angles are further used in coding of Arduino. Arduino is connected to Servomotors which are responsible for the moment of the fingers. The load capability of Arduino allows it to handle only 2 Servomotors which are used in the project. To overcome this problem Microcontroller PCA 9685 is used. The body of the project is made by using 10 Servomotors . The fingers of the model can rotate around the joints which are hinged in such a way that it can exactly replicate the human fingers. Threads are connected to Servomotors to cause movements of the fingers.

4.1 Software

4.1.1 Hand Detection

The image is captured using a camera. The OpenCV library is used the control the camera of the system . The captured BGR image is converted to equivalent RGB image to make it usable for the MediaPipe library. This conversion of formats is also done using OpenCV. This final RGB image is passed to the object of the MediaPipe library. The MediaPipe library plots 21 points on the hand.

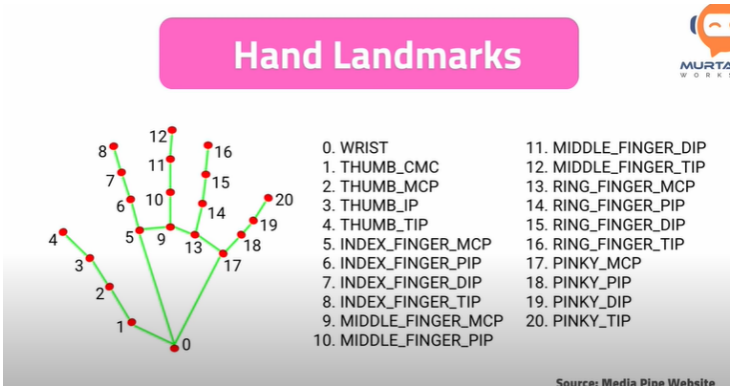


Fig 4.1

The position of the plotted points is shown in the Fig 4.1.
The coordinates of the 21 detected points are stored in a 10x3 2D array where each row stores the [x,y,z] coordinates of the points.

4.1.2 Calculations of angles

Ten angles are calculated by using the points which were plotted by the MediaPipe library .

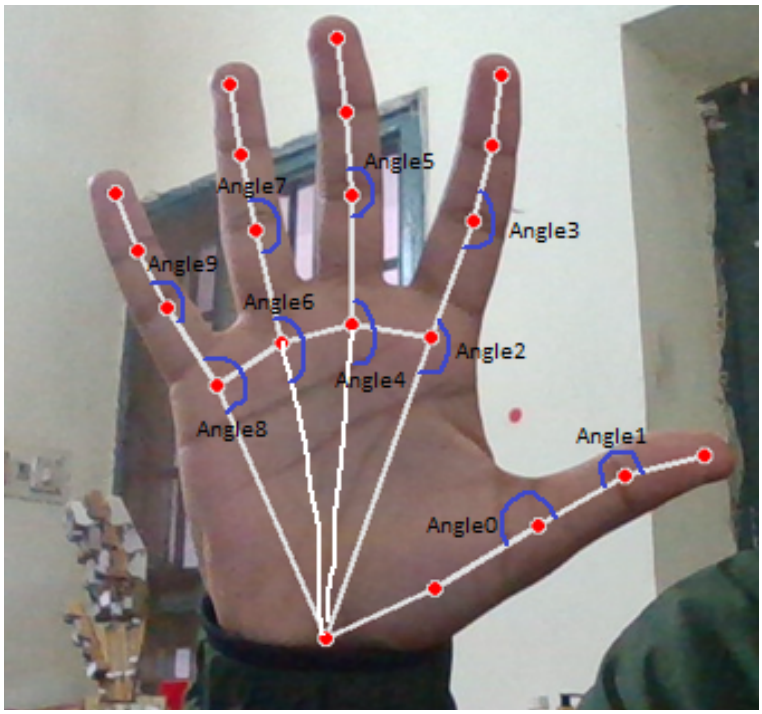


Fig 4.2

The angles which are to be calculated are shown in the Fig 4.2 .

The array of coordinates which were plotted using MediaPipe library is used to calculate all the angles.

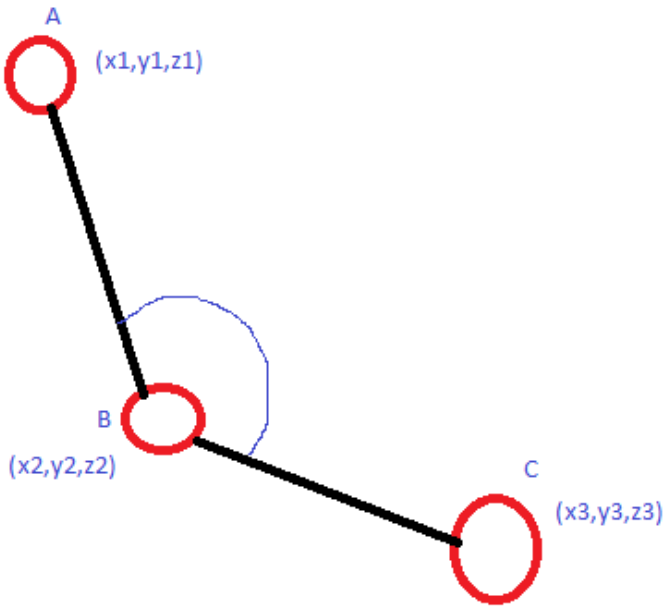


Fig 4.3

Fig 4.3 shows the angle which will be calculated .

For calculation of all the ten angles , a python module was created which uses the array of coordinates of the hand and gives a list as its output which contains all the angles which are to be calculated .

$$\text{Cos}\theta = \frac{a_1 . a_2 + b_1 . b_2 + c_1 . c_2}{\sqrt{a_1^2 + b_1^2 + c_1^2} . \sqrt{a_2^2 + b_2^2 + c_2^2}}$$

Eqn 4.1

The angle is calculated using the formula shown in Fig 4.4 where
a1 = x1 - x2
a2 = x3 - x2
b1 = y1 - y2
b2 = y3 - y2
c1 = z1 - z2

c2 = z3 - z2

4.1.3 Sending Data to the Arduino

A python module was created which uses “serial” library to communicate with the Arduino. The data transmission takes place at a baud rate of 9600.

To send the angles to Arduino, the list of angles is converted to a string of characters.

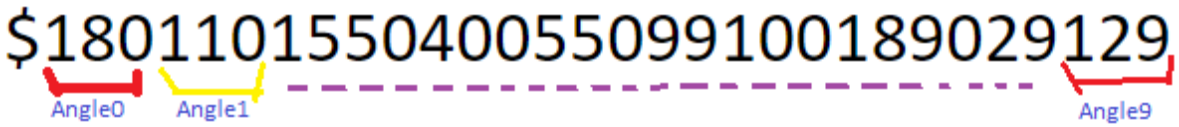


Fig 4.4

An example of generated string is shown in Fig 4.5

This generated string is then serially transmitted to Arduino using serial port.

4.1.4 Arduino Code

The string transmitted by the python code is received by the Arduino . The received string is then converted back to the list of angles of integer type. The list of angles created are then used to generate PWM (Pulse Width Modulated) signal to control the Servo motors . The generated PWM signals are then transmitted to PCA 9685 for the controlling the motors.

4.2 Hardware

The moving part of the hardware are its fingers. The fingers are constructed using cardboard cutouts. The Joints are hinged using rubber bands , tooth picks and screws. Each finger is controlled using 2 servo motors and 2 threads. Threads are connected between servomotors and the joints and are responsible for the moment of the fingers. One thread is used to control the angle of the upper two joints and the other one is used for lower joint .

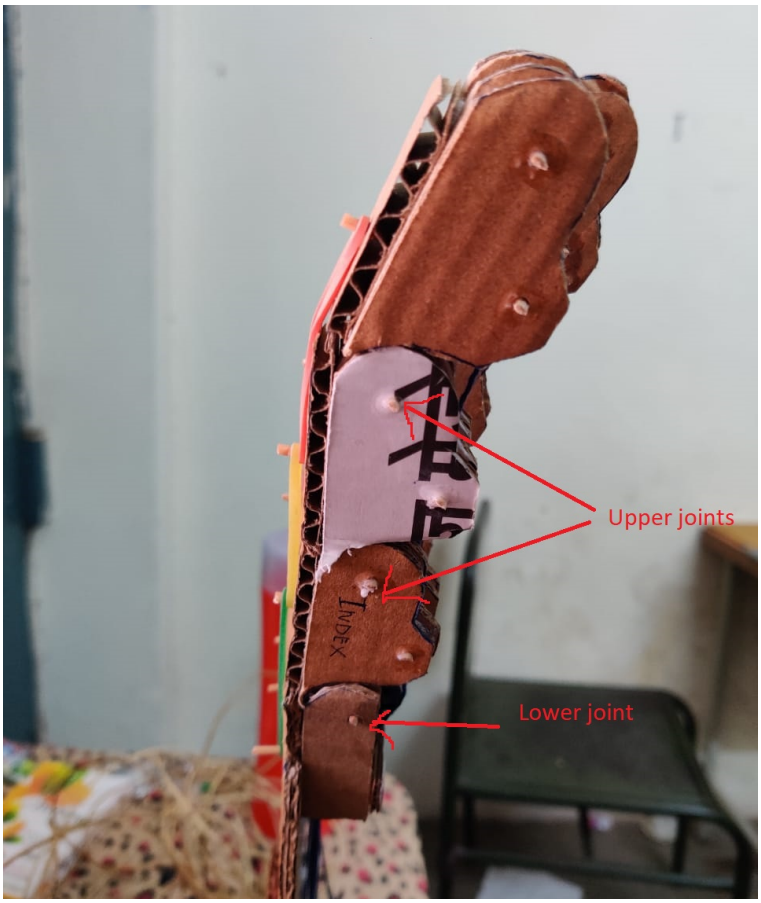


Fig 4.5

Fig 4.5 shows the upper and lower joints of the finger.

The Arduino receives the encoded string from the system then decodes it to provide instructions to the servomotors which moves the fingers with the help of threads.

Arduino is used along with PCA 9685 module to control the servomotors according to the instructions received. PCA 9685 is used due to its high current capacity and ability to control upto16 servomotors at a time which Arduino wasn't capable of doing alone .

4.3 Flow of the project

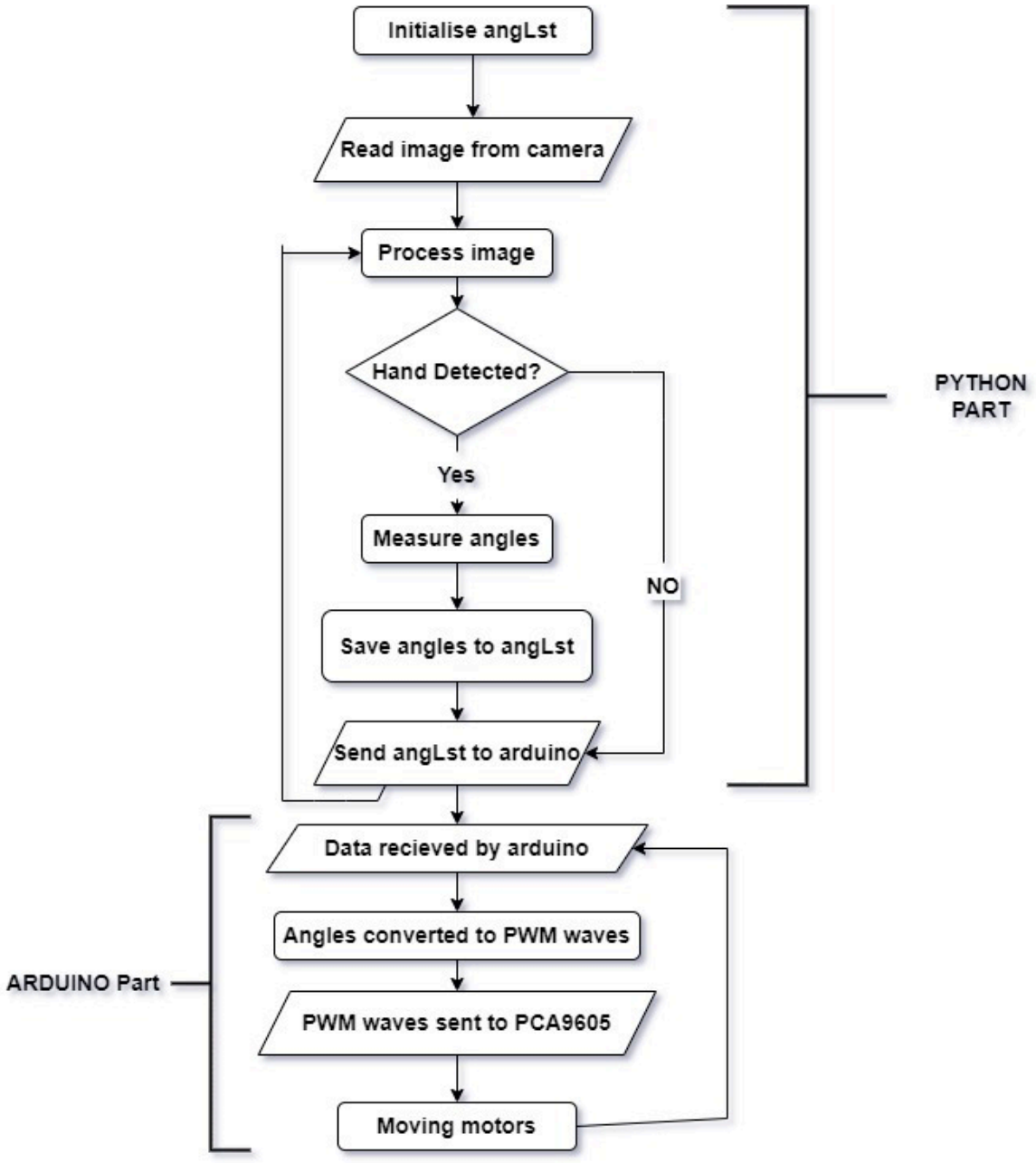


Fig 4.6

When the system starts the angle list is initialised to default. After this the image of the human hand is captured from the camera using OpenCV . Then the image is processed , if the hand gets detected angles are measured . The measured angles are saved to angle list and is sent to Arduino and if in case the hand is not detected the default list is sent back. The angles when they get received by the Arduino gets converted to PWM waves. These waves are sent to PCA 9605 and the motors get in motion according to the instructions sent to them.

CHAPTER 5
Results And Observation

Overview

The following chapter concludes whatever is learned from this project and what were our observation in the project.

5.1 Results

Complex computer vision-based gesture recognition systems are being developed that are capable of interpreting sign language in real time. Those system will be interfacing with neural networks such as Google's TensorFlow and system requirements on computer processing will be much higher because the application will be configured to operate on specialised GPUs . Considering, a low-cost gesture recognition application to be used as a control method for robotics, this project proves that simple control system can be built using computer vision.

We were able to design the software and also the hardware prototype for the mechanical hand . The software part was designed with the help of OpenCV . For the hardware prototype, Arduino and servo motors were used and for creating the body of the project , cardboard was used. We were able to record , copy and implement movements and gestures of the human hand accurately with the help of Motion understanding , gesture recognition , e estimation and Human Computer interaction (HCI). We were able to use all these Systems at one place i.e OpenCV. We used servomotors for each joint of the fingers of the hand.

5.2 Observation

The measured angles which are detected by the software are numbered as depicted in figure 5.1(a), 5.1(b).

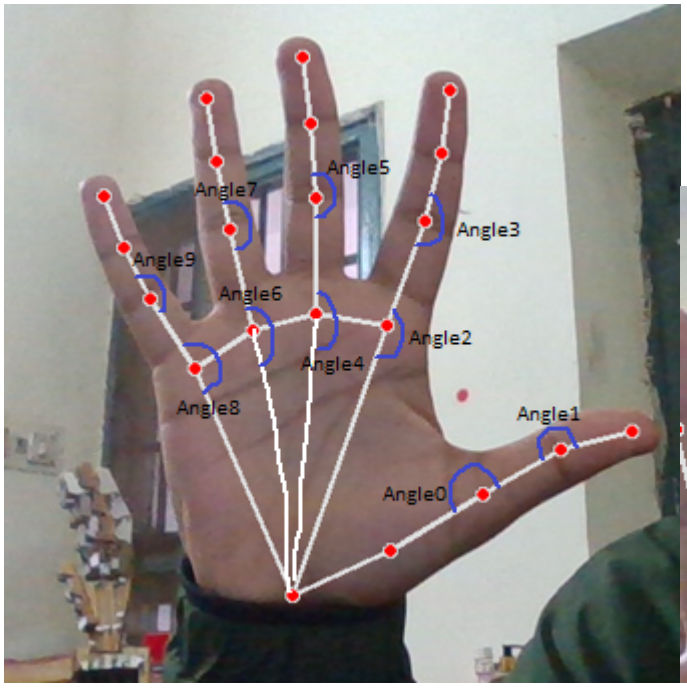


Fig 5.1(a)

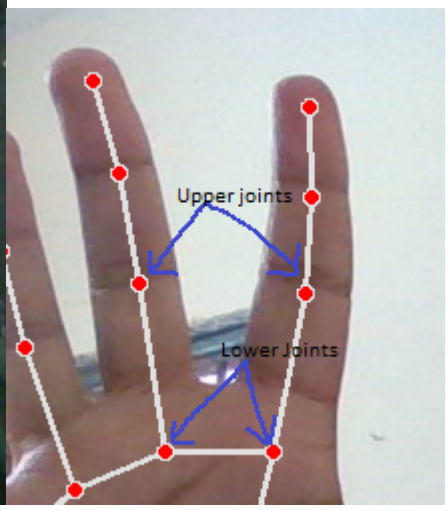


Fig 5.1 (b)

The output is displayed as a 1-d array with the syntax as follows-

AngLst =
[ThumbL,ThumbU,IndexL,IndexU,MiddleL,MiddleU,RingL,RingU,PinkyL,PinkyU]

Here ‘L’ designates the lower joint and ‘U’ designates upper joint of finger

Measured angles for different hand gestures-

1) Open palm gesture

Figure 5.2(a) shows the open hand gesture and figure 5.2(b) shows output on the terminal.

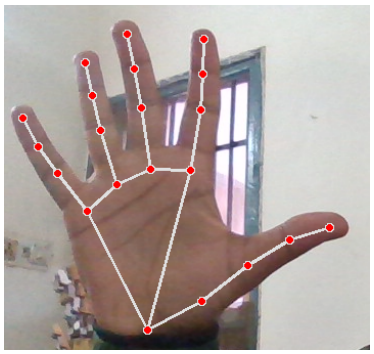


Fig 5.2(a)

```
[176, 164, 171, 169, 173, 173, 174, 175, 168, 171]
[176, 161, 171, 171, 173, 174, 174, 175, 169, 172]
[176, 164, 171, 171, 173, 173, 173, 174, 169, 173]
[175, 164, 171, 171, 173, 174, 174, 174, 169, 171]
[175, 162, 171, 171, 173, 173, 173, 175, 168, 172]
[176, 162, 171, 170, 173, 173, 173, 173, 167, 171]
[176, 162, 171, 169, 172, 174, 174, 175, 168, 170]
[176, 161, 171, 169, 173, 173, 174, 176, 169, 171]
[176, 163, 171, 171, 172, 173, 174, 174, 167, 170]
[175, 163, 170, 170, 172, 174, 173, 175, 168, 171]
[175, 162, 171, 171, 173, 173, 173, 174, 170, 173]
```

Fig 5.2(b)

2) Index finger closed gesture

Figure 5.3(a) shows the Index finger closed gesture and figure 5.3(b) shows output on the terminal.

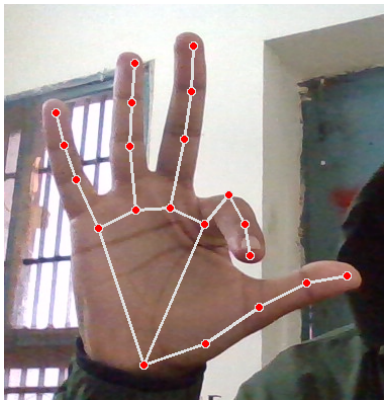


Fig 5.3(a)

```
[172, 161, 144, 109, 172, 174, 175, 172, 170, 169]
[172, 160, 143, 109, 172, 173, 175, 171, 172, 170]
[170, 163, 141, 113, 171, 173, 175, 172, 172, 169]
[172, 159, 143, 110, 172, 173, 175, 173, 171, 169]
[173, 159, 143, 111, 172, 173, 175, 173, 171, 170]
[172, 161, 142, 112, 171, 173, 175, 173, 172, 169]
[172, 162, 142, 109, 172, 174, 175, 172, 171, 170]
[172, 160, 142, 110, 172, 172, 175, 172, 171, 169]
[172, 159, 143, 112, 172, 173, 175, 173, 172, 169]
[172, 161, 142, 111, 172, 173, 175, 173, 170, 167]
[172, 161, 143, 110, 172, 173, 175, 172, 172, 168]
```

Fig 5.3(b)

3) “V” gesture

Figure 5.4(a) shows the “V” gesture and figure 5.4(b) shows output on the terminal.

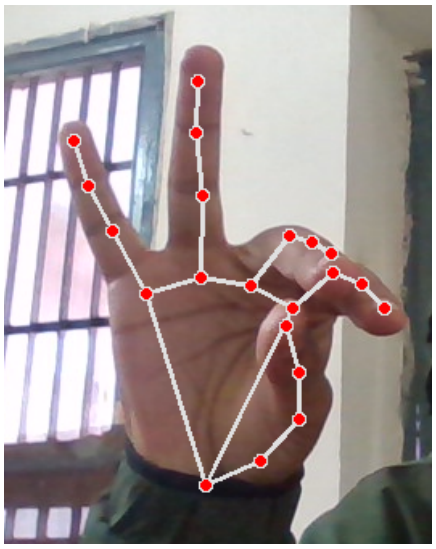


Fig 5.4(a)

```
[146, 163, 144, 131, 149, 128, 165, 174, 167, 179]
[145, 167, 142, 132, 150, 127, 164, 172, 168, 177]
[144, 164, 141, 137, 149, 130, 165, 174, 167, 177]
[142, 165, 146, 134, 150, 130, 165, 174, 168, 177]
[145, 165, 142, 132, 148, 129, 163, 174, 169, 178]
[147, 166, 140, 133, 146, 129, 164, 175, 168, 177]
[145, 167, 142, 134, 147, 131, 163, 173, 167, 177]
[145, 169, 143, 132, 148, 129, 164, 176, 167, 177]
[146, 162, 143, 136, 147, 131, 164, 174, 167, 177]
[143, 170, 143, 131, 148, 131, 165, 174, 167, 177]
[142, 169, 144, 133, 149, 131, 165, 174, 167, 178]
```

Fig 5.4(b)

4) Claw Gesture

Figure 5.5(a) shows the claw gesture and figure 5.5(b) shows output on the terminal.

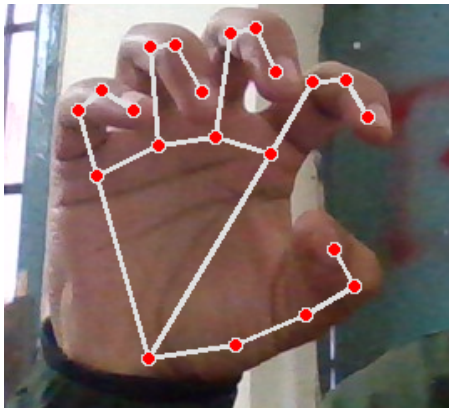


Fig 5.5(a)

```
[155, 99, 155, 143, 163, 125, 167, 132, 172, 130]
[156, 101, 155, 143, 164, 125, 169, 131, 173, 130]
[156, 102, 155, 143, 163, 125, 169, 128, 172, 130]
[156, 104, 155, 143, 163, 124, 168, 131, 171, 129]
[155, 104, 155, 142, 164, 127, 169, 130, 172, 128]
[159, 96, 154, 144, 162, 126, 167, 132, 173, 128]
[156, 103, 156, 140, 163, 122, 167, 125, 172, 129]
[159, 100, 155, 143, 163, 125, 168, 124, 171, 130]
[154, 100, 155, 141, 162, 124, 167, 129, 172, 128]
[158, 100, 155, 140, 163, 123, 168, 126, 173, 129]
[156, 100, 156, 140, 163, 123, 167, 127, 173, 129]
```

Fig 5.5(b)

Observed prototype state vs hand gestures -

Below diagrams depicts the state of the robotic arm prototype vs different hand gestures

1) Open palm gesture

Fig 5.6(a) shows the hand configuration for open palm and fig 5.6(b)

shows the output of hardware.

Fig 5.6(a)

Fig 5.6(b)

2) Closed fist gesture

Fig 5.7(a) shows the close fist configuration for closed fist gesture and fig 5.7(b)

shows the output of hardware.

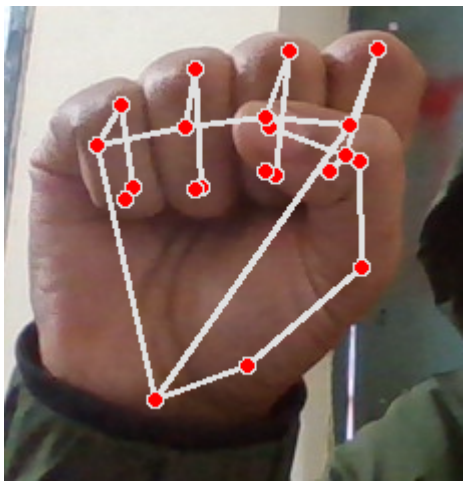


Fig 5.7(a)

5.7 (b)

3) “YO” gesture

Fig 5.8(a) shows the “YO” configuration for open palm and fig 5.8(b) shows the output of hardware.

Fig 5.8(a)

Fig 5.8(b)

CHAPTER 6
APPLICATIONS & LIMITATIONS

Overview

This chapter will lead to all the various use cases where this project can be used or may be used. It will also let us know what are the limitations of this robotic hand.

6.1 Applications

In this project the construction of a motion controlled robotic hand is considered. The aim is to build a robotic hand for which a high level of dexterity for the hand is desired.
Human hand helps us to lift objects and accounts for our grip strength. We use hand for various day to day purposes.
For this project we aimed at creating a robotic arm which can be controlled by a person according to his will. This will just act as a third hand for human beings. This project can be put into use many different places such as-

Places with unsafe work environments: places such with heavy machine works (ex- furnaces) where there is high risk of injury this arm can reduce the direct involvement of labour and instead the robotic arm can do the work and the labour can guide it with just it's hand gestures.

Precision surgery: surgeries which require pin point accuracy can be done using this robotic arm where the doctor will guide the arm

Lifting heavy weights: heavy weights which labour lift can causes them back problems so this arm can help to lift the heavy weights without injuring the labour.

War: this can also be used in war situations where a soldier can control this from back and without actually putting the soldiers directly in the battle ground thus reducing the loss of lives in a war

6.2 LIMITATIONS :

1)Lack of strength

Due to the use of cardboard , the prototype lacks strength to lift heavy objects. Prototype is also very vulnerable to moisture and external damages.

2)Inaccurate depth measurment

Current hand detection algoritham lacks the ability to measure the depth accurately . Due to this inaccuracy the error in measure measured angle increases drastically with increased dependancy on the depth.

3)Lack of proper hardware

Threads used in the prototype cannot exactly mimic the behaviour of muscles in human hands. Movement at one joint causes undesirable movement in other joint.

Arduino recieves data from computer through serial port cable which requires prototype and computer to be placed in close proxemity.This restricts the remote operating capacity of prototype.

4)Effectiveness
Current prototype only allows movement of fingers which limits it's effectiveness in some of the practical operation in current state.

CHAPTER 7
Accuracy & Future Scope

Overview
This unit will tell us about all the future aspects where this project can be used and what all improvements can be made so that we can use this project in various new fields. This chapter will also tell about accuracy of the overall project and accuracy of individual steps.

7.1Accuracy

The above project was tried to be a perfect project which could act just like a human hand. But still some loop holes are still left. Which we were not able to perfect.

The accuracy of hand detection using MEDIAPIPE was found to be 95.56%. The absolute error in measuring angles was found to be on average close to 9 degree.

When trying to detect hand gestures, cameras face challenges like background interference, image capture rate for real time processing, and 2-dimensional image representing of a 3-dimensional hand.The grip strength of the hand was very less because of the materials we used. The material used here was cardboard which was very bland and was not able to give a good grip strength to the project.

7.2 FUTURE SCOPE :

By taking inspiration from this prototype and with proper resources and technologies , full body suits can be created which will follow the instructions which will be coded in their program . These suits will be used everywhere humans can't go or is difficult for them to go . Human life will not be risked as these suits could be used in conditions which are not feasible for human bodies like in wars , space and places with extreme temperatures.

A camera can be induced in the hand module so that it can see what the hand sees this will make us remotely control the hand.

For increasing efficiency of the project we can introduce more joints in the hand which will increase the efficiency of the project and will be more versatile. But more joints will eventually make this project more complex.

The prototype if properly made and developed can be used in various purposes where human reach is not possible like

- Mining - These will be very useful to reach deep tunnels where human reach.
- Surgery-These units will be much more stable than Human hand.Surgeries would safer by the use of this.
- Farming- Farming will be more effective with these units as perfection will be attained.
- Space Exploration - Space research will be very effective with these.
- Industrial work - These will be very effective to do work at high temperatures and at places where human interaction is lethal.
- Precision work- We can use this project in places where high precision is required and by this we can eliminate the human error which can be caused in precision works.
- Clean room- This project can be further modified to be used in clean rooms also. This will make the human interaction in clean room negligible thus increasing the effectiveness of a clean room.
- Viruses-If we make a fully functional robot like this that is by inducing legs, arms etc.we can use it for delivering food and medicines affected with highly contagious virus such as ebola, corona. This will make the chances of virus being spread to the nurses very low.

Annexure

Codes-

1)Arduino code-

```
#include<Wire.h>
#include<Adafruit_PWMServoDriver.h>

#define MIN_PULSE_WIDTH 650
#define MAX_PULSE_WIDTH 2500
#define FREQUENCY 50
#define numOfValsRec 10
#define digitsPerValRec 3

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();

int valsRec[numOfValsRec];    //$0000000000
int stringLength = numOfValsRec * digitsPerValRec + 1;
int counter = 0;
bool counterStart = false;
String recievedString;

void setup()
{
  Serial.begin(9600);
  pwm.begin();
  pwm.setPWMFreq(FREQUENCY);
```

```
  for(int i=0;i<10;++i)
  {
    pwm.setPWM(i,0,toPulse(180));
    delay(500);
  }
}
void recieveData()
{
  while (Serial.available())
  {
    char c = Serial.read();

    if (c == '$')
    {
      counterStart = true;
    }
    if (counterStart)
    {
      if (counter < stringLength)
      {
        recievedString = String(recievedString + c);
        counter++;
      }
      if (counter >= stringLength)
      {
```

```
      //Serial.write("Recieved");
      for (int i = 0; i < numOfValsRec ; ++i)
      {
        int num = (i * digitsPerValRec) + 1;
                        valsRec[i] = recievedString.substring(num , num + digitsPerValRec).toInt();
        if(valsRec[i] > 180) valsRec[i] = 180;
        if(valsRec[i] < 0) valsRec[i] = 0;

      }

      recievedString = "";
      counter = 0;
      counterStart = false;
    }
   }
  }
}



int toPulse(int angle)
{
     int pulse_wide = map(angle, 0, 180, MIN_PULSE_WIDTH, MAX_PULSE_WIDTH);
   int pulse_width = int(float(pulse_wide) / 1000000 * FREQUENCY * 4096);
  return pulse_width;
```

```
}

void loop()
{
  recieveData();

  for(int i = 0;i<10;++i)
  {
    pwm.setPWM(i,0,toPulse(valsRec[i]));
  }
}
```

To Send Data to Arduino:

```
import serial
import cv2
import Calculate
import time

cap = cv2.VideoCapture(0)
mySerial = serial.Serial('COM3' ,9600 ,timeout = 1)
Cal = Calculate.controller()
sendLst = []

def releative(angle):
    return 2*(angle - 90)
```

```
while True:
    sucess, img = cap.read()
    cv2.waitKey(1)
    sendLst = Cal.generator(img)
    arduinoSend = "$"
    for i in range(0,10):
        tlst = []
        temp = sendLst[i]
        sendLst[i] = releative(sendLst[i])
        if sendLst[i]>180:
            sendLst[i] = 180
        if sendLst[i]<0:
            sendLst[i] = 0
        for j in range(0,3):
            tlst.append(sendLst[i]%10)
            sendLst[i] = sendLst[i]/10
        for j in range(0,3):
            c = str(int(tlst[2-j]))
            c = c[0]
            arduinoSend = arduinoSend + c
        sendLst[i] = temp
    mySerial.write(arduinoSend.encode())
    time.sleep(0.1)
    #print(mySerial.readline().decode('ascii'))
    print(arduinoSend)
```

Hand Tracking Module

```
import cv2
import mediapipe as mp
import time


class handDetector():
            def  _init_(self,static_image_mode=False,  max_num_hands=1,
model_complexity=1,  min_detection_confidence=0.5,
min_tracking_confidence=0.5):
      self.static_image_mode=static_image_mode
      self.max_num_hands=max_num_hands
      self.model_complexity=model_complexity
      self.min_detection_confidence=min_detection_confidence
      self.min_tracking_confidence=min_tracking_confidence

      self.mpHands = mp.solutions.hands
      self.hands = self.mpHands.Hands(self.static_image_mode,
            self.max_num_hands,
            self.model_complexity,
            self.min_detection_confidence,
            self.min_tracking_confidence)

      self.mpDraw = mp.solutions.drawing_utils

   def findHands(self, img, draw=True):
      imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
      results = self.hands.process(imgRGB)
      lmLst = []
      if results.multi_hand_landmarks:
         for handLms in results.multi_hand_landmarks:
            for id, lm in enumerate(handLms.landmark):
               h, w, c = img.shape
               cx ,cy = int(lm.x*w), int(lm.y*h)
               lmLst.append([cx, cy, lm.z*w])
            if draw:
                                    self.mpDraw.draw_landmarks(img  ,handLms,
self.mpHands.HAND_CONNECTIONS)
      cv2.imshow("Image",img)
      return lmLst
```

```
def main():
    cap = cv2.VideoCapture(0)
    detector = handDetector()
    lmLst=[]
    while True:
        success, img = cap.read()
        lmLst=detector.findHands(img)
        print(lmLst)
        cv2.waitKey(1)

if _name=="main_":
   main()
```

Calculation of angles

```
import cv2
import time
import HandTrackingModule as htm
import math
pi = 3.1415926535
detector = htm.handDetector()


def angle(c1, c2, c3):
   a1 = c1[0] - c2[0]
   b1 = c1[1] - c2[1]
   c1 = c1[2] - c2[2]
   a2 = c3[0] - c2[0]
   b2 = c3[1] - c2[1]
   c2 = c3[2] - c2[2]
        ang = (a1*a2 + b1*b2 + c1*c2)/(math.sqrt(math.pow(a1,2) +
math.pow(b1,2) + math.pow(c1,2)) * math.sqrt(math.pow(a2,2) +
math.pow(b2,2) + math.pow(c2,2)))
   ang = math.degrees(math.acos(ang))
   ang = int(ang)
   return ang

def angLst(lmLst):
   lst = []
```

```
    lst.append(angle(lmLst[1],lmLst[2],lmLst[3]))
    lst.append(angle(lmLst[2],lmLst[3],lmLst[4]))
    lst.append(angle(lmLst[0],lmLst[5],lmLst[6]))
    lst.append(angle(lmLst[5],lmLst[6],lmLst[7]))
    lst.append(angle(lmLst[0],lmLst[9],lmLst[10]))
    lst.append(angle(lmLst[9],lmLst[10],lmLst[11]))
    lst.append(angle(lmLst[0],lmLst[13],lmLst[14]))
    lst.append(angle(lmLst[13],lmLst[14],lmLst[15]))
    lst.append(angle(lmLst[0],lmLst[17],lmLst[18]))
    lst.append(angle(lmLst[17],lmLst[18],lmLst[19]))
    return lst



class controller():

    def _init_(self):
        self.currLst=[180,180,180,180,180,180,180,180,180,180]
        self.prevLst = self.currLst

    def generator(self,img):
        lmLst = detector.findHands(img)
        if len(lmLst):
            self.prevLst = self.currLst
            self.currLst = angLst(lmLst)
        return self.currLst


def main():
    cap = cv2.VideoCapture(0)
    control = controller()
    pTime = time.time()
    cTime = pTime
    while True:
        success, img = cap.read()
        cv2.waitKey(1)
        print(control.generator(img))

if _name_ == "_main_":
    main()
```

References

[1] Survey on Design and Development of competitive low cost Robot Arm With Four Degrees of Freedom by Ashraf Elfasahany. https://file.scirp.org/pdf/MME20110200004_76841505.pdf
[2] A survey on Arduino Controlled Robotic Arm by Ankur Bhargava. https://www.researchgate.net/publication/321990760_Arduino_controlled_robotic_arm
URN: urn:nbn:se:kth:diva-230144, 2018.
[3] Review on development of industrial robotic arm by Rahul Gautam https://www.researchgate.net/publication/322926656_Review_on_Development_of_Industrial_Robotic_Arm.
[4]Ranjith Kumar Goud and B. Santosh Kumar. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.676.5617
[5]Arpit Sharma, Ritesh Verma, Saurabh Gupta, Sukhdeep Kaur Bhatia.https://www.jetir.org/papers/JETIR2006065.pdf
[6] Arduino, What is arduino?, https : / / www . arduino . cc / en / guide / introduction, [Online; accessed 03-February-2020].
[7] ——, Arduino Uno Rev3, https://store.arduino.cc/arduino-uno-rev3, [Online; accessed 03-February-2020].
[8] ——, Arduino Nano, https://store.arduino.cc/arduino-nano, [Online; accessed 25-March-2020].
[9] TowerPro, MG996R Robot servo 180° Rotation, http : / / www . towerpro .
com.tw/product/mg995- robot- servo- 180- rotation/, [Online; accessed 07-February-2020].
[10] J. Gieras, PERMANENT MAGNET MOTOR TECHNOLOGY: DESIGN
AND APPLICATIONS. Jan. 2010, isbn: 978-1-4200-6440-7.
[11] M. Barr, “Pulse width modulation”, Embedded Systems Programming, pp. 103–
104, 2001.
[12] N. Semiconductor, nRF24 Series, https://www.nordicsemi.com/Products/

[13] R. Iyer, nRF24L01+ RF Module Tutorial, https://www.deviceplus.com/
how- tos/arduino- guide/nrf24l01- rf- module- tutorial/, [Online; accessed 07-February-2020], 2017.

L. minute engineers, How nRF24L01+ Wireless Module Works & Interface
with Arduino, https : / / lastminuteengineers . com / nrf24l01 - arduino - wireless-communication/, [Online; accessed 07-February-2020].
[15] S. Symbol, FLEX SENSOR FS, https://www.mouser.se/datasheet/2/381/Spectra_flex22-1203810.pdf, [Online; accessed 07-February-2020].
[16] (). Voltage dividers. [Online; accessed 27-May-2020], [Online]. Available:
https://learn.sparkfun.com/tutorials/voltage-dividers.
[17] Arduino, MPU-6050 Accelerometer + Gyro, https://playground.arduino.
cc/Main/MPU-6050/, [Online; accessed 11-February-2020].
[18] InvenSense, MPU-6000and MPU-6050 Product Specification Revision 4.2,
https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf, [Online; accessed 12-February-2020], 2013.
[19] Dejan, Arduino and MPU6050 Accelerometer and Gyroscope Tutorial, https:
//howtomechatronics.com/tutorials/arduino/arduino- and- mpu6050-accelerometer-and-gyroscope-tutorial/, [Online; accessed 22-January2020].
[20] K. Tuck, Tilt Sensing Using Linear Accelerometers, https://www.thierrylequeu.fr/data/AN3461.pdf, [Online; accessed 27-March-2020].
[21] G. Langevin, Inmoov-open source 3d printed life size robot, http://inmoov.
fr/, 2014.
[22] M. Kazi and M. Bill, Image by authors, 2020.
[23] KTH, Maskinelement: Handbok. Avd. för maskinelement, Institutionen för
maskinkonstruktion, Tekniska högsk., 2008.
[24] maniacbug, RF LIBRARY, https://github.com/maniacbug/RF24, [Online;
accessed 18-March-2020].
[25] Fritzing, version 0.9.3. [Online]. Available: https://fritzing.org/home/.