Smart Robotic Arm

Project Report submitted in the partial fulfilment

Of

Bachelor of Technology

In

Mechatronics Engineering

By

Miheer Diwan (H016)
Shubhankar Kulkarni (H024)

Under the supervision of

Prof. Dattatray Sawant
(Assistant Professor, Mechatronics, MPSTME)

Prof. Amey Raut
(Assistant Professor, Mechatronics, MPSTME)

SVKM's NMIMS University
(Deemed-to-be University)

MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT & ENGINEERING
Vile Parle (W), Mumbai-56

2021-22

CERTIFICATE

This is to certify that the project entitled **"Smart Robotic Hand"**, has been done by **Mr. Miheer Diwan** and **Mr. Shubhankar Kulkarni** under my guidance and supervision & has been submitted in partial fulfilment of degree of Bachelor of Technology in Mechatronics of MPSTME, SVKM's NMIMS (Deemed-to-be University), Mumbai, India.

Dattatray Sawant	**Amey Raut**	Examiner
Assistant Professor, Department of Mechatronics, MPSTME	Assistant Professor, Department of Electronics and Telecommunication Engineering, MPSTME	

Date:

Place: Mumbai

Dr. Venkatesh Deshmukh

(HOD Mechatronics)

Dr. Alka Mahajan

Dean, MPSTME, SVKM's NMIMS

(Deemed-to-be University)

ACKNOWLEDGEMENT

We want to thank our college, SVKM's NMIMS Mukesh Patel School of Technology Management & Engineering, the Dean, Dr. Alka Mahajan and the HOD of Mechatronics, Dr. Venkatesh Deshmukh for giving us the opportunity to work on this project.

We would also like to thank our project guides, Prof. Dattatray Sawant and Prof. Amey Raut for their constant guidance and support shown towards this project. Their knowledge and insight have helped us throughout this project.

Furthermore, we would also like to show our sincere gratitude to our parents and peers for their support and trust without which this project would not have been possible.

Miheer Diwan
B Tech. (MTRX)
Roll No.- H016

Shubhankar Kulkarni
B Tech. (MTRX)
Roll No.- H024

ABSTRACT

Humans working in hazardous work environments are prone to accidents which causes loss to human life. These fields not only endanger workers, but also damage property if not done properly. For example, in a radioactive facility, even if highly skilled professionals take charge of the experiment, there is a slight possibility of fumbling or doing something undesirable under pressure. Therefore, a device needs to be created to aid the user to do the same, without the risk of harming the human. This project deals with the development of a multipurpose robotic arm, which is operated in two modes. Firstly, the autonomous mode, which can accurately calculate the location of an object in 3D space, following the picking up of the object. The arm can vary the grip strength based on the weight and the material of the detected object. Secondly, the gesture control mode, which deals with controlling the arm using hand gestures. These hand gestures are fast and can perform the task in real time with high accuracy. These functions have various applications, one of them being manipulating objects in hazardous environments without human intervention.

Table of Contents

Topics	Page
List of Figures	**4**
List of Tables	**5**
Chapter 1 Introduction	
1.1 Background	**5**
1.2 Motivation and Scope	**5**
1.3 Salient Contributions	**6**
1.4 Advantages, Limitations and Applications	**6**
1.5 Organization of Report	**6**
Chapter 2 Literature survey	**8-12**
Chapter 3 Problem Definition	**13**
Chapter 4 Methodology	**14**
Chapter 5 Mechatronic System Analysis	**15**
5.1 Mechanical Design	**15 – 21**
5.2 Programming Analysis	**23 – 29**
5.3 Trajectory Generation	**30 – 32**
5.4 Drive System	**33 - 33**
Chapter 6 Testing and Results	**33 – 34**
Chapter 7 Advantages, Limitations and Applications	
7.1 Advantages	**35**
7.2 Limitations	**35**
7.3 Applications	**35**
Chapter 8 Conclusion and Future Scope	**36-37**
References	**38 - 40**
Appendix A: Soft Code Flowcharts	**41**
Appendix B: List of Components	**42**

List of the Figures

Fig. No.	Name of the figure	Page No.
1	Concept Drawing	16
2	Proposed Model Concept	17
3	Early Model	17
4	Finger	18
5	Thumb	19
6	Palm	19
7	Top View of the Forearm with Servo Motors	20
8	Base Assembly	21
9	CAD Model of Robotic Arm	22
10	Final Assembly	22
11	Extended Finger	23
12	Curled Finger	24
13	DLT Calculation	28
14	Stereo Calibration	29
15	PID Calculations PI	30
16	PID Calculations P	30
17	Timing Calculation for Trajectory	32
18	Updating Angles	32
19	Switched Mode Power Source	33
20	LM 2596 Buck Converter	34
21	Arduino PCB	35
22	Cable Management PCB	36
23	Arm-pose Estimation	37
24	Hand-Pose Estimation	37
25	Reaching the Can	38
26	Lifting the Can	38
27	Gesture-Controlled Mode	45
28	Autonomous Mode	45

List of the Tables

Fig. No.	Name of the figure	Page No.
1	Trajectory Generation	32

Chapter 1
Introduction

1.1 Background

Da Vinci designed the first sophisticated robotic arm in 1495 with four degrees-of-freedom and an analog on-board controller supplying power and programmability. von Kemplen's chess-playing automaton left arm was quite sophisticated. Unimate introduced the first industrial robotic arm in 1961, it has subsequently evolved into the PUMA arm. Over the years, humans have realized that robotic arms with sophisticated designs based on human hands, have a broader range of applications and are more dexterous. Since, robot hands are fundamental for performing tasks in several industries, extensive research has been done on them to make the robots as versatile as possible.

1.2 Motivation and Scope

This project was proposed to aid people in hazardous tasks, as well as other applications such as medical robots. Thus, this project deals with two fundamental problems which can be resolved using dexterous robots. The first problem encountered was eliminating presence of humans in hazardous environments but allowing the human to assume control to do a task. This project aims to eliminate this problem by introducing a gesture control mode, which provides the user to control the arm from a distance, without the user wearing any equipment. It lets the user control the hand freely with the help of cameras.

The second problem was the accuracy of the task done by the person. To solve this problem, the arm has an autonomous function, which tells the user to choose the object required and the arm will detect the object in 3D space. After detecting the object, the arm will pick up the object and control the grip strength based on the material and weight of the object. The arm can carry out the desired function several times, which can either be specified by the user, or can just place the object at a designated location.

1.3 Salient Contribution

- Carried out degree of freedom analysis of the system.
- Developed multiple CAD models for the proposed robotic arm.
- Calculation of Inverse Kinematics, Camera Triangulation and Trajectory generation were performed using a camera.
- Programming the bot for gesture-controlled mode.
- Performed extensive literature survey on humanoid robotic hands.

1.4 Advantages, Limitations and Applications

Applications:

- The robot can be used in the autonomous mode in hazardous environments like zones with radiation pollution etc.
- Gesture controlled mode is advantageous in situations where humans cannot be directly involved, and robots are not trained enough to perform the task. E.g., Manufacturing chipsets in the industry

Limitations:

- The robotic arm has less degrees of freedoms thus limiting its functionality.
- Area of effect of the hand is significantly reduced due to the scale of the proposed model.
- Depending on the type of motor used and the material of the robot, only specific objects can be carried and interacted with.

1.5 Organization of Report

This report introduces the topic of our project, followed by the literature survey done in this field. Furthermore, the problem statement is briefly described along with our solution for the same. We discuss in depth about the components and the working of the project. Finally, we also present some data obtained from extensive testing and simulation. To conclude, we wrap it all up with the future scope of this project.

Chapter 2

Literature survey

Manipulation of objects in the real world is an imperative feature. This enables us to displace objects within a given space without human interaction. For this to be possible, we need to take into consideration many factors, since these are the fundamentals of the robot. It is imperative to know the location of the object in the workspace since we need to reach the object space from the robot space [25]. Therefore, finger pointing detection is implemented to teach locations or objects. We determine the finger velocity and using this data we can determine the direction the finger is pointing to. Using this achievement, we can detect objects in 3D Space and help detect the object in the environment for picking and grasping.

Previous attempts at creating robot hands did not satisfy dexterous and versatile grasp tasks [19]. General imitation of a robot is much more difficult due to the motion constraints offered by human hands. A hybrid hand model was presented, which had ambidexterity. The KU hybrid hand can switch modes between human hand mode and conventional hand mode, which allowed the hand to grasp spherical objects with ease. However, it can only grasp objects up to a maximum 80mm radius, 60mm for the conventional hand mode. Moreover, LabView based stepper motor control robotic arm for medical procedures aimed at increasing the accuracy and the simplicity in control [9]. The individual angles of the motor were controlled by the user via LabView to effectively control the motion of the robotic arm.

Imitation of human gestures using neural networks is also being implemented, wherein we use programs like C++ to operate libraries such as OpenCV [21]. Depending on the gestures detected by the program, a crude effort was made to replicate these motions on a 3D printed hand. This hand was able to imitate the dexterity of a hand up to a certain level, since the program was only able to detect hand signs between the numbers 0 to 5 included. This is one of the limitations encountered.

Upon further research, a more sophisticated model of the hand was created, which was used to aid during laparoscopy [20]. HALS enables effective visual development by handling large organs smoothly by one hand, and it provides surgeons with the time required to master their skills. However, this instrument has some flaws. Therefore, a 3 finger 5 DOF hand was suggested, but was later discarded, since the difference in the number of fingers and the range of the fingers was an impairment. Hence, a 5-finger hand was proposed which grasped large organs during the surgery. This hand is limited to delivering a grasping force up to 30N, since it was required for grasping organs.

Vision based detection of hand gestures and movement were implemented [23]. This enables movement detection systems using two standard USB cameras. The hand position will decide the task allotted to the robot. Robot positions can be determined by these hand gestures. Softwares such as MATLAB were used to implement this. Therefore, this was used to detect hand gestures, which enabled disabled people to interact with the environment. Furthermore, a gesture controlled robotic arm was created for the improvement of the efficiency of robotic arms [13]. The arm gesture is recognized by an OpenCV model and promotes increase in the speed of actualization as well as accuracy and precision.

A similar approach was conducted using certain filters and programs to recognize hand gestures and determine the gesture pattern using the PCA algorithm [22]. This enables us to determine the direction the robot is supposed to move in. This model was tested on PUMA 762 robotic model. This can be implemented for complex robots as we see in future papers. Advancements in computer vision and robotics have aided to develop these robots and make them robust.

For preventing human intervention in hazardous environments, we can use robots which can do the work for us [3]. Since there are a lot of workplace accidents in hazardous environment, it is better to keep the human intervention to a minimum. We can, however, implement an external robotic hand which will mimic the user’s gesture. This will enable the workers to work from a remote location without physically interacting with the environment. Moreover, replacement of humans with mobile robots for tasks such as bomb detection in military sectors as well as in chemical industries for handling radioactive substances was done in [10]. Implementation of trackers makes the robot suitable for rugged terrains, but it is limited to the range of its Bluetooth sensor.

To decrease human involvement, gesture controlled robotic system is used to mimic user's movement inside the reactor for performing a certain task such as replacement of control rods [3]. The use of flex sensors along with connections to an Arduino enabled the user to do the same. This task will only decrease the effort required by the human to do the work whilst promoting safety of the worker. Furthermore, experimentation in deployment feasibility of robotic systems for Deactivation and Decommissioning tasks associated with hazardous nuclear site clean-up took place in [1]. Modular 17 DOF dual armed robot using multiple tools is used for the manipulation of heavy objects in a hazardous environment. It consists of a Magellan and a Space Ball controller. Successful experimentation results led to the prototyping of fully anthropomorphic 17 DOF dual robotic system.

We can also use mobile robots to perceive hazardous environments, so that we are able to automate tasks and eliminate human interaction [10]. An arm can be placed on the robot to interact with the environment and achieve certain tasks that are dangerous for humans. The outer build can withstand the environment. Also, it is imperative to see the speed and the range of the robot, which can be counted in one of the drawbacks of the robot. Similarly, a robotic arm was created for defuse bomb by synchronizing the arm movement with the movement of a bomb defusal officer who can control it from a distance [18]. It has a drawback of the distance it can travel, as well as the amount of rotation done by the robot.

To make the robot hand more efficient, we introduced haptic feedback to the user, which would allow the user to feel the object being picked up by the user [24]. This was also done in [14]. To make things easier, a decision of hand gesture controlled robotic hand was made. This would use smart watches and Bluetooth. Wireless transmission between the sender and receiver took place, allowing the accelerometer in the smart watches to assign a motion after a certain criterion is achieved. Similar research was conducted in [15], which used Brain Computer Interface (BCI) and implemented steady-state visual evoked potential (SSVEP) based BCI to transport objects to its desired destination. Brain signals are captured from the user by targeting the circles on a field, which is assigned a location and then the goal location is mentioned. This model does not require a lot of training, which is efficient in operation.

To avoid physical creation of the robot, simulation of robot in V REP without physically developing the robot can be done [12]. The main objective of this work was the development of a robotic prototype solution based on simulation that minimizes the radiation exposure of technicians in the nuclear medicine laboratories. Repeatability and accuracy are one of the key traits that make it a perfect fit for this field.

Training of individuals for the operation of heavy equipment was analyzed in [17]. Here, they researched how novice human operators can rapidly learn to control modern robots to perform basic manipulation tasks; also, how autonomous robotics techniques can be used for operator assistance, to increase throughput rates, decrease errors, and enhance safety. Humans are compared against autonomous machines for effective understanding. Human performance while operation was successfully evaluated. It was shown how autonomous control can reduce workload on an individual.

Some robots can also be used to train people [8]. One such example is the Biomedik Surgeon. This facilitates surgical specialties and techniques. This trains the user rom making an incision to creating a dissection. Such robots use an emulator which can help understand the integrities of surgical processes and the method used to implement them. In addition, low-cost robotic arm for assisting the doctor during operations as well as body checks was implemented to reach astute conclusions [5]. This combines servo motors with gain feedback to correctly perform a task.

Similar research was done at Korea Atomic Energy Research Institute for the development of remote robotic systems for use in a nuclear environment. Several robotic systems were described for using them in the DFDF (DUPIC Fuel Demonstration Facility) and PRIDE (PyToprocess Integrated inactive Demonstration facility). Remote clean-up in the nuclear plants was the major application they focused on throughout their research work to reduce personnel exposure dose rate and improve worker safety.
For assistance of robots for rehabilitation of patients, the process was successfully carried out by interpreting 3D kinetic and kinematic properties of a person's

movement with an arm and evaluates the performance of a recovering person [7]. This also analyses the trajectory of the hand, thereby aiding in better feedback. The difficulties of the exercises can also be adjusted by the preference of the physician.
Additionally, development of Interventional radiology (IVR) to avoid radioactive exposures to doctors was done. A parallel link mechanism is therefore suggested to overcome this problem. A mechanism like the Stewart Gough platform is created for the same for solving the problem. PD controller is implemented with the addition of torque feedback, which resulted in effective movement of the end effector.

Likewise, a 4-degree robotic arm with a vision system for OTOROB (Orthopedic Robot) was created [4]. This system enables the doctor to target a robotic arm on the patient. This promotes remote surgery with an increase in telecommunication systems. A flexible robotic system is proposed, which enhances the video capturing abilities.
Such steps in the field of Robotics have been taken over the past decade and these fields help us to manipulate objects in real time, since it is sometimes dangerous to have human intervention. All these processes contribute to the development of gesture-controlled robots and their various applications in different sectors. Additionally, robots were made for the purpose of self-feeding assistance for people with severe disabilities [6]. A 7 DOF arm is constructed by using technologies like Arduino mega. Simulations were performed on the arm for the effective performance of the end effector in ideal conditions. Eating food was successfully transferred using the 7 DOF arm. More than 80 percent of the food was successfully transferred.

Another research dealt with concepts seen in [6], which used effective four bar linkage with appreciable joint clearances [16]. This four-bar linkage uses CATIA and MATLAB software. In this paper designing of a robotic arm that can be applied in biopsy for diagnosis of cancerous and tumor cells by CT Image was done. Due to vibration present in the joints, the positioning of the end effector was not accurate.
All papers conclude to the idea that the project idea has not been considered before, i.e., there has not been a combination of two concepts together. This project is a multipurpose arm which can solve problems effectively. It also includes solving problems such as completion of hazardous tasks with or without human intervention but eliminates human presence.

Chapter 3
Problem Definition

To develop a smart robotic hand with dual modes of operation: Gesture Controlled Mode (GCM) and Autonomous mode. In gesture-controlled mode, the robot will track the user’s action in real time and mimic them to perform a specific task. In autonomous mode, the robot will identify objects, determine their location in 3-D space, and interact with them.

Chapter 4
Methodology

Implementation of gesture control on the robot is done using a linear approach. Firstly, the kinematics of the hand are calculated, giving us equations, which can be directly implemented in code. Secondly, the image of the camera is analyzed via a library called OpenCV. This image is passed onto a trained model to detect the landmarks present in the hand. These landmarks will be denoted by dots and these dots will relate to lines using OpenCV. Furthermore, this will also be done for the arm since the project entails the same. After the detection of landmarks, these landmarks can be passed on to the kinematics of the hand, feeding in the x and y coordinates of the hand.

Feeding the coordinates of the hand will let us calculate the angle made by the bottom most landmark and the entire finger. This angle can be fed to a servo motor code, which can change its degree of rotation as per the received angle. Moreover, the same process is done for the arm. The joint angles are calculated by the same kinematic code, leading us to get the joint angles. These angles are fed to a motor driver code to carry out the desired function.

Prior to this, these coordinates are also sent to a simulation program in MATLAB, which can tell us the feasibility of the robot. Thus, we can accurately predict the functioning of the model and propose viable solutions to problems we encounter in the simulation.

In autonomous mode, the first problem we need to tackle is determining the weight that the robot can lift without taking too much strain. All other parameters of the robot will be tuned in accordance with the max. weight that it will be capable of lifting.

The next step of the process will focus on object detection and identification in the 3-D space. Since the area of effect of the robot is quite small, the objects must be placed within a certain area. Two cameras (one on the base of the robot and one above the object area) will be utilized for this task. After the object is located the robot will reposition itself and pick up the object.

Chapter 5
System Analysis

5.1 Mechanical Design

To develop a dexterous robotic arm, it was necessary to have a well-designed mechanical system that mimics the human arm. Considering the complexity of replicating and producing a biomimetic human arm, we decided to exclude the abduction and adduction of the fingers and the thumb. This allowed us to reduce the complexity of the model without compromising too much on the functionality. The entire model of the proposed arm was developed from scratch using the software Solidworks and can be 3D printed with ease.

This project entails the design and manufacture of a robotic arm with 13 DOF mounted on a stationary frame. The proposed robot will have 4 fingers actuated with the help of high-tension cables connected to servo motors. The hand itself is connected to an elbow joint, which moves in the XY plane, which is further linked to the shoulder joint which moves in the XZ plane. The elbow and shoulder joints only exhibit rotational motion in their specific planes of motion. They are actuated with the help of high torque DC motors. The frame also mounts 2 cameras: one on the base and one on top, parallel to the ground. These cameras will help determine the exact location of the object placed in front of the robot. These coordinates will be relayed to the robot.

5.1.1 Early Iterations & Concepts Drawings

After performing extensive literature survey, we were able to get an idea of how a robotic hand works, the degrees of freedoms required for dexterous movement and the different actuation methods used for moving the fingers. The idea was to develop a robotic hand which consisted of with three fingers, one thumb and lacked a wrist joint. However, we later decided the add joint in the wrist as well as a dedicated joint for the base of thumb to provide better stability to the hand and to tackle problems while picking objects from the ground. We decided to actuate the robot with artificial tendons as it is a very efficient and dependable method used by many commercial prosthetics developers.

The idea was to connect high tensile strings at the tips of the fingers and connect them to servo motors placed in the forearm by means of a standard servo horn. Servo motors were chosen as they require less space, provide high torque, and can be programmed with ease. We decided to place the servo motors in the forearm instead of the palm as it would keep the surface of the palm free and thus give us more space to grasp objects. When the servo motor is actuated, the artificial tendons (i.e.,
high tensile strings) are pulled and curl the fingers inwards the palm.

Images of some early iterations of the CAD models and concept drawings are attached ahead, which outline the key ideas used in the final product.

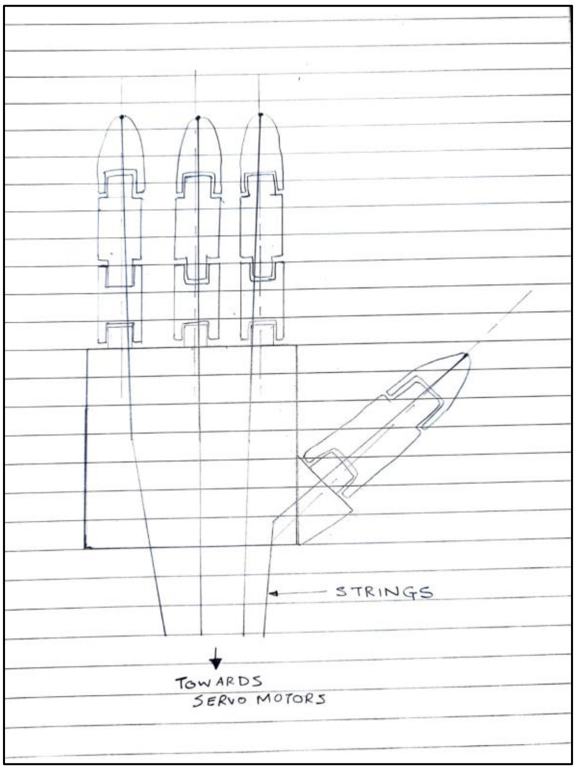


Figure 1: Concept Drawing

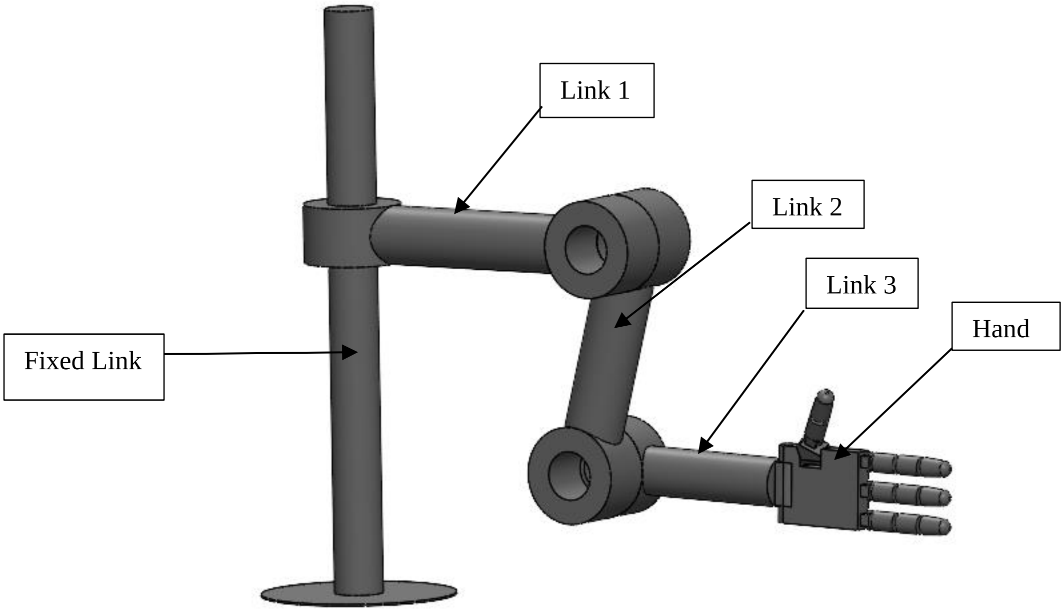


Figure 2: Proposed Model Concept

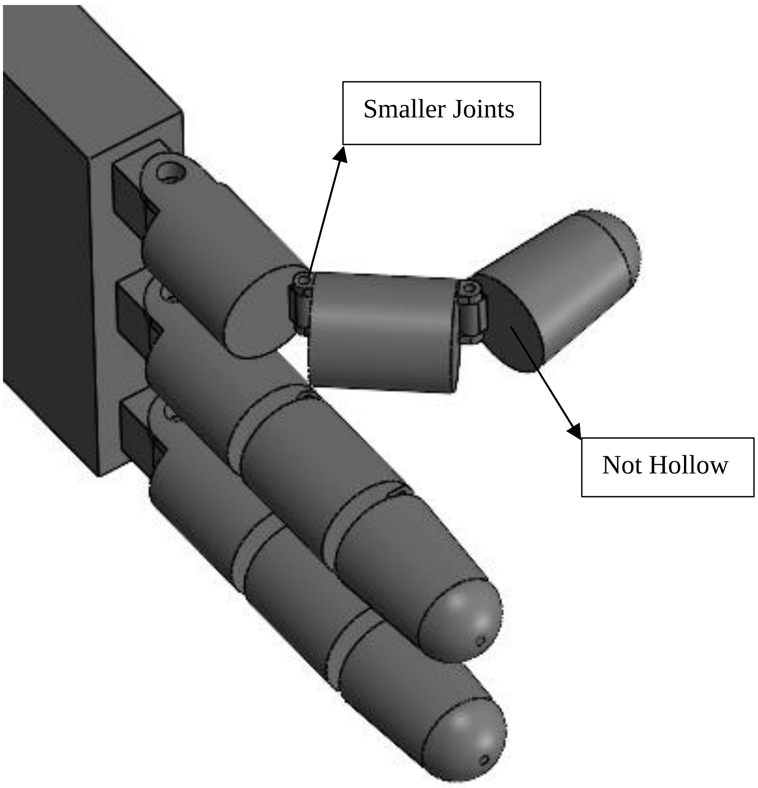


Figure 3: Early Model

5.1.2 CAD Model Analysis

The final CAD model was developed as a result of extensive prototyping and collecting data. Major problem encountered while developing the model was the weight of the hand. To achieve this, the links of the fingers, the palm and the forearm were made hollow. This helped reduce the overall weight of the mechanism to a mere 1.6 Kg overall We also had to ensure that the components designed were able to take the load and did not break under stress as most of the parts were 3-D printed using the material ABS. Stress analysis was performed on the parts using Solidworks which gave us an accurate estimate of the load bearing capacity.

1. **Fingers**

All fingers in the mechanism are of the same size and each finger consists of three links which are connected by means of dowel pins. The links have rotational freedom about these pins. The third link is connected to the palm in a similar manner. The final model of the finger showcases larger joints and a hollow body which reduces the weight of the mechanism. Each link also has a circular hole running down the middle of one half of the link. This groove is used to pass the high tensile thread to the palm and finally connect to the servo motors.

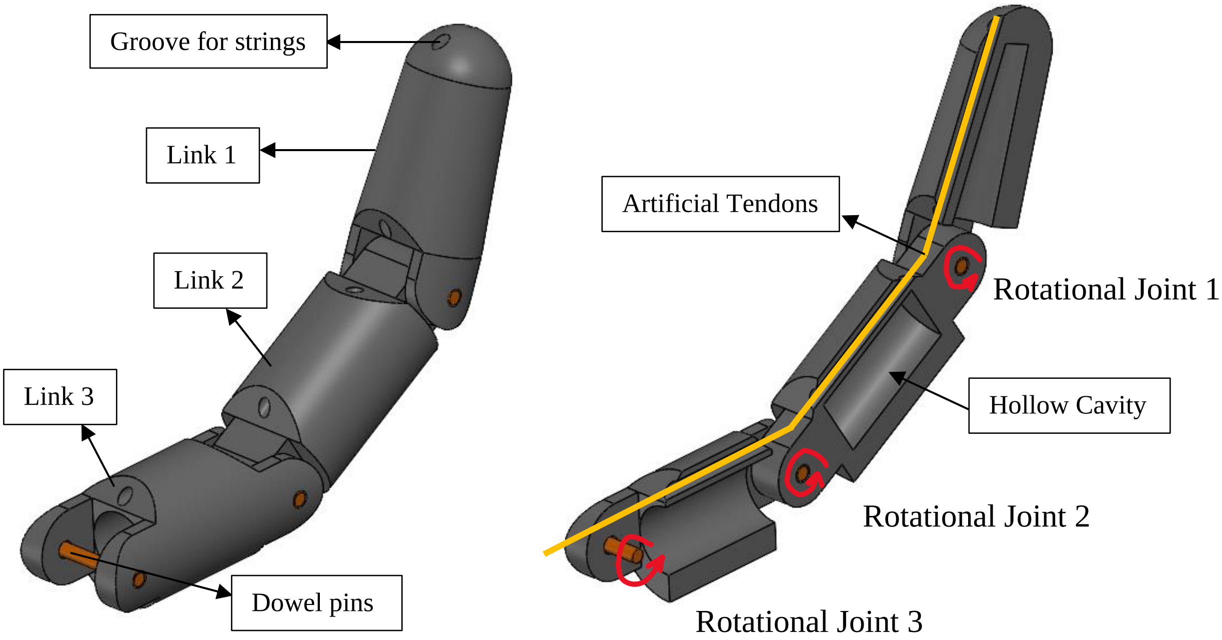


Figure 4: Finger

2. Thumb:

The thumb is similar in design to the other fingers and is actuated in the same way. However, it has two degrees of freedoms. An additional servo is fitted inside the palm which give more mobility to the thumb and helps to provide a better grip while grasping objects.

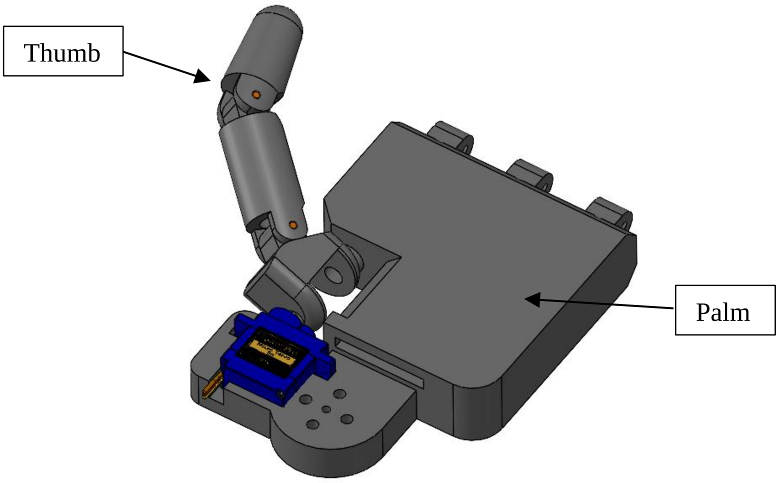


Figure 5: Thumb

3. Palm:

The palm houses the fingers and the thumb as well as a servo motor. The flat surface of the palm is useful for gripping objects securely. The palm also has dedicated grooves and slots for the artificial tendons of each finger. The base of the palm has a circular slot in its surface to accommodate a servo horn used to mount the hand mechanism onto the forearm. This servo motor also acts as the wrist joint in the assembly.

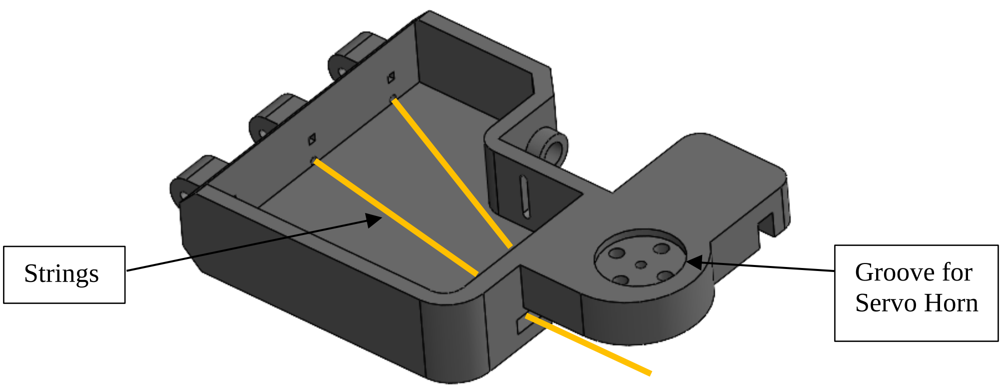


Figure 6: Palm

4. Forearm:

The forearm houses five servo motors. One of the five servo motors acts as the wrist joint while the other four are mounted on the forearm by means of a plate and are used for actuating the fingers.

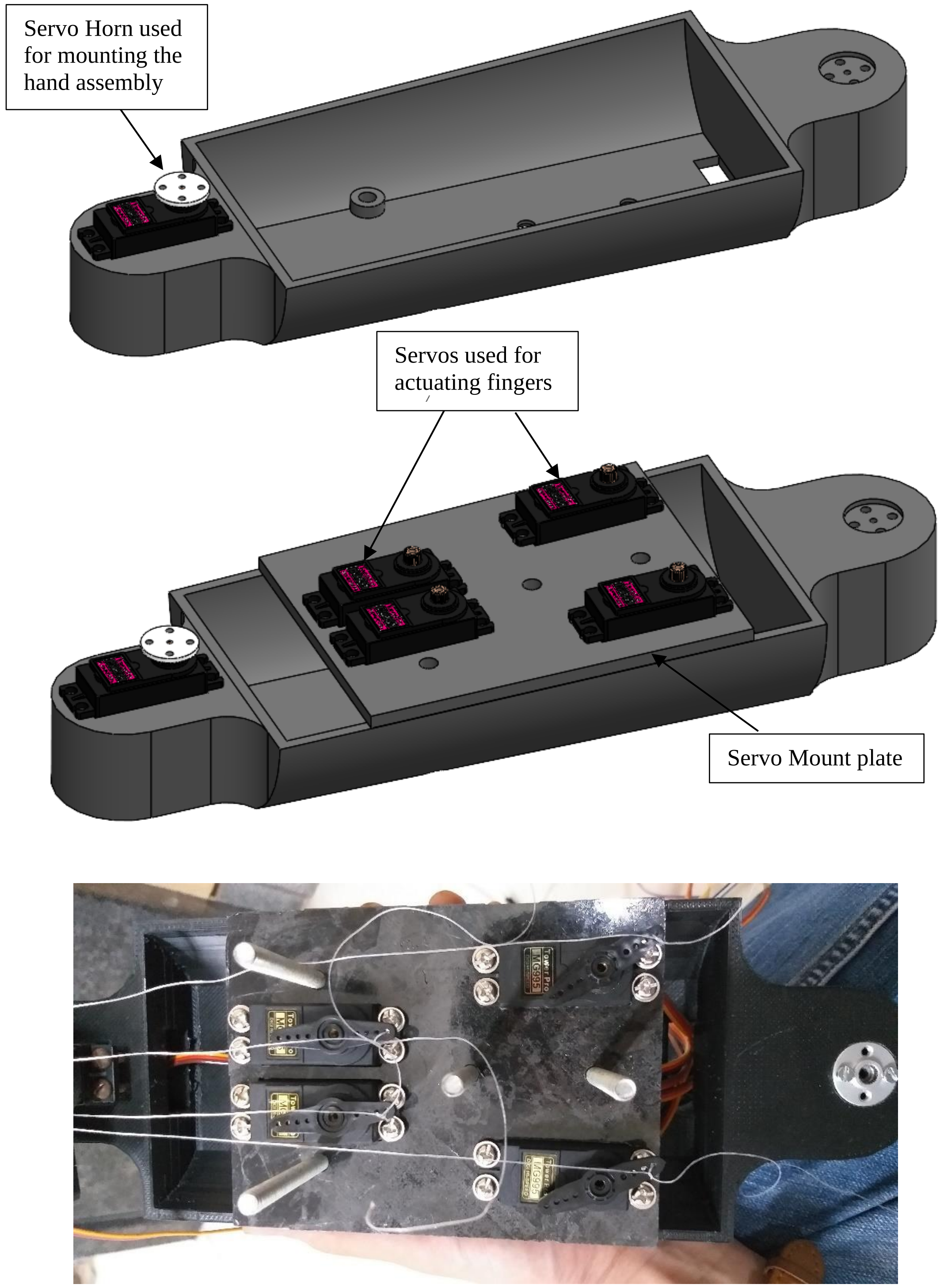


Figure 7: Top View of Forearm with Servo Motors

5. **Base:**

The base consists of a stepper motor and its housing. The stepper motor is mounted inside the housing using a L-bracket. A base plate is mounted on the stepper motor by means of a flange coupling. The base plate houses another servo motor which is connected to the bicep. The bicep is connected to the forearm. The stepper motor gives the entire assembly an additional degree of freedom.

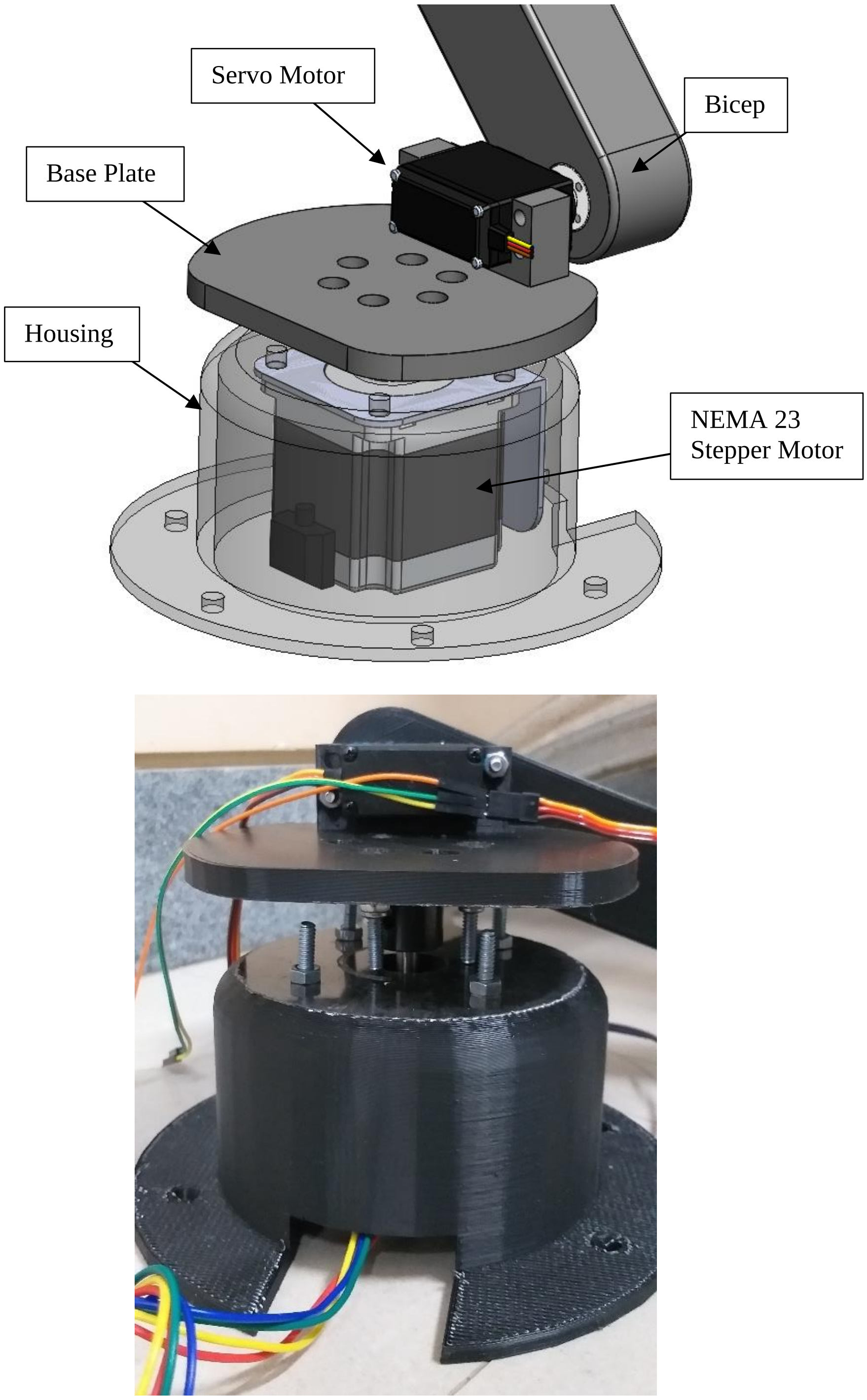


Figure 8: Base Assembly

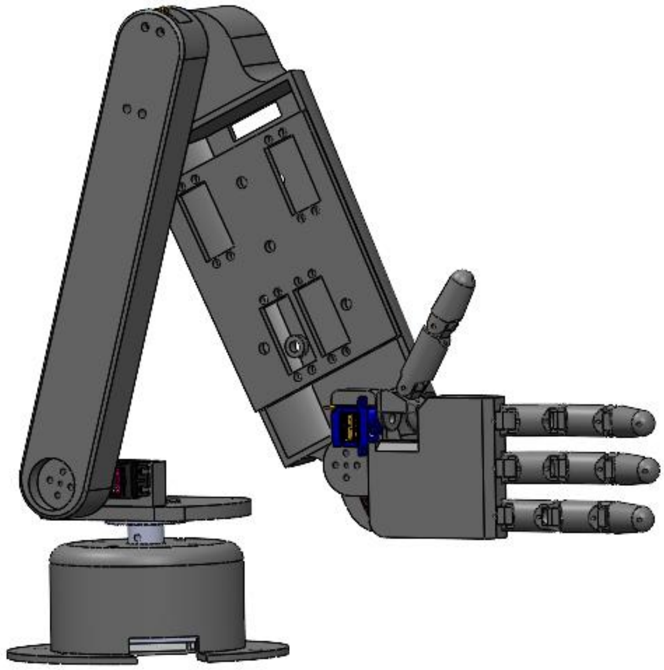


Figure 9: CAD Model of Robotic Arm

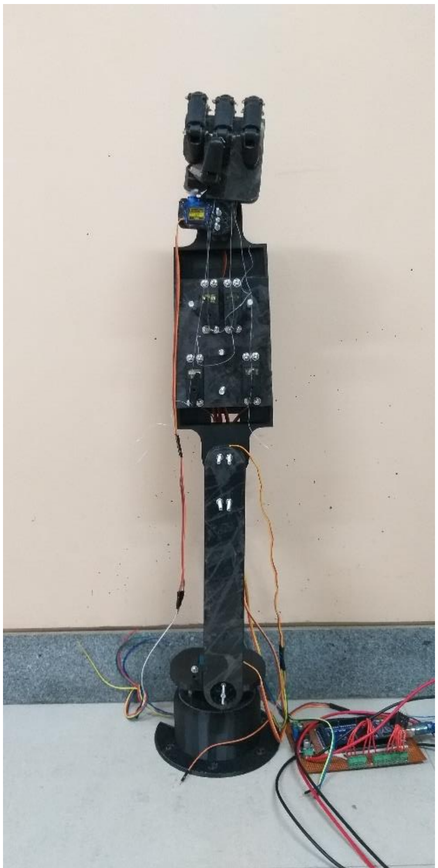


Figure 10: Final Assembly

Force Calculation:

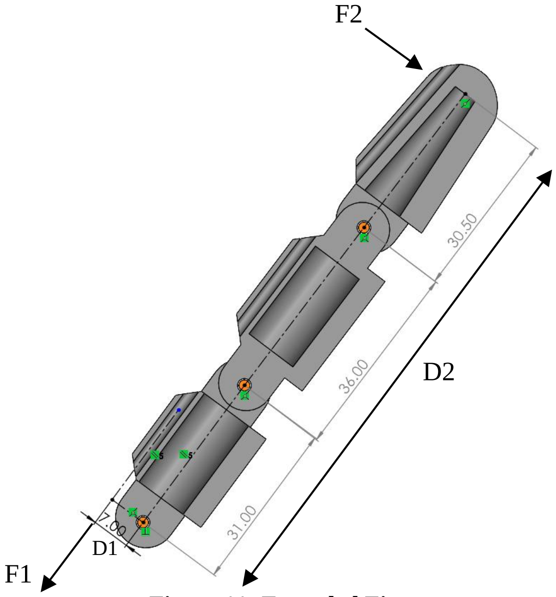


Figure 11: Extended Finger

When the fingers are completely extended and a force is applied at the tip, a moment is created at each joint of the finger. Since the perpendicular distance between the joint of the base and the tip of the finger is maximum, the moment about this joint will also be maximum. The servo motor used (MG995) has a stall torque of 10 Kg.cm = 1 N/m and the tendons are attached at a distance of 20 mm from the center of the servo by means of a horn. We start by determining the tension (F2) in the tendons used to pull the fingers.

$$\text{Torque}_{\text{servo}} \text{ x distance} = F_2$$

$$F2 = 1 \text{ N/mm x } 20 \text{ mm} = 20 \text{ N}$$

Now, since the load is being lifted by means of the tendons, the moments produced by the forces F1 and F2 about the base joint should be balanced out.

$$F_1D_1 = F_2D_2$$

$$F_2 = \frac{20\ N \times 7mm}{98} = 1.43 \text{ N} \approx 143\ gm$$

Therefore, the force exerted at the tip of the fingers when fully extended = 143 gm

However, the force applied at the finger tips while picking up an object increases as the fingers curl. This is because the perpendicular distance between the base joint and the tip decreases. Therefore, in this case:

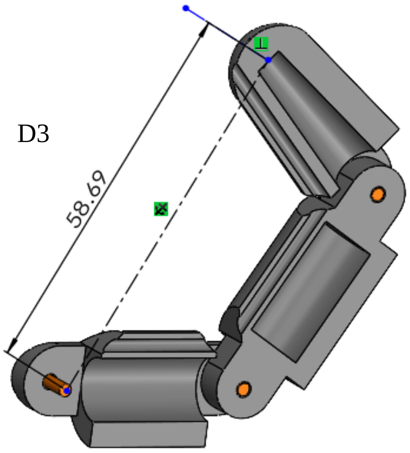


Figure 12: Curled Finger

$$F_1D_1 = F_3D_3$$

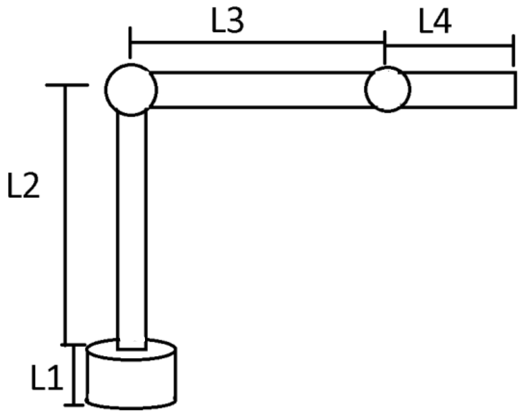
$$F_3 = \frac{20\ N \times 7mm}{60} = 2.33\ \text{N} \approx 233\ gm$$

Therefore, the maximum force applied at the tip of the fingers is when the fingers are fully curled up. This implies that an object with a maximum weight of 230 gm can be lifted using this mechanism.

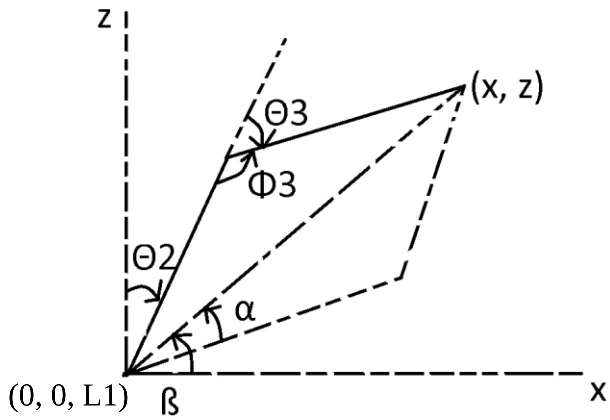
The servo motors used in the drive system have a stall torque of 12 Kg.cm which is sufficient for gripping and lifting objects.

5.2 Programming Analysis

5.2.1 Inverse Kinematics:



To calculate the inverse kinematics of the robot, we must note that there are infinite solutions to a 4 – DOF robot. Here, we have chosen to stabilize the link L4, which will ensure accurate picking of the object without toppling the object down. For calculating the inverse kinematics, we consider the following diagram:



We consider the angle Θ2 and Θ3 in this diagram since the relative position of Θ4 will be calculated after all the joint angles have been calculated, as this joint is static with respect to the entire robot.

Calculating the inverse kinematics of the robot:

$$\Theta_1 = \tan^{-1}(\frac{y}{x})$$

If L_1 is greater than z, the z coordinate considered for the further calculations will be $(L_1 - z)$, else it will be considered as $(z - L_1)$

$$\text{A} = \sqrt{x^2 + (L_1 - z)^2}$$

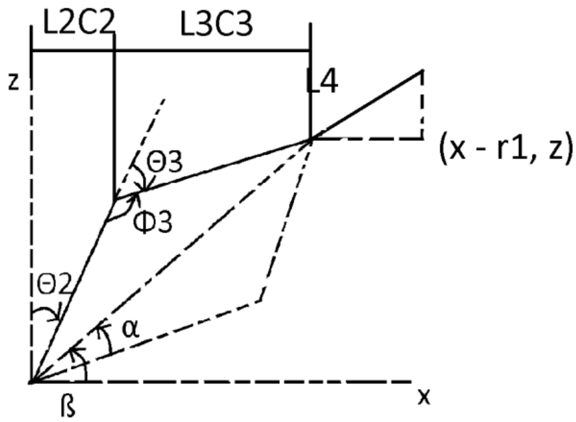
$$\Phi_3 = cos^{-1}(\frac{{L_2}^2 + {L_3}^2 - A^2}{2L_2L_3})$$

$$\Theta_3 = \pi - \Phi_3$$

$$ß = tan^{-1}(\frac{x}{L_1 - z})$$

$$\alpha = cos^{-1}(\frac{{L_2}^2 + A^2 - {L_3}^2}{2L_2A})$$

$\Theta_2 = ß \pm \alpha$ ($\Theta_3 < 0$, the sign is +. If $\Theta_3 > 0$, the sign is -)



To calculate the angle Θ_4, we should know that the end effector position is aligned with the previous joint. Hence, we need to calculate appropriate lengths to find the angle.

Here,

$$r_1 = L_2 \cos(\Theta_2) + L_3 \cos(\Theta_3)$$

Therefore,

$$\Theta_4 = cos^{-1}(\frac{x - r_1}{L_4})$$

Hence, we have calculated the desired angles required for reaching the desired end effector.

5.2.2 Object Triangulation:

In linear algebra, the Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into three matrices. It has some interesting algebraic properties and conveys important geometrical and theoretical insights about linear transformations. The DLT method is based on SVD. DLT is a method for calculating a matrix equation of the form A**x** = 0, where A is some matrix and **x** is the vector unknowns that we want. This problem setting occurs in many forms in photo-grammetry.

Suppose we have a 3D point in real space with coordinates given as **X** = [x, y, z, w] in homogeneous coordinates. Suppose we observe this point through two cameras, which have pixel coordinates **U**1 = [u1, v1, 1] for cameras #1 and **U**2 = [u2, v2, 1] for camera #2. Using the camera projection matrix P1, we can write **U**1 as:

$$U_1^{\rightarrow} = \alpha P_1 X^{\rightarrow}$$

In a triangulation problem, we do not know the coordinates of **X**. But we can determine the pixel coordinates and assume we found the projection matrix through camera calibration. Our task is then to determine the unknowns in **X**. Since **U**1 and P1**X** are parallel vectors, the cross product of these should be zero. This gives us:

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} x \begin{bmatrix} p_1 X^{\rightarrow} \\ p_2 X^{\rightarrow} \\ p_3 X^{\rightarrow} \end{bmatrix} = \begin{bmatrix} v_1 p_3 X^{\rightarrow} - p_2 X^{\rightarrow} \\ p_1 X^{\rightarrow} - u_1 p_3 X^{\rightarrow} \\ u_1 p_2 X^{\rightarrow} - v_1 p_1 X^{\rightarrow} \end{bmatrix} = \begin{bmatrix} v_1 p_3 - p_2 \\ p_1 - u_1 p_3 \\ u_1 p_2 - v_1 p_1 \end{bmatrix} X^{\rightarrow} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Since we have two cameras, we can extend the matrix to have more rows. In fact, we simply add on more rows for any number of views. This gives us the equation:

$$AX^{\rightarrow} = \begin{bmatrix} v_1 p_3 - p_2 \\ p_1 - u_1 p_3 \\ u_1 p_2 - v_1 p_1 \end{bmatrix} X^{\rightarrow} = 0$$

In camera triangulation, we are given A and we want to determine **X**. In this setting, we use DLT to determine **X**.

We want to obtain the non-trivial solution of an equation of the form A**x** = 0. In the real world, there can be some noise, so we write the equation as A**x** = **w**, and we solve for **x** such that **w** is minimized. The first step is to determine the SVD decomposition of A.

$$Ax^{\rightarrow} = USV^T x^{\rightarrow}$$

Our goal is to minimize **w** for some **x**. This can be done by taking the dot product:

$$w^{\to T} w^{\to} = (x^{\to T} USV^{T}) \,.\, (USV^{T} x^{\to}) = x^{\to T} V S^{2} V^{T} x^{\to}$$

Remember that U and V are orthonormal matrices and S is a diagonal matrix. Moreover, the entries on the diagonal of S are decreasing, so that the last entry on the diagonal is the minimum value. These are guaranteed by the SVD decomposition. Exploiting the property that V is an orthonormal matrix, if we simply select **x** to be one of the column vectors of VT:

$$v_i^{\to T} V S^{2} V^{T} v_i^{\to} = s_i^{2}$$

In the above equation, the i[th] diagonal entry of S is written as s_i. Since our goal is to minimize **w**T**w**, this tells us that it is equivalent to choosing the smallest value of S2 by selecting the corresponding **v**i column vector of VT as **x**. In other words, the minimum value is obtained if we choose the last column vector of VT as **x**. Thus, we have solved the A**x** = **w** equation in the presence of noise. If there is no noise, this SVD method will still work.

As you can see in the code, the above-mentioned method is being implement:

```
def DLT(P1, P2, point1, point2):

  A = [point1[1]*P1[2,:] - P1[1,:],
      P1[0,:] - point1[0]*P1[2,:],
      point2[1]*P2[2,:] - P2[1,:],
      P2[0,:] - point2[0]*P2[2,:]
     ]
  A = np.array(A).reshape((4,4))

  B = A.transpose() @ A
  from scipy import linalg
  U, s, Vh = linalg.svd(B, full_matrices = False)

  print('Triangulated point: ')
  print(Vh[3,0:3]*2.2/Vh[3,3])
  return Vh[3,0:3]/Vh[3,3]
```

Figure 13: DLT Calculation

For calculating the necessary stereo-camera parameters, the cameras need to be calibrated individually as well as together. The reason for incorporating stereo vision is for the sole purpose of estimating the depth, as well as the 3D location of the object, so that the robot hand can easily maneuver over the goal position. Here is a snippet of the stereo calibration:

```
def stereo_calibrate(mtx1, dist1, mtx2, dist2, frames_folder):

    images_names = glob.glob(frames_folder)
    images_names = sorted(images_names)
    c1_images_names = images_names[:len(images_names)//2]
    c2_images_names = images_names[len(images_names)//2:]
    print(c1_images_names, c2_images_names)
    c1_images = []
    c2_images = []
    for im1, im2 in zip(c1_images_names, c2_images_names):
        _im = cv.imread(im1, 1)
        c1_images.append(_im)

        _im = cv.imread(im2, 1)
        c2_images.append(_im)


    criteria = (cv.TERM_CRITERIA_EPS + cv.TERM_CRITERIA_MAX_ITER, 100, 0.0001)

    rows = 6
    columns = 8
    world_scaling = 1.

    objp = np.zeros((rows*columns,3), np.float32)
    objp[:,:2] = np.mgrid[0:rows,0:columns].T.reshape(-1,2)
    objp = world_scaling* objp
```

Figure 14: Stereo Calibration

5.2.3 Trajectory Generation:

For an effective trajectory to be generated, another method is being used to create computer generated angles for the robot. Here, we are incorporating the matrices of the end effector and based on these matrices, we are calculating the Body Jacobians of the robot. These Jacobians are then passed on to an inverse kinematics calculator, which can predict the next state of the robot, as well as provide the angles for this state. (Note that this next state can be predicted using a trajectory generator) After calculating the next state, the PID stability parameters are taken into consideration, which help improve the overall stability of the robot. Various tests conducted on PID controller have shown the following output data.

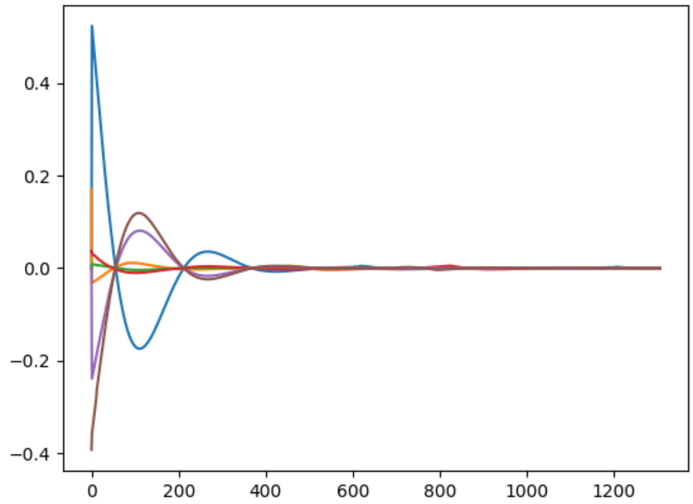


Figure 15: PID Calculations PI

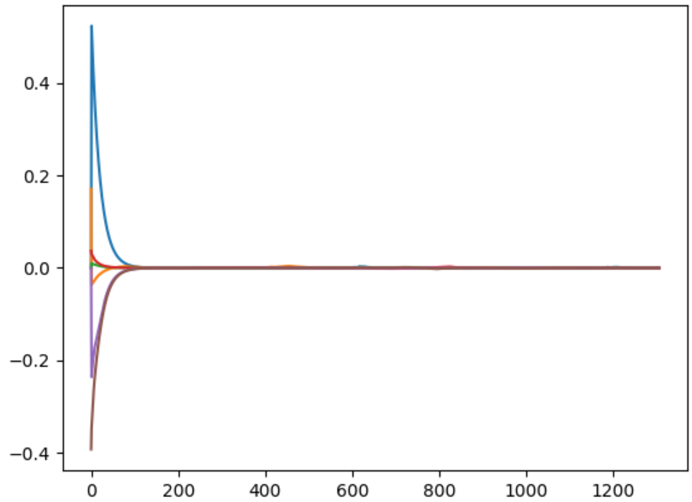


Figure 16: PID Calculations P

This is a sample for the calculated trajectory:

0.403102	0.591638	-0.69819	-0.86603	1
0.404198	0.588638	-0.70009	-0.86603	1
0.405288	0.585632	-0.70198	-0.86603	1
0.406372	0.582621	-0.70386	-0.86603	1
0.407449	0.579605	-0.70572	-0.86603	1
0.408519	0.576583	-0.70758	-0.86603	1
0.409583	0.573556	-0.70942	-0.86603	1
0.41064	0.570525	-0.71125	-0.86603	1
0.411691	0.567489	-0.71307	-0.86603	1
0.412734	0.564448	-0.71488	-0.86603	1
0.413771	0.561403	-0.71667	-0.86603	1
0.414802	0.558353	-0.71846	-0.86603	1

Table 1: Trajectory Generation

Here, each value pertains to the respective joint position and the last column signifies whether the gripper should be open or closed.

To calculate the trajectory, fixed points need to be provided to the robot to estimate the basic trajectory needed to reach the object. Here is a snippet of the same:

```
Tse_init_q = np.array([Tse_init[0][-1], Tse_init[1][-1], Tse_init[2][-1]])  #initial location and rotation of gri
T_standoff_init_q = np.array([T_standoff_init[0][-1], T_standoff_init[1][-1], T_standoff_init[2][-1]])  #initi
T_standoff_final_q = np.array([T_standoff_final[0][-1], T_standoff_final[1][-1], T_standoff_final[2][-1]])
T_grasp_q = np.array([T_grasp[0][-1], T_grasp[1][-1], T_grasp[2][-1]])  #initial location and rotation of gr
T_release_q = np.array([T_release[0][-1], T_release[1][-1], T_release[2][-1]])  #final location and rotation

d1 = np.linalg.norm(Tse_init_q - T_standoff_init_q) #distance from the initial gripper and hovering gripp
d2 = np.linalg.norm(T_standoff_init_q - T_grasp_q) #distance from the hovering gripper and grasping g
d5 = np.linalg.norm(T_standoff_init_q - T_standoff_final_q) #distance from the initial hovering gripper
d6 = np.linalg.norm(T_standoff_final_q - T_release_q) #distance from the hovering gripper and final gra

dtotal = d1 + d2*2 + d5 + d6*2 #total distance measured. Since d2 and d6 are written twice in the co

tgrip_open = 0.63   #time required to open gripper
tgrip_close = 0.63  #time required to close gripper

#Timing of each segment calculated based on the fraction of distance from the total distance
t1 = d1 * (Ttotal - tgrip_open - tgrip_close) / dtotal
t2 = d2 * (Ttotal - tgrip_open - tgrip_close) / dtotal
t4 = t2
t5 = d5 * (Ttotal - tgrip_open - tgrip_close) / dtotal
t6 = d6 * (Ttotal - tgrip_open - tgrip_close) / dtotal
t8 = t6

dt = 0.01
```

Figure 17: Timing Calculation for Trajectory

To update the angles, the following function is used:

```
def AngleUpdate(angle, speed, dt):

    curr_angles = []
    for i in range(len(angle)):
        curr_angles.append(angle[i] + speed[i] * dt)

    return curr_angles
```

Figure 18: Updating Angles

5.3 Electronics Analysis

5.3.1 Power Source:

To power the entire robot, it was imperative that we provide sufficient voltage to power all the motors, because if sufficient voltage isn't received, the motors start stalling and can possibly lead to failure. For this purpose, a Switched Mode Power Supply (SMPS) is used to provide 24V 15A DC power directly from the mains (220V AC), which ensures a constant supply to the motors. Even if the motors were to stall and pull more than required current, the 15A output from the SMPS will ensure that there is no failure of components.

Figure 19: Switched Mode Power Source

Since the NEMA 2 Stepper motor requires a starting voltage of 18V, this is the ideal power source to carry out all functionality for the same. But the servo motors are limited to an operation voltage of 8.4V and 7.2V. Therefore, we need to step down the received 24V using a Buck Converter. LM 2596 is used for the same. This converter performs the above-mentioned process and ensures that constant supply is received. We require two of these, since there are two different voltage requirements for the servo motors attached.

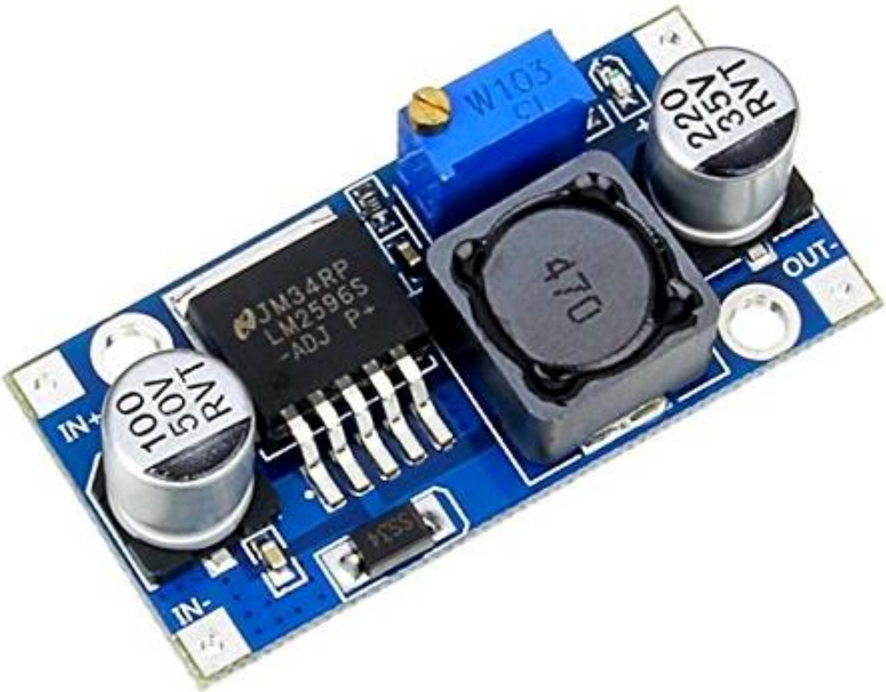


Figure 20: LM 2596 Buck Converter

To connect rest of the components to the Arduino Mega, we use appropriate wiring from the SMPS to the Buck Converters, which are sent to individual Servo Motors and a Stepper Motor. The following connections are made to the Arduino:

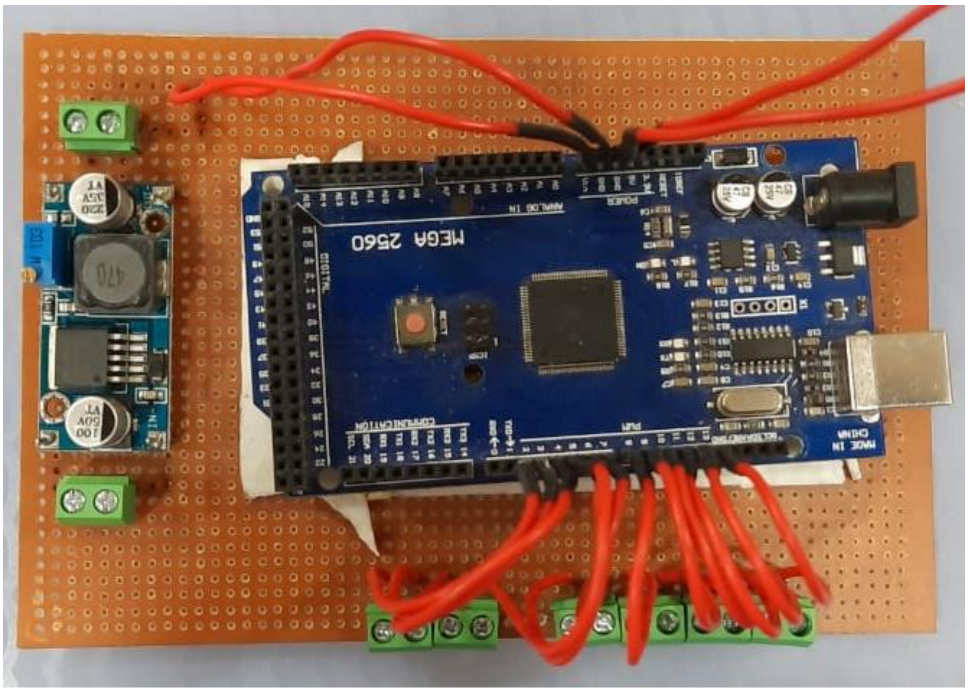


Figure 21: Arduino PCB

Additionally, an extra connection board has been created to manage the wires attached to the 3D Printed Model, to facilitate longer connections of the wires. Here is the PCB for the same:

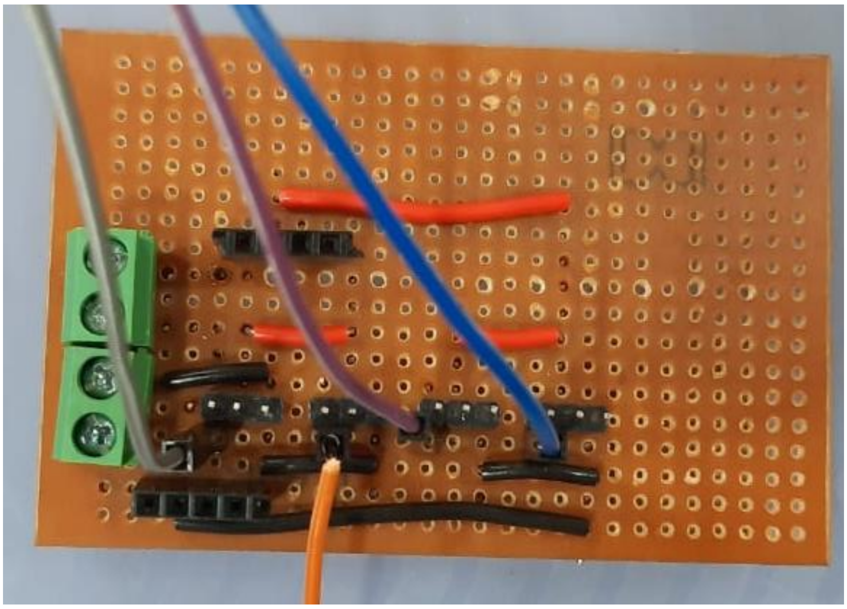


Figure 22 Cable Management PCB

Chapter 6

Testing and Results

6.1 Landmark Detection

After the testing of the sample code, we get the following results:

Detection of the landmarks on the arm are shown in the below image, clearly indicating a joint angle between the forearm and the femur.

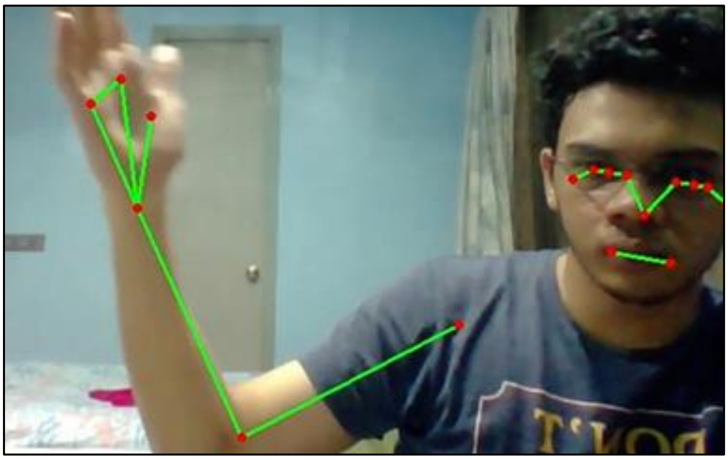


Figure 23: Arm-pose Estimation

Furthermore, the landmarks on the fingers are effectively displayed in the test image. These landmarks can be distinguished by the color of the finger. To calculate the joint angles, we need to calculate the angle between the red landmarks on the bottom of the fingers and the rest of the finger, giving us a joint angle for the motor.

Figure 24: Hand-Pose Estimation

6.2 Test: Picking up Cans

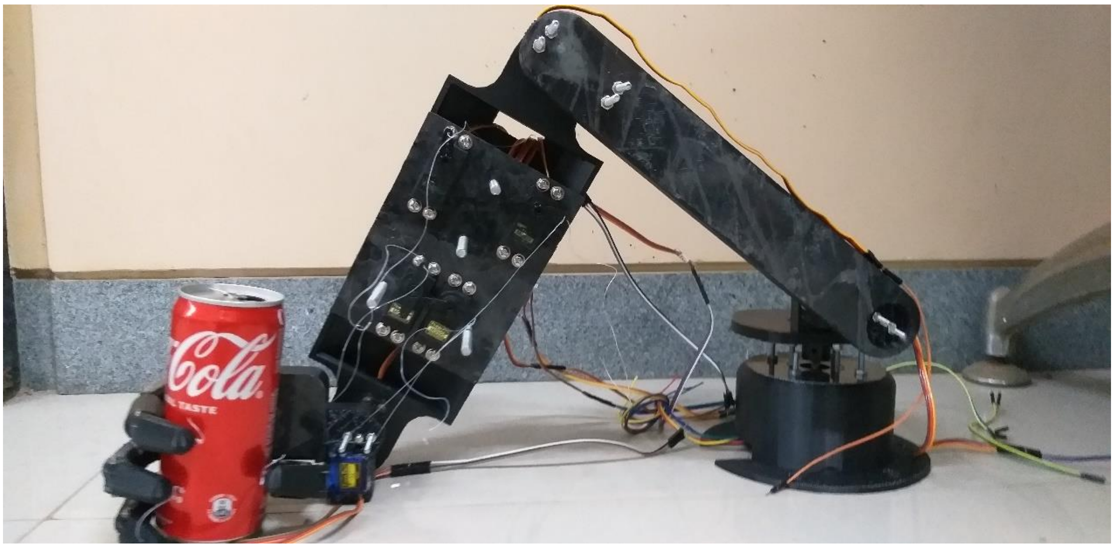


Figure 25: Reaching the Can

Figure 26: Lifting the Can

Chapter 7

Advantages, Limitations and Applications

Applications:

- Hazardous Environments:
  This arm can be used in environments which are inaccessible to humans to perform tasks with speed and precision. For e.g., Radioactive environments or emergency situations.
- Medical Domain:
  The AI based object identification capabilities of the arm will prove to be useful in the medical field. For e.g., Passing a specific tool to the surgeon during a surgery.
- Manufacturing Industries:
  The robot can be used in the autonomous mode to perform repetitive tasks in industries. Owing to its dexterity, it has a wider range of applications than a usual robotic arm.

Advantages:

- Reduces risk to human life as they do not have to be exposed to harsh environments.
- Repetitive tasks can be done faster with more accuracy and precision by using the robot in the autonomous mode.
- Gesture controlled mode is advantageous in situations where humans cannot be directly involved, and robots are not trained enough to perform the task. E.g., Manufacturing chipsets in the industry.

Limitations:

- The robotic arm has less degrees of freedoms thus limiting its functionality.
- Area of effect of the hand is significantly reduced due to the scale of the proposed model.
- Depending on the type of motor used and the material of the robot, only specific objects can be carried and interacted with.

Chapter 8

Conclusion and Future Scope

Conclusion:

Through this project, we have addressed and presented a solution for the problem that causes an approximate of 12.6 million deaths per year. Our robot, in comparison to the conventional methods used for gesture control, does not incorporate the use of any external gear, which not only facilitates ease of operation, but also puts the user out of harm’s way. Furthermore, this robot can successfully pick up and place an object, without the user having to define the location of the object. This method simplifies the task and causes minimal human error. Additionally, the stability of the end effector causes the robot to always grip the object parallel to the ground, rather than picking up an object at a particular angle. Overall, we believe that this robot can be a significant contribution to the industry, which eliminates the risk of a human to be exposed to hazardous environments, being adept enough to decrease the death rate caused by industrial accidents and unhealthy environments.

Future Scope:

- More degrees of freedom can be implemented to improve the functionality of the robot and make it more humanoid, thus, increasing its dexterity.
- The material used is ABS and it is prone to breaking. Therefore, a better material could be utilized which would strengthen the product.
- Voice Control can also be implemented in the project which will make it viable in many industries. E.g., Passing a specific tool to the surgeon during a surgery.
- Variable grip-strength mechanism based on material detection to enhance the interaction of the robots with different objects.

References

[1] D. J. Cox, "Mock-up of hazardous material handling tasks using a dual-arm robotic system," Proceedings of the 5th Biannual World Automation Congress, 2002, pp. 527-532, doi: 10.1109/WAC.2002.1049491.

[2] K. Kim and H. Kim, "Remote robotic systems for nuclear environment application," ICCAS 2010, 2010, pp. 1843-1846, doi: 10.1109/ICCAS.2010.5670160.

[3] T. D. Bendale and V. J. Kharat, "A design of gesture controlled mobile robot with robotic arm for nuclear environment," 2017 International Conference on Nascent Technologies in Engineering (ICNTE), 2017, pp. 1-6, doi: 10.1109/ICNTE.2017.7947918.

[4] M. Mariappan, T. Ganesan, M. Iftikhar, V. Ramu and B. Khoo, "A design methodology of a flexible robotic arm vision system for OTOROB," 2010 International Conference on Mechanical and Electrical Technology, 2010, pp. 161-164, doi: 10.1109/ICMET.2010.5598341.

[5] Y. Wu, M. Wang and N. M. Mayer, "A new type of eye-on-hand robotic arm system based on a low-cost recognition system," 2017 International Conference on Advanced Robotics and Intelligent Systems (ARIS), 2017, pp. 110-114, doi: 10.1109/ARIS.2017.8297200.

[6] S. Gushi, Y. Shimabukuro and H. Higa, "A Self-Feeding Assistive Robotic Arm for People with Physical Disabilities of the Extremities," 2020 5th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), 2020, pp. 61-64, doi: 10.1109/ICIIBMS50712.2020.9336390.

[7] E. Saraee, A. Joshi and M. Betke, "A therapeutic robotic system for the upper body based on the Proficio robotic arm," 2017 International Conference on Virtual Rehabilitation (ICVR), 2017, pp. 1-2, doi: 10.1109/ICVR.2017.8007498.

[8] J. Cornejo, J. A. Cornejo-Aguilar and R. Palomares, "Biomedik Surgeon: Surgical Robotic System for Training and Simulation by Medical Students in Peru," 2019 International Conference on Control of Dynamical and Aerospace Systems (XPOTRON), 2019, pp. 1-4, doi: 10.1109/XPOTRON.2019.8705717.

[9] A. K. Sahoo, B. Brahma and A. Pattanaik, "Design & Development of Robotic Arm for Medical Surgery," 2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC), 2019, pp. 105-110, doi: 10.1109/ICPEDC47771.2019.9036490.

[10] P. C. Patic, M. Mainea, L. Pascale and G. Mantescu, "Designing a Mobile Robot used for Access to Dangerous Areas," 2017 International Conference on Control,

Artificial Intelligence, Robotics & Optimization (ICCAIRO), 2017, pp. 60-65, doi: 10.1109/ICCAIRO.2017.21.

[11] T. Inoue, T. Matsuno, A. Yanou, M. Minami and T. Hiraki, "Development of a minimally invasive Robotic Interventional Radiology for treatment of lung cancer - Manufacture of a basic mechanism and verification experiment-," The SICE Annual Conference 2013, 2013, pp. 2646-2651.

[12] G. Guevara, J. Lima, P. Leitão and M. do Carmo Baptista, "Development of a robotic prototype system for the preparation and partition of radioactive products," 2015 IEEE International Conference on Industrial Technology (ICIT), 2015, pp. 173-178, doi: 10.1109/ICIT.2015.7125095.

[13] P. Atre, S. Bhagat, N. Pooniwala and P. Shah, "Efficient and Feasible Gesture Controlled Robotic Arm," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 1-6, doi: 10.1109/ICCONS.2018.8662943.

[14] S. A. Stansfield, "Knowledge-based robotic grasping," Proceedings., IEEE International Conference on Robotics and Automation, 1990, pp. 1270-1275 vol.2, doi: 10.1109/ROBOT.1990.126173.

[15] X. Chen, B. Zhao and X. Gao, "Noninvasive Brain-computer Interface Based High-level Control of a Robotic Arm for Pick and Place Tasks," 2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2018, pp. 1193-1197, doi: 10.1109/FSKD.2018.8686979.

[16] G. Mohapatro, S. S. Kamlesh and R. Mishra, "Optimized path tracing analysis and application of four-bar linkage and CT guided robotic arm," 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017, pp. 967-972, doi: 10.1109/CCAA.2017.8229947.

[17] N. Marturi et al., "Towards advanced robotic manipulation for nuclear decommissioning: A pilot study on tele-operation and autonomy," 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA), 2016, pp. 1-8, doi: 10.1109/RAHA.2016.7931866.

[18] A. Sabarivani, A. Anbarasi, A. Vijayaiyyappan and S. Sindhuja, "Wireless Synchronization of Robotic Arm with Human Movements Using Aurdino for Bomb Disposal," 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), 2018, pp. 229-234, doi: 10.1109/ICSSIT.2018.8748690.

[19] H. Jeong and J. Cheong, "Design of hybrid type robotic hand: The KU hybrid HAND," 2011 11th International Conference on Control, Automation and Systems, 2011, pp. 1113-1116.

[20] K. Yoshida, H. Yamada, R. Kato, T. Seki, H. Yokoi and M. Mukai, "Development of five-finger robotic hand using master-slave control for hand-assisted

laparoscopic surgery," 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2016, pp. 5124-5127, doi: 10.1109/EMBC.2016.7591880.

[21] M. C. Ergene, A. Durdu and H. Cetin, "Imitation and learning of human hand gesture tasks of the 3D printed robotic hand by using artificial neural networks," 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2016, pp. 1-6, doi: 10.1109/ECAI.2016.7861177.

[22] J. L. Raheja, R. Shyam, U. Kumar and P. B. Prasad, "Real-Time Robotic Hand Control Using Hand Gestures," 2010 Second International Conference on Machine Learning and Computing, 2010, pp. 12-16, doi: 10.1109/ICMLC.2010.12.

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

[24] S. Bularka, R. Szabo, M. Otesteanu and M. Babaita, "Robotic Arm Control with Hand Movement Gestures," 2018 41st International Conference on Telecommunications and Signal Processing (TSP), 2018, pp. 1-5, doi: 10.1109/TSP.2018.8441341.

[25] S. van Delden and M. A. Umrysh, "Visual detection of objects in a robotic work area using hand gestures," 2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE), 2011, pp. 237-242, doi: 10.1109/ROSE.2011.6058529.

Appendix A: Soft Code Flowcharts

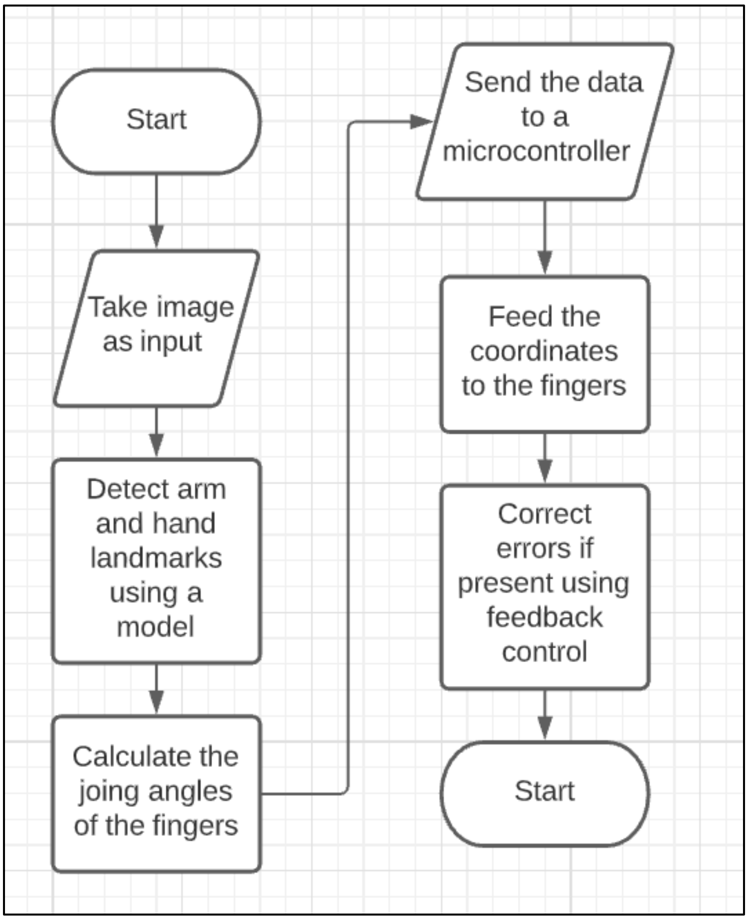


Figure 26: Gesture-Controlled Mode

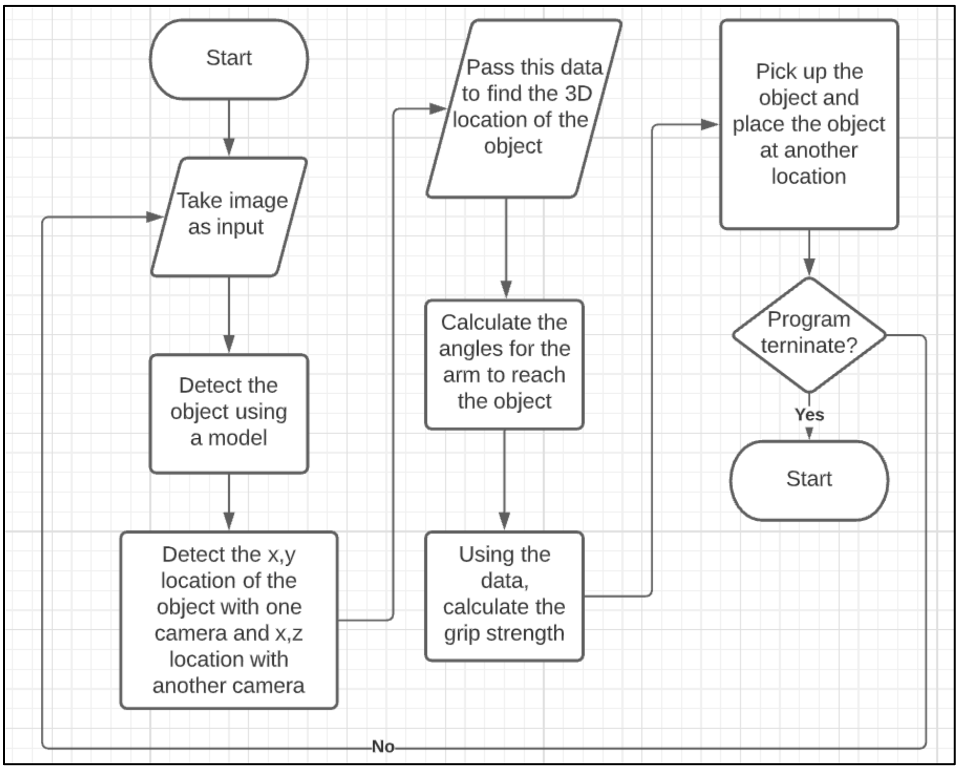


Figure 27: Autonomous Mode

Appendix B: List of Components

1. Arduino Mega
2. Solidworks
3. MATLAB
4. Python
5. CoppeliaSim
6. Servo Motor
7. Stepper Motor
8. Camera
9. Laptop
10. 3-D Printer & Material
11. High-Tension Cable
12. SMPS 24V/15A
13. LM 2596 Buck Converter