Smart Robot Hand

Shubhankar Kulkarni
Department of Mechatronics Engineering
SVKM's NMIMS Mukesh Patel School of Technology Management and Engineering
Mumbai, India
kulkarnishubhankar10@gmail.com

Miheer Diwan
Department of Mechatronics Engineering
SVKM's NMIMS Mukesh Patel School of Technology Management and Engineering
Mumbai, India
miheer.diwar@gmail.com

Dattatray Sawant
Department of Mechatronics Engineering
SVKM's NMIMS Mukesh Patel School of Technology Management and Engineering
Mumbai, India
dattatray.sawant@nmims.edu

Amey Raut
Department of Mechatronics Engineering
SVKM's NMIMS Mukesh Patel School of Technology Management and Engineering
Mumbai, India
amey.raut@nmims.edu

***Abstract*— This paper entails the design, simulation and implementation of a human inspired robotic hand with a dual-mode operation. The robot can mimic the motions and gestures of the user's hand, as well as detect and manipulate an object in 3D space using color detection and triangulation.**

***Keywords*— robotics, object detection, triangulation, mechanics.**

I. Introduction

This project was proposed to aid people in hazardous tasks, as well as other meticulous applications such as handling medical devices and microchip manufacturing. This robot works interchangeably in two modes. In the first mode, hand gestures of the user are recorded using a camera and translated into signals for the hand to mimic. In the second mode, the robot works autonomously to detect and triangulate the position of the object in 3D space. Using this data, the robot then accurately grasps and manipulates the objects. This project aims to reduce the number of deaths caused by working in a hazardous environment [1]. An estimate of 2 million deaths occurs every year due to work occurring in hazardous environments. To reduce this, our smart robot hand can be placed in the environment and a person can remotely control the motion of the robot instead of travelling to an environment unfit for work.

This paper is divided into different sections—mechanical design, object manipulation, and robot kinematics. The methods used for this project are considered and discussed in depth in the following paper.

II. Mechanical Design & CAD Model

The proposed design of the robot consists of 3 links—the upper arm, the forearm and the hand connected to each other by means of servo motors. These servo motors serve as the shoulder, elbow and wrist joints respectively. The arm itself is mounted on a baseplate, connected to a Stepper motor which provides revolute motion to the arm.

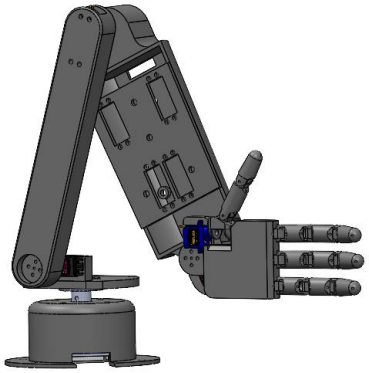

Figure 1. Complete CAD Model

A. Fingers

The robot hand features three symmetric tendon-driven fingers and a thumb actuated with the help of high-tension cables and servo motors. The abduction and adduction motion of the fingers was also excluded to reduce the complexity of the model without compromising too much functionality.

Each finger consists of three links which are connected by means of dowel pins [2]. The links have rotational freedom about these pins which gives each finger a single degree of freedom. The third link is connected to the palm in a similar manner [3].

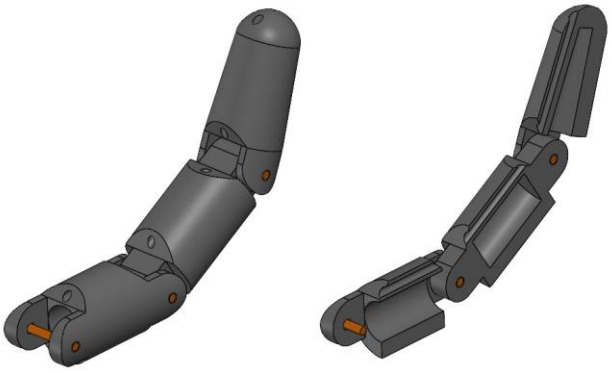

Figure 2. Finger Orthogonal and Cross-Sectional View

B. *Thumb and Palm*

The palm houses the fingers and the thumb. It is connected to the forearm by means of the wrist joint. The thumb is similar in design to the other fingers and is actuated in the same way. A small servo motor (SG 90) fitted inside the palm, offers the thumb an additional degree of freedom. This allows the thumb to stay in opposed and non-opposed positions and helps in grasping objects.

C. *Forearm*

The high-tension strings and the four servo motors (MG 995) used for moving the fingers are mounted in the forearm cavity. The strings are connected to the servo horns. The fingers curl when the servo horns are fully extended. An additional servo motor which serves as the wrist joint is also housed in the forearm cavity.

D. *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 [4]. 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.

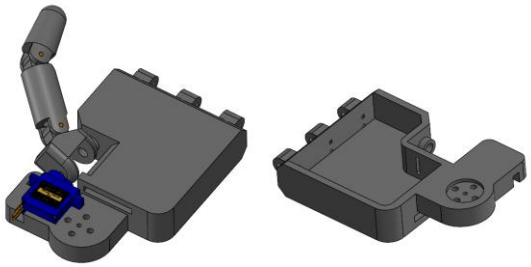

Figure 3. Palm

Figure 4. Forearm

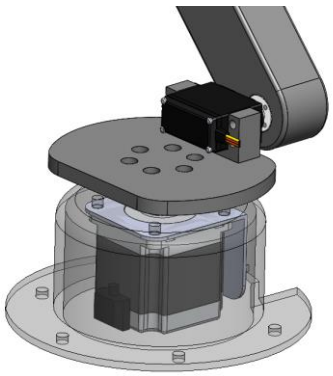

Figure 5. Base

III. Force Calculation

When the fingers are completely extended and a normal 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 (MG995) has a stall torque of 1 N/m and the tendons are attached at the tip of the servo horn, ~20 mm from the center. To determine the tension (F_2) in the tendons used to pull the fingers,

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

$$F_2 = 1 \text{ N/mm} \times 20 \text{ mm} = 20 \text{ N}$$

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.

$$F_1 \times D_1 = F_2 \times D_2$$

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

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

However, the force applied at the fingertips 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, as observed in Figure 6 and Figure 7 [5]:

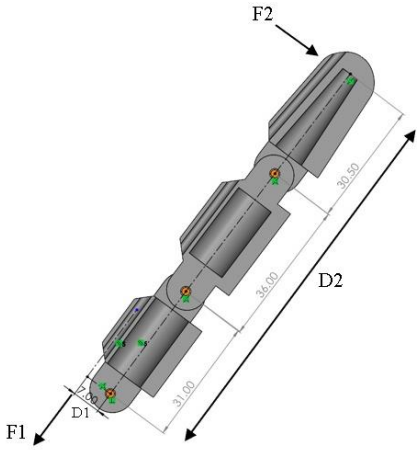

Figure 6. Force exerted on an extended finger

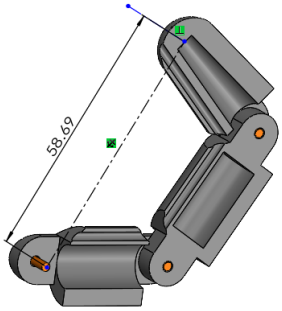

Figure 7. Force exerted on a curled finger

$$F_1 \times D_1 = F_3 \times D_3$$

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

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.

IV. Hand Motion Detection

One of the primary functions of this project is detecting hand gestures and reciprocating the same on the robot. In order to do so, one must detect the landmarks present on the hand to accurately calculate the angle made during the curling of the fingers.

A. Hand Landmark Detection

To implement this system, the Google hand detection API was used, which effectively detected the joints and the links between each finger, which can be observed in Figure 8. Using this data, the joint angles were calculated using the inverse tangent rule [6]. After obtaining the angle between the fingers, this angle is sent to a microcontroller, which actuates the fingers based on the angle of actuation calculated by the program.

B. Arm Landmark Detection

A similar process is used for the upper body as well. In this process, it is imperative that we keep in mind which arm is shown to the camera, i.e., the right or the left arm. If the right arm is shown to the camera, the right-side joint angles need to be calculated, otherwise the left-side joint angles need to be calculated. Using the Google API used for the finger landmark detection, we detect the landmarks on the arm, which enables us to get the estimated coordinates of the joints from the camera and allows us to calculate the angles between the joints for mimicking the motion of the arm onto the robot. We have therefore completed the first essential feature of the robot arm.

V. Object Detection

The second essential feature of this project is autonomous object manipulation in 3D space using cameras. This task involves two essential tasks, namely the detection of the object and the triangulation of the object. In the next few paragraphs, these steps are described in detail.

Figure 8. Hand Landmark Detection

A. Object Detection

For this task, to facilitate an easier detection of the object, an object of uniform color was chosen, which could be easily distinguished in the background. This color was segregated from the image by using an HSV Mask of that particular color. Any other background colors were discarded. The object with the maximum area was chosen in the image, which indicated that the object of interest was located in the camera.

In order to find the location of the object in 3D, it was essential that the cameras be calibrated before use, as well as the exact location of the object be determined and sent to the triangulation algorithm.

To find the exact location of the object in the camera frame, 50 iterations were taken into account. These images had their respective object of interest marked with its center. To determine that we are using the center of the object and not a false positive, an average of these 50 iterations is taken to get the closest possible center value of the object of interest.

B. Camera Calibration

When one wants to find the depth information of an object in an image, it is essential that we know the parameters of the camera. This allows one to map the pixel coordinates and the camera coordinates in the image frame.

It is also essential that one find the extrinsic parameters of the object, which are the orientation and location of the camera with respect to a world coordinate system.

In order to calculate these parameters, one must first assume real world coordinates as a point of reference. Therefore, a checkerboard, rather the intersections of a checkerboard are used as points of reference. The general idea of camera calibration is given by Equation (1) and Equation (2). Here (X_w, Y_w, Z_w) are the 3D points in world coordinates and (u, v) are image coordinates, **P** is the projection matrix consisting of an intrinsic matrix **K** and an extrinsic matrix [**R**|**t**].

$$\begin{bmatrix} u' \\ v' \\ z' \end{bmatrix} = P \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1)$$

$$P = K\ x\ [R|t] \quad (2)$$

Since two cameras are being used, we use stereo calibration to calibrate the cameras in order to know the pose and

location of one camera with respect to the other. This information helps us to feed the coordinates of an object with respect to one camera to the robot, instead of finding the location of each camera.

C. Stereo Calibration

Having the parameters of the camera and the necessary image coordinates, let $i_{pl} = (u_l, v_l)^t$ be the observation of an The solution of the given problem lies on the epipolar line of the K_l camera, which can be seen in Figure 10.

D. 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 shown in Equation (4).

$$U_1^{\rightarrow} = \alpha P_1 X^{\rightarrow} \tag{4}$$

In a triangulation problem, we don't know the coordinates of **X**. But we can determine the pixel coordinates and also 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 Equation (5).

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} x \begin{bmatrix} p_1 X^{\rightarrow} \\ p_2 X^{\rightarrow} \\ p_3 X^{\rightarrow} \end{bmatrix} = \begin{bmatrix} v_1 p_3 X^{\rightarrow} - p_2 X^{\rightarrow} \\ p_1 X^{\rightarrow} - u_1 p_3 X^{\rightarrow} \\ u_1 p_2 X^{\rightarrow} - v_1 p_1 X^{\rightarrow} \end{bmatrix} = \begin{bmatrix} v_1 p_3 - p_2 \\ p_1 - u_1 p_3 \\ u_1 p_2 - v_1 p_1 \end{bmatrix} X^{\rightarrow} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{5}$$

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 (6).

$$AX^{\rightarrow} = \begin{bmatrix} v_1 p_3 - p_2 \\ p_1 - u_1 p_3 \\ u_1 p_2 - v_1 p_1 \end{bmatrix} X^{\rightarrow} = 0 \tag{6}$$

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, as shown in Equation (7).

$$Ax^{\rightarrow} = USV^T x^{\rightarrow} \tag{7}$$

Our goal is to minimize **w** for some **x**. This can be done by taking the dot product as found in Equation (8).

object in the image of the left camera K_l and $i_{pr} = (u_r, v_r)^t$ the observation of the same object of the right camera K_r. For two corresponding points i_{pl} and i_{pr} in the image of the left and right camera, Equation (3) holds.

$$i_{pr}{}^t F i_{pl}{}^t = 0 \tag{3}$$

$$w^{\rightarrow T} w^{\rightarrow} = (x^{\rightarrow T} USV^T) . (USV^T x^{\rightarrow}) = x^{\rightarrow T} VS^2 V^T x^{\rightarrow} \tag{8}$$

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, we obtain Equation (9).

$$v_i^{\rightarrow T} VS^2 V^T v_i^{\rightarrow} = s_i^2 \tag{9}$$

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.

VI. Robot Inverse Kinematics

For the robot to reach the goal destination, we use inverse kinematics of the robot to feed the goal position to the robot by rotating the joint angles in such a fashion that it does not violate its workspace and reaches the desired end effector location.

When location *(x, y, z)* is given to the robot, Equations (10) can be used to find the desired angles, as observed in Figure 9 [7].

$$\theta_1 = \tan^{-1}\left(\frac{x}{y}\right) \tag{10}$$

We consider the angle θ_2 and θ_3 in Figure 11 since the relative position of θ_4 will be calculated after all the joint angles have been calculated, as this joint is static with respect to the entire robot.

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)

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

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

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

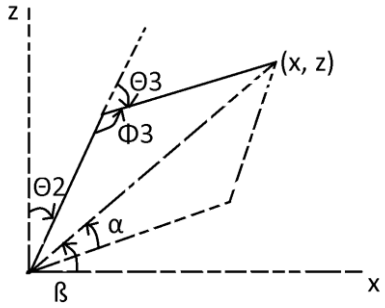

Figure 9. Inverse Kinematics Calculation

$$ß = tan^{-1}\left(\frac{x}{L_1 - z}\right)$$

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

$$\Theta_2 = ß \pm \alpha$$

In order to calculate the angle θ_4, we should know that the end effector position is aligned with the previous joint, as shown in Figure 10. Hence, we need to calculate appropriate lengths to find the angle.

Therefore, we can say that the length r_1 is equal to

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

And hence the value of θ_4 is equal to

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

VII. Observations

On implementing the hardware and software on a 3D model, the grasping motion in Figure 11 and Figure 12 was accomplished.

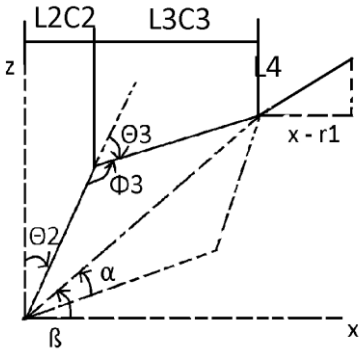

Figure 10. θ_4 Calculation

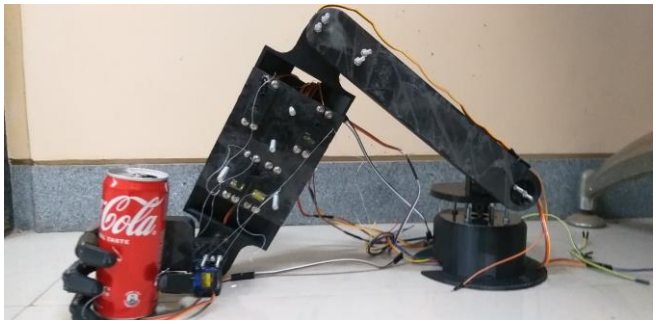

Figure 11. Grasping an Object

Figure 12. Picking up an Object

Figure 13. Gesture Control

Further simulation implementation of gesture control is shown in Figure 13.

Upon constant loading and de-loading, it was observed that the motors were not able to sustain the weight of the object, given the weights of the 3d printed parts each motor held. It led us to the conclusion, that a 3d part of higher in-fill percentage is required and motors of higher torque are desirable.

VIII. Conclusion and Future Scope

Through this paper 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.

ACKNOWLEDGMENT

This project was done as a final year project at our study in SVKM's NMIMS Mukesh Patel School of Technology Management and Engineering. I would like to thank Prof. Dattatray Sawant and Prof. Amey Raut for their constant support and aid throughout the project.

REFERENCES

[1] 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.

[2] 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.

[3] U. Rai, M. Patil, A. P. Singh and W. Arif, "An IoT based wireless robotic-hand actuation system for mimicking human hand movement," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-6, doi: 10.1109/INCET49848.2020.9154028.

[4] M. A. Norizan, F. Ali, N. Abas, H. Jamaluddin and M. F. Juhari, "Design and development of RH-2000 Robotic Hand," 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), 2015, pp. 149-153, doi: 10.1109/IRIS.2015.7451602.

[5] M. E. Hussein, "3d printed myoelectric prosthetic arm," Ph.D. dissertation, Thesis Bachelor degree Engineering (Mechatronics), 2014.

[6] Lin, Chern-Sheng & Chen, Pei-Chi & Pan, Yu-Ching & Chang, Che-Ming & Huang, Kuo-Liang. (2020). The Manipulation of Real-Time Kinect-Based Robotic Arm Using Double-Hand Gestures. Journal of Sensors. 2020. 1-9. 10.1155/2020/9270829.

[7] 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.