Mehran University Research Journal of Engineering and Technology

https://doi.org/10.22581/muet1982.2814

2025, 44(1) 20-33

Design, analysis, and model fabrication of an industrial manipulator for quality control purposes using computer vision as a sorting mechanism

Muthair Saeed [a, *], Haris Imran [a], Liaquat Ali Khan [b], Kamran Nazir [b]

[a] *Mechanical Engineering Department, National University of Technology, I-12, Islamabad, Pakistan*

[b] *Marine Engineering Department, Military Technological College Muscat, Oman*

[*] Corresponding author: Muthair Saeed, Email: muthair.saeed34@gmail.com

Received: 04 May 2023, Accepted: 26 December 2024, Published: 01 January 2025

KEYWORDS

Industrial Automation

Structure Optimization

Topology

Image Processing

Quality Control

Computer vision

ABSTRACT

The industrial manipulator uses complex control systems and requires specialist equipment for object detection and sorting. These are expensive and complicated. Installing and maintaining these manipulators require professional personnel and equipment. The manipulator design in this project is fully modular, easier to automate, and inexpensive, which is designed to enable industrial automation in developing countries. Using Topology in conjunction with Fusion 360 and ANSYS material consumption is reduced by up to 30% while keeping the material strength and stress constant. 3D printing parts help in reducing manufacturing complexity, time, and material consumption by reducing the overall manufacturing cost. With a Workspace of two feet and an operating temperature range from 10-60 °C, it is designed to deliver 3 DOF. To prevent the system from overheating, an effective cooling system for the controller and driver, is also developed using CFD analysis. The manipulator is linked with several sensors and a camera, to provide full autonomous control. As a result, it can be utilized in various industrial applications for precise and quick output. In this project, a manipulator is designed to sort objects as they come off the conveyor and pick and place them at the design location. A conveyor is intended for item movement, both manipulator and conveyor are controlled by a microcontroller and a PC via Computer vision. Initially sorting of objects is performed through both color sensors. Then, using computer vision, an algorithm is trained to identify any defective item among the bunch using a camera and place it at the designated location. This computer vision algorithm can detect an object at a range of 3 feet and provides proper control with an accuracy of 90-95%, depending on the amount of information about the object collected and analyzed. This algorithm is also adaptable, as it can identify any object once it has been trained.

1. Introduction

An articulated robotic arm is a programmable robot, resembling a human arm having only rotatory-type joints. The articulated robotic arm design is made with a combination of rotatory joints at all links, which help it achieve the full range defined by the manufacturer. A combination of programming and trajectory generation helps these robots to reach and follow complex pathways easily. Articulated Robotic Arms are commonly used in manufacturing, assembling, and pick/place processes. Multiple articulated robots are also used along the manufacturing line to complete numerous tasks with precision. Robotics is an integral part of every manufacturing industry around the world. Robotics help perform different manufacturing and material handling processes in these industries efficiently and accurately. The use of robotics has increased manufacturing rates and decreased manufacturing time.

2. Literature Review

[1] The first robotic arm was launched in 1961. George Devol invented the robotic arm and filed his invention for a patent in 1954. He received a patent for the robotic arm in 1961. George Devol cofounded a company Unimate with Joseph Engelberger and became the world's first company to successfully manufacture and sell articulated robotic arms to the world for industrial applications. Today modern industries are adopting and integrating robotic arms in manufacturing processes and assembly lines. A lot of research [5] is being done by engineering institutes and industries themselves to develop precise, robust, efficient, and inexpensive robotic arms.

Considering the importance of robotic arm and their potential to be adopted and implemented further in manufacturing industries and other fields like medical fields for operating on patients, a 3-DOF robotic arm was developed in research. This robotic arm is designed explicitly for pick-and-place applications. The structure of this robot is designed and made with the modern techniques of Shape Optimization processes; thus, its lightweight frame and quick movement enable the users to clamp welding tools, cutting tools, painting tools, and an extruder to make it a 3D-Printer making it suitable for more applications other than pick and place. This type of manipulator is commonly used in different industries, the difference is that this robotic arm will be inexpensive and provide easy control as compared to old techniques used by previous researchers [2]. This manipulator will be controlled using the modern technique of image processing using Computer Vision which is easy and cheaper to implement on a robotic arm, but whether it's reliable or not this top will also be covered in this paper.

The concept behind the replacement of humans in vehicle manufacturing facilities [7] with lower costs, precise working, and faster manipular to make the operations safer and increase mass production is also working in the current industry perfectly but for fully autonomous control and better accuracy in the industry, integrating Computer vision is much needed. The research work [8] of adding a vision source that leads to the automatic operation of the robot was helpful. It is used as a basic guideline for integrating the computer vision process. As in this approach, image processing of the objects is carried out through MATLAB with a camera connected to the PC, and digital information about the positions of the objects is obtained and then processed through the Visual Basics environment and used, which is somewhat similar but older method of image process which take a lot more data and time to process as compare to the computer vision approached us in this project.

Željko Šitum et. al. work on antagonistic muscle pair was used in a rotational sense to produce a required torque on a pulley. The concept, operating principles, and elementary properties of pneumatic muscle actuators are explained. Different conceptions of the system realizations are analyzed using the morphological-matrix conceptual design framework and a top-rated solution was practically realized. [9]. Hieu Giang Le et.al. review the demand for robotics manipulators in industry and their demand in the growing economy. To solve such challenging issues, substantial research efforts have been focused on automating tasks and eliminating the need for manual Labor. The Robotics field spans different disciplines, including mechanical engineering, control systems, robotics, and artificial intelligence. [10].

3. Background

3.1Problem Statement

Modern technological products are becoming more complex and intricate every year. The manufacturing of these devices requires manufacturing processes to be very precise and must have high repeatability to acquire mass production for products like cars, rocket engines, mobile phones, etc. Due to these reasons, human labor is being replaced by robots and robotic arms. However, these robotic arms have expensive initial purchasing costs and maintenance costs. These manipulators are also based on complex systems that are difficult to understand. That is why “there is a need for a robust and efficient robotic arm that is also easy to manufacture and maintain thus allowing industries to adopt them more easily”.

3.2 Purpose and Specifications Of The Purposed Solution

The purpose of this project is to develop a fast, accurate, inexpensive, and easy-to-manufacture sensor and AI-controlled Articulated Robotic arm that will be used for identifying and picking and placing a desired product in any process or manufacturing industry. This specific project is designed to be used for Quality control in industries. These Articulated Robotic arms will provide better accuracy in identifying and detecting the desired object and placing it in the desired place thus reducing human effort, error, and product cost in the industry.

Some specifications of the solution include:

- This design is based on RRR an Articulated Robotic Arm which is simple and easy to manufacture and requires fewer parts.
- The links in this manipulator are optimized using topology, which reduces material consumption and the overall cost of the product.
- Fully 3D printed parts provide rapid and complex design in less time, with fewer parts meaning manufacturing, assembly costs, and time will be cut by half.
- Microcontroller or computer-based control system for reliable performance.
- Image processing and other sorting sensors are used for reliable and quick sorting purposes.

4. Process Flow Chart Of Manipulator Design And Control

All design criteria for the robotic arm are included in this research. The position of the end effector is then determined using MATLAB's kinematic modeling method. In this project, Kinematic modeling plays an essential role in the design and control of the manipulator. Robot arm kinematics examines the geometric motion of a robot arm from a fixed reference of coordinates as a time function without considering the actions that generate the motion. The model developed on MATLAB is also used to find joint velocities and torque required for motor selection. After mathematical modeling, a CAD model is developed and structure optimization of this manipulator is started, after finalizing the CAD model manufacturing and control system design take place.

5. Mathematical Modelling Manipulator

Details of mathematical modelling according to the concept of Robotics and simulation of design model from MATLAB is as follows.

5.1 Workspace

The workspace of a robotic manipulator is the set of points that can be reached by its end-effector. In other words, it is simply the 3D space in which the robot mechanism works. It is important to note that no kinematic solution exists for the manipulator's configuration or joint space for any desired end-effector position outside of the workspace.

5.2 DH Parameters

Following Denavit Hartenberg DH table is used to find joint parameters of RRR type 3-DOF manipulator.

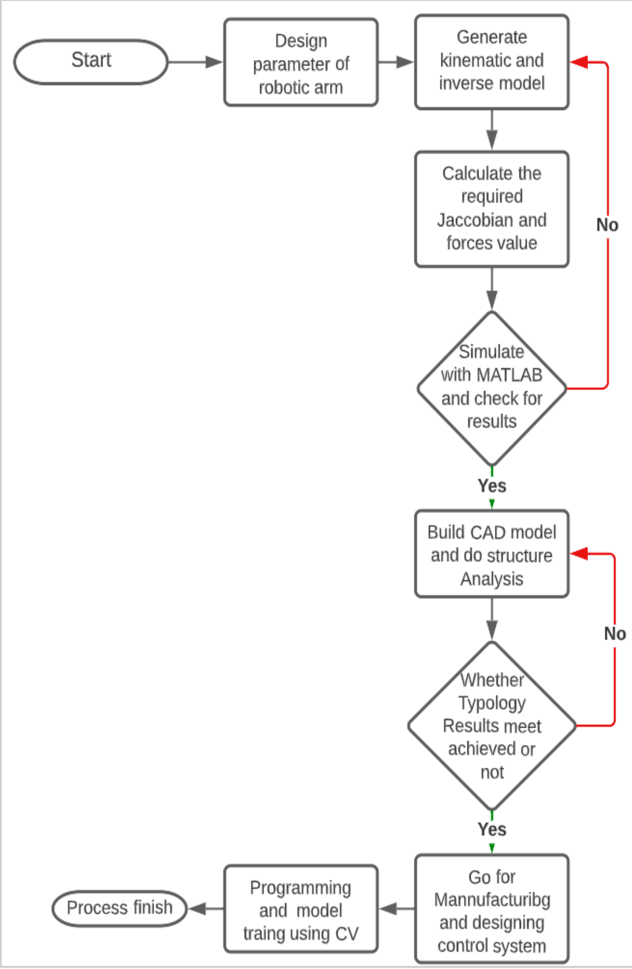

Fig. 1. Flow Chart For Manipulator Design And Control

Table 1

DH Parameters

DH Table RRR with 3-DOF						
J	theta	d	a	alpha	offset	J
1	Q1	10	0	$\pi/2$	0	1
2	Q2	0	35	0	0	2
3	Q3	0	30	0	0	3

The transformation matrix obtained by solving using this DH table will help in obtaining the forward kinematics.

5.3 Forward Kinematics

Forward kinematics is used to find the position and orientation of the end-effector in the C-space once the parameters of the actuators such as rotation angles are given. When the user defines the rotation angle of all joint actuators the robot moves in that position and the end effector reaches a specific location having specific coordinates at that position.

Forward kinematics for this manipulator is found by using the transformation matrix that's found from the above DH table by using general transformation matrix form and at angles

$$(\boldsymbol{\theta 1} = 0,\ \boldsymbol{\theta 2} = 90,\ \boldsymbol{\theta 3} = 45)$$

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & \alpha_{i-1} \\ \cos(\alpha_{i-1})\sin(\theta_i) & \cos(\alpha_{i-1})\cos(\theta_i) & -\sin(\alpha_{i-1}) & -d_i\sin(\alpha_{i-1}) \\ \sin(\alpha_{i-1})\sin(\theta_i) & \sin(\alpha_{i-1})\cos(\theta_i) & \cos(\alpha_{i-1}) & d_i\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The transformation matrix from Base to the End-effector for RRR manipulator is for forward kinematic will be

$$ {}^{0}_{3}\boldsymbol{T} = \begin{vmatrix} -0.7074 & -0.7068 & 0 & -22 \\ 0 & 0 & 1 & 0 \\ 0.7068 & -0.7074 & 0 & 66 \\ 0 & 0 & 0 & 1 \end{vmatrix} $$

The forward position of the manipulator according to this transformation matrix is obtained by MATLAB and is shown in Fig. 2.

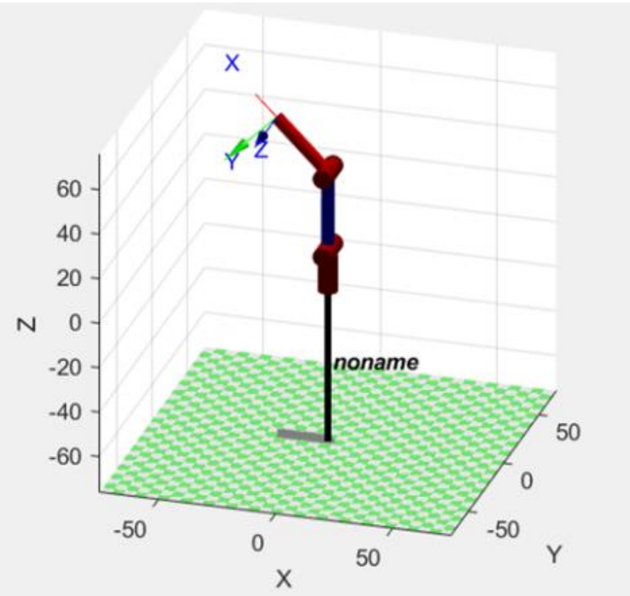

Fig. 2. Forward Kinematics Graph

5.4 Inverse Kinematics

Inverse Kinematics is used to find the backward motion of the robot after moving it forward with the help of forward kinematics. In inverse kinematics, the end effector coordinates in space are known and thus joint angles are found to move the robot to the previous position. Inverse kinematics can give several possibilities for the robot to move backward, and the programmer must choose the best one. The angle of the tool w.r.t. to the world frame can also be written as:

$$P_x = \cos\theta_1\,(a_3\cos(\theta_2+\theta_3) + a_2\cos\theta_2)$$
$$P_y = \sin\theta_1\,(a_3\cos(\theta_2+\theta_3) + a_2\cos\theta_2)$$
$$P_z = a_3\sin(\theta_2+\theta_3) + a_2\sin\theta_2 + a_1$$

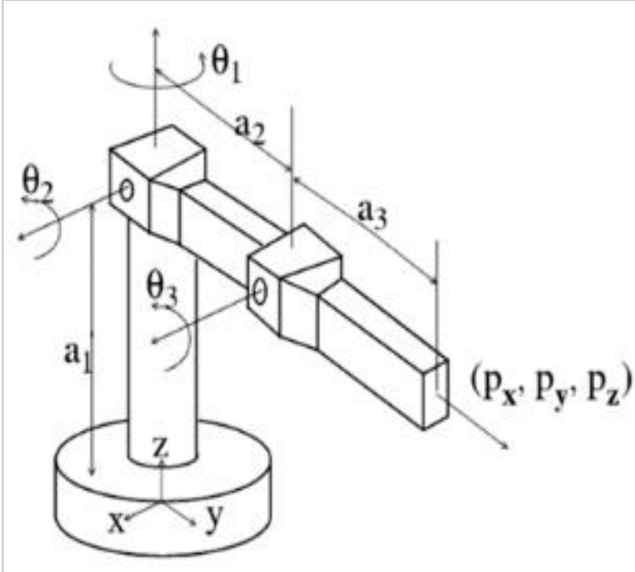

Fig. 3. Manipulator Kinematics Parameters

The inverse kinematics for this manipulator is calculated at the position

$$\alpha = \sqrt{p_x^2 + p_y^2}$$
$$\theta_1 = atan\left({}^{p_y}/{}_{p_x}\right)$$
$$\theta_2 = atan\left[\frac{\beta}{\pm\sqrt{p_x^2+\alpha^2-\beta^2}}\right] - atan\left[\frac{\alpha}{p_z}\right]$$
$$\theta_3 = atan\left[\frac{p_z - a_2\sin\theta_2}{\alpha - a_2\cos\theta_2}\right] - \theta_2$$

Table 2

Inverse Kinematics

Position	Values (cm)	Angles obtained by inverse kinematics are	Angles	Values Deg
Px	-22		$\theta 1$	0
Py	0		$\theta 2$	90
Pz	66		$\theta 3$	45

5.4.1 GUI design on MATLAB

A Graphical User Interface (GUI) development environment offers a set of tools to generate (GUI). It greatly facilitates the operation of designing and building GUIs. A GUI example has been prepared with the help of [3] GitHub code for 3 DOF Robots including the forward and inverse kinematics. GUI is given in the following figures. Push buttons, sliders, axes, etc. can be added to it. Additions are visible on the m. file simultaneously as a function.

Robotics Toolbox is embedded into GUI. The results are compared with the expressions obtained in the analytical solution. It is proved that the same results are obtained by the robotic toolbox and the analytical solution.

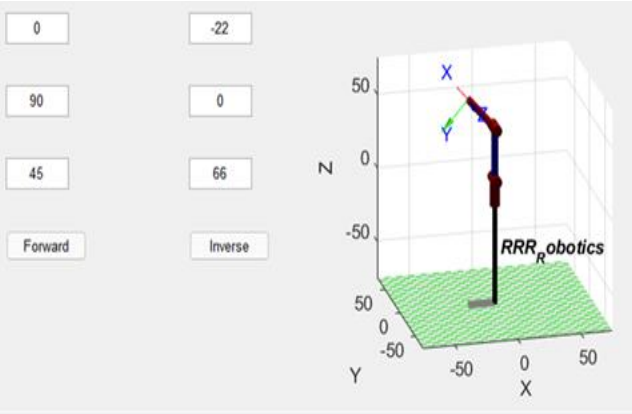

Fig. 4. GUI Layout And Interface

5.5 Jacobian

Jacobian is a Matrix in robotics that provides the relation between joint velocities (q) and end-effector velocities (X) of a robot manipulator. If the joints of the robot move with certain velocities, then we might want to know with what velocity the end-effector would move. Here is where Jacobian comes to our help. The relation between joint velocities and end-effector velocities is given below.

$$X = J\dot{q}$$

- q is the column matrix representing the joint velocities. The size of this matrix is nx1. 'n' is the number of joints of the robot.
- X is the column matrix representing the end-effector velocities. The size of this matrix is mx1. 'm' is 3 for a planar robot.
- J is the Jacobian matrix which is a function of the current pose. Size of Jacobian matrix is (m x n).

Since J is related to linear velocities of the end-effector due to joint velocities, we can get the J` by derivation of the position functions for x, y, and z of the end-effector w.r.t joint variables [q1, q2, q3] is shown above in Fig. 4. at (θ1= 0, θ2=90, θ3=45) Then Arm Jacobian is:

$$J_{arm} = \begin{bmatrix} -s_1(a_2c_2 + a_3c_{23}) & -c_1(a_2s_2 + a_3s_{23}) & -a_3c_1s_{23} \\ c_1(a_2c_2 + a_3c_{23}) & -s_1(a_2s_2 + a_3s_{23}) & -a_3s_1s_{23} \\ 0 & a_2c_2 + a_3c_{23} & a_3c_{23} \end{bmatrix}$$

$$J = \begin{vmatrix} -0.0000 & -56.9206 & -21.9206 \\ -21.9198 & -0.0000 & -0.0000 \\ 0.0000 & -21.9198 & -21.9200 \end{vmatrix}$$

5.6 Singularities

As shown in research [4] Journal of the University of Babylon for Engineering Sciences, determined that a Jacobian base with an end-effector equal to zero gives us Singularities. Two types of singularities can occur in our manipulator.

$$Det\ (J) = [\boldsymbol{l_2C_2 + l_3C_{23}}][\boldsymbol{l_3\ l_2\ S_3}]$$

$$l_3 \text{ and } l2l_3 \text{ and } l_2 = 0 \ldots \text{ gives lose link length}$$

$$S_3 = 0 \ldots \text{Elbow arm fully extended}$$

Elbow Singularity- Occurs when Sinϑ_3 becomes 0 or π.

Shoulder Singularity- $[\boldsymbol{l_2C_2 + l_3C_{23}}]$ It occurs when the wrist center interests the axis of the base rotation z_o, No matter how ϑ_1 rotates, the solution will be infinite.

5.7 Torque

For manipulator torque calculation we use the following relation

$$\begin{pmatrix} \tau x \\ \tau y \\ \tau z \end{pmatrix} = \boldsymbol{J} \times \begin{pmatrix} \boldsymbol{Fx} \\ \boldsymbol{Fy} \\ \boldsymbol{Fz} \end{pmatrix}$$

To estimate the torque required at the Base joint, the worst-case scenario must be chosen. In Fig. 2 the manipulator position at an angle ($\theta 1 = 0$, $\theta 2 = 90$, $\theta 3 = 90$) is given and this is the position where maximum torque can occur.

The only force acting on the manipulator is the weight it carries with the end-effector whose direction is along the Y-axis, so forces along the X-axis and Z-axis are zero. The maximum capacity of this manipulator is considered (5N) and after considering the safety factor and end-effector weight the total force is 7.5N. So, the torque obtained for the base motor is -232.4983 N cm. While considering the gear ratio at link-2 which is 1:4 the torque required at the motor will be 52 N cm.

6. Design and Analysis

The CAD model has been developed using Solid Works. Each part has been made individually and then assembled in Solid work with pin joints, merge and sliding mechanisms, etc. A detailed CAD model of the manipulator, conveyor belt, and control box follows.

Fig. 5. CAD Model

6.1 Topology/Shape Optimization of Links

The procedure of topology and shape optimization is performed on link-2 and link-3 with the help of Fusion 360 and Finite Element Analysis using the ANSYS Solid structure toolbox. This optimization will allow the link to carry the same load using 30% less material which will reduce cost, weight, and time for 3D printing.

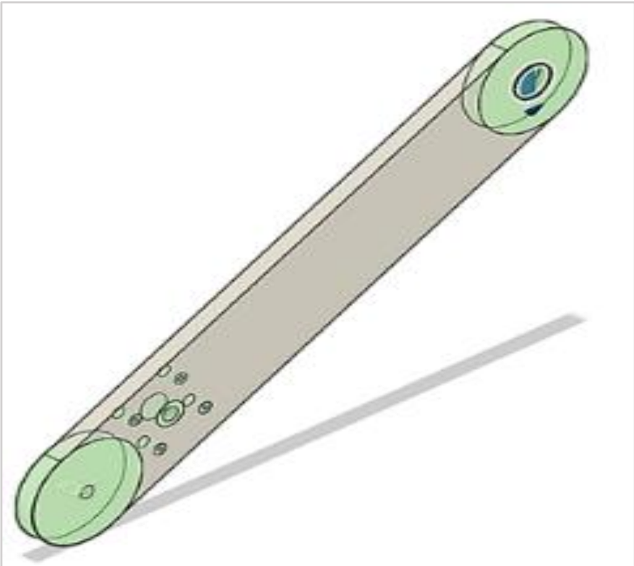

Fig. 6. Initial Process of Topology

6.2 Link-2 Shape Optimization And Analysis

Link-2 is fixed at one end where it will be attached to the shaft and a force of -15N is applied on the end where link-3 will be attached. The green regions show preserved regions where no material removal and Topology Optimization will be performed. The criteria given for topology is to maximize the link stiffness and remove approximately 30% material

Note that the Topology optimization operation only gives us a mesh, so the CAD model is made from that mesh. Some refinements in the model such as applying fillets to sharp corners are also performed during the making of CAD model from mesh. The mass of original Link-2 found by SolidWorks Mass Properties is 225g. The mass of Optimized Link-2 by SolidWorks Mass Properties is 164g. Thus, after Topology Optimization and adjustments in the optimized link. We have reduced link-2 mass by approximately 28%.

Fig. 7. Topology Process

Now for structure Analysis mesh generation process is performed using FEA on ANSYS. Sectioning is done on link-2 to ease the processes of making structured mesh and refining mesh gradually from holes to outward direction. Forces applied on link-2 is -15N where link-3 will be attached and -1.5N force is applied on all small 4 holes where the motor assembly will be attached. The left end is fixed.

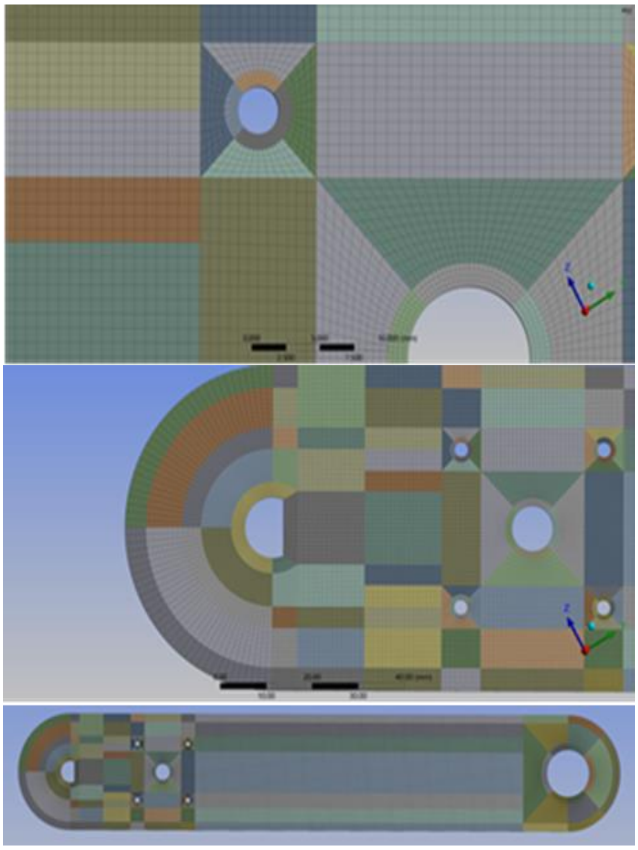

Fig. 8 Original Link-2 Meshing

Ansys Analysis is done for Original Link-2 and Optimized link-2 to observe the change in deflection after removal of 28% mass. For (Original Link-2) Mesh Independence is achieved at about (2511443) Nodes at which the Maximum Total Displacement is 0.56403mm.

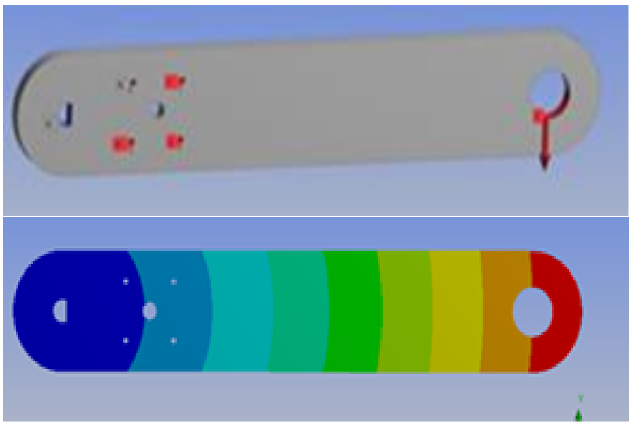

Fig. 9. Total Deformation Result of Original Link-2

Red colored areas are under high stress and the blue colored area is under low stress. It is obvious that it is the joint area.

Table 3

Original link-2 Mesh independence

No	Number of nodes	Deformation (mm)	Number of elements
1	2511443	0.56403	589274
2	2212841	0.56399	517036
3	2046967	0.56394	478062
4	1688826	0.56387	392256
5	1516800	0.56382	352128
6	1021242	0.56373	234000
7	842629	0.56353	192730
8	494582	0.56324	110848
9	333156	0.56284	73556
10	228703	0.56251	49578
11	139281	0.56211	29380
12	75104	0.56103	15248

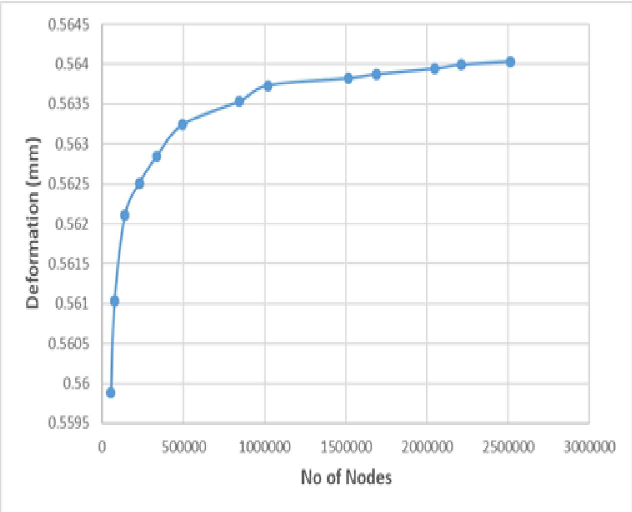

Fig. 10. Original Link-2 Mesh Independence

For (Optimized Link-2) Mesh Independence is achieved at about (1991322) Nodes at which the Maximum Total Displacement is (0.68052) mm.

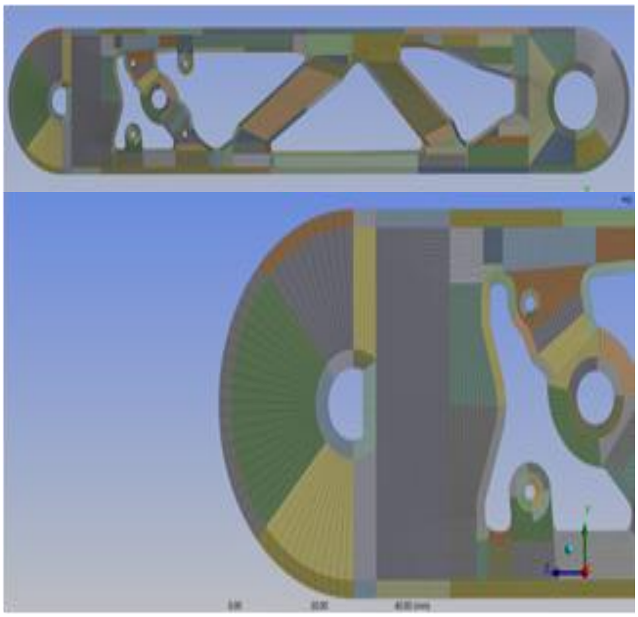

Fig. 11. Optimized Link-2 Meshing

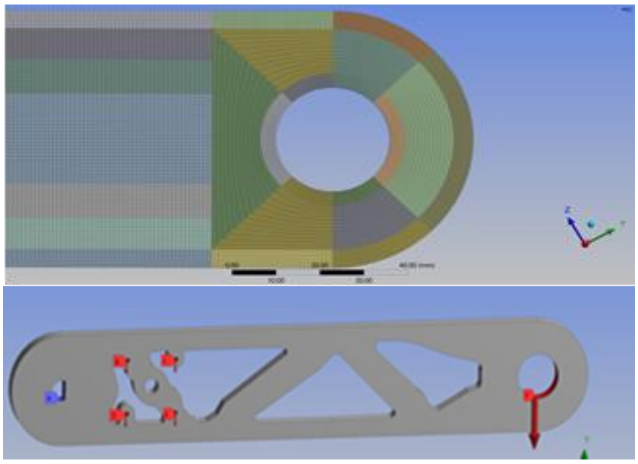

Fig. 12. Location Of Fixed Support And Forces Applied

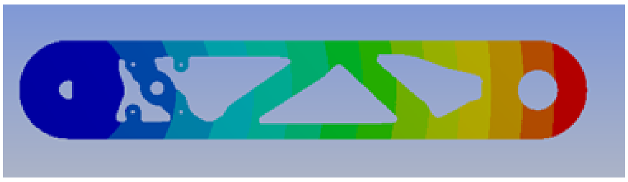

Fig. 13. Total Deformation of Optimized Link-2

Red colored areas are under high stress and blue colored areas are under low stress. It is obvious that it is the joint area.

Table 4

Optimized Link-2 mesh independence

No	NUMBER OF NODES	DEFORMATION (mm)	NUMBER OF ELEMENTS
1	1991322	0.68052	460600
2	1724609	0.68049	396903
3	1660637	0.68045	382044
4	1357801	0.68039	310332
5	1220002	0.68033	278472
6	826578	0.68018	185720
7	673400	0.68005	150700
8	413287	0.67994	90392
9	270773	0.67982	58086
10	188576	0.67961	39588
12	121147	0.67919	24675
13	64886	0.67798	12568

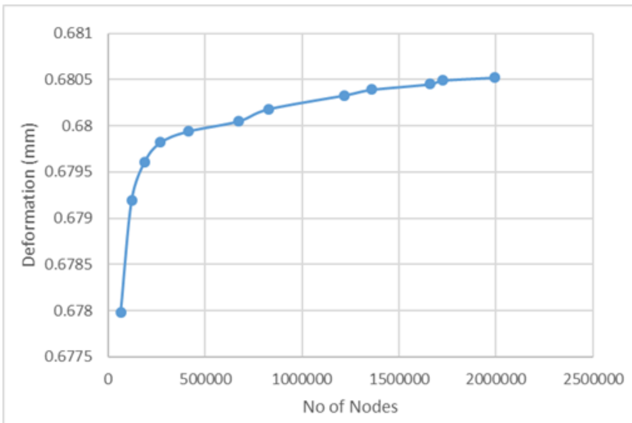

Fig. 14. Optimized Link-2 Mesh Independence

7. CFD Analysis of Control Box

CFD analysis is done on a controller and motor driver's enclosure box with dynamics meshing. This box has a fan attached outside which causes forced airflow. The purpose is to observe the flow pattern of air under the influence of a cooling fan at a particular position. The simulation will help observe flow patterns of air when the fan sucks air outside from the box and while air is entering the box through the vent due to the pressure differences created inside the box due to the cooling fan. The fan is given -265rad/sec velocity with which the fan sucks air outside the box. Two boundary conditions at Inlet and Outlet are defined one as pressure-outlet (as no velocity inlet is required), the fan will automatically suck air from the surrounding, and the other as gauge pressure is set to zero meaning the atmospheric pressure is 1atm. After calculating the required time steps for setup, the velocity contours, streamlined air flow animations, and volumetric renderings for air velocity are made for visualization and analysis of airflow in this case. The case is run with 0.0005s time step having 40 iterations, each time for suitable calculations. These settings helped achieve airflow visualization better. The setup in which the fan forces air outside the box is suitable for airflow cooling purposes inside the electronic box and it excellently stops air accumulation inside the box.

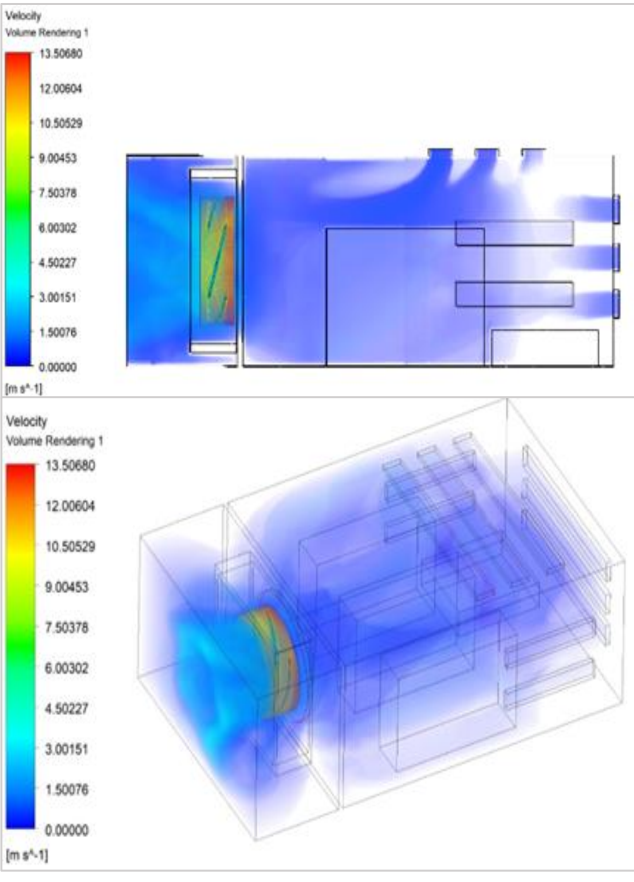

Fig. 15. Volume Rendering

7.1 Cell Zone Conditions

In cell zone conditions both fluid domain and rotating fan enclosure are selected. The fluid domain is kept stationary while the rotating fan enclosure cell zone is given mesh motion with a rotation axis of Y. The Cartesian coordinate is made at the center of the fan-rotating mesh. The rotation axis direction along which the rotating fan enclosure will move is set at Y with -141rad/. The negative sign is added so the fan will rotate in an anticlockwise direction about Y and suck out air from inside the electronic box and exhaust it into the atmosphere. In this CFD analysis as a rotating fan influences the fluid domain and Mesh motion is used for movement.

7.2 Results and Discussion

The results of contours and streamlines and volumetric renderings we can observe when the fan forces air outside the box the air stream enters through inlets due to pressure difference continuously and does not accumulate inside the box. This allows for continuous heat removal from drivers. The streamlines also show that the placement of inlets allowed the air to move around all 3 stepper motors on all sides allowing for good heat removal. The max velocity of air is 13.5m/s as it interacts with fan blades. Fan blades force air out with increased velocity of air. Thus, the fan reduces pressure inside the box which causes more air to enter through the vents.

After performing Ansys Fluent dynamic mesh analysis for forced air flows out of the box it is clear that this setup is efficient for airflow cooling of the electronic box for heat removal purposes. The fan moved air inside the box consistently, did not allow air to accumulate, created pressure difference to allow fresh air to come inside the box continuously, and distributed air stream flow around the 3 stepper motor drivers more equally.

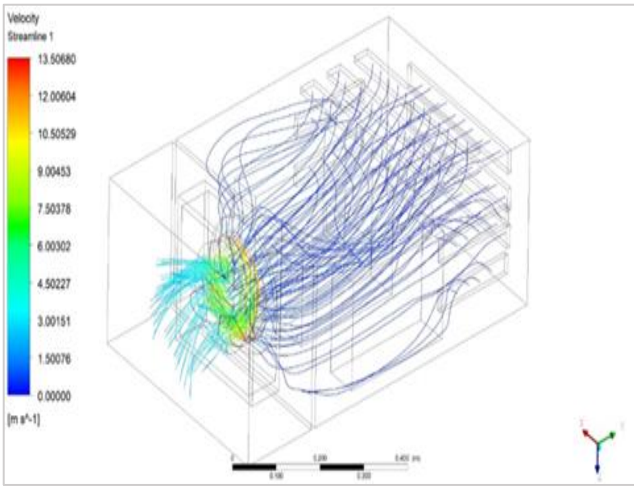

Fig. 16. Velocity Streamline

8. Manipulator Control System

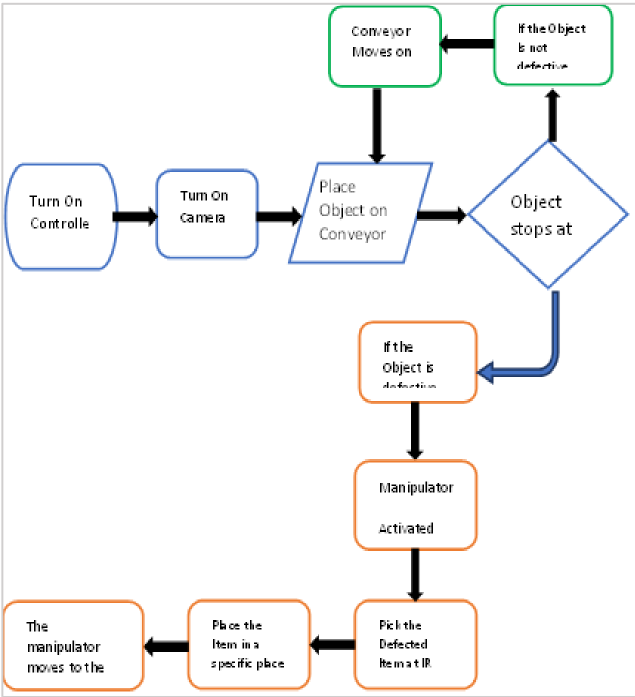

Fig. 17. Manipulator Workflow Chart

8.1 Motors And Control System

According to calculated torque and including the safety factor stepper motors like Nema-23 and Nema -17 are used in all links as these motors provide precise positioning, great speed control, and repeatable movement. For Gripper, a servo motor is used as this motor provides a high ratio of torque to inertia, more constant torque at higher speed, and closed-loop control which helps in gripping objects of any shape with constant force.

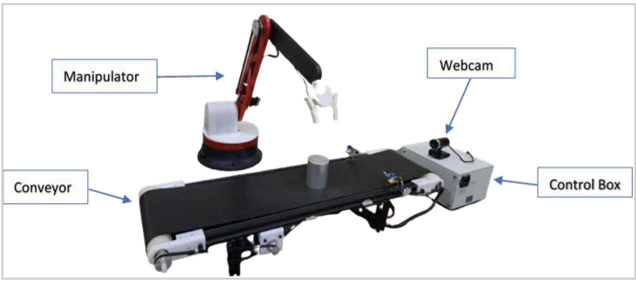

Fig. 18. Project Setup

The following are the types of controllers and control systems used in this project.

1. Arduino for sensor data accusation and motion control (Kinematic codes)
2. Python and Open CV for image processing.

8.2 Micro-Controller

There are different types of microcontrollers available in the market. In this project, Arduino is selected because it's simple to program and link with different electronic components and is easily available in the market at a relatively cheaper price. Arduinos are also easy to link different languages and applications which makes them quite flexible for programming and usage.

In this project, two Arduino UNO are used for system control. One for controlling the manipulator and the other for the conveyor belt. Arduino Mega can be used instead but the conveyor is used as a separate system from our manipulator. So, the manipulator can be used for different applications independently.

First Arduino is used to control the motion and path trajectory of the manipulator. The trajectory plan is done by using inverse kinematics. So, only Jacobian (joint velocity), acceleration, and position of every link motor are specified, and Arduino converts it into forward kinematics and approaches these angles given by position points. It also controls the logic of our sorting mechanism which in this case is color base sorting by using a color sensor that detects the color depending on a specified wavelength in the program and provides a specific signal to Arduino and this signal will determine which path to follow.

Second Arduino is used to control the motion of our conveyor belt which is equipped with one stepper motor for conveyor movement. This stepper motor can move the belt at various speeds but for the current setup, we have set the speed to 0.2 m/s (for demonstration purposes). One IR sensor is also used in this conveyor belt to stop the motion of the conveyor as some things encounter it. This will allow the system to detect the color of the object and the manipulator to pick the object and put it into a designated place. As the object is picked by the manipulator now the conveyor starts again new objects come to the conveyor.

8.3 Python And OpenCV For Image Processing

The color sorting concept is also implemented using OpenCV (python) in which we have trained our algorithm for detecting and sorting different colors based on the trained model. All this is done by training our system for both positive and negative data. In positive data, we have 400-1000 images of color that we want to choose and negative data includes colors that we don't want to choose then we train our algorithm by using Harr Cascade GUI and making a harr cascade file of that trained data. Then this trained data is imported into the OpenCV codes. In OpenCV codes, webcam parameters and desired object that needs to be detected is defined, then a while loop that will link with Arduino through the Serial package in Arduino, OpenCV is linked with our Arduino program with the help of the serial package and links the results of OpenCV with the specific trajectory of the manipulator.

8.4 Specific Image Detection

In a review of some previous research [6] . A specific defect in a particular product in the mass production line can be detected using Computer Vision and a manipulator can pick that defect. Item and remove it from the production line. So, ensure the quality control on a conveyor line. This can be achieved by training algorithms with positive images of defects and negative of non-defective items and things related to that defective item. Then this train Harr Cascade can be imported into Python and by using OpenCV through webcam that specific defect can be detected, and defective items can be picked by manipulator with specified trajectory.

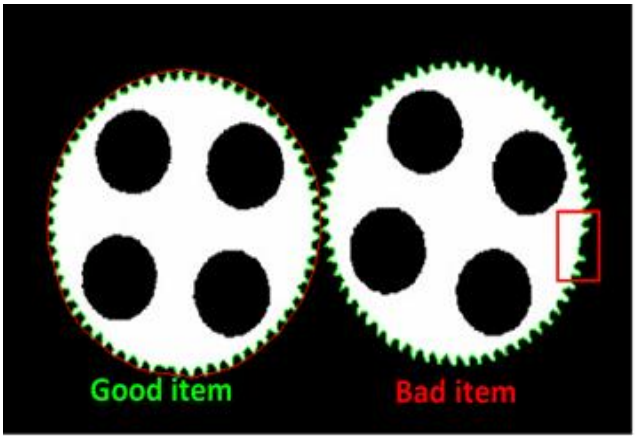

Fig. 19. Quality Control using Computer Vision

For this purpose, the algorithm is trained to detect a specific defective item. For simplification purposes, a simple Stop Traffic sign is taken as a defective item, and the algorithm is trained with positive images of a stop signs and negative images of things related to the top traffic signs. In Python (OpenCV) the harr cascade of train items is imported and linked with Arduino through the serial package. A while loop has been created to link the logic of defective items with a manipulator. Objects can be easily sorted through a webcam and a signal is sent to Arduino for running a specific trajectory to remove the defective item from the conveyor belt.

Fig. 20. Live Object Detection Using Webcam

9. Results / Presentation Of Findings

The following are the findings of the project including both hardware and software results in a detailed manner.

9.1 Hardware Results

The design of the manipulator was studied in a detailed manner before going into the manufacturing stage so there isn't any problem related to manipulator movement during operation. The manipulator performs smooth movement in all directions during both low and high-speed operations. Manipulator motors work on the optimum loading providing the necessary torque required for the smooth operation of the manipulator. The details about manipulator movement and step calculation are as follows. (Noting that each step is equal to 1.8 degrees of revolution).

Table 5

Link steps calculations

Link motor	Micro-stepping (driver)	Steps required for 360 degrees	Gear ratio	Total step required for 1 revolt
Link 1 (base)	1/8 step= 0.125 degree per step	2880	5	14400
Link 2 (shoulder)	1/8 step= 0.125 degree per step	2880	4	11520
Link 3 (elbow)	1/8 step= 0.125 degree per step	2880	1.5	4320

The shape optimization has helped us in reducing the weight of linkages and loading on the motors. This allowed low power consumption and smooth movement of the manipulator. The gripper uses using servo motor so angle calculation for movement is very simple. The gripper design is optimum according to manipulator size and loading capacity as there isn't any issue related to opening and closing of gripper jaws and rubber padding on the gripper provides reliable grip during operation.

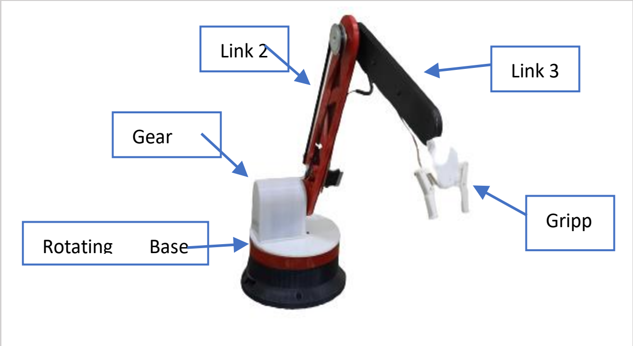

Fig. 21. Manipulator Main Parts

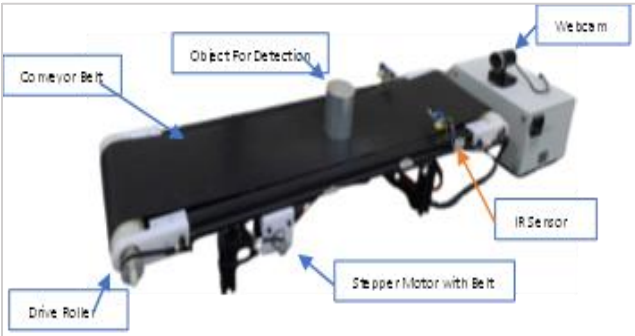

Fig. 22. Conveyor Main Parts

Manipulator hardware is properly sinks with software as it performs pick and place operation responsively and proper current and voltage are provided for all components of the manipulator. Along with the manipulator, the conveyor belt is also working properly as the conveyor stepper motor is programmed to run continuously with a defined step rate until some IR sensor is triggered. If the IR sensor is engaged the conveyor will remain stopped. This allows the sorting mechanism to sort the object and signal the manipulator to pick and place it in the specified place.

The maximum workspace use in this project is around 46 cm Feet and the designed Workspace is about 61 cm feet as it depends on the tilt angle of the factor. The end factor used in this project is a gripper with a gear mechanism. Some positions of manipulators in the workspace and overall workspace bubble were used in this study.

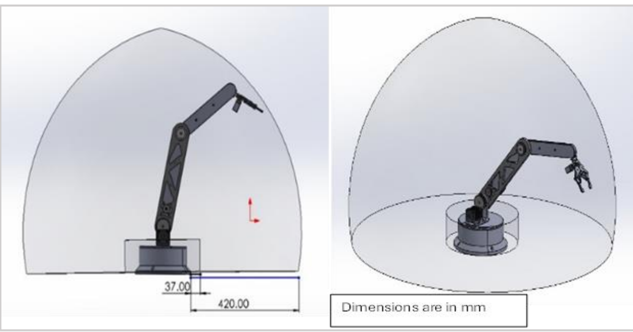

Fig. 23. Workspace Bubble

9.2 Software Results

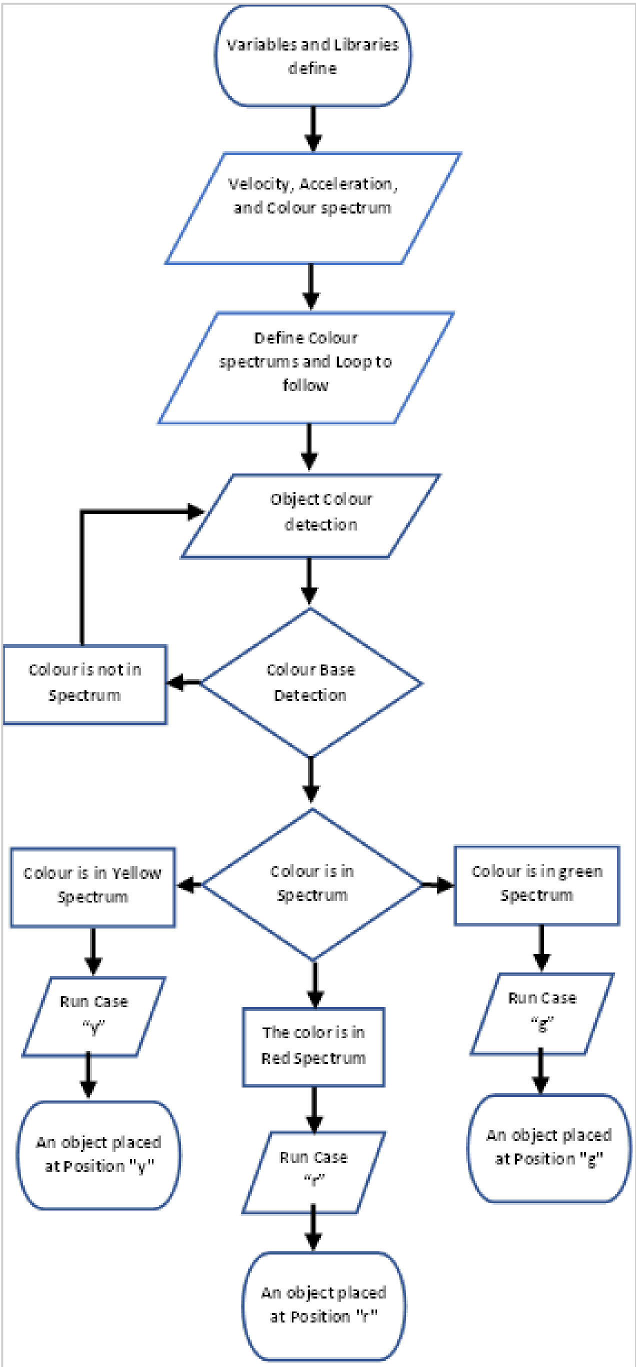

Fig. 24. Color Sensor Workflow Chart

As in this study, the task is to sort the objects and pick different objects to different locations based on their type. A microcontroller (Arduino) is used for motion control and logic control of both manipulator and conveyor the results are very pleasing. The manipulator motion is smooth, and the velocity and acceleration of manipulator links are properly controlled by Arduino as required by the user. The logic of both color sensors and Computer vision is properly working with the help of Arduino. The color sensor detects colors defined in specific wavelengths and controls the different trajectories based on the color of the object on the conveyor it runs that stored trajectory. This type of control allows quick sorting of objects.

Fig. 25. Color Base Sorting

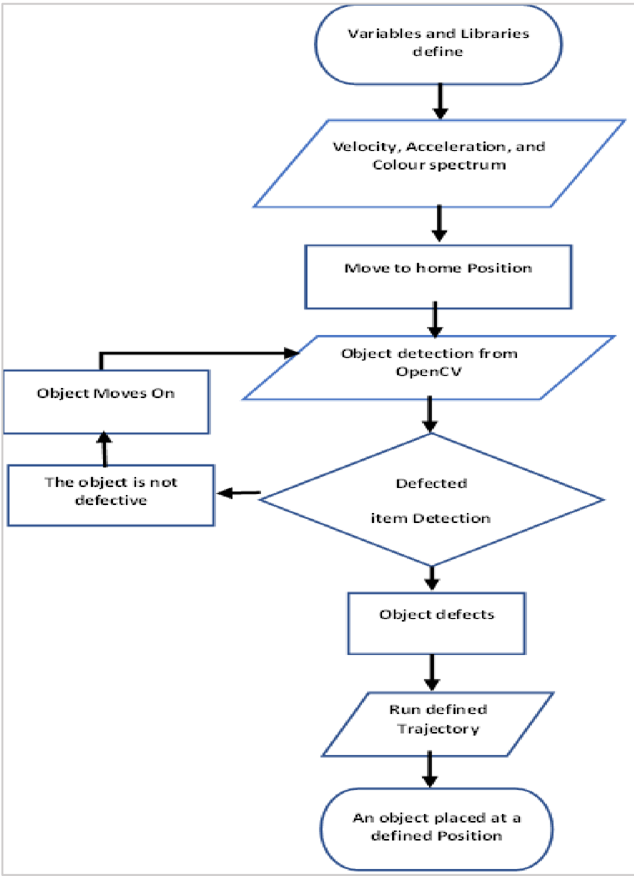

Fig. 26. Computer Vision Control Flowchart

Where the defective item detection through computer vision is properly working with Arduino. Whenever a defective item is detected through the webcam the OpenCV sends a signal to the manipulator to perform a predefined trajectory. In this case, the stop sign is defined as a defective item so whenever the camera detects this sign it labels it as a target on the computer screen and sends a signal to Arduino to run the planned trajectory to move this item from the conveyor and for any item without this sign (defect) Is allowed to pass through the conveyor.

Fig. 27. Object Detection And Placement

The conveyor is also linked with a separate Arduino that controls the stepper motor and link the stepper motor with the IR sensor on the conveyor to stop and start the conveyor whenever there is a change in value of IR sensor as its value changes between 1 and 0.

9.3 System Results and Accuracy

To test the accuracy of the system different tests have been performed. Some of the tests include testing the system in different lighting conditions to check the accuracy of sensors and cameras. The conclusion from the lighting conditions is explained in the project limitations. Moreover, for Object detection, numerous sorting attempts have been performed to check working proper working of the manipulator (as the accuracy of our computer vision program is around 96%). For this purpose, 4 sets of testing is performed and each set includes 25 attempts to sort and place the defective object the results for Object-based Image processing sorting and placement attempts are shown in Table 6.

Table 6

Accuracy of test result

Set	Attempts	Right	Wrong	Accuracy
1	25	22	3	88%
2	25	25	0	100%
3	25	23	2	92%
4	25	24	1	96%

From the table and graph, one can see that there are some false triggers in our system which is because of camera sensitivity, ambient lighting, and limited data of the trained model. The overall average accuracy of our program which is at 96%.

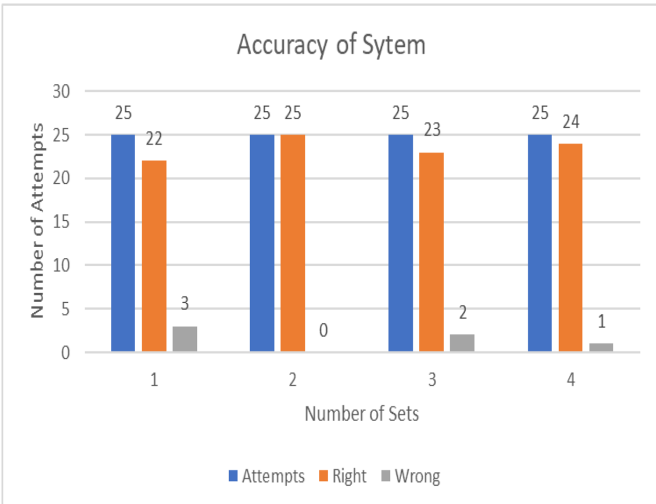

Fig. 28. System Accuracy Testing

The false triggers in the system can be eliminated by changing the setup to optimum environmental conditions and changing the focal length of the camera according to the distance of object identification.

Along with accuracy testing, the environment factor is also calculated to provide the best result for manipulator operation. Following data obtained from studying of different environmental factors affecting the manipulator Computer vision base sorting, these values are tested to provide the best sorting result.

Table 7

Calculate the optimum value for Environmental factors

Factor	Min Range	Max Range
Illuminance Level	100 lux	1000 lux
Contrast Ratio	10:1	100:1
Noise Reduction	30	70
Distance Range	1 cm	30 cm

After integrating this value in the setup, results are obtained with 98% accuracy, which is a good sign when using a simple microcontroller instead of industrial PLCs.

After that, the Manipulator was also tested with color-based sorting to check the accuracy in detecting different colors. Under the lighting condition of 50-70 lumens per square foot, there isn't any error in color detection and sorting as the wavelength of colors is defined for this type of lighting condition. So, the color base sorting has 100 % accuracy if the lightning conditions are met.

10. Limitations And Solutions

10.1 Trajectory Plan

A proper predefined trajectory plan is required for manipulator motion. As if an object is needed to pick and place from a specific location. A predefined trajectory must be calculated and stored in the Arduino.

10.2 Environmental Limitations

A specific environment must be provided for the proper working of the equipment. As IR sensors work on IR radiation, working sunlight can cause issues with IR sensors working. The same camera and the color sensor work based on color detection based on color spectrum distribution. So, working in sunlight and bright light can cause issues with proper object detection and can cause false detection. So an indoor environment with cool light is better for proper working equipment.

10.3 Load Capacity

The motor calculations are done for a maximum carrying load of 500g. This limit can be increased by using higher torque stepper motors like Nema-23 or Nema-44 stepper motor instead of Nema-17. Also, the gripper servo motor can be replaced with a higher torque motor.

10.4 Material Limitation

As most of this manipulator is 3D printed with PLA, it works in temperatures below 45 Degree Celsius. So, for operation at a temperature of 45 Degrees Celsius, other materials should be used. Where ABS can be used to increase the operation temperature range.

11. Conclusion And Future Recommendation

11.1 Conclusion

The purpose of this project was to familiarise the local industries and engineers of Pakistan with the fact that automation and robotics an inevitable change that are needed to compete with international products. The aim was to study the industrial manipulator model process, analyze the efficient structure for manipulator design, and then design an inexpensive and easy-to-control manipulator that can perform real tasks and could be used to perform the subject tasks without error or exhaustion with the help of computer vision. Therefore, this industrial manipulator was designed using 2 links only with rotation at the base. Such a simplistic design was adopted to be used for a wide range of tasks such as pick and place, sorting, spot welding, and arc welding. Painting at an accurate and low cost. The workspace, singularities, dynamics analysis (Jacobians), forward as well as reverse kinematics, and torques calculations were done. The motor selection based on torque calculations selecting link lengths based on the workspace CAD model for the manipulator has been made using SolidWorks software. The model includes a conveyor belt, the control box, and the manipulator: including the base and the gripper. For the control box, a CFD analysis was conducted, using Ansys Software for optimum cooling to increase the efficiency and safety of the manipulator. The CAD model was then put in a software, called Fusion 360, for topology analysis on the 2 links for reduction in weight and cost while maintaining the strength. The design was then again refined using solid works and then imported to Ansys software to ensure the stresses and strains on the joints and holes did not reach the yield strength. The manipulator was 3D printed with a honeycomb pattern to reduce weight and increase strength and rigidity. An IR sensor was employed on the conveyor machine to stop the machine for ease of pickup of the part. Arduino Microcontrollers were used to power and move the manipulator and the conveyor. For the sorting mechanism, a color sensor was employed as well as a camera for the computer vision aspect of the project for shape detection. The color sensor detects the colors red, blue, and yellow and sorts them to their respective places while the camera detects a certain stop sign sticker to sort that object to its designated place. The upgradation of the microcontroller from Arduino to Raspberry will further enable us to perform a greater amount of sorting with much more accuracy while doing so due to greater computational power and memory. The gripper sorts picks and places the object with an accuracy of 95%. Hence a low-cost 2-link manipulator has been designed, simulated, and implemented successfully. Taking this design technique one can make a manipulator of any other work like spot welding, arc welding, painting, etc.

11.2 Future Recommendations

Several modifications and improvements can added in the future, some of them are the following.

11.2.1 Proper GUI with multiple modes

The Robotic Arm currently has 2 modes for sorting applications. The first mode is color detection mode in which a color sensor is used to differentiate between different objects. The first mode uses Arduino Uno and Arduino sketch programming to run commands and control the Robotic Arm. The second mode is image recognition mode in which image processing with the help of Python (CV module) is used to recognize objects based on their shape, edge, text, and color. Both these modes require separate programs for running, thus in the future a custom Graphical User Interface can be developed which will enable both programs and other features to be integrated into one GUI, and switching between them becomes seamless.

11.2.2 6 DOF for robotic arm

The robotic arm currently has a 3-DOF design meaning it has 3 links of revolute type. The robotic arm can be upgraded to a 6DOF design which will give greater mobility and ability to follow complex trajectories and paths. This is especially useful in medical operation robotic arms.

11.2.3 Modular end-factor

Currently, the Robotic Arm has one gripper which enables it to pick and place objects and sort them. In the future, a universal clamp can be designed which will be fixed on link 2. Several attachments can then be used with this clamper such as a welder, cutter, FDM nozzle for 3D-Printing and laser cutting, etc. This modular end-effector design will allow for easy swap between different end-effectors and increase the application of Robotic Arm.

12. References

[1] Robotics history at IEEE web;https://robots.ieee.org/robots/unimate/

[2] Patil, Madhav. “PLC-based Robot Manipulator Control using Position-Based and Image-based Algorithm”. Global *Journals of Research in Engineering*, 17(H1), 25–33. (2017).

[3] Ahmed Abdel Basit; GUI Design for Manipulator Kinematics; https://github.com/AhmedAbdelbasit/Robotic-Manipulator.

[4] H. M. Alwan and Z. H. Rashid, “Investigation and Simulation the Kinematic Singularity of Three Links Robot Manipulator with Spherical Wrist”, *JUBES*, vol. 26, no. 8, pp. 77–87, Aug. 2018.

[5] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, “Substantial capabilities of robotics in enhancing industry 4.0 implementation”, Cognitive Robotics, vol. 1, pp. 58–75, 2021.

[6] G. Alnowaini, A. Alttal, and A. Alhaj, "Design and simulate robotic arm with computer vision for the inspection process", *International Conference of Technology, Science and Administration (ICTSA)*, 2021, pp. 1-6,

[7] G. H. Nasir and K. Jármai, “Research and application of industrial robot manipulators in vehicle and automotive engineering, a survey”, Vehicle and Automotive Engineering 2: Proceedings of the 2nd VAE2018, Miskolc, Hungary, Springer, 2018, pp. 611–623.

[8] M. H. Ali, K. Aizat, K. Yerkhan, T. Zhandos, and O. Anuar, “Vision based robot manipulator for industrial applications”, Procedia computer science, vol. 133, pp. 205–212, 2018.

[9] Ž. Šitum, S. Herceg, N. Bolf, and Ujević Andrijić, Željka, “Design, construction and control of a manipulator driven by pneumatic artificial muscles”, Sensors, vol. 23, no. 2, p. 776, 2023.

[10] H. G. Le, N. L. Ho, and T. Dao, “Design synthesis, modeling, control strategies, and fabrication methods of compliant grippers for micromanipulation and micromanipulator: A comprehensive review”, Robotics and ComputerIntegrated Manufacturing, vol. 92, p. 102893, 2025.

