



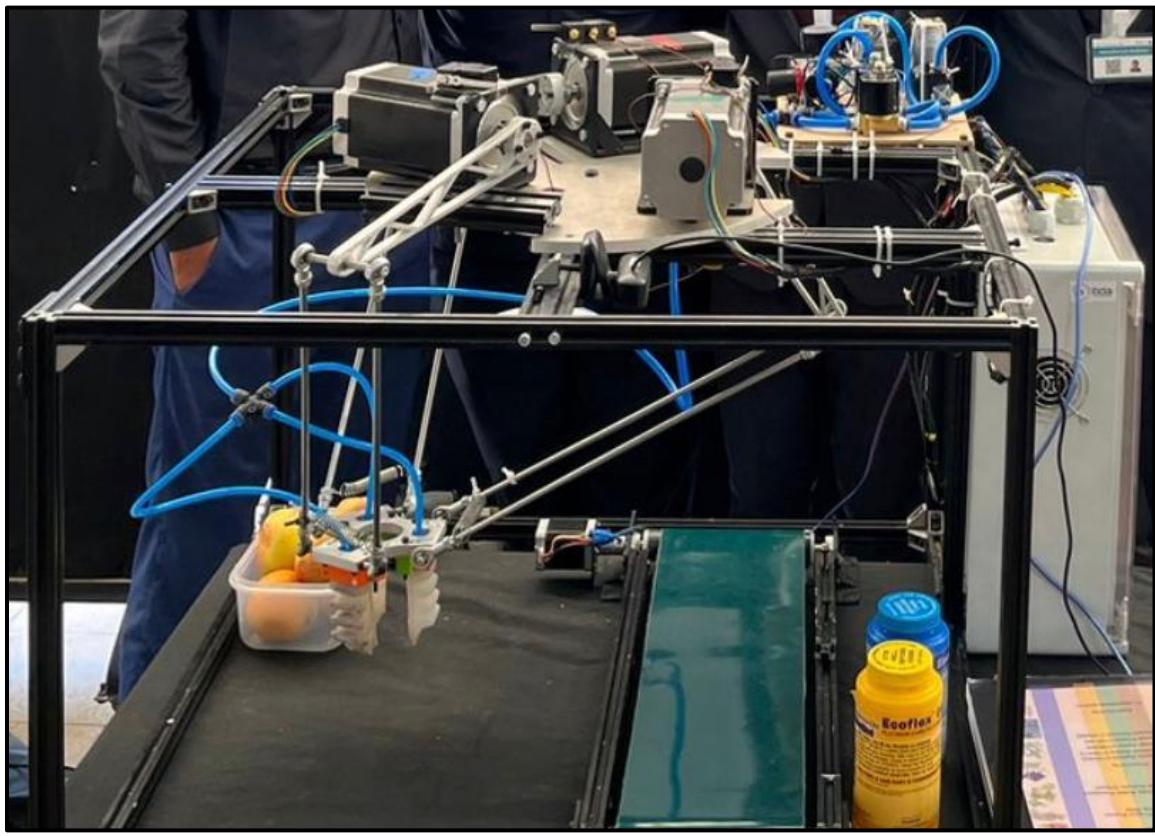
Cairo University  
Faculty of Engineering  
Department of Mechanical Engineering  
Mechatronics Program

# Pick & Place Delta Robot with a Soft Robotic Gripper

a Bachelor's Thesis Submitted by:

Abdulhamid Walid Abdulhamid  
Abdulrahman Mohamed Ismail  
George Ayman Mounir  
Hamza Walid Elreedy  
Mohamed Adel Hamzawi

Under the Supervision of: Dr. Abdelfattah Ibrahim



Final Project Image

## **Abstract**

This paper summarizes the whole process of innovating, designing, manufacturing and implementing our sorting and packaging system. In this report, we start by introducing our system, including an overview of the system components and features of the system, then we state our motivation behind choosing this topic as our graduation project.

We go through each segment individually. Starting with our main system manipulator “Delta Robot”, going through the details of design and fabrication with an iterative approach. Moreover, we will discuss the robot’s kinematic analysis and synthesis, and how we used it to come up with adequate robot dimensions that corresponds to the desired workspace. Then, the report mentions the whole process of calculating the motor sizing and specifications needed for our robot, using torque calculations and scenario analysis. We will also discuss the stepper motors’ control and how we used trajectory planning to limit motor jerk and vibrations. The robot simulation using SIMSCAPE environment will also be discussed, whether it’s the methodology or benefits of the simulation process.

Next is the soft robotic gripper section. We give a quick overview on soft robotics technology, literature review and the recent developments in the industry. Then comes the design methodology and iterations of our soft gripper, along with material selection based on hyperplastic material modelling, with the help of Finite Element Analysis using Ansys software. We will talk about our pneumatic control unit and how we use it to drive the gripper to inflate and deflate on command. And lastly, we will discuss the fabrication process of the soft gripper, including the construction of the mold and the casting process.

Then, we will discuss the computer vision of the system in details and system integration including, embedded systems design, electrical integration of the system, system workflow and the graphical user interface.

A github repository for all of the optimization, design, and real-time operations code is available at: <https://github.com/Abdulhamid0701/GP---FLEXPICK.git>

## Acknowledgements

We would like to thank the administration of Cairo University, Faculty of Engineering for their support of this project. We would also like to thank **Prof. Chahinaz Saleh** and **Dr. Moataz AlSisy** for managing and coordinating the graduation project processes.

We would very much like to express our sincere gratitude to our supervisor, Dr. Abdelfattah Ibrahim, for his guidance and support throughout the project. His expertise and knowledge were essential to the success of our project. We are extremely grateful for their help and support.

Special thanks to our graduation project sponsor & supporters Sector B5 and ITIDA, which are a part of the Egyptian ministry of information technology initiative Egypt Makes Electronics “EME” for their invaluable help and support throughout the project.

# Contents

Introduction.....	14
1.1    Project Overview .....	14
1.2    Motivation.....	14
1.3    Engineering Approach .....	15
1.4    Report Layout .....	16
Delta Robot.....	17
2.1    What is a Delta Robot Configuration .....	17
2.2    Kinematic Analysis & Dimensional Synthesis .....	18
2.2.1    Inverse & Forward Kinematics .....	18
2.2.2    Workspace Formulation.....	24
2.2.3    Final Dimensions & Parameters .....	29
2.3    Mechanical Design.....	30
2.3.1    Material Selection.....	30
2.3.2    Design Iterations & Components Optimization.....	31
Validation with normal loading for the whole robot.....	35
Validation with beam test (as if it was a beam fixed on two supports).....	35
Other Active Arm Design Modifications .....	36
2.3.2    Frame Design .....	38
2.4    Actuator Sizing & Selection.....	41
2.4.1    Torque Calculation Methodology.....	41
2.4.2    Load Torque Calculation .....	41
2.4.3    Inertial Torque Calculation .....	42
2.4.3    Final Motor Selection .....	43
2.5    Simulation & Control .....	44

2.5.1	SIMSCAPE Simulation .....	44
2.5.2	Trajectory Planning and Point to Point Motion.....	46
2.5.3	Scenario Analysis.....	52
2.6	Problems Faced .....	57
	Soft Robotic Gripper .....	58
3.1	What is a Soft Robot.....	58
3.2	Our Project Application .....	59
3.3	Pneumatic Networks Construction .....	60
3.3.1	Choosing an Actuator Construction.....	60
3.3.2	PneuNets Principle .....	60
3.4	Gripper Design Methodology .....	62
3.5	Material Selection & Hyperelastic Material Modeling .....	63
3.5.2	Material Selection.....	63
3.5.3	Hyperelastic Material Modeling .....	64
3.5.4	Selected Model Calibration.....	67
3.6	Gripper Design Optimization using Finite Element Analysis (FEA) .....	69
3.6.1	Design Objectives & Specifications .....	69
3.6.2	Finite Element Analysis (FEA) Setup .....	71
3.6.3	Actuator Cross Section Study.....	73
3.6.4	Final Gripper Design .....	78
3.7	Pneumatic Controller .....	79
3.7.1	Requirements.....	79
3.7.2	Implementation .....	79
3.8	Design & Simulation Validation .....	82
3.9	Fabrication .....	83
3.9.1	Mold Design .....	83
3.9.2	Connectors & Assembly.....	87

3.9.3	Step-by-Step Fabrication & Assembly Procedure .....	89
Computer Vision System .....	92	
4.1	Introduction to Computer Vision in Robotics .....	92
4.1.1	Overview .....	92
4.1.2	Significance of Computer Vision in Robotics .....	92
4.1.3	Principles of Computer Vision .....	93
4.1.4	Applications of Computer Vision in Robotics .....	93
4.2	Computer Vision System Purpose & Objectives .....	95
4.2.1	Purpose .....	95
4.2.2	Objectives.....	95
4.3	Used Hardware.....	97
4.3.1	Rapoo C260 Webcam .....	97
4.3.2	Purpose & Role .....	98
4.3.3	Placement & Configuration .....	98
4.3.4	Compatibility .....	98
4.4	Software and Libraries.....	99
4.4.1	OpenCV (Open-Source Computer Vision Library).....	99
4.4.2	YOLO (You Only Look Once).....	99
4.4.3	Integration with OpenCV .....	100
4.5	YOLO Model and COCO Dataset .....	101
4.5.1	YOLO (You Only Look Once).....	101
4.5.2	Key Features of YOLO.....	101
4.5.3	COCO (Common Objects in Context) Dataset.....	101
4.5.4	Key Features of COCO Dataset .....	102
4.5.5	Integration with YOLO .....	102
4.6	Coordinate Localization and Conversion.....	103
4.6.1	Coordinate Localization:.....	103

4.6.2	Coordinate Conversion .....	104
4.6.3	Benefits and Importance:.....	105
4.7	Communication with Arduino .....	106
4.7.1	Serial Communication Protocol .....	106
4.7.2	Data Transmission to Arduino .....	107
4.7.3	Utilizing Data in Arduino.....	107
4.8	Algorithm Description (Pseudocode) .....	109
4.9	Conclusion .....	111
4.9.1	Performance and Results .....	111
	System Integration .....	112
5.1	Embedded System Design.....	112
5.1.1	Functional Requirements.....	112
5.1.2	Performance Requirements .....	113
5.1.3	Hardware Interface .....	115
5.1.4	Embedded Software.....	116
5.2	Electrical Integration.....	119
5.3	Graphical User Interface.....	120
	Conclusion .....	121
6.1	Market Analysis .....	121
6.1.1	Integrated System Commercialization.....	121
6.1.2	Delta Robot Commercialization.....	122
6.1.3	Soft Robotic Gripper Commercialization .....	123
6.2	Project Cost Analysis .....	125
6.2.1	Introduction .....	125
6.2.2	Scope.....	125
6.2.3	Methodology.....	125
6.2.4	Delta Robot.....	126

6.2.5	Soft Robotic Gripper .....	129
6.2.6	Total Integrated System.....	129
6.2.7	Market Comparison.....	129
	Hornet 565 4 axis industrial parallel delta robot .....	130
6.2.8	Limitations & Conclusion .....	132
6.3	Future Work .....	133
	Appendix.....	137
	Appendix B: EcoFlex 00-30 Material Data.....	137
	Appendix B: Purchased Parts List .....	139
	Appendix C: Delta Robot Mechanical Working Drawings.....	141

## List of Figures

Figure 1 Delta Robot Kinematic Diagram .....	17
Figure 2 Delta Robot Kinematic Loop.....	19
Figure 3 Dimensional Synthesis Methodology .....	24
Figure 4 Delta Robot Complete Workspace .....	26
Figure 5 Delta Robot Complete Workspace, Top View .....	27
Figure 6 Delta Robot Selected Workspace Section .....	28
Figure 7 Delta Robot Selected Workspace Section, Side View .....	28
Figure 8 Delta Robot Design Iteration 1.....	31
Figure 9 Delta Robot Design Iteration 2 Model .....	32
Figure 10 Delta Robot Design Iteration 3 Model .....	33
Figure 11 Delta Robot Final Iteration Model .....	34
Figure 12 Travelling Platform Design Modifcattions & Gripper Fixation.....	37
Figure 13 Frame Initial Analysis Loading .....	38
Figure 14 Frame Initial Model .....	38
Figure 15 Frame Analysis Iteration 1 Stress & Deformation Results.....	39
Figure 16 Frame Second Iteration Model .....	39
Figure 17 Frame Iteration 2 Analysis Results .....	40
Figure 18 Load Torque Calculation Delta Robot Configuration .....	42
Figure 19 Simcscape Simulation program on Simulation .....	44
Figure 20 SIMSCAPE Simulation Screenshot .....	45
Figure 21 Trajectory Planning Methodology.....	46
Figure 22 Cubic Spline Interpolation Results Plots.....	48
Figure 23 Visualization for Triangular Speed Profile .....	50
Figure 24 Visualization for trapezoidal speed profile .....	51
Figure 25 Scenario Analysis Methodology.....	52
Figure 26 Scenario Analysis Operating Points .....	53
Figure 27 Scenarios Simulation Simulink .....	54
Figure 28 Different Applications of Soft Robots [7] & [8] .....	58
Figure 29 Different Examples of Pre-Existing Soft Robotic Gripper .....	59
Figure 30 Side Section of PneuNets [10].....	61
Figure 31 PneuNets Illustration Diagram .....	61

Figure 32 Gripper Design Methodology Illustration Diagram .....	62
Figure 33 Stress-Strain Results for Different Materials used for Soft Robotic Applications [12] .....	64
Figure 34 NeoHookean Hyperelastic Material Mode .....	65
Figure 35 Yeoh 2nd Order Entry on Ansys.....	67
Figure 36 Mooney Rivlin 3 Parameter Entry on Ansys.....	68
Figure 37 Mooney Rivlin 5 Parameter Entry on Ansys.....	68
Figure 38 Contact Force Illustration, from [20] .....	70
Figure 39 Bending Angle Illustration, from [20] .....	70
Figure 40 Ansys FEA Mesh Settings .....	71
Figure 41 FEA Pressure Input Loading .....	72
Figure 42 Chamfered Chambers Design From [10], which was a gripper designed to handle bakery products.....	73
Figure 43 Chamfered Design of a Real Product of Gripper Used. Source & Manufacturer: Data sheets from SoftGripping GmbH, Germany .....	73
Figure 44 Soft Gripper iteration 0 3D model on SolidWorks.....	74
Figure 45 Gripper Iteration 0 Working Drawing .....	75
Figure 46 Gripper Iteration 0 Simulation.....	75
Figure 47 Rectangular Cross Section Collapse [12].....	76
Figure 48 Hybrid Cross Section of Soft Gripper.....	77
Figure 49 Gripper Final Design CAD Model .....	78
Figure 50 Pneumatic Control System Schematic.....	80
Figure 51 Electro-Pneumatic Circuit Schematic.....	81
Figure 52 Validation Results under negative pressure supply .....	82
Figure 53 Gripper Validation Results under Positive Pressure Supply .....	82
Figure 55 Mold Model: Lower Part 2 .....	84
Figure 54 Mold Model: Upper Part 1.....	84
Figure 56 Mold Model: Base Part3 .....	84
Figure 57 Molding Phase 1 Assembly Illustration .....	85
Figure 58 Figure 36 Molding Phase 2 Assembly Illustration .....	85
Figure 59 Soft Gripper Air Connector .....	87
Figure 60 Gripper End Effector Assembly Jig.....	88
Figure 61 Fruit Detection using Computer Vision.....	96
Figure 62 Rapoo C260 Webcam.....	97

Figure 63 Results & Outputs .....	103
Figure 64 Bounding Box .....	104
Figure 65 Getting ruler length in pixels .....	105
Figure 66 Output of the scale factor code.....	105
Figure 67 Computer Vision Code Flow Chart.....	110
Figure 68 Arduino MEGA Development Board .....	114
Figure 69 Embedded Hardware Architecture .....	115
Figure 70 Computer Vision Controlled Mode Flowchart .....	118
Figure 71 System Electrical Connections Schematics .....	119
Figure 72 Screenshot from the System's Graphical User Interface.....	120

## List of Tables

Table 1 Final Delta Robot Workspace Desired Selection.....	25
Table 2 Final Delta Robot Dimensions .....	29
Table 3 Delta Robot Material Selection Tables .....	30
Table 4 Scenario Analysis Inputs Table .....	53
Table 5 Scenario Analysis Results: Starting Point A .....	54
Table 6 Scenario Analysis Results: Starting Point B .....	55
Table 7 Scenario Analysis Results: Starting Point H .....	55
Table 8 Reviewed Literature Hyperelastic Calibration Models Formulated Comparison...	66
Table 9 Gripper Designed Parameters.....	69
Table 10 Gripper Iteration 0 Details .....	74
Table 11 Delta Robot Competition Price Comparison .....	130
Table 12 Soft Robotic Gripper Market Price Comparison .....	131

# **Chapter 1**

# **Introduction**

## **1.1 Project Overview**

By using a delicate grasp for the items along with high speed and precision, the sorting and packing system seeks to increase profit and decrease losses.

The subsystems that have been put in place do this. The delta robot configuration achieves high speed because of its lightweight linkages, which are composed of lightweight materials. The system was also accelerated by the arrangement of all actuators in one plane. Precise control provided to the actuators yields accuracy. Computer vision and the pretrained model recognize the thing being caught, and then the filtering and sorting process is carried out. Regarding the soft grip, the gripper is composed of "Eco Flex 30," a material that is soft and flexible with several benefits, including food-grade compatibility, chemical resistance, non-slip grip, flexibility, and tear resistance

## **1.2 Motivation**

As a team, we simply aimed for a project that gives us both opportunities, research and development, as well as solving an industrial problem. We wanted to delve into three emerging fields, Soft Robotics, Machine Learning & Model-Based Design, as well as combining them with disciplines we study such as robotics, electronics, mechanical design and other disciplines.

Moreover, we needed to combine our passion in these disciplines in a project that solves a problem in the industry, which why we chose this project, an Industrial Human-Inspired Pick & Place Robot which makes use of computer vision to sort items and position them with a delta robot that is equipped with a soft robotics gripper. But what is Soft Robotics in the first place, what makes this robot Human-Inspire, and how this project through the implementation of Artificial Intelligence innovation and Robotic Automation can boost production lines & decrease labor? The urge to solve these questions made us choose this project particularly for our thesis.

Another motivation for us to choose this project is its potential to grow as a business idea that we can implement in the future. According to our market research we concluded that soft robotics; especially soft grippers is a growing market, especially in Egypt where there are very few companies manufacturing these types of grippers.

### **1.3 Engineering Approach**

Our engineering approach to this project follows a systematic methodology that encompasses several key phases. Initially, thorough research and analysis were conducted to understand the requirements and constraints of achieving sorting task with a delta robot, a soft gripper and computer vision.

The next step in the design phase was to conceptualize the system architecture, choose suitable components to go along with the primary components mentioned above, and simulate the system dynamics utilizing the knowledge we had gained from our university studies.

An iterative method was used throughout the implementation process, enabling quick prototyping and improvement of the hardware and software components of the system before simulating it in a Simulink environment to make sure the required requirements are satisfied.

To make sure the sorting system satisfies the required performance and specifications, a thorough performance evaluation and optimization were carried out.

Our engineering strategy makes use of a wide range of ideas and techniques from different engineering specialties. The delta robot and robotic soft gripper were developed using mechanical design principles to ensure their resilience, precision, and dependability when handling objects of various sizes and forms. To perform precise and seamless sorting operations, algorithms for motion planning and trajectory optimization were developed with the help of inverse kinematics. Moreover, the system's ability to observe and adjust to its surroundings was made possible by the integration of sophisticated sensors and vision systems, which improved overall performance and adaptability.

## 1.4 Report Layout

**Chapter 1:** Introduction of our thesis; including project overview going through the whole process of sorting, with a snippet of some of the hardware components used in the system

**Chapter 2:** A brief introduction of delta robot configuration, and the reasons behind choosing it as the manipulator of the system. Also, we will go through the whole process of designing and implementation of delta robot. Starting with kinematic analysis and synthesis; including forward and inverse kinematics, workspace formulation and dimensional synthesis. Moreover, we will talk about delta robot's mechanical design, including design iterations and validation, frame design and fabrication. Furthermore, we go through the process of motor sizing and selection. Then, we will go through the robot control, point to point movement and trajectory planning. Lastly, we will discuss robot simulation and some of the problems we faced with the robot and how we solved them.

**Chapter 3:** We start with a brief introduction on soft robotics and literature review and recent developments. Then we will discuss the gripper's material selection and modeling, design and finite element analysis. Then we will demonstrate the pneumatic control system and pneumatic circuits and lastly, the fabrication of the gripper.

**Chapter 4:** A brief demonstration of our computer vision system

**Chapter 5:** System integration of our hardware and software components, along with the microcontroller interfacing, to control and communicate between those components. A brief discussion of the design and implementation of conveyor belt, which is a vital component for the demonstration of the final product. The schematics of the electrical circuits and the electrical control unit are also discussed.

## Chapter 2

# Delta Robot

### 2.1 What is a Delta Robot Configuration

As shown in the following figure, the delta robot consists of main parts and they are as follows:

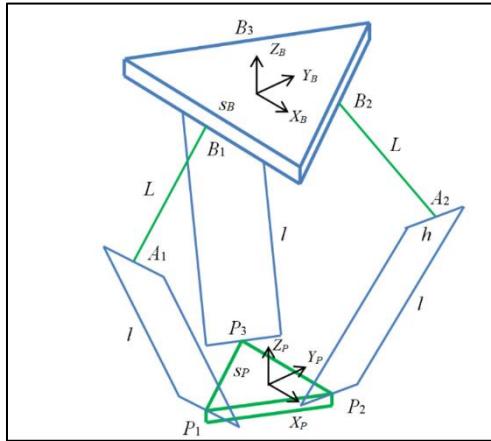


Figure 1 Delta Robot Kinematic Diagram

“Base Platform” where the actuation system is mounted,

“Active Arm” which transfers the movement from the actuating system to the rest of the system, “Passive Arm” which receives the movement transferred from the active arm to direct the end effector,

“Travelling Platform”, it is the part which is directed by the passive arms and connects them together, the end effector is mounted on this platform.

The workspace in which the robot operates is conical shaped with a circular base. We chose the circular base to be (200 mm) and the height is not determined yet, it will be considered later, but the movement from the zero position for end effector to the belt carrying the products will be (300 mm).

## 2.2 Kinematic Analysis & Dimensional Synthesis

In this section of the report, we present the kinematic model formulated for

### 2.2.1 Inverse & Forward Kinematics

It is necessary to derive the kinematic model for our robot. Kinematic model is a relationship between the robot's end effector and its joint angles controlled by three motors. The following can be derived from the kinematic model:

- Inverse kinematics (sec. of this report)
- Forward kinematics (sec. of this report)
- Workspace analysis (sec. of this report)
- Path and trajectory planning (sec. of this report)

#### *Initial desired outcomes*

- 1) Using forward kinematics, we needed to Deduce the robot's desired workspace of radius 25mm and to use the workspace boundaries to come up with an initial links lengths to start our first CAD model iteration.
- 2) We needed inverse kinematics to control the robot motion given end effector co-ordinates, the inverse kinematic model calculates the desired motor position.
- 3) We needed forward kinematics to calibrate our robot and analyze errors. This is done by comparing the desired end effector co-ordinates with the calculated one. Thus, allowing us to identify and compensate for discrepancies caused by factors such as joint flexibility, manufacturing tolerances, or sensor inaccuracies. Thus, enhancing robot accuracy

### 2.2.1.1 Inverse Kinematics

This chapter shows the inverse kinematic modeling and analysis for the delta robot.

#### Kinematic position model

In order to get the kinematic position model of the delta robot, we start by getting the following three vector-loop closure equations:

$$\vec{B}_i + \vec{L}_i + \vec{l}_i = \vec{P}_P + \vec{P}_i$$

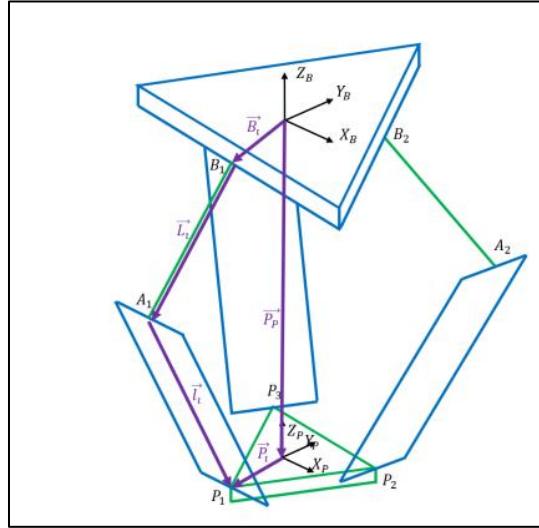


Figure 2 Delta Robot Kinematic Loop

Knowing that the length of  $\vec{l}_i$  is constant (the virtual length through the center of each parallelogram), we can deduce that:

$$l_i = l_i = P_P + P_i - \vec{B}_i - \vec{L}_i$$

Both sides could then be squared for the convenience of getting rid of any square roots as follows:

$$l_i^2 = l_{ix}^2 + l_{iy}^2 + l_{iz}^2$$

Where  $P_P = [x \ y \ z]^T$  and  $\vec{L}_i$ ,  $P_i$ , and  $\vec{B}_i$ , could be calculated to be:

$$P_1 = \begin{bmatrix} 0 \\ -u_P \\ 0 \end{bmatrix}, P_2 = \begin{bmatrix} \frac{1}{2}S_P \\ w_P \\ 0 \end{bmatrix}, P_3 = \begin{bmatrix} -\frac{1}{2}S_P \\ w_P \\ 0 \end{bmatrix}$$

$$\vec{B}_1 = \begin{bmatrix} 0 \\ -w_B \\ 0 \end{bmatrix}, \quad \vec{B}_2 = \begin{bmatrix} \frac{\sqrt{3}}{2}w_B \\ \frac{1}{2}w_B \\ 0 \end{bmatrix}, \quad \vec{B}_3 = \begin{bmatrix} -\frac{\sqrt{3}}{2}w_B \\ \frac{1}{2}w_B \\ 0 \end{bmatrix}$$

$$\vec{L}_1 = \begin{bmatrix} 0 \\ -L\cos \theta_1 \\ -L\sin \theta_1 \end{bmatrix}, \quad \vec{L}_2 = \begin{bmatrix} \frac{\sqrt{3}}{2}L\cos \theta_2 \\ \frac{1}{2}L\cos \theta_2 \\ -L\sin \theta_2 \end{bmatrix}, \quad \vec{L}_3 = \begin{bmatrix} -\frac{\sqrt{3}}{2}L\cos \theta_3 \\ \frac{1}{2}L\cos \theta_3 \\ -L\sin \theta_3 \end{bmatrix}$$

Substituting we obtain the kinematic position model equations:

$$\begin{aligned} 2L(y+a)\cos \theta_1 + 2zL\sin \theta_1 + x^2 + y^2 + z^2 + a^2 + L^2 + 2ya - l^2 &= 0 \\ -L(\sqrt{3}(x+b) + y + c)\cos \theta_2 + 2zL\sin \theta_2 + x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2xb + 2yc - l^2 &= 0 \\ L(\sqrt{3}(x-b) - y - c)\cos \theta_3 + 2zL\sin \theta_3 + x^2 + y^2 + z^2 + b^2 + c^2 + L^2 - 2xb + 2yc - l^2 &= 0 \end{aligned}$$

### Inverse kinematics position model

The kinematic position model can be simplified as:

$$E_i \cos \theta_i + F_i \sin \theta_i + G_i = 0$$

where:

$$\begin{aligned} E_1 &= 2L(y+a) & F_1 &= 2zL & G_1 &= x^2 + y^2 + z^2 + a^2 + L^2 + 2ya - l^2 \\ E_2 &= -L(\sqrt{3}(x+b) + y + c) & F_2 &= 2zL & G_2 &= x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2(xb + yc) - l^2 \\ E_3 &= L(\sqrt{3}(x-b) - y - c) & F_3 &= 2zL & G_3 &= x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2(-xb + yc) - l^2 \end{aligned}$$

Using the tangent half-angle substitution we define  $t_i = \tan(\frac{\theta_i}{2})$  then  $\cos \theta_i = \frac{1-t_i^2}{1+t_i^2}$  and  $\sin \theta_i = \frac{2t_i}{1+t_i^2}$  to get:

$$G_i - E_i t^2 + 2F_i t_i + G_i + E_i = 0$$

Which is solved for  $t$  using the quadratic formula as follows:

$$t_{i\pm} = \frac{-F_i \pm \sqrt{E_i^2 + F_i^2 - G_i^2}}{G_i - E_i}$$

We then get  $\theta_i$  by inverting the tangent half-angle substitution:

$$\theta_i = 2\tan^{-1} t_i$$

### 2.2.1.2 Forward Kinematics

In this section we will be deriving the direct position kinematics of the delta robot. We did not derive the direct velocity and acceleration kinematic models since they were not needed. Since the delta robot motion is only translational, the direct kinematics only needs to determine the position of the end-effector, in our case this is the vector  $PP$  which is shown in figure 2.5. Looking at figure 2.6, we can define point  $A_{iv}$  where:

$$A_{1v} = \begin{bmatrix} 0 \\ -w_B - L\cos \theta_1 \\ -L\sin \theta_1 \end{bmatrix} \quad A_{2v} = \begin{bmatrix} \frac{\sqrt{3}}{2} w_B + L\cos \theta_2 \\ \frac{1}{2} w_B + L\cos \theta_2 \\ -L\sin \theta_2 \end{bmatrix}$$

$$A_{3v} = \begin{bmatrix} -\frac{\sqrt{3}}{2}(w_B + L\cos \theta_3) + \frac{S_P}{2} \\ \frac{1}{2}(w_B + L\cos \theta_3) - w_P \\ -L\sin \theta_3 \end{bmatrix}$$

From there,  $P_P$  can be found by getting the point of intersection of three spheres centered at the points  $A_{iv}$  and having a radius of  $l$ . The solution to this problem would yield two points, from which the correct point could be easily identified. After obtaining the points  $A_{iv}$ .

### Three-Spheres Intersection Algorithm

Let us assume that the three spheres have the center coordinates and radii  $[x_i \ y_i \ z_i]^T$  and  $r_i$  respectively. The equations of the three spheres are therefore:

$$\begin{aligned} x - x_1^2 + y - y_1^2 + z - z_1^2 &= r_1^2 \\ x - x_2^2 + y - y_2^2 + z - z_2^2 &= r_2^2 \\ x - x_3^2 + y - y_3^2 + z - z_3^2 &= r_3^2 \end{aligned}$$

Subtracting these three equations we get:

$$\begin{aligned} a_{11}x + a_{12}y + a_{13}z &= b_1 \\ a_{21}x + a_{22}y + a_{23}z &= b_2 \end{aligned}$$

where:

$$\begin{aligned} a_{11} &= 2(x_3 - x_1) & a_{21} &= 2(x_3 - x_2) \\ a_{12} &= 2(y_3 - y_1) & a_{22} &= 2(y_3 - y_2) \\ a_{13} &= 2(z_3 - z_1) & a_{23} &= 2(z_3 - z_2) \end{aligned}$$

$$\begin{aligned} b_1 &= r_1^2 - r_3^2 - x_1^2 - y_1^2 - z_1^2 + x_3^2 + y_3^2 + z_3^2 \\ b_2 &= r_2^2 - r_3^2 - x_2^2 - y_2^2 - z_2^2 + x_3^2 + y_3^2 + z_3^2 \end{aligned}$$

Then we solve for  $x$

$$\begin{aligned} x &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}}y - \frac{a_{13}}{a_{11}}z \\ x &= \frac{b_2}{a_{21}} - \frac{a_{22}}{a_{21}}y - \frac{a_{23}}{a_{21}}z \end{aligned}$$

Following that, we subtract these two from each other we get:

$$y = a_4z + a_5$$

where:

$$a_5 = \frac{a_3}{a_1} \quad a_4 = \frac{a_2}{a_1} \quad a_3 = \frac{b_2}{a_{21}} - \frac{b_1}{a_{11}} \quad a_2 = \frac{a_{13}}{a_{11}} - \frac{a_{23}}{a_{21}} \quad a_1 = \frac{a_{22}}{a_{21}} - \frac{a_{12}}{a_{11}}$$

We then substitute with 2.13 in 2.12 to obtain:

$$x = -a_6 z + a_7$$

where:

$$a_7 = \frac{b_2 - a_{22}a_5}{a_{21}} \quad a_6 = \frac{a_{22}a_4 + a_{23}}{a_{21}}$$

Substituting  $y$  and  $x$  gives us a single quadratic equation in terms of  $z$

$$a_0 z^2 + b_0 z + c_0 = 0$$

where:

$$\begin{aligned} a_0 &= a_6^2 + a_4^2 + 1 \\ b_0 &= -2a_6a_7 + 2a_6x_1 + 2a_4a_5 - 2a_4y_1 - 2z_1 \\ &= 2(a_6x_1 - a_7) + a_4(a_5 - y_1) - z_1 \\ c_0 &= a_7^2 - 2a_7x_1 + x_1^2 + a_5^2 - 2a_5y_1 + y_1^2 + z_1^2 - r_1^2 \\ &= a_7(a_7 - 2x_1) + a_5(a_5 - 2y_1) + x_1^2 + y_1^2 + z_1^2 - r_1^2 \end{aligned}$$

Using the quadratic formula, we solve for  $z$  then use 2.13 and 2.14 to get  $y$  and  $x$  respectively.

$$\begin{aligned} z_{\pm} &= \frac{-b_0 \pm \sqrt{b_0^2 - 4a_0c_0}}{2a_0} \\ y_{\pm} &= a_4z_{\pm} + a_5 \\ x_{\pm} &= -a_6z_{\pm} + a_7 \end{aligned}$$

It is important to note that this method for getting the intersection of three spheres introduces algorithmic singularities. Algorithmic singularities are singularities that are the result of the algorithm used and not real physical limitations. As such, we modified the manipulation of the equations in the three spheres intersection algorithm form. To make sure that the algorithmic singularities would not occur within the limits of our robot's targeted workspace. Such modifications changed the conditions under which we would be dividing by zero. This is done by isolating a different variable first.

## 2.2.2 Workspace Formulation

### 2.2.2.1 Objectives

- A key step as a start of the iterative process of selecting the linkages dimensions since different linkages length combinations would result in different end effector reachable points
- Determination of achievable positions of the robot and the singularities.
- Determination of extreme positions that will be the input to the calculation of the maximum acceleration required to compute motor torque requirements

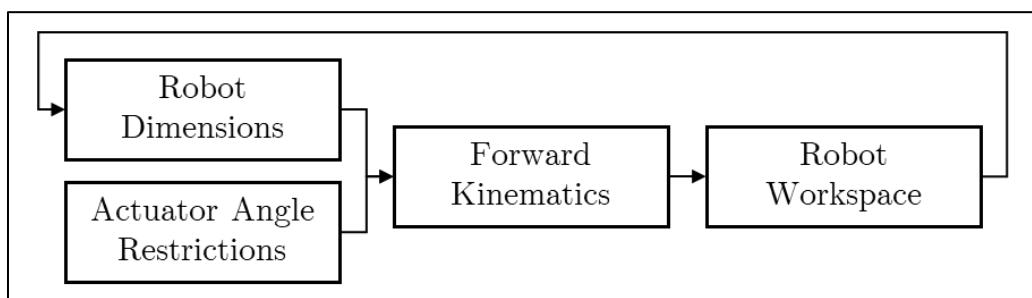


Figure 3 Dimensional Synthesis Methodology

### 2.2.2.2 Inputs & Constraints

- Common delta robot configurations found in the industry.
- Soft gripper overall length. The vertical range of the robot should allow for the gripper to move freely and not hit any of the materials found in the workspace. Since it is now not selected yet, we decided to allow for a safe range which assumes that the gripper will be no longer than 100mm. This means that the total overall vertical range of the robot should allow for at least 200mm, to accommodate for ease of motion.
- Rod ends (spherical joints) maximum angle limitation.
- Position limitations for the motor due to the fixation, this selected to be from -60 to +60 degrees.

### **2.2.2.3 Methodology**

The workspace of the robot is plotted using a simple program written on MATLAB. The inputs to the program are the robot dimensions, the forward kinematics function that is written on MATLAB as well, this forward kinematics function is based on the Forward Kinematic Position Model mentioned. The program logic simply is to plot a range of XYZ coordinates corresponding to all possible combinations of the three motors angular positions. NOTE: the home position, i.e. (0,0,0) is taken at the center of the upper plate of the robot.

### **2.2.2.4 Results & Final Work Area**

The final workspace obtained was a concave-like shape fig. (4), so in order to satisfy the requirements mentioned, we decided to allow the robot to work in a selected area in fig. (6), as a cylinder to allow for ease in control and location of coordinates. Final ranges obtained are:

**Table 1 Final Delta Robot Workspace Desired Selection**

<b>X – Position Range</b>	<b>-200mm to +200mm</b>
<b>Y – Position Range</b>	<b>-200mm to +200mm</b>
<b>Z – Position Range</b>	<b>-550mm to -350mm</b>

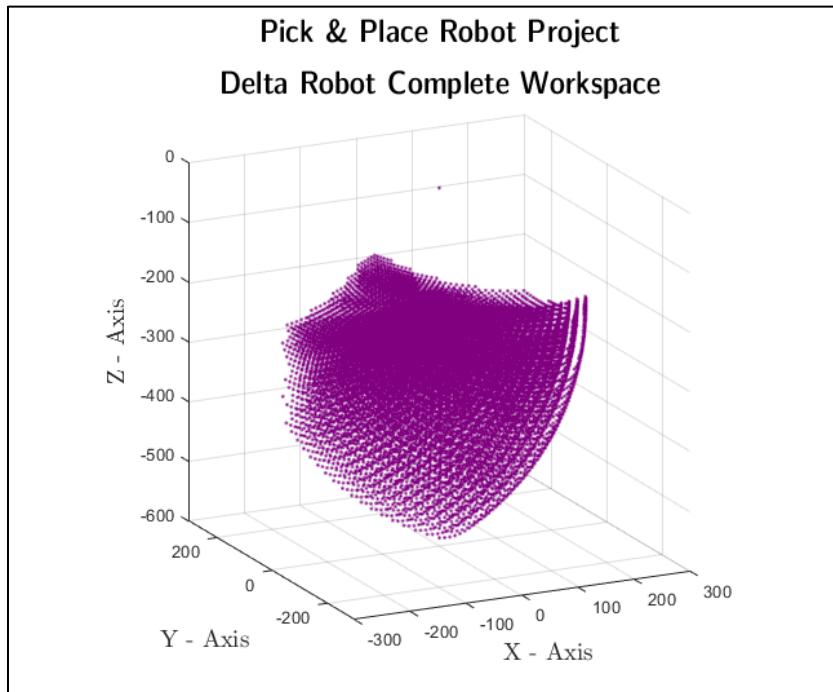


Figure 4 Delta Robot Complete Workspace

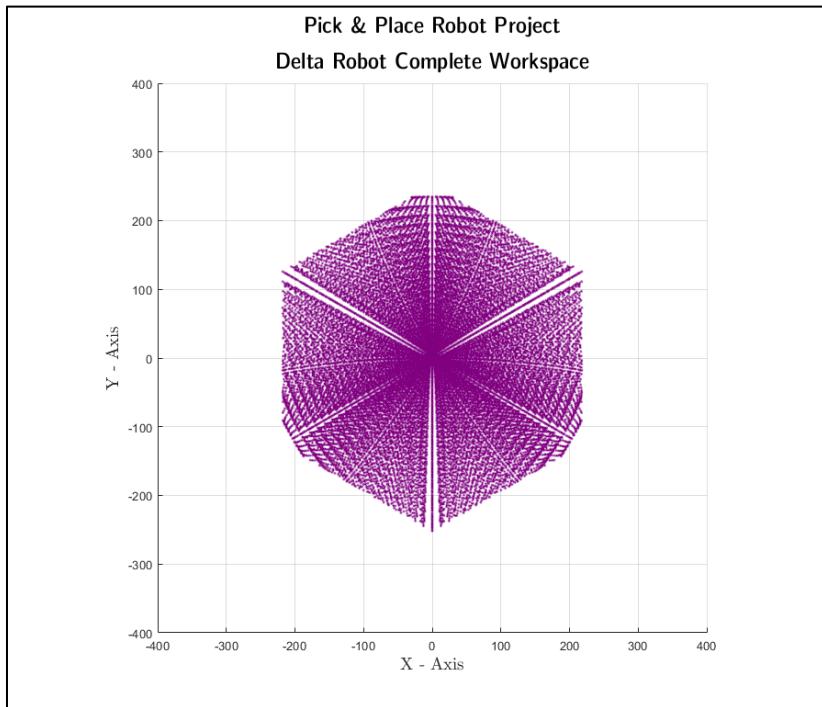


Figure 5 Delta Robot Complete Workspace, Top View

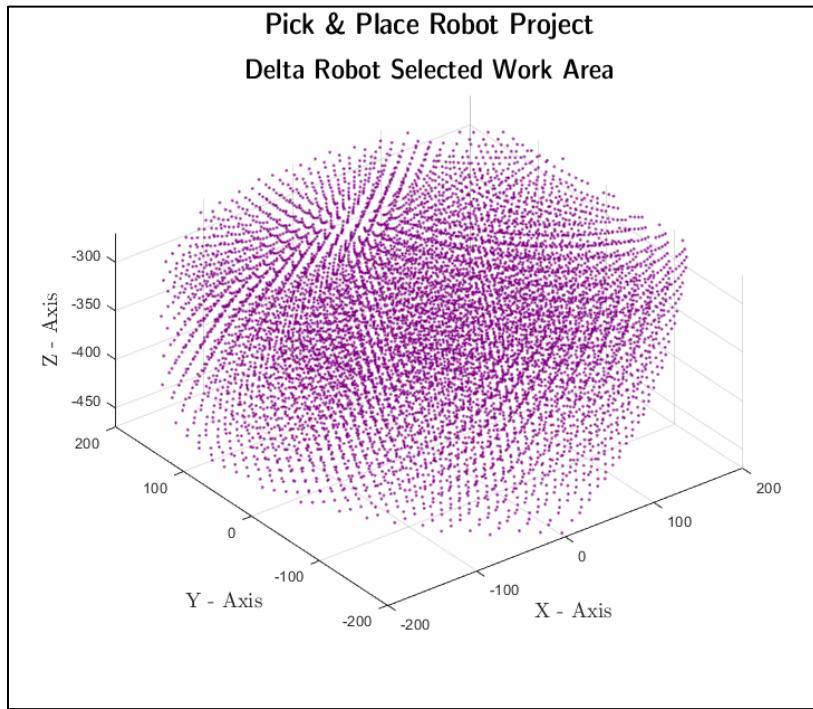


Figure 6 Delta Robot Selected Workspace Section

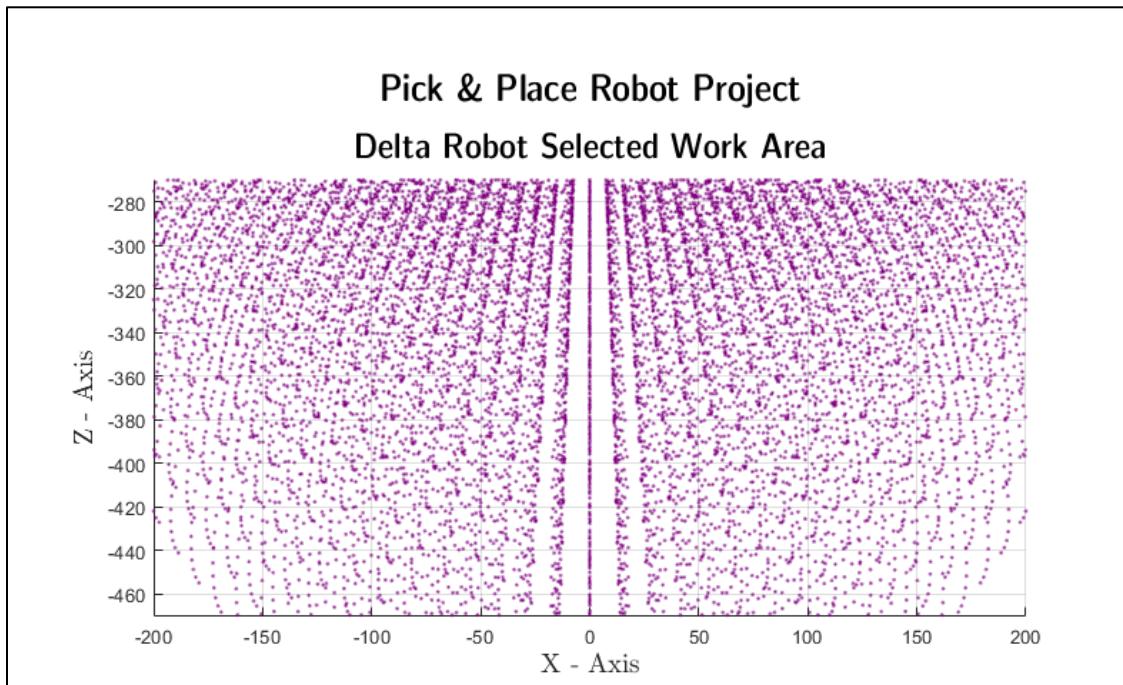


Figure 7 Delta Robot Selected Workspace Section, Side View

### **2.2.3 Final Dimensions & Parameters**

The final dimensions of the delta robot after optimizing the robot's workspace are presented in the following table:

Table 2 Final Delta Robot Dimensions

<b>Active Arm Length (l)</b>	230 mm
<b>Passive Arm Length (L)</b>	436 mm
<b>Base Platform Side Length</b>	385 mm
<b>Travelling Platform Side Length</b>	86 mm

## 2.3 Mechanical Design

### 2.3.1 Material Selection

Composites like carbon fiber and aluminum are ideal for delta robot construction. Because of the extremely low inertia system these materials offer, the delta robot can operate at very fast speeds.

However, as the goal of our project was to install the required system in the FMCG sector, the material that was selected had to be safe and hygienic to come into touch with the products.

Therefore, stainless steel was the safer option. Nonetheless, using aluminum won't be hazardous because there isn't any direct contact between the arms and the food.

In addition to the sanitary criterion, which we prioritized, there were additional considerations that influenced our decision. These factors are shown in the decision matrix below:

Table 3 Delta Robot Material Selection Tables

Criteria	Weighting	Options					
		Stainless Steel		Aluminum		Carbon Fiber	
		Score	Total	Score	Total	Score	Total
Hygiene	4	3	12	2	8	1	4
Availability	3	3	9	2	6	1	3
Machining	2	2	4	3	6	1	2
Cost	2	2	4	3	6	1	2
Weight	4	2	8	3	12	4	16
	Total		37		38		27

Aluminum is the optimum choice for our application, still we need to identify the grade of aluminum as it would affect the machining and cost.

### **2.3.2 Design Iterations & Components Optimization**

Each iteration differs from the previous one not in too many details but crucial ones, some dimension changes to identify the workspace in which the system operates. Other dimensions are important for stress analysis and to prevent failure of the system.

In another iteration the end effector and actuating system will be added after specifying the forces and torques the system is subjected to, so the design of some parts like the “Base Platform”, “Active Arm” or “Base Platform”.

#### **2.3.2.1 Iteration 1**

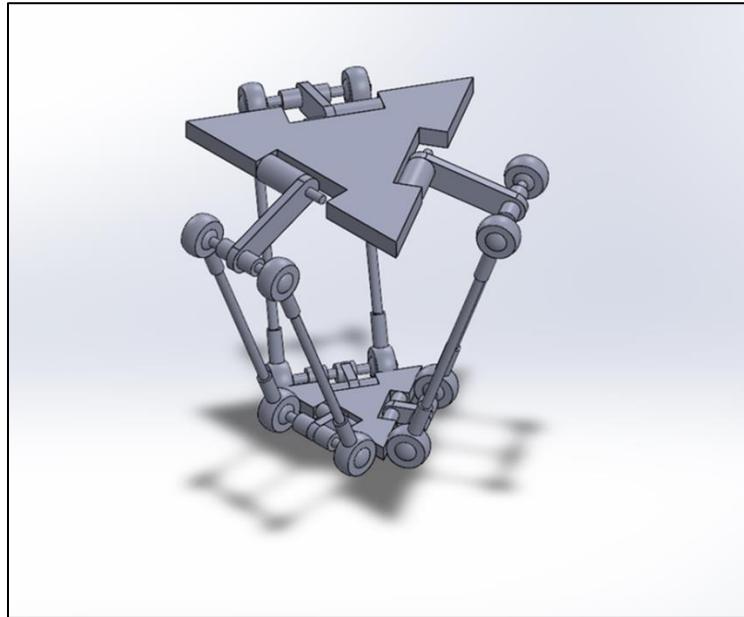


Figure 8 Delta Robot Design Iteration 1

In this iteration we tried to visualize the robot and how it will move in its workspace, so its mechanical components and parts were chosen and designed as shown,

For the passive arms to achieve their movement properly we coupled them in spherical joints to allow the side-to-side movement and rotate around the pin. All dimensions used in this iteration were arbitrary for visualization, the cylindrical part linked with the “Active Arm” is considered as the motor and its shaft that actuates the active arm.

#### 2.3.2.2 Iteration 2

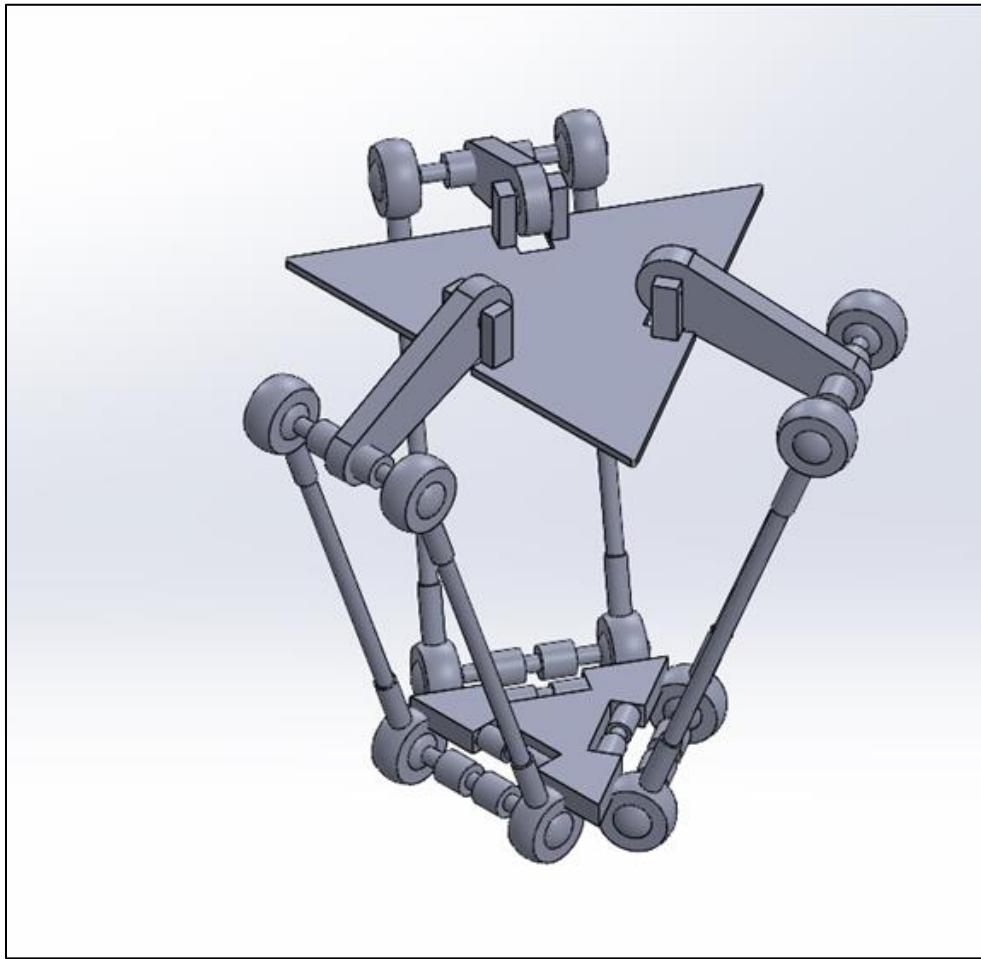


Figure 9 Delta Robot Design Iteration 2 Model

In order to delve into this iteration, we used a tool box on MATLAB to determine the robot's dimensions: Active arm length, Passive arm length, median length of both Base Platform and Travelling Platform. The input was the dimensions of the robot's workspace previously mentioned in the beginning of this section.

Modifications in this iteration are as follows:

- 1- The Active Arm is tapered rather than cylindrical, which helps to reduce the torque generated by bringing its center of gravity (CG) closer to the CG of the motor.
- 2- The spherical joints used in the first iteration are replaced with the Rod End bearing for its availability in the market and for its advantages needed in this project which are as follows: it is a self-aligning plain bearing that uses a spherical inner ring which has the same level of accuracy and hardness as bearing steel balls. With the combination of a spherical inner ring whose sliding surface is mirror-finished and a rationally designed holder, the Rod End ensures play-free, extremely smooth rotation and oscillation.
- 3- Flanges are to be added to the traveling platform to be coupled with the passive arm
- 4- The Travelling Platform is to be cut from the middle to allow installation of the end effector with its connections.

#### **2.3.2.3 Iteration 3**

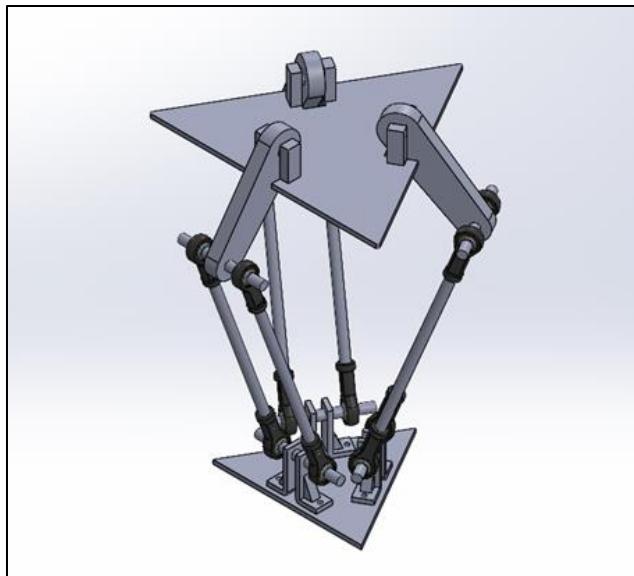


Figure 10 Delta Robot Design Iteration 3 Model

In order to improve the design furthermore, we changed the fixation design to be more practical and easier to assemble and disassemble when maintenance is needed. We analyzed the stress on this design and it was too safe. So, in the next iteration we will reduce unnecessary materials to reduce cost.

#### **2.3.2.4 Iteration 4**

In this iteration we took into consideration the market constraints and removed excess material that won't affect the safety of the structure.

another modification was in the traveling platform's fixation, we redesigned it to be easier in manufacturing and assembly

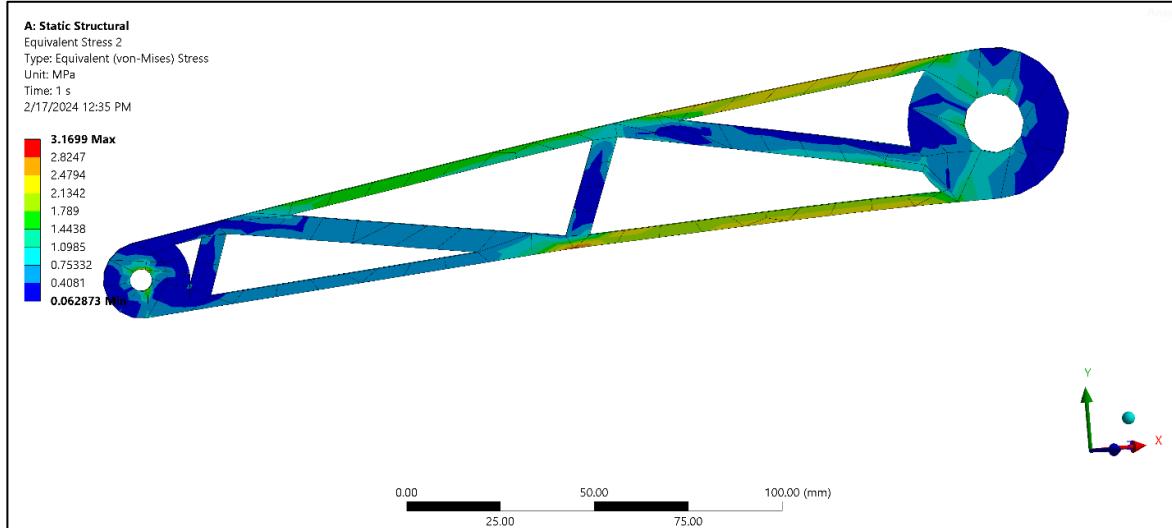


Figure 11 Delta Robot Final Iteration Model

#### **Main Modifications**

- Due to market-constraints, design remodified to reduce weight and inertial loads.
- Active arm
- Platform
- For weight-reduction, the active arm changed to be made from aluminum alloy 5086. Less weight means less torque requirement from the motor.
- For assembly purposes, the traveling platform remodified.
- The active arm's structure now is truss shaped to reduce stresses.

Validation with normal loading for the whole robot



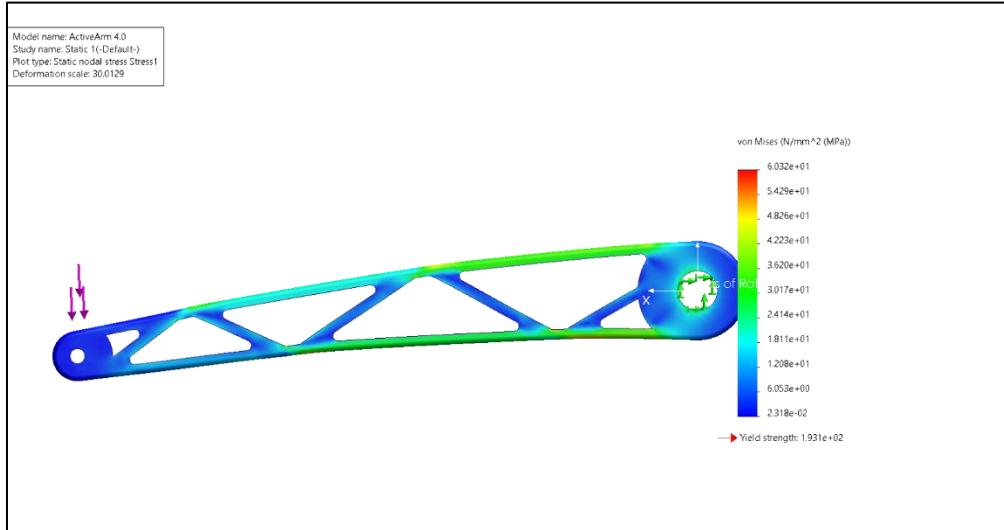
Validation with beam test (as if it was a beam fixed on two supports)

Study is carried with gravity and a weight added to the free end of the active arm, so we could see the behavior with the subjected load plus its own weight when it is fixed from the motor end.

200N, SolidWorks

Maximum – 60 MPa

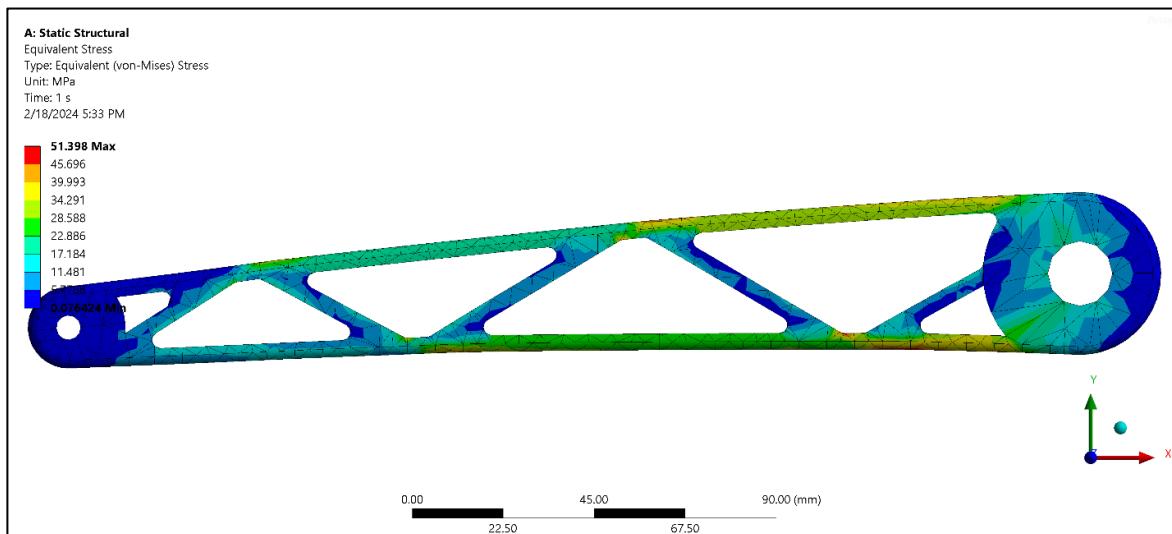
Aluminum Yield Strength – 190 MPa



### 200N, Ansys Static Structural

Maximum – 51 MPa

Aluminum Yield Strength – 190 MPa



### Other Active Arm Design Modifications

- Addition of a keyway and holes for connection with the flanged coupling.
- Transition fit.
- Laser cutting.

## Travelling Platform (End-Effector) Design

Traveling Platform Design Considerations.

- Multiple fixation configurations with gripper connector to allow the interaction with different picked objects.
- Holes for gripper's pneumatic air connector.

Three fixation sizes are available according to the desired gripping size.

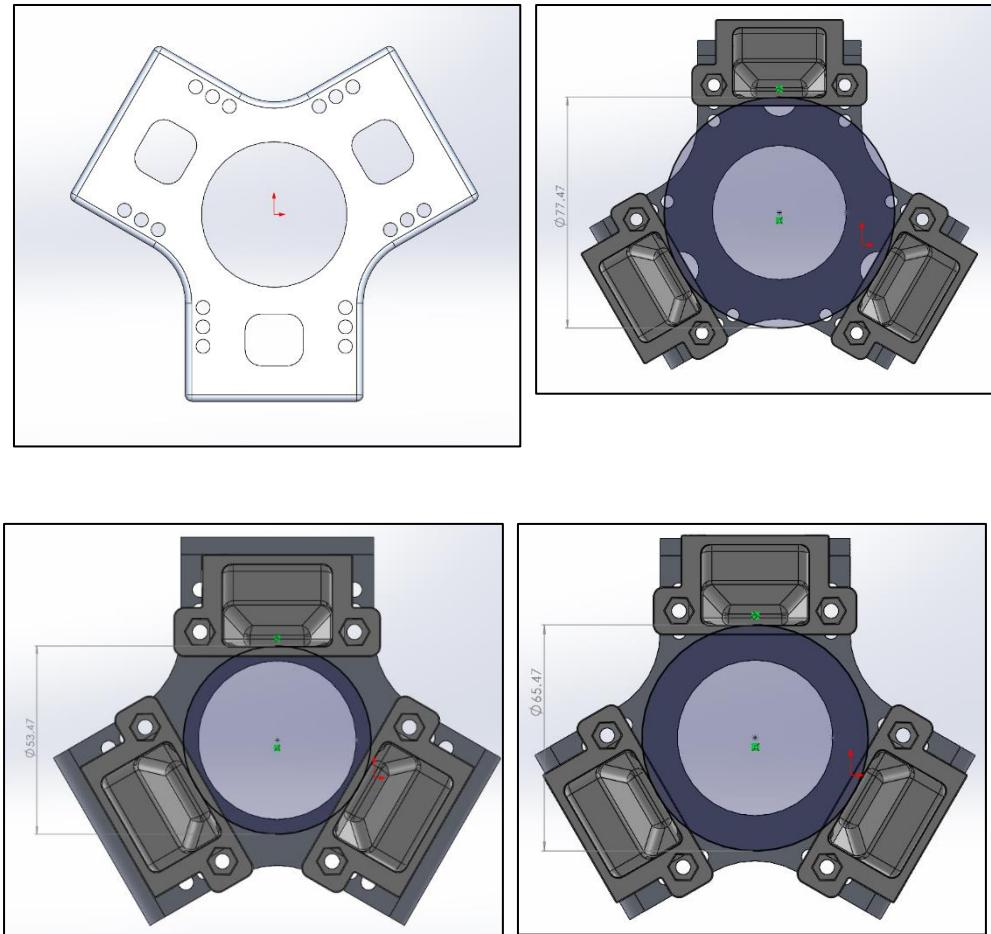


Figure 12 Travelling Platform Design Modifications & Gripper Fixation

### **2.3.2 Frame Design**

This subsection demonstrates the process of the designing the main frame of the delta robot, going through type, material and analysis.

#### **2.3.2.1 Aluminum Profile Extrusions**

The frame is selected to be fabricated from standard aluminum profile extrusions. This choice was made due to their durability and high strength, as well as that being standardized makes assembly process much easier due to the wide range availability of their assembly accessories, which makes them suitable for mounting other components in the project as well. Profile sizes 2020 and 2040 are used in the frame and their material is 6061 Aluminum alloy.

#### **2.3.2.2 Frame Design Iteration 1**

The frame was constructed on SolidWorks software and the software was used for the analysis as well. The frame was initially (figure ((14)) constructed from 2020 profiles only and then analyzed by fixing the robot fixed base platform on the top of the frame and loading it with 30Kgs (300N) plus its own weight (figure ((13))). The 30Kgs are chosen since the three motors are 5Kg each, the total weight of the robot is approx. 2Kg without loading.

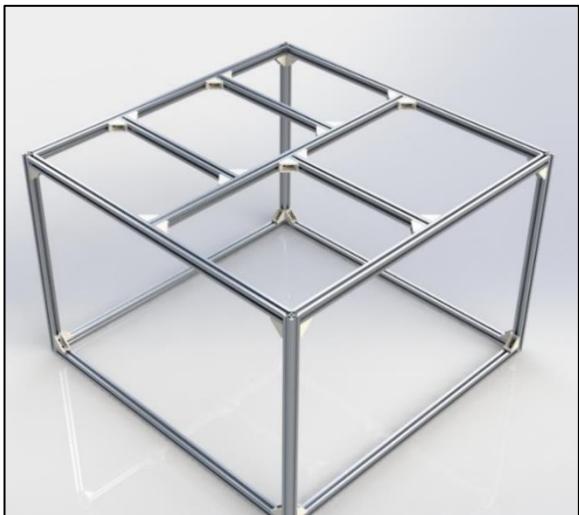


Figure 14 Frame Initial Model

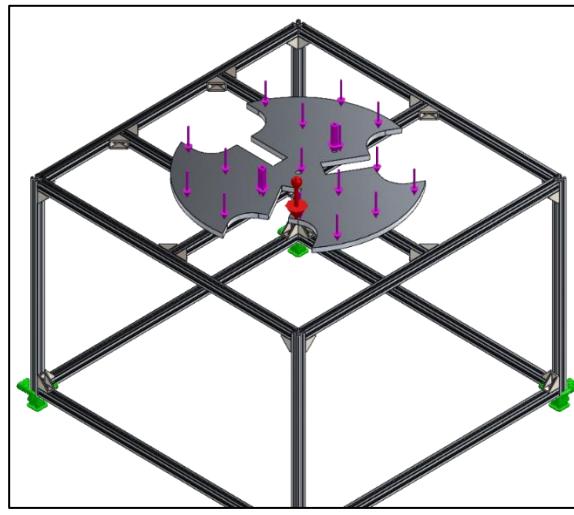


Figure 13 Frame Initial Analysis Loading

### 2.3.2.3 Iteration 1 Results

The initial iterations results presented in figures (15) were: Maximum Stress 49MPa & Maximum deformation of 0.8mm. The stress results are safe (yield strength of profile material is 276 MPa), yet the upper beams that are connected to the base platform exhibit very high deformation, therefore next design is to use 2040 profile for the upper-mid beams only and the remaining of the frame is unchanged as 2020 profile.

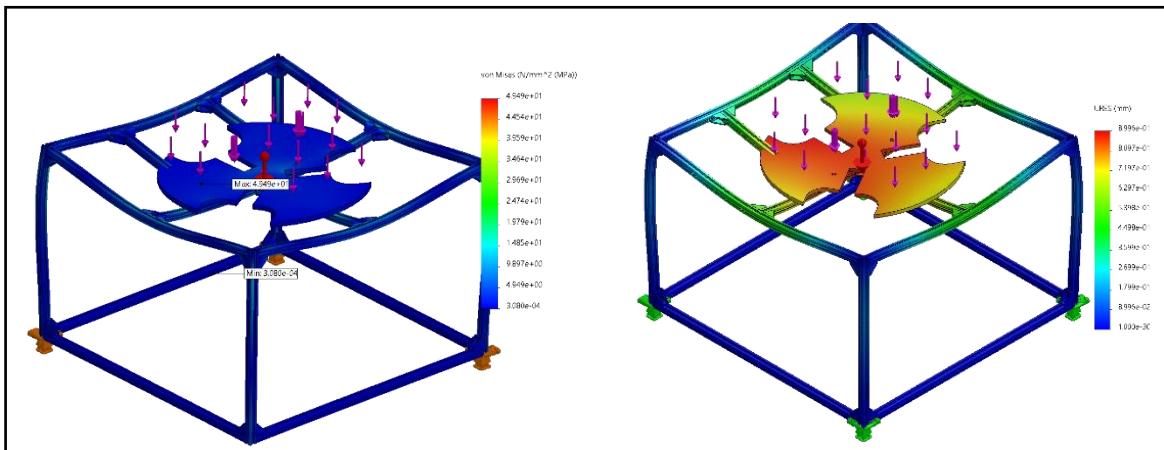


Figure 15 Frame Analysis Iteration 1 Stress & Deformation Results

### 2.3.2.4 Frame Design Iteration 2

As mentioned, the second iteration was required to strengthen out the part that the robot's base will be fixed to. To do this, 20x40 aluminum profiles were used for the middle beams, see figure (16). The second analysis was done likewise, loading with 300N on the robot's fixed base.

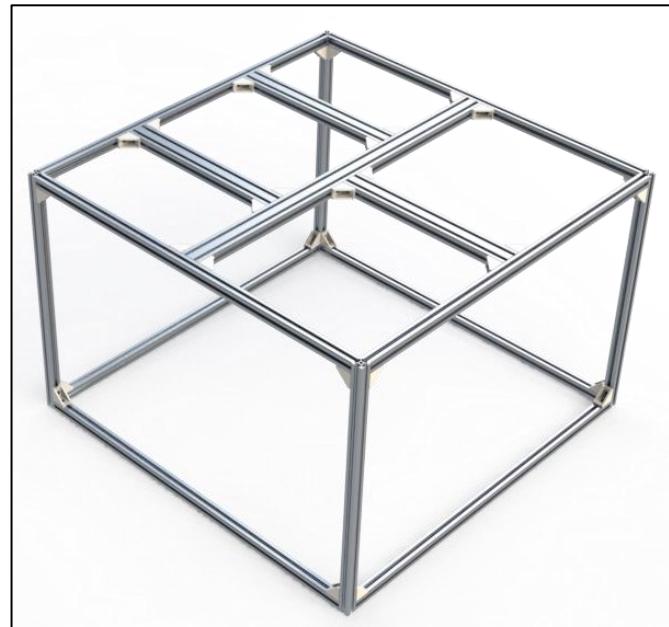


Figure 16 Frame Second Iteration Model

### 2.3.2.5 Iteration 2 Results

The second iterations results presented in figures (17) were: Maximum Stress 29MPa & Maximum deformation of 0.6mm (on the mid-upper beam). This considered to be the final iteration and this was backed up by the fact that according to the international code council (ICC) catalogues for acceptable aluminum structural loading, the acceptable deflection for aluminum beams is  $L/60$ , in our case the length is 700mm, therefore the acceptable deformation is 11.66mm.

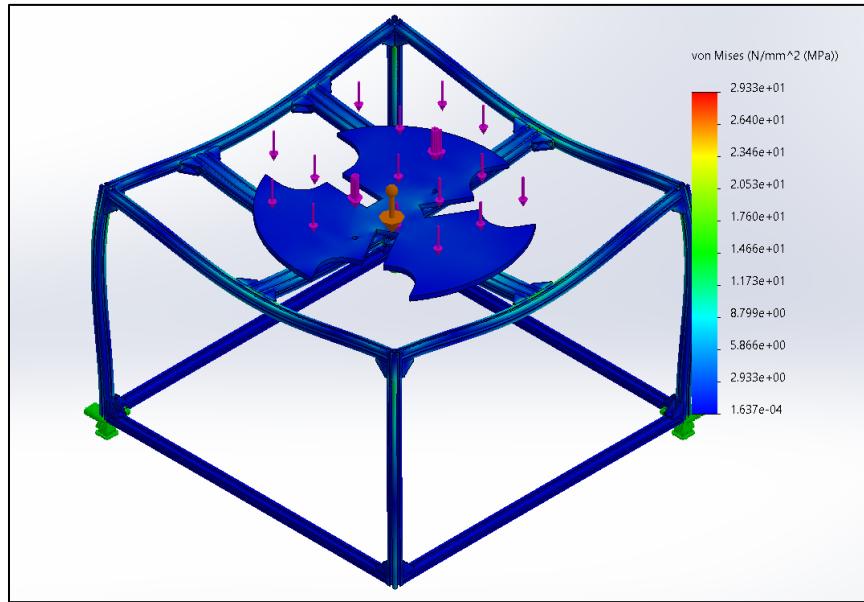


Figure 17 Frame Iteration 2 Analysis Results

### 2.3.2.6 Final Frame Dimensions

The frame dimensions were chosen to account for approximately the total delta robot workspace as well as extra space for mounting other components. The dimensions are (LxWxH) 700x700x500 in millimeters.

## 2.4 Actuator Sizing & Selection

### 2.4.1 Torque Calculation Methodology

The approach is to find the torque requirements from the motor is to calculate each of the torque components and maximize each of them to find the maximum torque required from the motor.

$$T_{tot} = T_L + T_{acc} + T_{ext}$$

Where:

- $T_{tot}$  is the total torque requirement from the motor.
- $T_L$  is the load torque on the motor due to the static loads.
- $T_{acc}$  is the acceleration torque or the inertial torque required to accelerate the masses attached to the motor.
- $T_{ext}$  is the external torque which comes mainly from frictional losses in spherical joint, this torque is compensated with a safety factor at the end of the calculation.

NOTE: All calculations are calculated per 1 motor only.

Therefore, it is necessary to calculate the two components, load and inertial torques, the next subsections present how each of these two components were found.

### 2.4.2 Load Torque Calculation

The load torque component is simply the torque due to the static load. This component varies with the orientation of the linkages, since each orientation would result to load transfers between the linkages, thus the motors. Since we are only requiring the maximum value, we could say that this value is when the active arm is horizontal and the passive arm is vertical, as shown in the below fig. (). This assumption is valid since at that position, the vertical load at the end point of the active arm will be maximized and thus maximum load torque on the motor. We assumed that this motor at the position shown in the figure carries the total weight as well.

Thus, the maximum load torque could be computed as:

$$T_{L_{Max}} = (2m_1g + M_{tot}g).L + \frac{m_2gL}{2} + M_{tot}gr_b$$

Where:

- $M_{tot}$  is the total end-effector weight plus the weight of the handled object. This assumed to be 1kg. It is important to note that this is an exaggerated weight for safety purposes only.
- $m_1$  is the mass of passive arm and is equal to 90 grams.
- $m_2$  is the mass of the active arm and is equal to 109 grams.
- $L$  is the length of the active arm and is equal to 220mm.
- $r_b$  is the distance from the lower platform to the spherical joint and is equal to 103.923mm



Figure 18 Load Torque Calculation  
Delta Robot Configuration

This results to a value of:  $T_{L_{Max}} = 2.7 \text{ N.m}$

#### 2.4.3 Inertial Torque Calculation

Please refer to sec. 2.5.3 of this report before seeing this section.

Calculating the maximum acceleration of the motors is a tricky part, as we lack the control over the maximum acceleration because of the trajectory planning method we used (Cubic polynomial interpolation) (sec. of this report). The trajectory planning provides self-generated velocity and acceleration profiles according to the inputs it receives, which means various input data results in various acceleration profiles which makes the absolute maximum acceleration that can be achieved very unpredictable.

To overcome the above issue, we started with the Scenario Analysis (sec 2.5.3) process. This process involved computing different trajectory planning outcomes corresponding to different movements and motion combinations on the computed workspace. The scenario with the absolute maximum acceleration the output that we are looking for that will be used in the torque calculation.

Thus, the maximum inertial torque could be calculated as:

$$T_{acc_{Max}} = J_{tot} \alpha_{max}$$

Where:

$\alpha_{max}$  is the maximum angular acceleration found from the trajectory planning and is equal to:

$J_{tot}$  is the total mass moment of inertia around the axis of rotation of the motor, this is calculated using the CAD model on SolidWorks to find the mass moment of inertia of each part around that axis then add them together. That is for one motor only (end effector plus item weight still assumed to be 1kg):

$$J_{tot} = 2J_{passive\ arm} + J_{active\ arm} + J_{end\ effectot+item\ weight}$$

That resulted in value of:  $J_{tot} = 0.54\ Kg.m^2$

### 2.4.3 Final Motor Selection

#### 2.4.3.1 Final Torque Results

From the obtained results in sec. 2.4.2 and sec. 2.4.3 and sec. 2.5.3, the total torque requirement is:

$$T_{tot_{max}} = T_{L_{max}} + J_{tot}\alpha_{abs_{max}}$$

This result in:  $T_{tot_{max}} = 3.5\ N.m$

## 2.5 Simulation & Control

### 2.5.1 SIMSCAPE Simulation

Although there are many adopted simulation software that proven to provide accurate results other than MATLAB and Simulink, our choice is backed up by several factors that led to choosing to work with MATLAB and Simulink:

- Familiarity and ease of use. MATLAB requires basic programming skills and provides a wide range of toolboxes to facilitate the simulation process.
- Having the ability to integrate all of the robot control in the future on one program, for example testing the embedded microcontroller code could be easily done on Simulink.
- Having the ability to test conducted work on the spot, such as the scenario analysis that is mentioned in this report.
- Having the ability to test control design.

SIMSCAPE have the capabilities of simulating the behavior of (position, velocity, acceleration, forces and torques), which are important to validate our control circuits)

We used the SolidWorks cad model and exported it using SIMSCAPE Multibody extension. This creates an XML file containing rigid parts, joints and other robot parameters. Then, we import the XML file on MATLAB Simulink, and start configuring those parameters.

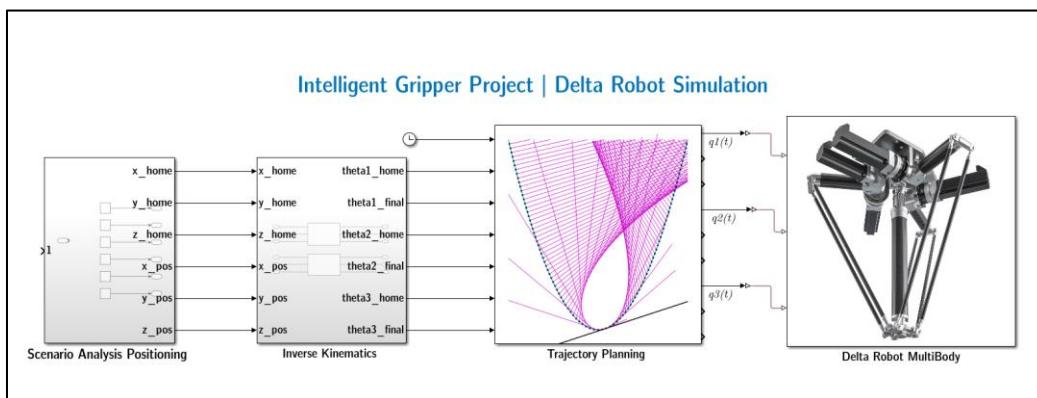


Figure 19 Simscape Simulation program on Simulation

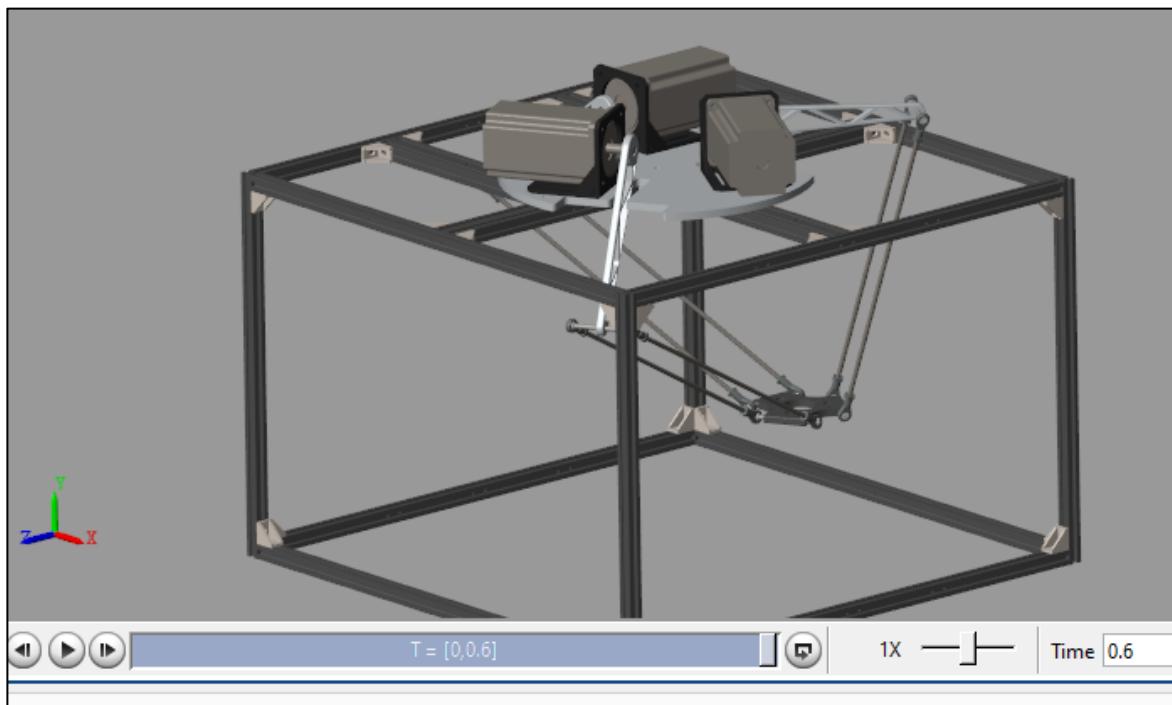


Figure 20 SIMSCAPE Simulation Screenshot

## 2.5.2 Trajectory Planning and Point to Point Motion

Trajectory planning was necessary in both the simulation and the Realtime implementation phases of the project, yet, it is necessary to clarify first how we aimed to achieve motion from one point to another, and how trajectory planning helped us achieve this motion in a synchronized, vibration-free fashion. Figure () shows how the point-to-point motion is achieved, first the desired end effector positions are converted into actuator angles using inverse kinematics, then, motor speeds are computed using trajectory planning.

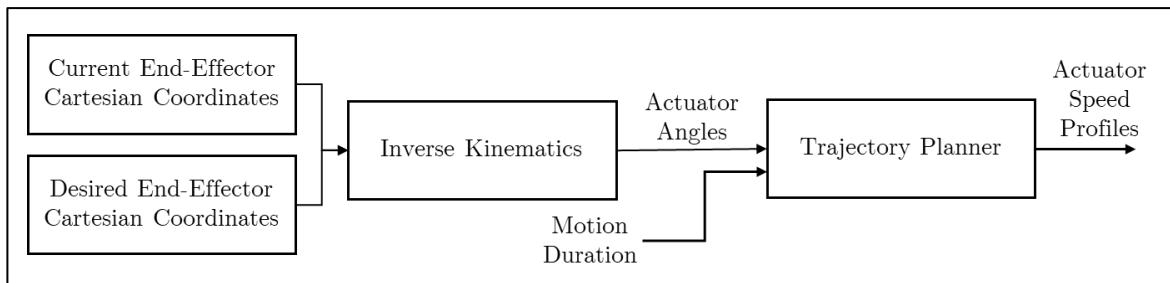


Figure 21 Trajectory Planning Methodology

To summarized we needed to design our trajectory planning as soon as possible for the following reasons:

- Calculate inertial or the acceleration torque for the motor selection.
- Robot motion simulation
- Robot motion control in real-time.

### 2.5.2.1 First Approach: Cubic Spline Interpolation

We first decided to use Cubic polynomial interpolation to generate our trajectory.

Cubic polynomial interpolation is a method that generates position, velocity, and acceleration profiles.

The idea is to create a third-degree polynomial relationship between motor angular position( $q$ ) and time ( $t$ ) such that:

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

we incorporated this equation in a matrix form:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_1 & t_1^2 & t_1^3 \\ 0 & 1 & 2t_1 & 3t_1^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q(t_0) \\ q(t_0) \\ q(t_1) \\ q(t_1) \end{bmatrix}$$

For us to be able to solve the matrices for the constants ( $a_0$  to  $a_3$ ), 4 boundary conditions are required. These conditions are the initial position  $q(t_0)$ , initial velocity  $\dot{q}(t_0)$ , final position  $q(t_1)$  and final velocity  $\dot{q}(t_1)$ . In addition to  $t_0$  and  $t_1$  which are the starting and final time in reference to the reference time of the system.

We obtain the constants vector by the following relation:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_1 & t_1^2 & t_1^3 \\ 0 & 1 & 2t_1 & 3t_1^2 \end{bmatrix}^{-1} \begin{bmatrix} q(t_0) \\ q(t_0) \\ q(t_1) \\ q(t_1) \end{bmatrix}$$

We use a MATLAB script to calculate the constants using the above equation then create our profiles.

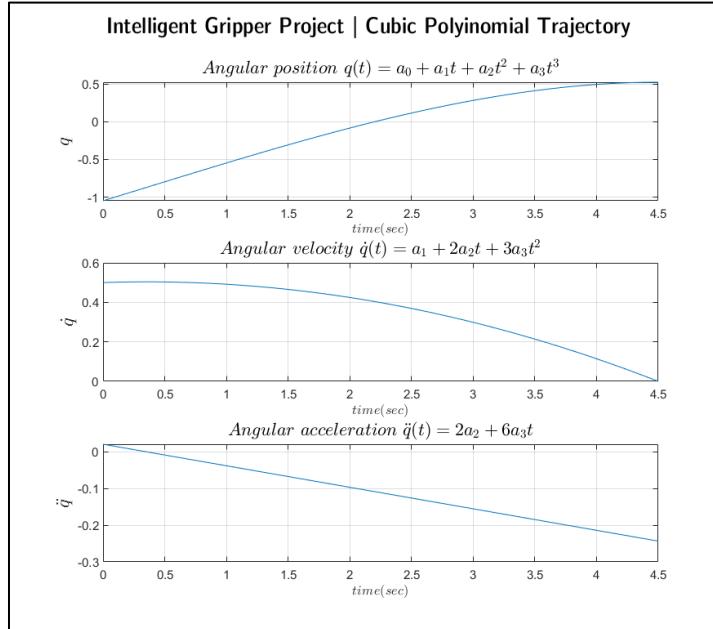


Figure 22 Cubic Spline Interpolation Results Plots

Why didn't this approach work?

After implementing this approach on the robot in real life we encountered some huge problems regarding the robot's point to point movement. The problems are summarized as follows:

- Inaccuracy: There was a high degree of inaccuracy in the robot movement as the robot failed to reach the exact goal location. This is because the Cubic polynomial interpolation incorporates a non-linear speed profile as shown in figure 7. This nonlinear profile is very hard to apply on our stepper motors.
- Jerky movement: Also related to the non-linearity of the speed profile of the Cubic polynomial, the self-generated acceleration profile which we have no control on, introduced areas of high accelerations and decelerations which causes the motor to jerk due to sudden change in its speed.

#### 2.5.2.2 Second Approach: Linear Speed Profile (Triangular & Trapezoidal)

In this section, we present the alternative approach of performing the trajectory planning process.

It is required to design a control architecture to allow the robot to be positioned between specified points in the workspace, which are specified by the inverse kinematics. In addition, the control procedure must account for the vibrations accompanied with the robot motion which are primarily due to the stepper motor motion nature. Finally, after the motors are able to move to specified positions accurately and without vibration, the three stepper motors must be synchronized to start and stop at the same time, regardless of their motion stroke.

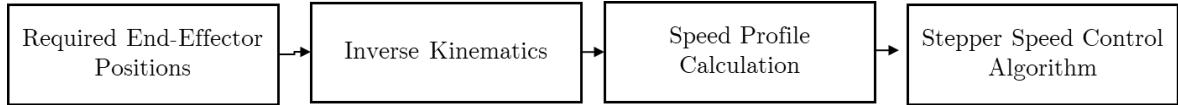
To achieve these requirements, we could divide the delta robot control procedure into two main procedures:

1. Integration between inverse kinematics & stepper motor interfacing on the embedded software.
2. Inclusion of an algorithm that allows the stepper motors to move with a specified speed profile (trapezoidal and triangular) to allow for acceleration and deceleration throughout the point – to – point movement to achieve a smooth, vibration-free positioning.

Initially, we moved the three stepper motors from point to point with constant speed, which resulted in two major issues, the first one is the three motors not being synchronized with each other; thus, the robot kinematics are not satisfied.

The second, and most important issue is the vibrations. Predominantly, the stepper motors motion nature involves discrete steps rather than continuous rotation. This step-by-step motion can result in vibrations at low speeds as the delay between each step becomes high, thus, the vibration become observable. Another vibrational issue is that if the motors start at stop at the same constant speed, the robot end effector stops instantaneously from this constant speed to zero, which result in very high vibrations due to the motors inertia.

To resolve these issues, the stepper motors must move with a speed profile that involves acceleration and deceleration, to reduce the effects of vibrations and jerk. To achieve this speed profile, a control algorithm was used based on an application note named ‘AVR446: Linear speed control of stepper motor’ presented by ATMEL [7], which introduces an algorithm that allow variation of stepper motor speed in real-time operations using 16bit timer interrupts.



The generation of the profile itself was quite easy. A linear speed profile is characterized by the maximum angular velocity as well as the acceleration and deceleration required. To calculate these, the required angular stroke by the actuator is formulated using inverse kinematics and then the following equations are used:

$$\omega = \text{angular stroke} * \text{duration} * 0.5$$

$$\alpha = \omega * 0.5$$

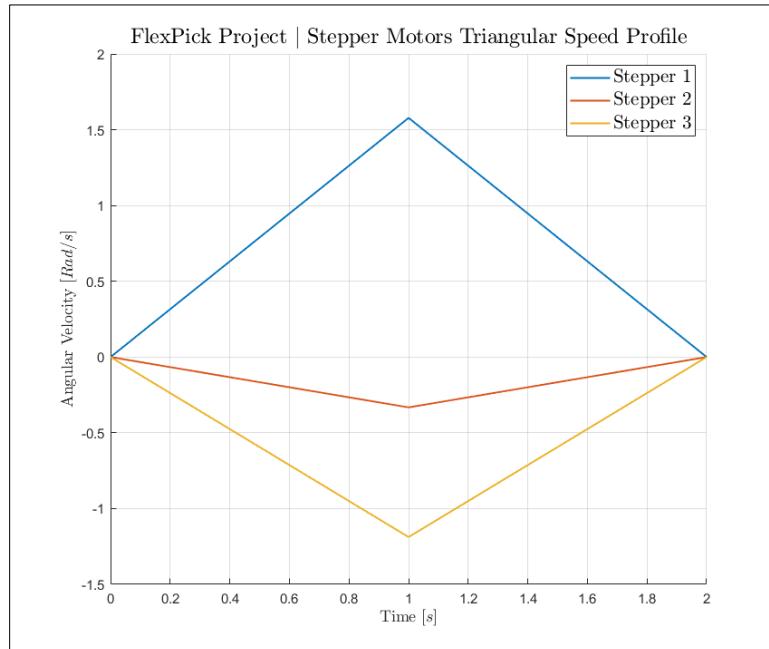


Figure 23 Visualization for Triangular Speed Profile

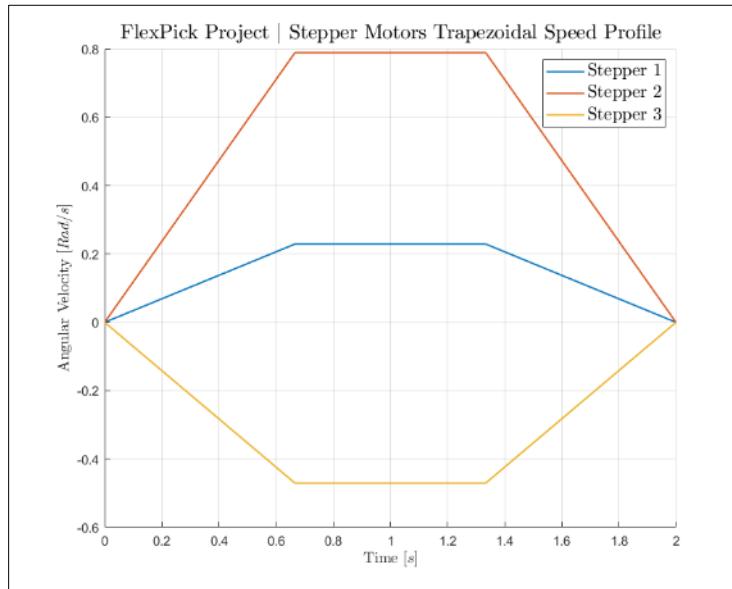


Figure 24 Visualization for trapezoidal speed profile

### 2.5.3 Scenario Analysis

For the current project stage, it is a necessity to start to select the motors that will drive the system. The motor selection is usually based on a complete simulation that outputs the torques, speeds, and accelerations required from the motor. However, as mentioned in the previous section, there are still some errors in the simulation program, this required us to use an alternative fast way to find maximum acceleration (thus maximum torque) and maximum speed requirements from the motor.

#### 2.5.3.1 Methodology

It is impossible to compute all possible speeds and accelerations required on all of the workspace, however, a common methodology used and found in the literature is *Scenario Analysis*, which simply tries different combination motions between the extremes of the workspace, i.e. moving the end effector to extreme positions, which we formulated in sec. II of this report. The process simply is to take in different start and end points on the boundaries of the workspace, feed them to the inverse kinematics, which alongside with a reasonable acceleration time, will be fed into the trajectory planning, which will output the speed and acceleration of this scenario, then different scenarios are tried and the one with maximum acceleration and speed is selected to be the motor requirement.

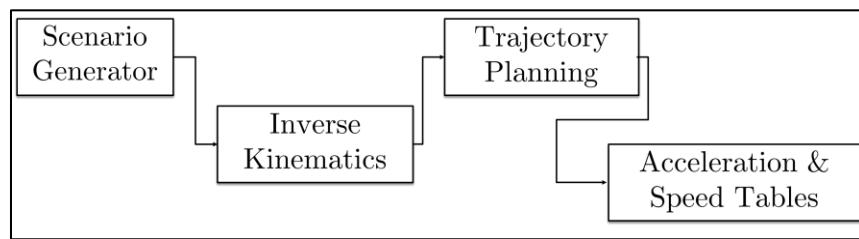


Figure 25 Scenario Analysis Methodology

#### 2.5.3.2 Operating Points

As shown in sec. II of this report, the operating working space is selected to be a cube like area in the middle of the overall reachable points. This area was 250 x 250 mm in the X\_Y plane and had an elevation range (Z-range) from -350 to -550. If we selected to work at

constant. The following figure (26) shows each point that will be used in this analysis on the workspace:

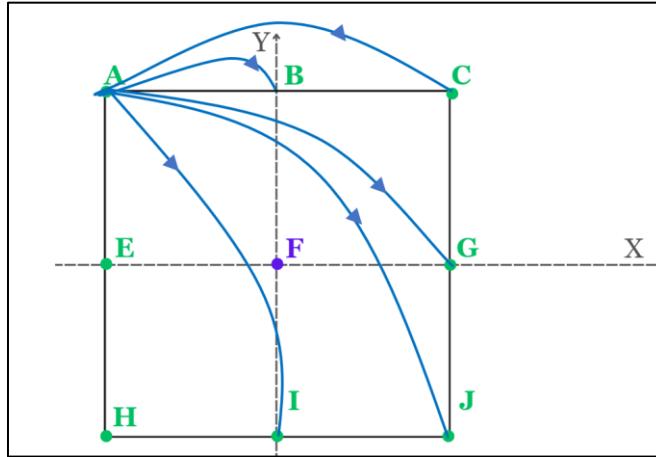


Figure 26 Scenario Analysis Operating Points

Where the point "F" represents the home position and the green points denoted from A-J are the different points that lie on the boundaries of the robot workspace and the arrows represent the "scenarios" that the robot may conduct.

The following inputs table summarizes each scenario with its starting and ending points, equivalent motor angular positions (initial and final). It is important to note that not all scenarios here are computed since only 3 starting points are required (A, E, H) and the other will be redundant since the workspace is a cube. (All dimensions are in mm and degrees).

Table 4 Scenario Analysis Inputs Table

Point	X	Y	Z	$\theta_1$	$\theta_2$	$\theta_3$
A	-200	200	-500	58.88043	46.87	1.242215
B	0	200	-500	49.29285	14.63068	14.63068
C	200	200	-500	58.88043	1.242215	46.87
E	-200	0	-350	-3.1956	25.97988	-32.8094
F	0	0	-350	-16.7188	-16.7188	-16.7188
G	200	0	-350	-3.1956	-32.8094	25.97988
H	-200	-200	-550	22.49883	71.18815	39.37846
I	0	0	-500	17.95421	17.95421	17.95421

J	200	-200	-500	11.02249	28.59179	64.14451
---	-----	------	------	----------	----------	----------

### 2.5.3.3 Scenarios Simulation Program

The simulation program is a simple MATLAB code that sets an initial position, set all possible end positions, and input each combination as a pair to the simulation program and allow it to run and take back from it the maximum values. The simulation program is computed in Simulink as shown in the below figure, the inputs from the running MATLAB code are fed to the inverse kinematics, which feeds in equivalent initial and final angular positions for the motor and feed them into the trajectory planning. The MATLAB code then takes the outputs of the trajectory planning and get its maximum and record it in the results table as the maximum speed and acceleration for this scenario.

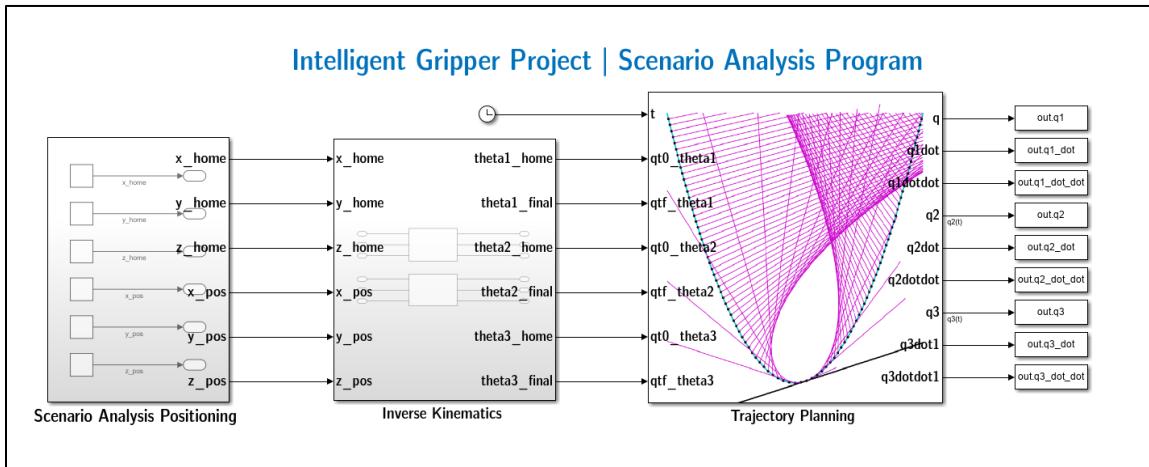


Figure 27 Scenarios Simulation Simulink

### 2.5.3.4 Results

Table 5 Scenario Analysis Results: Starting Point A

Scenario	$\omega_{1max}$	$\omega_{2max}$	$\omega_{3max}$	$\alpha_{1max}$	$\alpha_{2max}$	$\alpha_{3max}$
<b>A - B</b>	0	0	0	0	0	0
<b>A - C</b>	0.251	0.844	0.351	1.004	3.376	3.376
<b>A - E</b>	0	1.195	1.195	0	4.778	4.778
<b>A - F</b>	1.625	0.547	0.891	6.501	2.188	2.188
<b>A - G</b>	1.979	1.665	0.47	7.917	6.659	6.659
<b>A - H</b>	1.625	2.086	0.648	6.501	8.344	8.344

<b>A - I</b>	0.952	0.637	0.998	3.81	2.547	2.547
<b>A - J</b>	1.071	0.757	0.438	4.286	3.028	3.028

Table 6 Scenario Analysis Results: Starting Point B

Scenario	$\omega_{1_{max}}$	$\omega_{2_{max}}$	$\omega_{3_{max}}$	$\alpha_{1_{max}}$	$\alpha_{2_{max}}$	$\alpha_{3_{max}}$
<b>E - A</b>	1.625	0.547	0.891	6.501	2.188	2.188
<b>E - B</b>	1.374	0.297	1.242	5.497	1.188	1.188
<b>E - C</b>	1.625	0.648	2.086	6.501	2.591	2.591
<b>E - F</b>	0	0	0	0	0	0
<b>E - G</b>	0.354	1.118	0.421	1.416	4.471	4.471
<b>E - H</b>	0	1.539	1.539	0	6.156	6.156
<b>E - I</b>	0.673	1.184	1.89	2.691	4.734	4.734
<b>E - J</b>	0.554	0.21	1.329	2.215	0.84	0.84

Table 7 Scenario Analysis Results: Starting Point H

Scenario	$\omega_{1_{max}}$	$\omega_{2_{max}}$	$\omega_{3_{max}}$	$\alpha_{1_{max}}$	$\alpha_{2_{max}}$	$\alpha_{3_{max}}$
<b>H - A</b>	0.952	0.637	0.998	3.81	2.547	3.994
<b>H - B</b>	0.701	1.481	0.648	2.806	5.923	2.592
<b>H - C</b>	0.952	1.831	0.196	3.81	7.325	0.785
<b>H - E</b>	0.673	1.184	1.89	2.691	4.734	7.559
<b>H - F</b>	1.027	2.301	1.469	4.107	9.206	5.874
<b>H - G</b>	0.673	2.723	0.351	2.691	10.891	1.403
<b>H - I</b>	0	0	0	0	0	0
<b>H - J</b>	0.119	1.394	0.561	0.476	5.575	2.244

All values are in rad/s and rad/s<sup>2</sup>. Acceleration time is assumed to be 1 sec. for each scenario.

### 2.5.3.5 Absolute Maximums

From the results obtained, the absolute maximum speed and acceleration values which will affect the motor selection are:

$$\alpha_{abs_{max}} = 10.8 \frac{rad}{sec^2}$$

$$\omega_{abs_{max}}=2.722\frac{rad}{sec}$$

## **2.6 Problems Faced**

### **Mechanical assembly:**

- Some rod ends got stiff, and its solution was lubrication.
- For every two passive arms, their rod ends have to be facing each other to achieve proper movement. The solution was to fasten rod ends for one passive arm and use it as a reference to fasten the others.

### **Vibration tuning:**

Due to acceleration and deceleration vibration occurs.

To tune it we installed springs between each two passive arms to reduce vibrations as much as possible.

## Chapter 3

# Soft Robotic Gripper

### 3.1 What is a Soft Robot

Soft robotics is a new class of robotics that is bio-inspired from humans and biological creatures. Soft robots could be best described as soft materials where motion is achieved by elastic deformation in the structure itself. Compared to rigid robots, they have more flexibility, adaptability, sensitivity as well as infinite DOFs. Unlike traditional rigid robots, soft robots are made from soft, hyperplastic, extensible and compliant materials, which makes them have almost infinite degrees of freedom, flexible, lightweight, energy efficient, and more importantly for our application, compatible with any geometric irregularities. These stated features make soft robotics best suited for various applications in several field such as biomedical, industrial applications, exploration, and other applications, see figure (28).

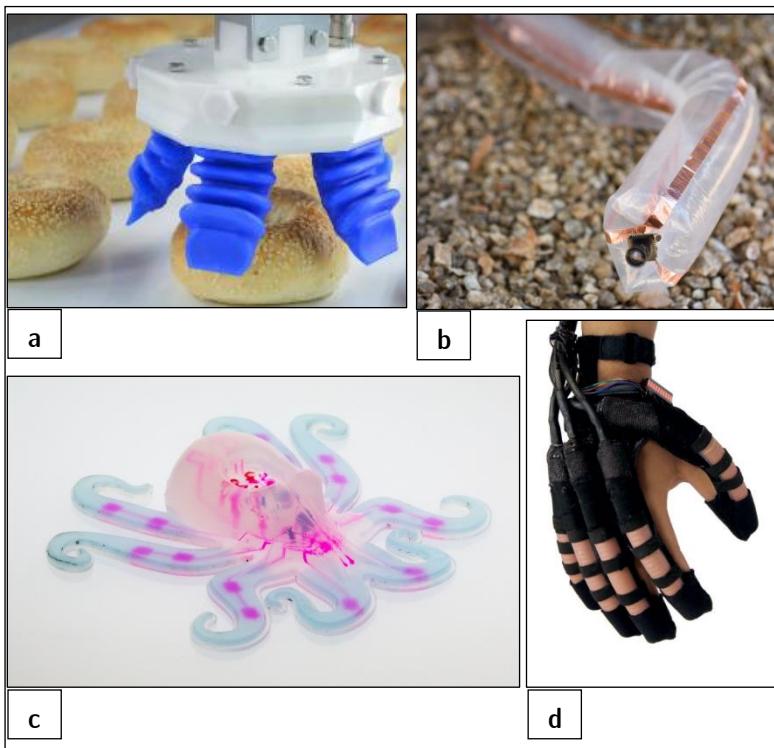


Figure 28 Different Applications of Soft Robots [7] & [8]

a) Industrial Soft Robotic Grippers utilized in food packaging systems (soft robotics inc.)

b) Exploration soft robot (vine robot) used in

### 3.2 Our Project Application

As mentioned, in sec.1 of this report, we were aiming to implement a project that both solves an industrial problem or provide a novel industrial solution, as well as enabling us to explore a new, forthcoming field. Therefore, we decided to implement in our pick and place system a soft robotic industrial gripper. Such an application would be beneficial for simply any pick and place operation that requires delicate and sensitive gripping on the picked item without causing any deformation to it while having approximately no prediction what does the geometry of the picked item is formed, it simply adapts to it. As shown in the figure (29), a soft gripper easily adapts to the shape of the picked item, unlike traditional motor-actuated rigid grippers, which requires accurate calculations of the correct geometry of the picked item, as well being soft, the soft gripper won't cause any damage to the picked item. Soft grippers could include embedded sensors inside of them, which makes them capable of even more delicate gripping. This field is yet undeveloped commercially, especially in the region, so we decided to step into it, academically and commercially.

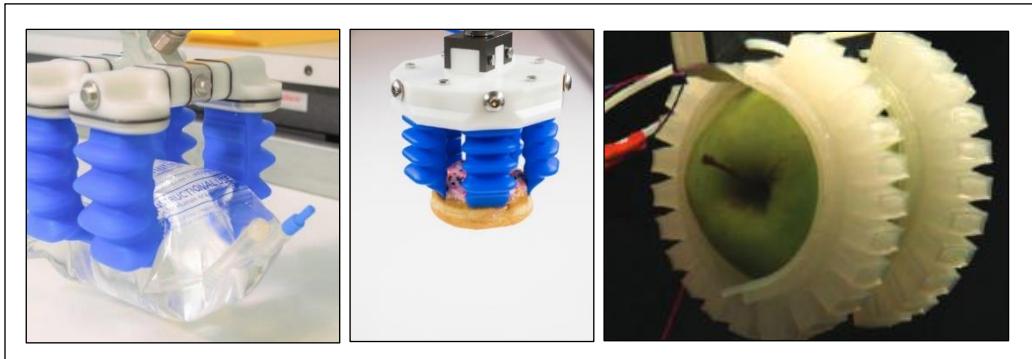


Figure 29 Different Examples of Pre-Existing Soft Robotic Gripper

We finally could summarize the benefits of implementing such grippers as:

- Adaptability to different geometries and surface irregularities in food products or harvested corps. (mimicking the human hand)
- Lower possibility of product damage as most of these products are fragile or deformable.
- Better accommodation for moisture as less electronic parts is used.
- Operate at high speeds with more accuracy and less damage.

### **3.3 Pneumatic Networks Construction**

#### **3.3.1 Choosing an Actuator Construction**

To be able to integrate the soft gripper with the robotic positioning system (delta robot) and the computer vision system, we need to establish a well – defined model for the gripper in order to be able to control this model via a pneumatic control circuit. There are several models that were available in the literature regarding soft robotic actuators, each with a different construction. Our main focus was on researching the models that utilized pneumatic actuation.

Pneumatic soft actuation itself is a whole topic, with many constructions available in the literature, however, we chose to work with a construction called the Pneumatic Networks or the “PneuNets” (explained in the next section) since it is the most researched and developed among the soft robotic field, enabling us to have a well-developed base to work on. In addition, PneuNets are easily modeled via Finite Element Analysis, unlike the other constructions which poses difficulties in modeling and could involve the use of empirical approximations to find a model for the actuator.

A great contributor to our decision of working with PneuNets is also the fact that almost all products available in the market, which included in the market analysis section, are designed with the PneuNets actuator construction. These solutions deployed PneuNets primarily due to its ability to produce high curvature and high gripping forces at relatively low cost (pressure). Thus, working with it will be very beneficial, as we could try to reverse engineer some of their properties and try to improve it.

#### **3.3.2 PneuNets Principle**

Perhaps the first time the Pneumatic Networks or the PneuNets was introduced was by the Harvard Whitesides Mechanical Engineering Laboratory [9], where they first introduced the concept of PneuNets.

PneuNets are simply elastomers that have channels or chambers in which pressurized air could pass through. Areas with lower stiffness in the channel will expand or strain more, giving the bending required. In homogeneous elastomers, lowest stiffness is at the thinnest areas. Thus, the curvature depends on the geometry of the channels and the material properties. Upon pressurization, the thinner parts elongate or expand or become “thinner” and the volume of inside the channel increases. These channels are spanning a rectangular slab which is bent around the channel’s axis when channels are strained. Fabricating PneuNets from composite materials where the young modulus of the bottom part or the slab is higher than the young modulus of the channel material would cause the bottom part to slightly bend and the channel to expand. The stiffer the bottom slab is, the less bending it could accommodate, thus they could bend for the same pressure to a smaller radius of curvature, or they’re said to be more agile (achieve more complex motion faster).

PneuNets themselves could be designed in various constructions, there is no perfect design available for them, every design available in the literature was made for its specific purposes or for illustration only. However, there several factors that we could take into consideration when designing our actuator, which are discussed in the next section of the report.

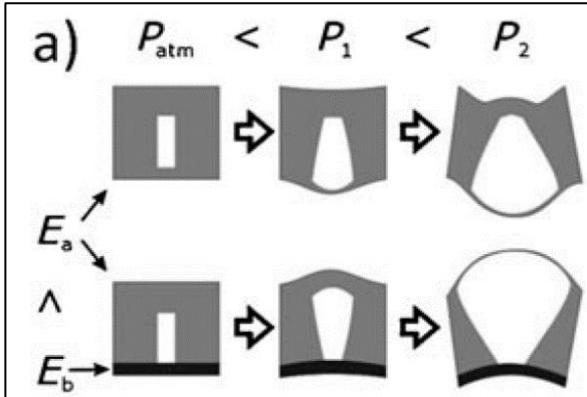


Figure 31 PneuNets Illustration Diagram

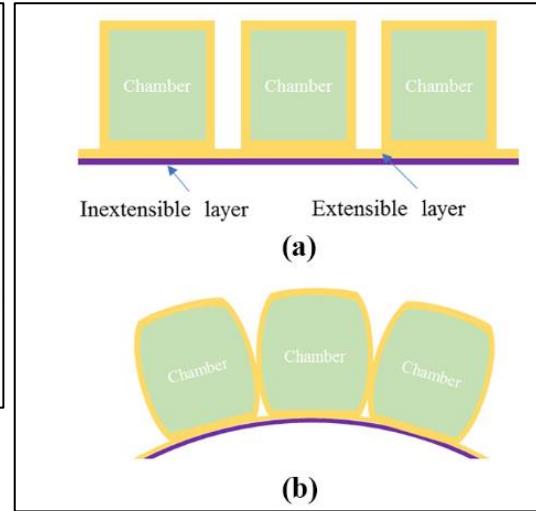


Figure 30 Side Section of PneuNets [10]

### 3.4 Gripper Design Methodology

This subsection of the report discusses and summarizes the procedures & processes took whilst designing, modeling, and fabricating the soft robotic gripper. These processes helped us design a gripper capable of handling various weights and objects sizes, specifically food items. Modeling the gripper's behavior was necessary as well in order to optimize its design to follow the design requirements, these optimizations are done by changing the deign materials and evaluating bending angle and force results.

Due to the poor material availability & time constrictions, we decided to not consider modeling and experimental testing only for making design decisions, yet we decided to include another design decision maker, which is previous design considerations, in which for each reviewed design we reviewed specific parameters and include them design modifications. These considerations are summarized in section () of this report.

The below diagram (figure (28)) illustrates the gripper design methodology.

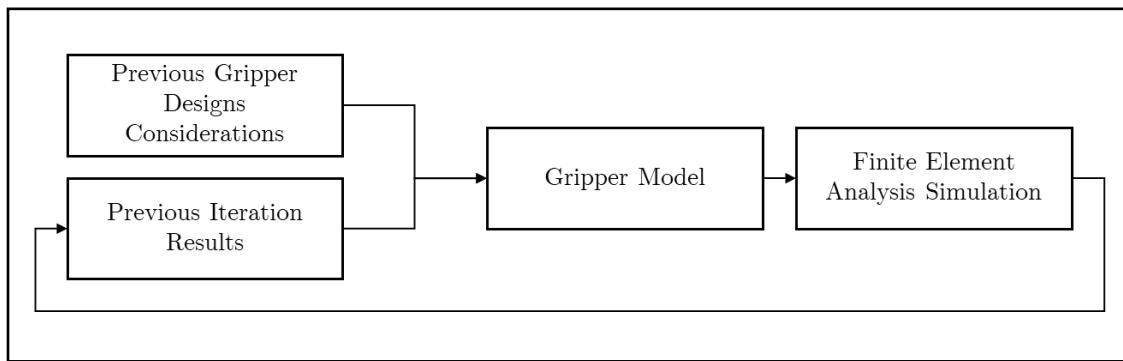


Figure 32 Gripper Design Methodology Illustration Diagram

Finite Element Analysis Simulation (FEA) was performed to evaluate the gripper behavior throughout the design process (sec. 3.6)

## **3.5 Material Selection & Hyperelastic Material Modeling**

This subsection of the report discusses the process of selecting and modeling the soft gripper's material. In this subsection we discuss how we modeled the material as well, and how we categorized it using hyperelastic material strain energy models found in the literature to achieve accurate modeling results that would help us in the design process.

### **3.5.2 Material Selection**

For such application, in which high, large deformation is required against pneumatic actuation (regularly could achieve  $>500\%$  strain [11]), as well as being rubber like, flexible, and bio-compatible. According to our literature review [9 - 14], silicone rubbers could be the best fit in such material requirement, since they combine all the stated features as well as being commercially available.

To select a suitable silicone rubber for our application, we reviewed a paper published by Marechal et al. [12], in which they published a complete framework to clarify the process of material selection and modeling for silicone rubbers in soft robotics. They stated the material that were used the most in the literature as well as performing tensile stress tests to calibrate each material model (discussed more in sec. 3.5.3). The following figure (33), shows a complete stress – strain curve for the most used silicone rubber.

We could observe from the figure that the material EcoFlex 0030 (produced by Smooth On inc.), could undergo large amounts of stretching whilst having minimal stresses, which means it would require less pressure for a pneumatic or fluidic actuator to inflate a structure based on deformable membranes. In addition, EcoFlex series is widely commercially available, easy to use [15], and food-safe [15].

For these reasons, we chose Smooth On's EcoFlex 0030 to be our soft robotic gripper material. A detailed table of the material properties could be found in the appendix of this report.

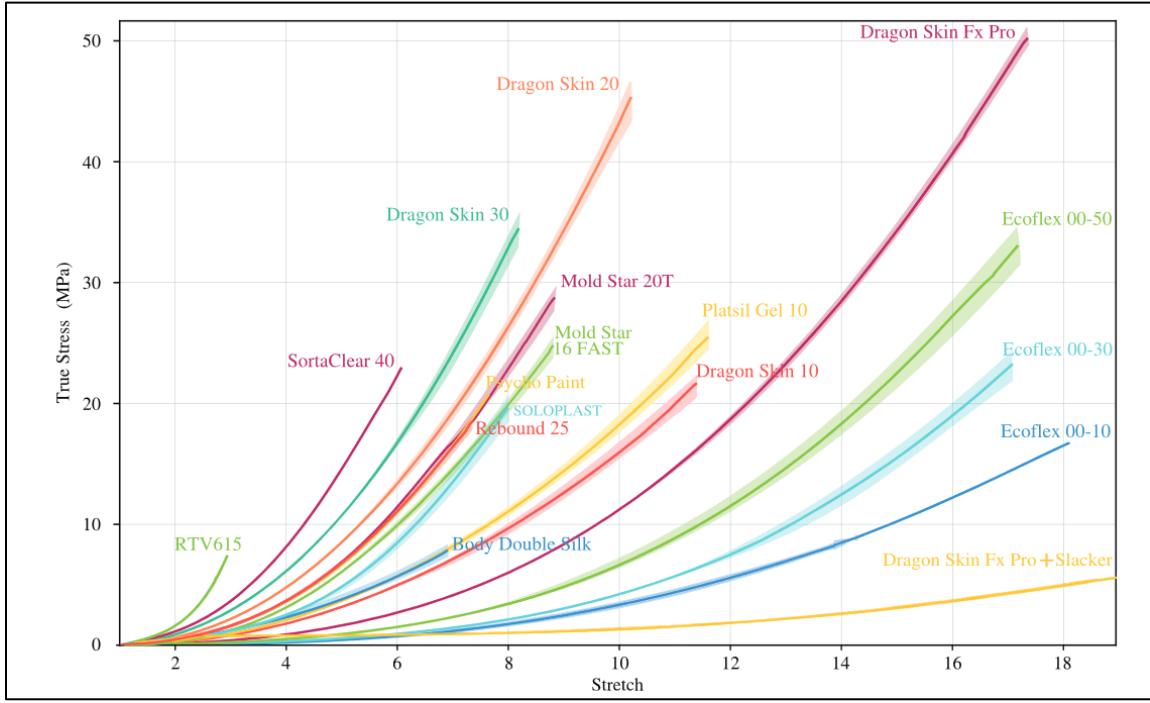


Figure 33 Stress-Strain Results for Different Materials used for Soft Robotic Applications [12]

### 3.5.3 Hyperelastic Material Modeling

#### 3.5.3.1 Introduction & Significance

To accurately model the gripper and simulate its behavior, stress – strain data must be present as an input to the simulation. Due to the extreme non-linear behavior of silicone rubbers, information available by the manufacturer, which is only shore hardness, is not enough to have such data. Therefore, in the process of modeling such materials, it was found in the literature [12],[14] that hyperelastic strain energy material modeling is used to have stress-strain data of silicone rubbers. Hyperelastic material modeling is simply performing uniaxial and biaxial tensile test, record data, and try to fit a polynomial model, i.e., generate some constants, that could be best used to describe the stress-strain curve of the silicone rubber. There are several models existing that could be used to calibrate stress-strain data, perhaps the most known one is the Mooney-Rivlin model and the Neo-Hookean model, which

are curve fitting techniques based on strain energy functions and they could be from 3 to up to 9 parameters functions. An example of such models is available in figure (34)

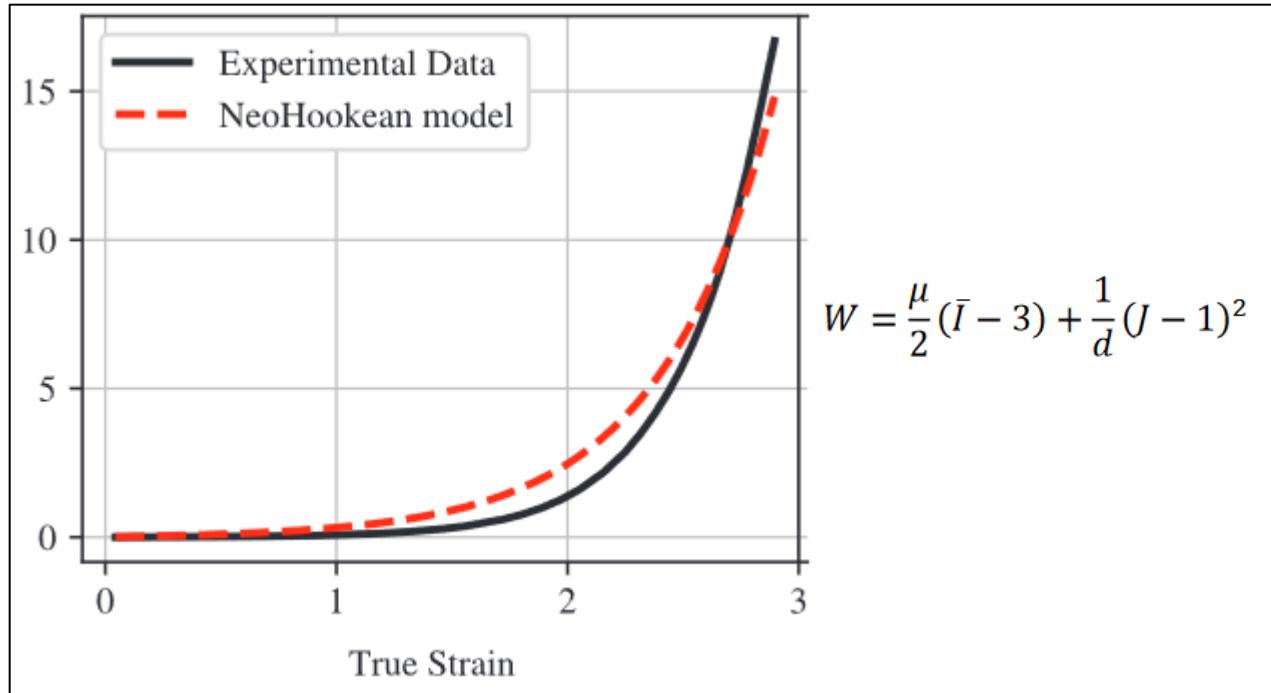


Figure 34 NeoHookean Hyperelastic Material Mode

### 3.5.3.2 Model Selection Process & Experimental Data

To choose a hyperelastic material model for our material, EcoFlex 0030, we had to approach two different approaches, the first one was to perform our own uniaxial and biaxial tensile tests and calibrate one of the available models to our test data. However, throughout our literature review, and since the material is widely available in soft robotic applications, we found various of previous projects & publications performing these tests on the same material, and providing calibration curve results, perhaps the two most important publications were [12] & [13]. The first one [12] as mentioned previously, performed stress strain curves on various materials and calibrated to almost all common-known material models, the second one, [13], did not perform a test, yet, the publishers some of the calibrated models found in the literature, as well as software used to calibrate them.

Therefore, to select an accurate material model to start our gripper modeling iterations, we performed our literature review on these models, constituted a table (table ) that summarizes each reviewed model, software used to calibrate the test data, and we rated its accuracy based upon how accurate was it to the experimental test data available in the reviewed paper (if exists). We then selected the model with the highest accuracy rating.

Table 8 Reviewed Literature Hyperelastic Calibration Models Formulated Comparison

EcoFlex 00-30 Previously used Constitutive Hyperelastic Material Models					
Model	Model Coefficients	Software	Test Type	Accuracy	Paper(s)
Yeoh 2nd Order	$D1 = D2 = 0$ $C10 = 9392.0915 \text{ Pa}$ $C20 = 50.93876918 \text{ Pa}$ $\text{Res.} = 3.000602064$	Ansys	Curve fitting with Ansys against a pre-defined test data from another paper	3	
Mooney Rivlin 9 Parameter	$D1 = 0$ $C01 = -1678.878751 \text{ Pa}$ $C02 = 946.0670013 \text{ Pa}$ $C03 = 0.001098034 \text{ Pa}$ $C10 = 9546.512215 \text{ Pa}$ $C11 = -1654.735597 \text{ Pa}$ $C12 = -0.072750706 \text{ Pa}$ $C20 = 1343.537247 \text{ Pa}$ $C21 = -234.700926 \text{ Pa}$ $C30 = 38.6465242 \text{ Pa}$ $\text{Res.} = 1.477312654$	Ansys	Curve fitting with Ansys against a pre-defined test data from another paper	5	[14]
Mooney Rivlin 3 Parameter	$C10 = 1.8 \text{ e-2 MPa}$ $C01 = -4.69 \text{ e-2 MPa}$ $C20 = 4.34 \text{ e-5 MPa}$	Python Program	Biaxial & Uniaxial Testing	5	[12]
Yeoh 3rd Order	$C1 = 1 \text{ e-1 MPa}$ $C2 = 1.2 \text{ e-2 MPa}$ $C3 = 4.96 \text{ e-5 MPa}$	Python Program	Biaxial & Uniaxial Testing	3	

Ogden	$\mu_1 = 22 \text{ kPa}$ $\mu_2 = 0.4 \text{ kPa}$ $\mu_3 = -2 \text{ kPa}$ $a_1 = 1.3$ $a_2 = 5$ $a_3 = -2$	Abaqus	Uniaxial Test	5	[16]
Ogden	$\mu_1 = 0.001887 \text{ MPa}$ $\mu_2 = 6.67e-5 \text{ MPa}$ $\mu_3 = 0.003574 \text{ MPa}$ $a_1 = -3.848$ $a_2 = 0.663$ $a_3 = 4.225$ $D_1 = 2.93 \text{ MPa}$ $D_2 = 0 \text{ MPa}$ $D_3 = 0 \text{ MPa}$	Abaqus	Biaxial & Uniaxial Testing	3	[17],[18],[13]

Accuracy Levels					
Excellent	5	Good	4	Fair	3
Poor	2				

### 3.5.4 Selected Model Calibration

The final selected model is the Mooney-Rivlin 3 Parameter model calibrated in [14] and Yeoh 2<sup>nd</sup> (table first & third entries). The model constants are entered on the engineering data of the simulation program on Ansys (see figures 33,34,35).

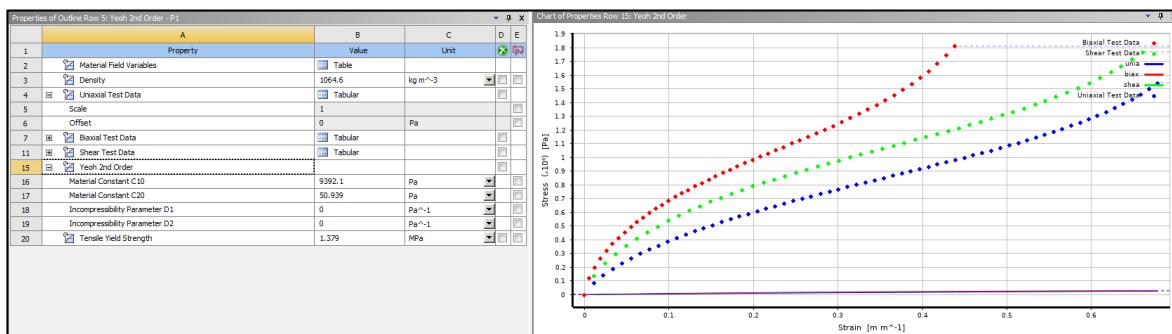


Figure 35 Yeoh 2nd Order Entry on Ansys

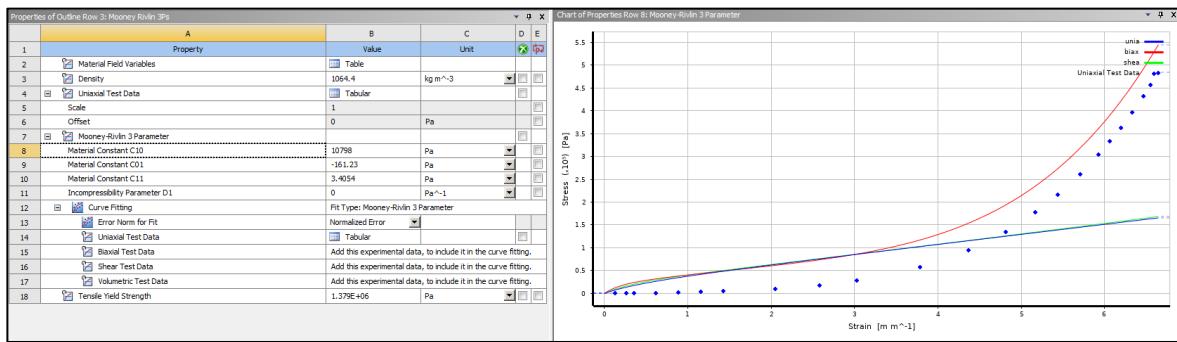


Figure 36 Mooney-Rivlin 3 Parameter Entry on Ansys

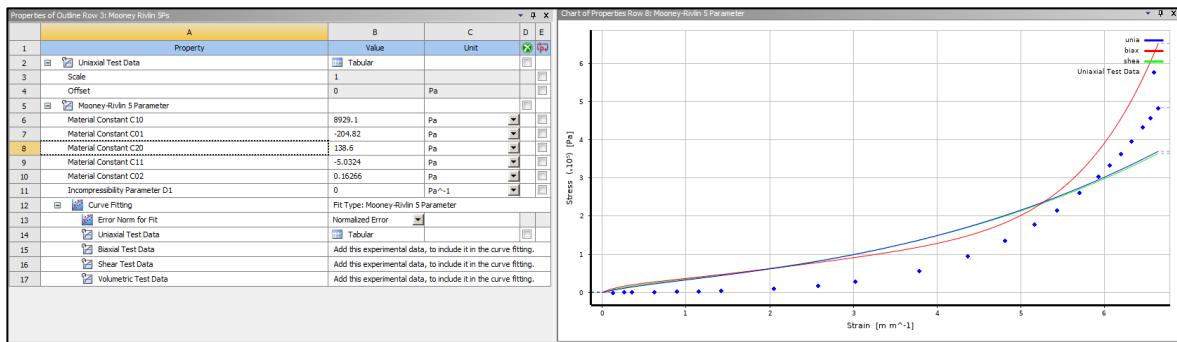


Figure 37 Mooney-Rivlin 5 Parameter Entry on Ansys

## 3.6 Gripper Design Optimization using Finite Element Analysis (FEA)

### 3.6.1 Design Objectives & Specifications

This section of the report discusses the process of designing the construction of the soft robotic gripper. The gripper is constituted from three pneumatically actuated chambers (PneuNets) assembled together on the end effector of the delta robot. To optimized the design of these chambers, i.e., their internal dimensions, finite element analysis is used as the modeling technique to evaluate each design modification. A common evaluation attributes found in the literature are the bending angle of the gripper as well as the contact force with the gripped object (see figures(39) (38)). In this section, we present how we aimed to optimize these two outputs by changing the following parameters:

Table 9 Gripper Designed Parameters

Gripper Designed Parameters		
Parameter Description	Notation	Units
Length of one actuator	$L$	-
Cross Section of one Actuator (geometry type and its area)	$CSA$	$\text{mm}^2$
Number of chambers or segments per actuators	$n$	-
Chamber wall thickness	$t$	$\text{mm}$
Inflating Pressure	$P_{IN}$	KPa

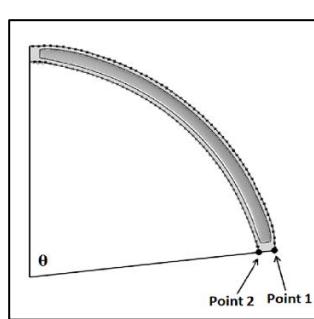


Figure 39 Bending Angle Illustration, from [201]

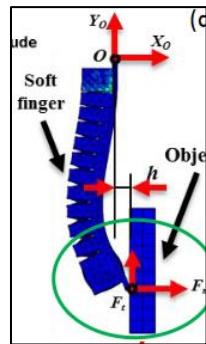


Figure 38 Contact Force Illustration, from [201]

### 3.6.2 Finite Element Analysis (FEA) Setup

To evaluate each design using FEA, several steps were conducted prior to the start of the simulation.

#### 3.6.2.1 Material Definition

As mentioned previously, this might be the most important step in the simulation, we define the material as shown in the previous material modeling section figures (40), we enter the hyperelastic material calibration model constants as well as the available data from the manufacturer (density & Tensile strength) found in the appendix of this report.

#### 3.6.2.2 Model Entry

We output the gripper CAD model from SOLIDWORKS software as a step file. We then imported it in the geometry section in Ansys software.

#### 3.6.2.3 Mesh Settings

We use a non-linear quadratic mesh with a mesh size varying from 3mm to 12mm (this range was found through reviewing previous literature).

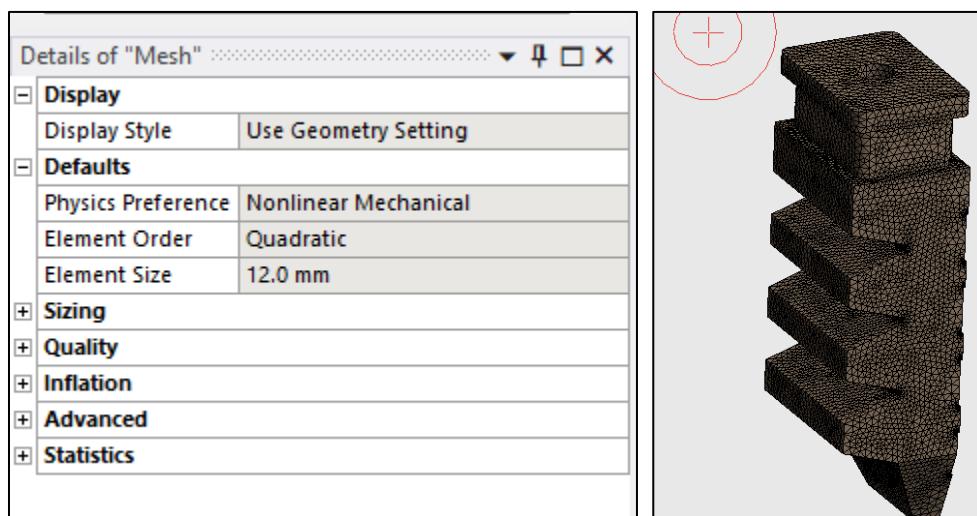


Figure 40 Ansys FEA Mesh Settings

### **3.6.2.4        Input Pressure Load**

The actuation pressure is loaded normally (ramp input) against the inner walls of the actuator.

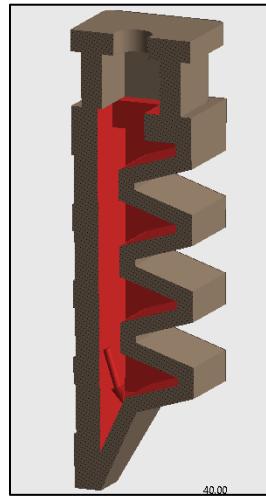


Figure 41 FEA Pressure Input Loading

### 3.6.3 Actuator Cross Section Study

#### 3.6.3.1 Design Choices

There are several design constructions for PneuNets we came across in the reviewed literature.

##### *Rectangular Chambers*

The basic construction for PneuNets, and the most used, is the rectangular construction, which is the one we used in our first iteration. This construction is the simplest to manufacture and to design as it includes minimum number of designed parameters. (Figure 39).

##### *Semi-Circular Chambers*

They are used with if the material's elasticity modulus is not high enough to handle stress concentrations at corners.

##### *Chamfered Rectangular Chambers*

This type is used extensively with the specific application industrial grippers since it provides relatively good gripping force with less pressure requirement than the one associated with the rectangular chambers.

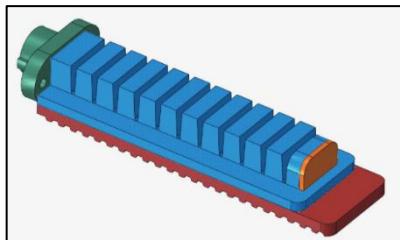


Figure 42 Chamfered Chambers Design From [10], which was a gripper designed to handle bakery products



Figure 43 Chamfered Design of a Real Product of Gripper Used. Source & Manufacturer: Data sheets from SoftGripping GmbH, Germany

### 3.6.3.2 Initial Conceptual Design (Rectangular)

The first design iteration we implemented for the gripper was the basic, rectangular, soft pneumatic actuator. We modeled the actuator using SolidWorks prior to performing FEA simulation.

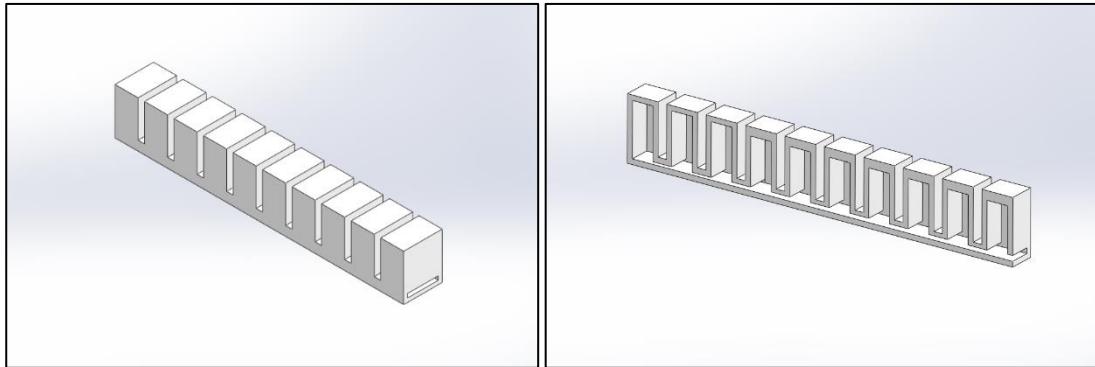


Figure 44 Soft Gripper iteration 0 3D model on SolidWorks

### Iteration Details

Table 10 Gripper Iteration 0 Details

<i>Designed Parameters of Iteration 0</i>
$N = 1$
$L = 75 \text{ mm}$
$CSA = 12 \times 10 \text{ mm}^2$
$n = 10$
$CD_i = 4 \times 8 \times 8.5$
$CD_o = 12 \times 10 \times 1$
$t = 1$
$P_{IN} = 0 \text{ to } 100 \text{ kPa (Ramp Inputted)}$

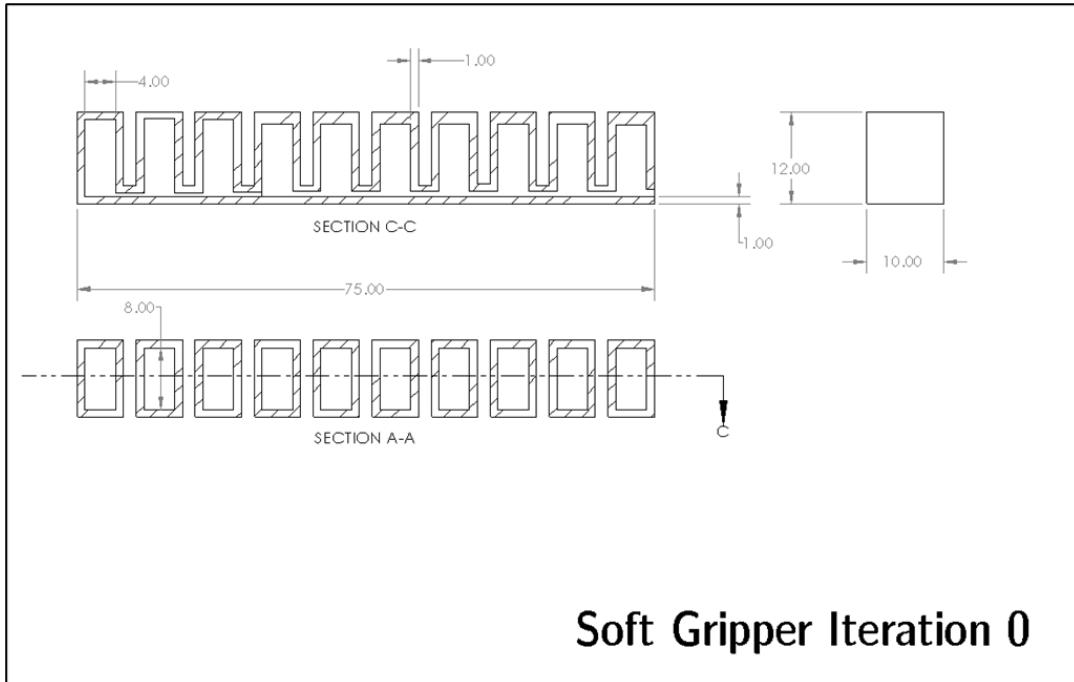


Figure 45 Gripper Iteration 0 Working Drawing

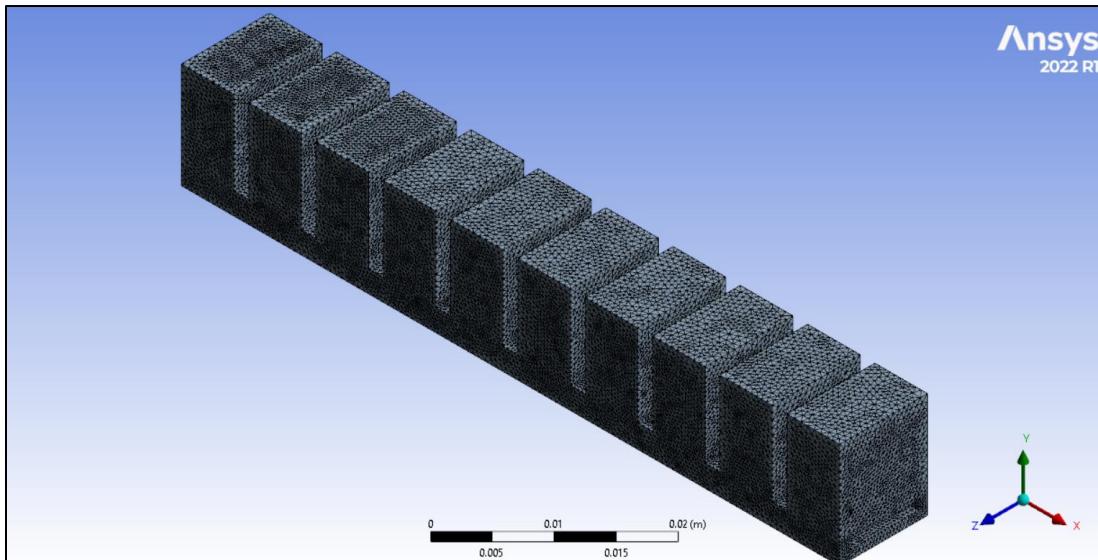


Figure 46 Gripper Iteration 0 Simulation

## Problems with the Rectangular Design

Under vacuum, the rectangular shape does not allow its walls to collapse; hence, the bending achieved in this type of soft actuator is very low. As seen in Figure 43, when the side walls tend to collapse inward, which stops the top wall to collapse inside and forces it to bend outward. Some bending is still achieved, but the geometrical constraints do not allow the top wall to collapse as much, and as a result, significant bending is not achieved. However, this is the case only in vacuum conditions. When under positive pressure, this shape helps a lot.

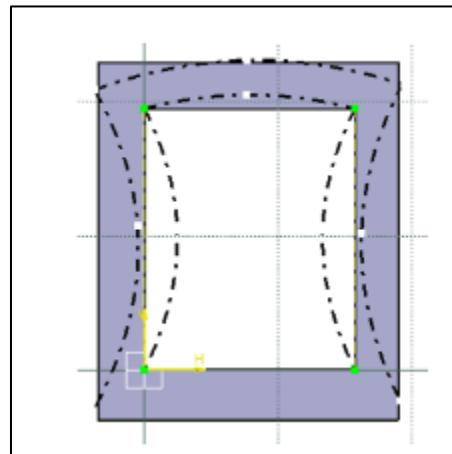


Figure 47 Rectangular Cross Section Collapse [12]

### 3.6.3.3 Hybrid Design (Chamfered)

The cross-sectional profile for the chamber should have the properties and advantages of the rectangular profile with the additional characteristic of working better under vacuum conditions. The top part can be modified for better performance to eliminate the problem of not collapsing from the rectangular cross-section. Instead, we can introduce a triangle-like structure to the rectangular profile and make a hybrid profile with a triangle on top of a rectangle, as shown in Figure 44. It can be observed from Figure 44 that the proposed design performs very well under a vacuum as the design of the actuator allows the side walls to collapse with negligible geometric resistance. The only obstruction this design has, is undergoing a bi-stable state under a vacuum state.

Nevertheless, because of the material properties of silicone elastomer, especially Eco-Flex 00-30, this has an insignificant effect on the soft actuator. Therefore, not only this design performs well under vacuum, but it also performs very well under positive pressure

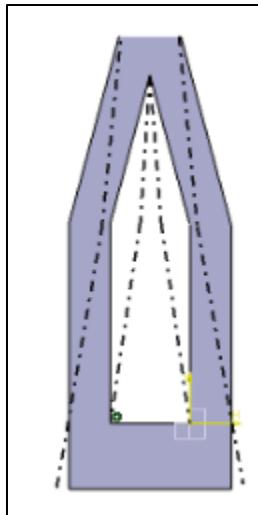


Figure 48 Hybrid Cross Section of Soft Gripper

### 3.6.4 Final Gripper Design

The final design of the gripper was performed to be with the hybrid cross section mentioned previously, as well as some extra modifications (figure 45) (simulation results on the final design are in section 3.8):

- Addition of a tip at the end of the gripper to help facilitate the gripping action of objects.
- Making the bottom layer textured so it could extra contact with the gripped object.
- Addition of an extra rectangular part at the top of the soft finger so it can incorporate the installation of 3D printed air connector.
- Decreasing the number of chambers to be 4 chambers.
- Changing the thickness of the chambers to be 2mm.
- Changing the thickness of the bottom layer to be 4.6mm

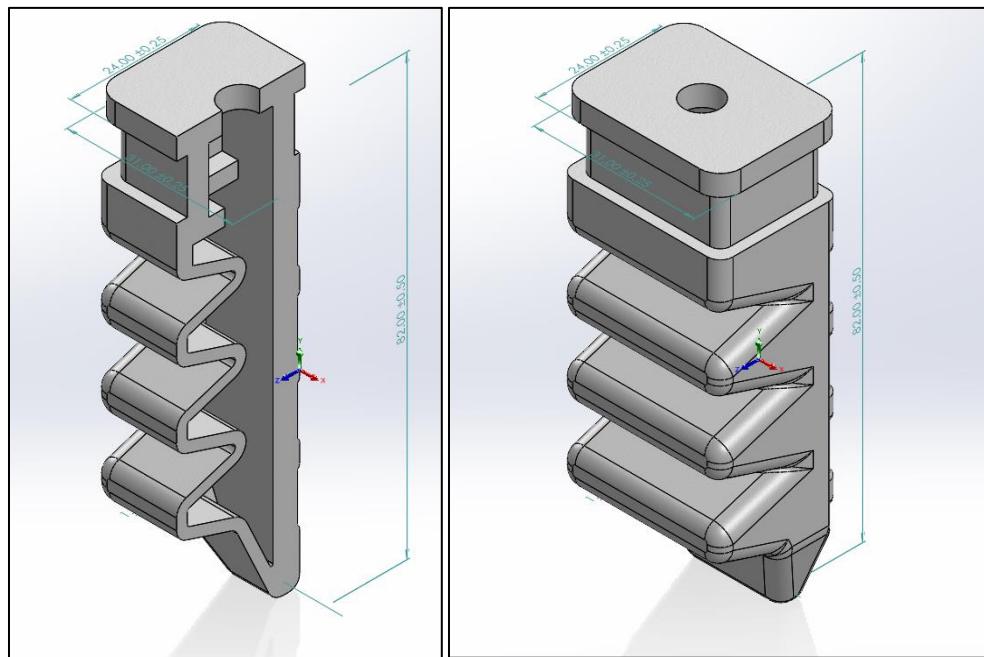


Figure 49 Gripper Final Design CAD Model

## 3.7 Pneumatic Controller

### 3.7.1 Requirements

It was required to design a pneumatic control system that allow the gripper to achieve the following states:

1. *Deflated State*; in this state, the pressure will be drawn from the soft robotic gripper, in other words, negative pressure is supplied to it. This is done to allow the gripper to open the gripping area by bending outwards so it could be prepared for gripping the item.
2. *Inflated State*; in this state, positive pressure will be supplied to the gripper allowing it to bend inwards, gripping the desired object. In this state, air must continuously be supplied to the gripper until pressure inside the gripper reaches a desired setpoint, after this point, pressure must be maintained.
3. *Idle State*; in this state neither positive nor negative pressure is supplied to the gripper, it will be present whenever the gripper is not required to pick an item.

### 3.7.2 Implementation

#### 3.7.2.1 Pneumatic Circuit Description & Specifications

To achieve the desired requirements, we first needed to simplify our circuit and construct it on a simulation software (Automation Studio) before implementing it. The following figure (50) shows the schematic diagram for the pneumatic circuit. One compressor is used for the positive air pressure supply, one vacuum pump is used for negative air pressure supply, two 2/2 solenoid valves are used to control the input to the gripper, whether it's the positive or negative pressure line. A pressure regulator is used to regulate the positive pressure line to the pressure required by the gripper.

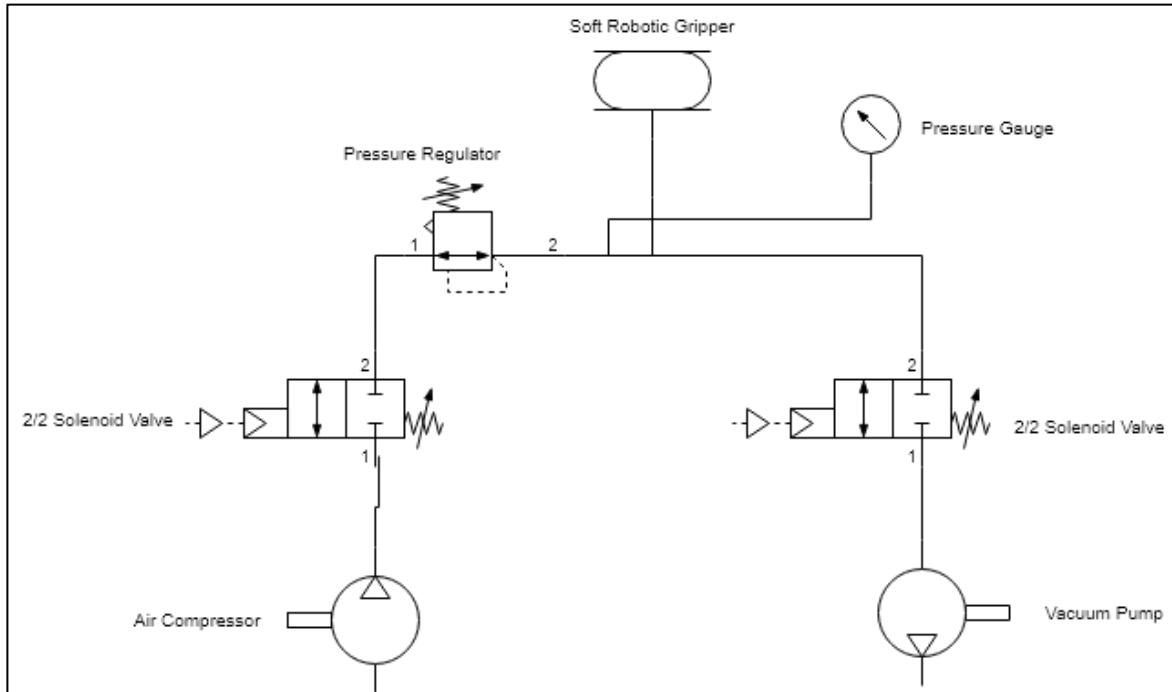


Figure 50 Pneumatic Control System Schematic

To achieve the functionality desired in the pneumatic circuit, positive and negative air supply is required. We used a TOPSFLO 12VDC Medical Air Pump. This pump could be used as a normal air pump (compressor) as well as a vacuum pump, therefore, two pumps of this type were used in the circuit. The negative pressure line were left as rated by the pump manufacturer, which is -70 KPa, and the positive pressure line is regulated by the pressure regulator, based upon the picked item required pressure. For the solenoid valves, two 24VDC  $\frac{1}{4}$  in. NC AirTac solenoid valves were used.

### 3.7.2.2 Electro-Pneumatic Control

To achieve complete control over the pneumatic unit based upon desired actions, i.e. when to inflate, deflate, or stay idle, the circuit is controlled by the main system microcontroller (Arduino board) and a 4 – channel relay module for the four components of the circuit. A schematic diagram for the circuit connection is illustrated in figure (51).

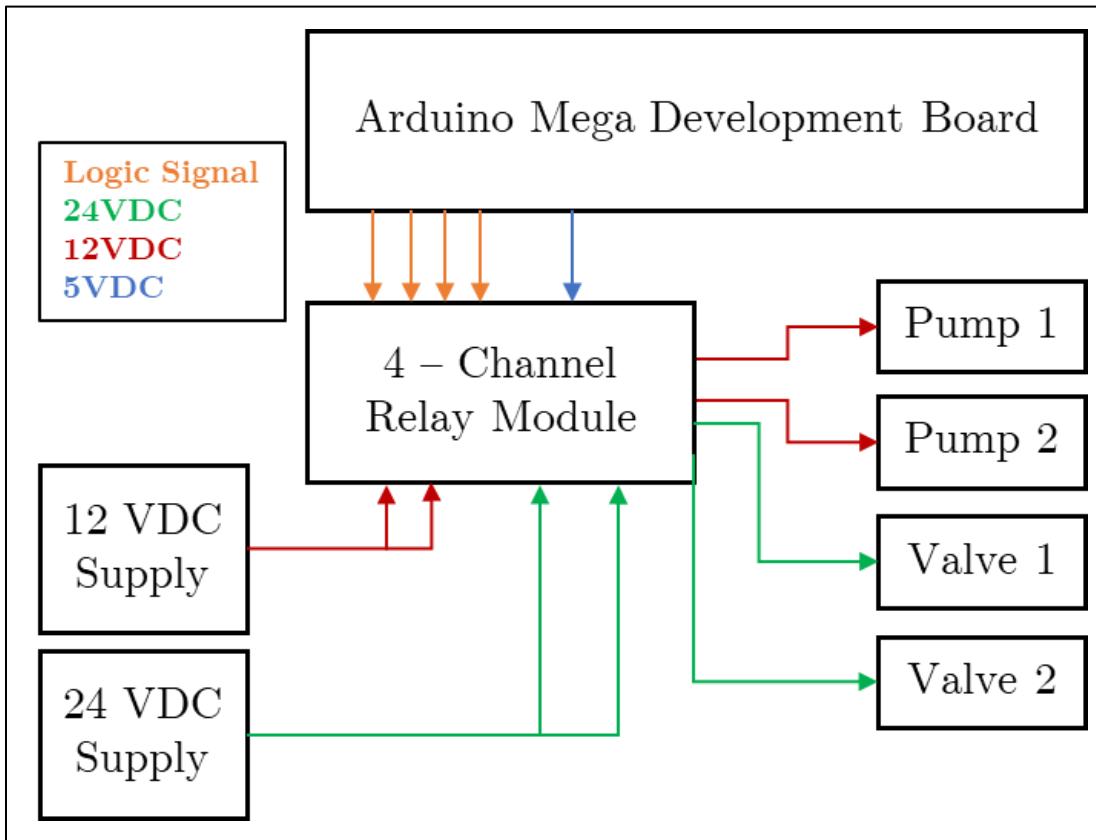


Figure 51 Electro-Pneumatic Circuit Schematic

### 3.8 Design & Simulation Validation

This section shows the validation of the simulation results of the soft robotic gripper.

The following figure (52) shows resultant similar behavior between the simulation and the real time gripper in case of vacuum or negative pressure is applied. Input to both is -70kPa

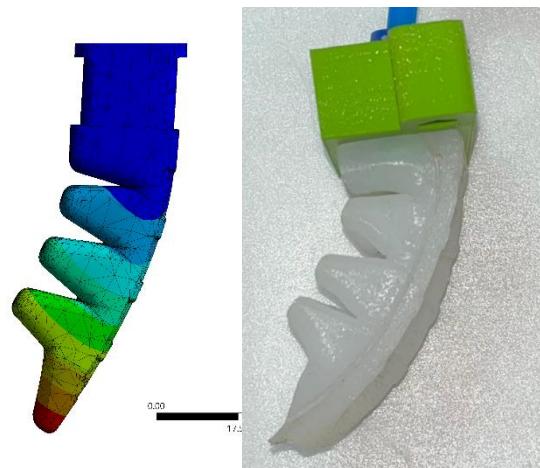


Figure 52 Validation Results under negative pressure supply

The following figure (53) shows resultant similar behavior between the simulation and real time gripper in case of positive pressure is supplied. Input to the simulated model is 50KPa. Input to the realtime gripper is 30KPa.

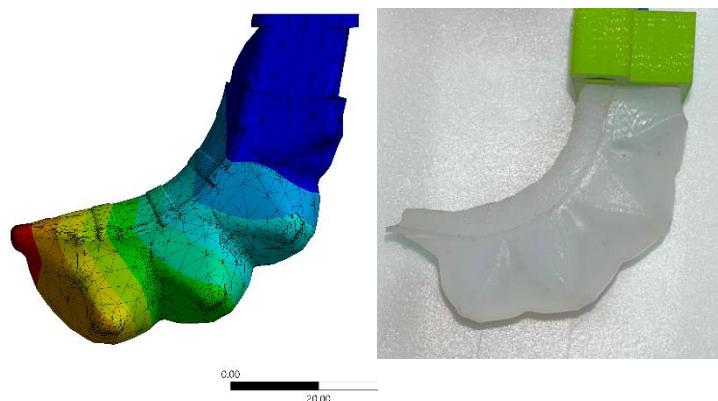


Figure 53 Gripper Validation Results under Positive Pressure Supply

## 3.9 Fabrication

The soft robotic gripper or the pneumatic soft fingers are fabricated by casting the silicone elastomer (EcoFlex 0030) into 3D printed molds. In this section, we present the design of the mold and how it was formulated. We document a step-by-step fabrication procedure as well for the soft gripper.

### 3.9.1 Mold Design

#### 3.9.1.1 Requirements

Apart from the main requirement, which is curing silicone into the desired shape of the soft gripper, there are other mold design requirements that were formulated:

1. Easy removal process to not damage the cured gripper.
2. Easy molding process, ensuring minimal waste of material.
3. Removal process should not damage the mold itself, so it could be reusable for multiple times.
4. Mold should incorporate a cavity to install the air connection fitting, without any extra space for air leakage.

#### 3.9.1.2 Design

The mold was prototyped on SolidWorks software with tooling split cavity tool around the final designed gripper in sec. 3.6. The mold is divided into 3 main parts that allow a 2 step molding process:

1. **Mold Upper Part:** this part is responsible for the first molding phase; it is responsible for making the rubber take the inner/outer geometry of the pneumatic chambers, its outer dimensions are less than inner dimensions of the Mold Base (Pt3.) with the value of the thickness of the required pneumatic chamber fig(54).

2. **Mold Lower Part:** this part is responsible for the second molding phase; it is responsible for making the bottom layer for the soft finger. This is performed on two steps since between the first and the second phase, the air connector is installed fig(55).
3. **Mold Base:** this is the common part between the two phases of the mold process, it is cavitated with dimensions that are higher than the first part (upper part) inner dimensions with the value of the desired thickness value. It is used for the second molding process as well to combine the first half part with the bottom layer of the pneumatic soft actuator.

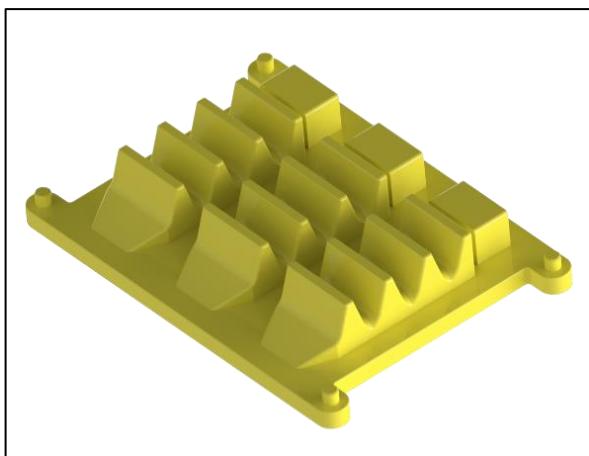


Figure 55 Mold Model: Upper Part 1

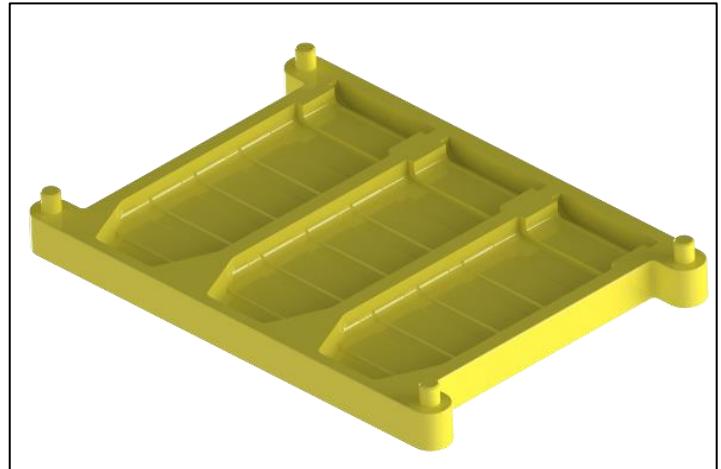


Figure 54 Mold Model: Lower Part 2

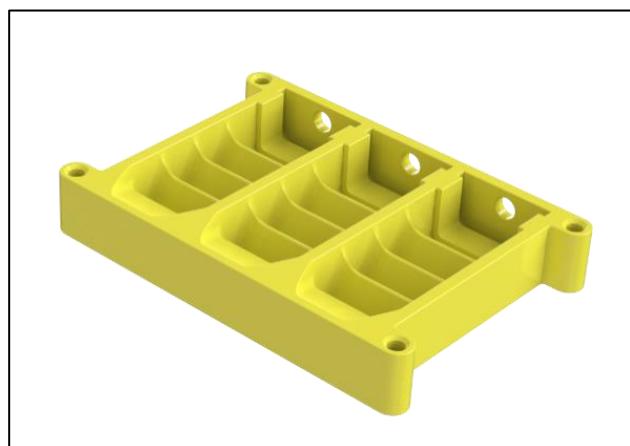


Figure 56 Mold Model: Base Part3

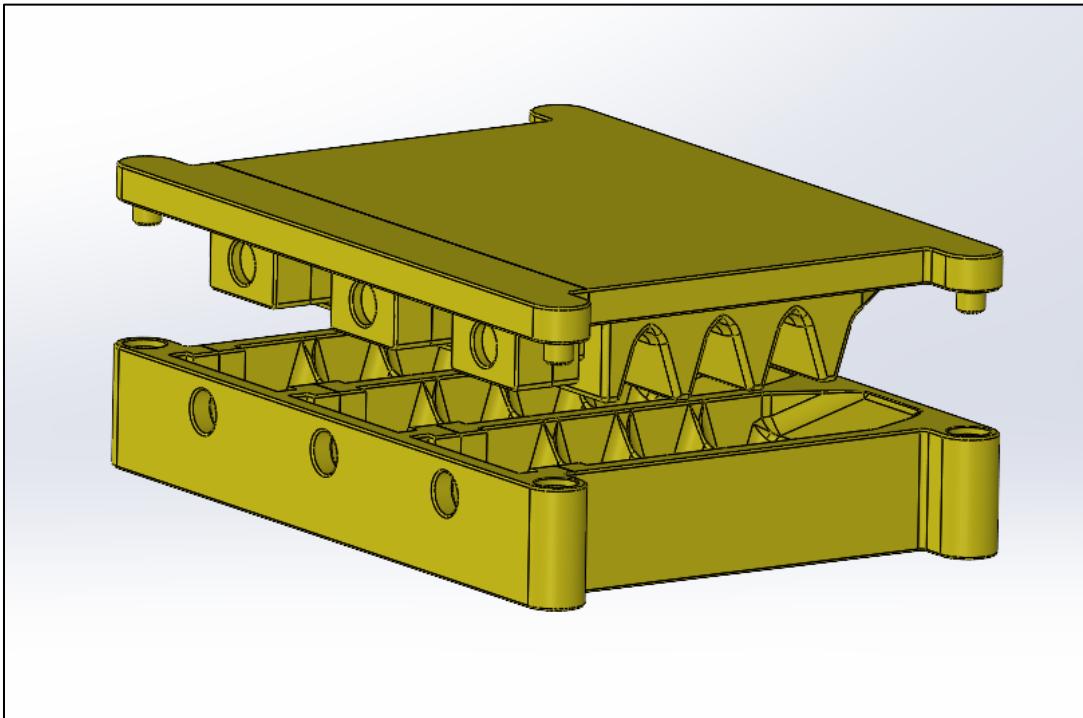


Figure 57 Molding Phase 1 Assembly Illustration

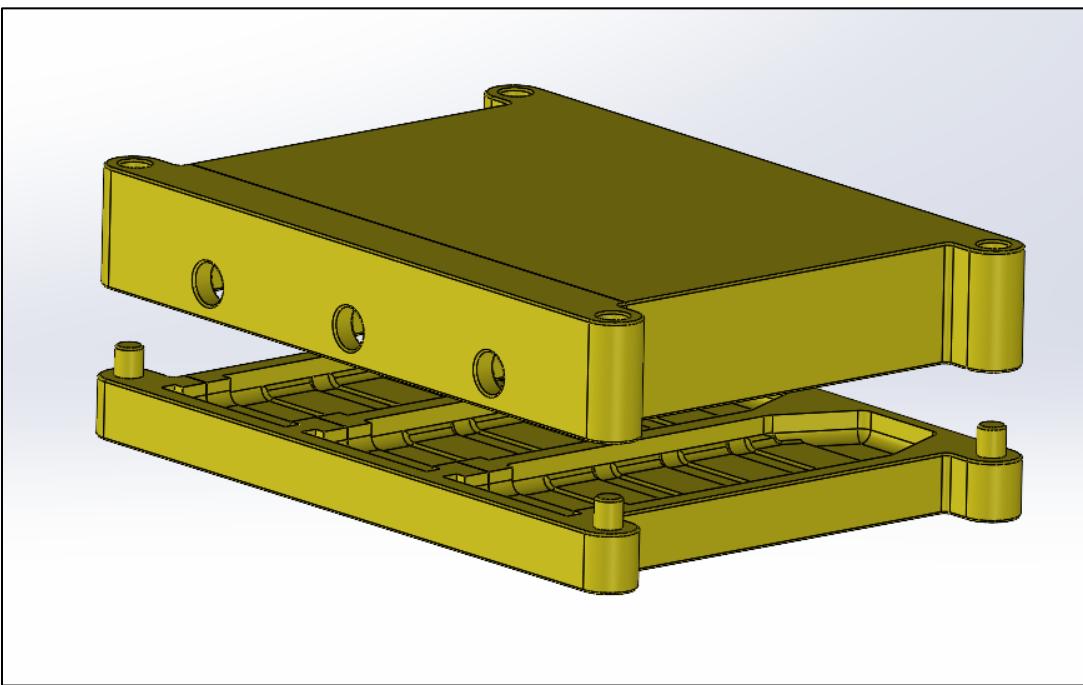


Figure 58 Figure 36 Molding Phase 2 Assembly Illustration



### **3.9.2 Connectors & Assembly**

In this section, we present how we designed the assembly accessories required to assemble the soft gripper and attach it to the robot's end effector.

#### **3.9.2.1 Air Connection Fitting Connector**

We designed a threaded 3D printed square block to be inserted inside the gripper to ensure the air connection fitter available in the market.

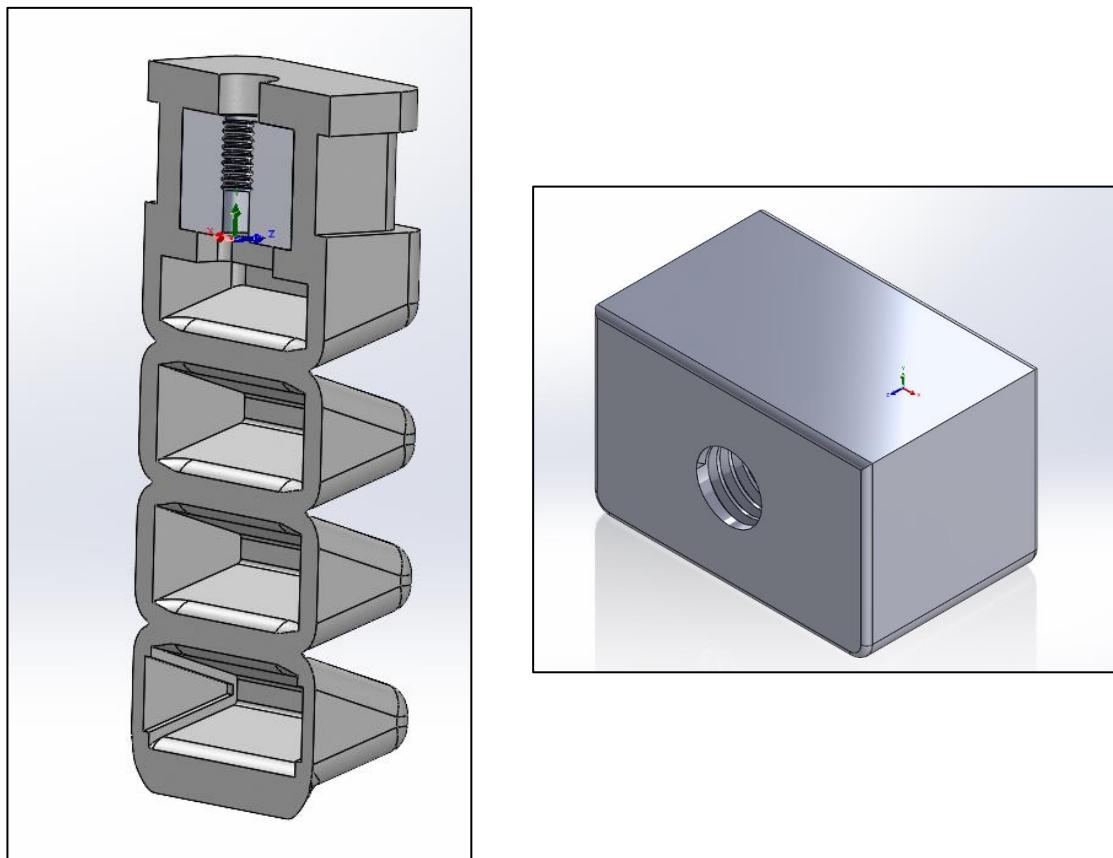


Figure 59 Soft Gripper Air Connector

### 3.9.2.2 End-Effector Assembly Jig

An 3D printed connector was designed to attach the soft fingers to the delta robot's end effector,

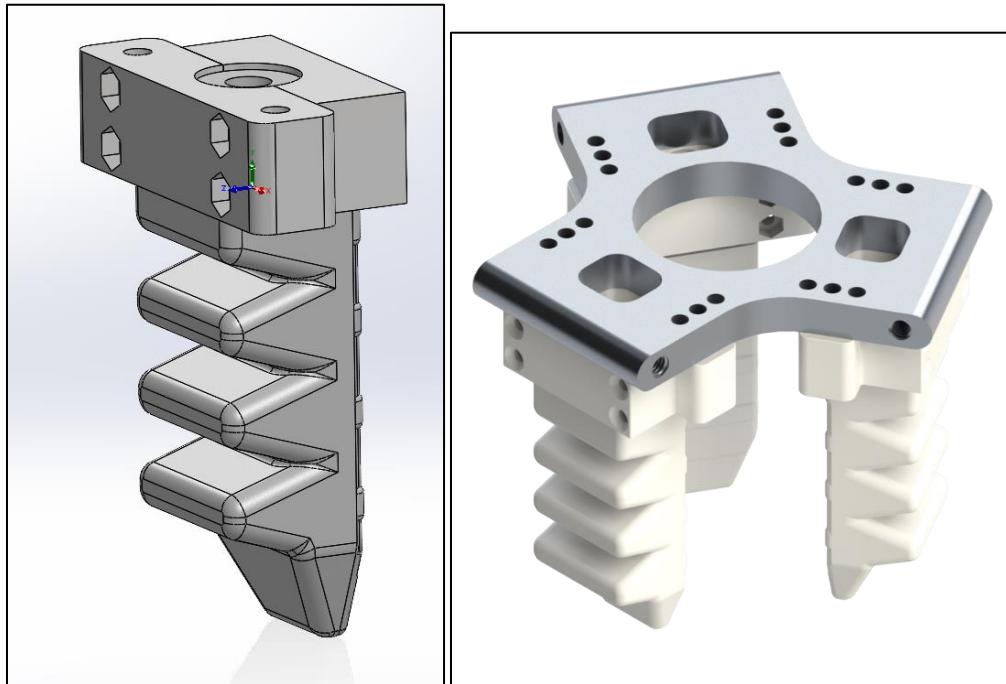


Figure 60 Gripper End Effector Assembly Jig

### **3.9.3 Step-by-Step Fabrication & Assembly Procedure**

This section documents the step-by-step fabrication procedure of the soft robotic gripper.

#### **1 – Preparing the silicone rubber**

In this step, the silicone rubber EcoFlex 0030 is prepared by adding equal amounts of each part A and B into a cup then mixing them thoroughly



#### **2 – Bubble removal**

In this step, the silicone mixture is inserted inside a custom made vacuum chamber for 2 or 3 minutes to remove any bubble residuals from the mixing procedure. The vacuum chamber is produced by connecting the vacuum pump to a sealed lunchbox



### **3 – Molding Phase 1**

In this step, the first molding phase (mentioned previously) is performed by pouring silicone rubber inside the base mold then fixing the upper mold to it.

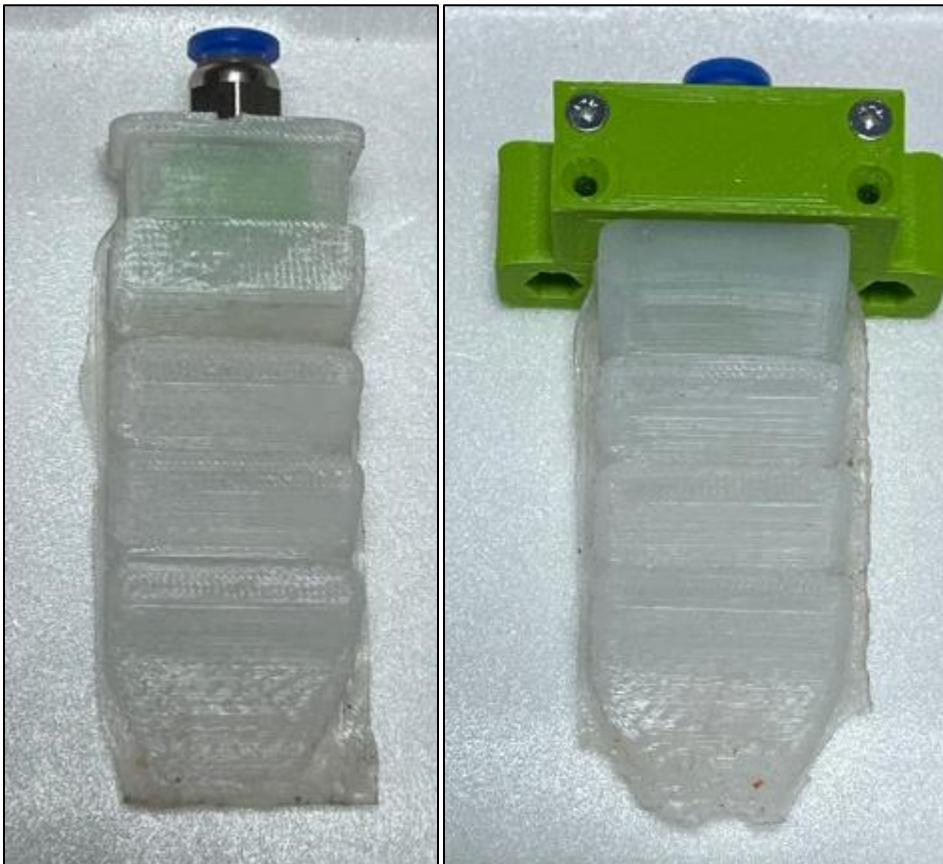


### **4 – Molding Phase 2**

After 4-5 hrs, the first mold is removed and the air connector is inserted to the cured part. Steps 1 – 3 are performed similarly with the bottom layer mold and its left for the same amount of time,



**5 – Connecting the pneumatic fittings and the end effector jig**



## **Chapter 4**

# **Computer Vision System**

### **4.1 Introduction to Computer Vision in Robotics**

#### **4.1.1 Overview**

Computer vision is a crucial component of modern robotics, enabling robots to interpret and understand visual information from the surrounding environment. This technology mimics the human visual system, allowing robots to perform complex tasks that require visual perception. In our delta robot project, computer vision is employed to detect and locate fruits within the robot's workspace, enabling precise pick-and-place operations. This introduction delves into the significance, principles, and applications of computer vision in robotics, with a focus on its implementation in our project.

#### **4.1.2 Significance of Computer Vision in Robotics**

Computer vision transforms robotic systems by providing the ability to:

- **Recognize Objects:** Identify and classify objects within the environment, a fundamental capability for many robotic applications.
- **Understand Context:** Interpret scenes and contexts, allowing robots to make informed decisions based on visual data.
- **Interact with Dynamic Environments:** Adapt to changes and interact with moving objects, enhancing the versatility and flexibility of robots.
- **Achieve Precision and Accuracy:** Enable precise measurements and control, crucial for tasks requiring high accuracy, such as manufacturing and assembly.

#### **4.1.3 Principles of Computer Vision**

Computer vision operates on several core principles and techniques, including:

1. **Image Acquisition:** Capturing images or video frames using cameras or other imaging devices. In our project, a Rapoo C260 webcam is used to provide real-time visual input.
2. **Preprocessing:** Enhancing and preparing images for analysis. This can include resizing, normalization, filtering, and noise reduction to improve the quality and usability of visual data.
3. **Feature Extraction:** Identifying and extracting relevant features from images. Techniques such as edge detection, corner detection, and texture analysis help in recognizing and describing objects.
4. **Object Detection:** Locating and identifying objects within an image. Advanced algorithms like YOLO (You Only Look Once) are employed to detect and classify multiple objects in real-time.
5. **Segmentation:** Dividing an image into meaningful regions or segments. This helps in isolating objects from the background for further analysis.
6. **Recognition and Classification:** Assigning labels to detected objects based on their features. This step involves comparing extracted features with known patterns to identify the object.
7. **Localization:** Determining the position and orientation of objects in the image. This involves calculating the coordinates and dimensions of bounding boxes around detected objects.

#### **4.1.4 Applications of Computer Vision in Robotics**

Computer vision has a wide range of applications in robotics, including but not limited to:

- **Automated Inspection:** Used in manufacturing for quality control and defect detection.

- **Autonomous Navigation:** Enables robots and vehicles to navigate through environments without human intervention.
- **Human-Robot Interaction:** Facilitates communication and collaboration between humans and robots by recognizing gestures, faces, and movements.
- **Pick-and-Place Operations:** Used in logistics and warehousing for sorting and handling items.
- **Agriculture:** Employed in tasks such as fruit picking, crop monitoring, and pest detection.
- **Healthcare:** Assists in medical imaging, surgery, and patient monitoring.

## **4.2 Computer Vision System Purpose & Objectives**

### **4.2.1 Purpose**

The primary purpose of the computer vision system in our delta robot project is to enable the robot to autonomously identify and locate various fruits within its working area. This capability is crucial for automating the fruit handling process, allowing the robot to perform tasks such as picking and placing fruits with precision and efficiency. By leveraging computer vision, we enhance the robot's ability to interact with its environment in a dynamic and adaptable manner.

### **4.2.2 Objectives**

The specific objectives of the computer vision system are as follows:

1.     Object Detection
  - Accurate Identification.
  - Real-Time Processing.
2.     Coordinate Localization
  - Pixel Coordinates Determination.
  - Real-World Conversion.
3.     Communication with Robotic Control System
  - Data Transmission.
  - Timely and Reliable Communication.

#### **4.2.2.1 Object Detection**

Object detection is the foundational capability of the computer vision system. It involves:

- Model Selection.
- Model Training and Configuration.
- Detection Process.

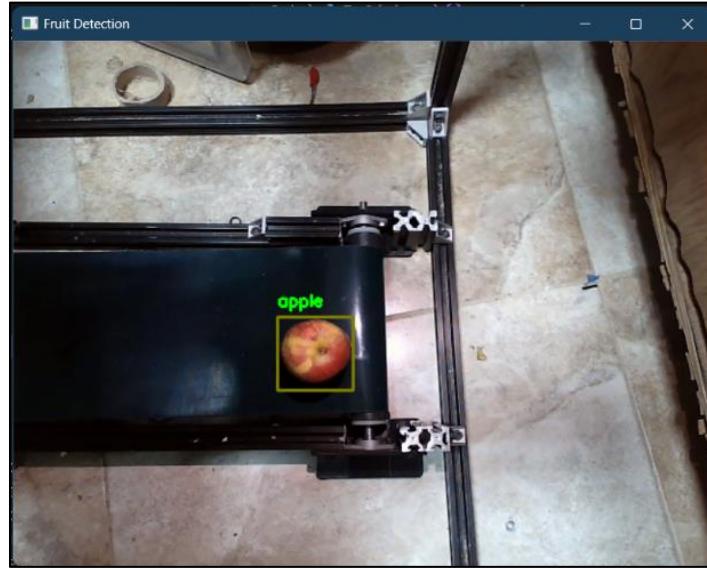


Figure 61 Fruit Detection using Computer Vision

#### 4.2.2.2 Coordinate Localization

After detecting the fruits, the next step is to determine their coordinates:

- Bounding Box Calculation.
- Center Coordinates.
- Scale Factor Application.

#### 4.2.2.3 Communication with the Robot Control System

Effective communication between the computer vision system and the robotic control system is essential for coordinated operation:

- Data Formatting: Formatting the detected fruit type and coordinates into a string that can be easily transmitted and interpreted by the Arduino. The format typically includes the first letter of the fruit's name, followed by the x and y coordinates.
- Serial Communication: Establishing a serial communication link between the computer vision system and the Arduino controller. The data is transmitted over this link to guide the robot's actions.

## 4.3 Used Hardware

### 4.3.1 Rapoo C260 Webcam



Figure 62 Rapoo C260 Webcam

The primary hardware component utilized in our project for visual input is the Rapoo C260 webcam. This webcam is chosen for its high-resolution imaging capabilities and reliability, making it suitable for real-time object detection tasks. Here's an overview of the Rapoo C260 webcam:

- Resolution: The webcam boasts a high resolution, providing clear and detailed images necessary for accurate object detection.
- Frame Rate: It supports a high frame rate, ensuring smooth and fluid video streaming, which is essential for real-time processing.
- Mounting Flexibility: The webcam is designed to be easily mountable, allowing for flexible placement and adjustment to capture the desired field of view.
- Compatibility: It is compatible with a wide range of operating systems, making it easy to integrate into our project environment.
- Reliability: The Rapoo C260 webcam is known for its reliability and durability, ensuring consistent performance during prolonged use.

#### **4.3.2 Purpose & Role**

The Rapoo C260 webcam serves as the primary visual input device for the computer vision system in our delta robot project. Its role is crucial in capturing live video frames of the robot's workspace, which are then processed by the object detection algorithm (YOLO) to detect and localize fruits.

#### **4.3.3 Placement & Configuration**

The webcam is strategically positioned to provide a top-down view of the delta robot's working area, ensuring comprehensive coverage of the space where fruits are expected to be manipulated. It is securely mounted at an optimal height and angle to capture clear and unobstructed images of the fruits and surrounding environment.

#### **4.3.4 Compatibility**

The Rapoo C260 webcam seamlessly integrates with the computer vision system implemented using OpenCV and the YOLO object detection algorithm. It serves as the primary source of visual input, delivering real-time video frames to the system for processing. The high-resolution images captured by the webcam enable the YOLO algorithm to accurately detect and localize fruits within the robot's workspace.

## 4.4 Software and Libraries

### 4.4.1 OpenCV (Open-Source Computer Vision Library)

OpenCV is a powerful open-source computer vision and machine learning software library. It provides a comprehensive set of tools and functions for image processing, computer vision, and machine learning tasks. In our delta robot project, OpenCV plays a central role in implementing various computer vision algorithms and techniques.

#### Features of OpenCV:

1. Image Acquisition: OpenCV allows for easy acquisition of images from cameras, videos, and files, making it suitable for real-time applications like ours.
2. Image Processing: It offers a wide range of image processing functions, including filtering, edge detection, morphological operations, and color space conversion. These functions are essential for preprocessing images before object detection.
3. Object Detection and Recognition: OpenCV provides support for object detection algorithms such as Haar cascades, HOG (Histogram of Oriented Gradients), and deep learning-based methods like YOLO (You Only Look Once). We utilize OpenCV's capabilities for object detection in our project.
4. Feature Extraction and Matching: OpenCV includes algorithms for feature extraction and matching, which are useful for tasks such as image registration and object tracking.
5. Integration with Python: OpenCV has extensive Python bindings, making it easy to use in Python-based projects like ours. Python's simplicity and versatility make it an ideal choice for rapid prototyping and development.

### 4.4.2 YOLO (You Only Look Once)

YOLO is a state-of-the-art real-time object detection algorithm known for its speed and accuracy. It frames object detection as a single regression problem, directly predicting bounding boxes and class probabilities from image pixels. YOLO is employed in our project for its ability to detect multiple objects in real-time, making it suitable for applications where speed is crucial.

#### **Features of YOLO:**

1. Real-Time Detection: YOLO can detect objects in real-time with impressive speed, making it suitable for applications requiring fast processing.
2. Single-Pass Inference: Unlike traditional object detection methods that use multiple stages and classifiers, YOLO performs object detection in a single pass through the neural network, resulting in faster inference times.
3. High Accuracy: Despite its speed, YOLO achieves high accuracy in object detection, making it suitable for a wide range of applications, including ours.
4. Multiple Object Detection: YOLO can detect multiple objects within an image, providing bounding box coordinates and class probabilities for each detected object. This capability is crucial for our project, where we need to identify and localize multiple fruits simultaneously.

#### **4.4.3 Integration with OpenCV**

In our project, we leverage OpenCV's capabilities for image processing and YOLO for object detection. OpenCV provides functions for image acquisition, preprocessing, and visualization, while YOLO handles the actual object detection task. The integration of these two libraries allows us to develop a robust and efficient computer vision system for our delta robot project.

## 4.5 YOLO Model and COCO Dataset

### 4.5.1 YOLO (You Only Look Once)

YOLO is a state-of-the-art object detection algorithm known for its speed and accuracy. Developed by Joseph Redmon et al., YOLO approaches object detection as a single regression problem, directly predicting bounding boxes and class probabilities from image pixels. The YOLO algorithm divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell, enabling it to detect multiple objects in real-time with impressive speed.

### 4.5.2 Key Features of YOLO

1. Real-Time Detection: YOLO can detect objects in real-time with remarkable speed, making it suitable for applications requiring fast processing, such as ours.
2. Single-Pass Inference: Unlike traditional object detection methods that use multiple stages and classifiers, YOLO performs object detection in a single pass through the neural network, resulting in faster inference times.
3. High Accuracy: Despite its speed, YOLO achieves high accuracy in object detection, making it suitable for a wide range of applications.
4. Multiple Object Detection: YOLO can detect multiple objects within an image simultaneously, providing bounding box coordinates and class probabilities for each detected object.

### 4.5.3 COCO (Common Objects in Context) Dataset

The COCO dataset is a large-scale object detection, segmentation, and captioning dataset. It is widely used for training and benchmarking object detection algorithms like YOLO. The

COCO dataset contains images with complex scenes, including a diverse set of object categories, making it suitable for training robust object detection models.

#### **4.5.4 Key Features of COCO Dataset**

1. Large-Scale: The COCO dataset consists of tens of thousands of images, each annotated with multiple object instances across various categories.
2. Diverse Object Categories: It includes a wide range of object categories, covering common objects found in everyday scenes, such as people, animals, food, vehicles, and household items.
3. Rich Annotations: Each image in the COCO dataset is annotated with bounding box coordinates and class labels for all object instances present in the scene. Additionally, it provides segmentation masks for pixel-level object delineation.
4. Standard Benchmark: The COCO dataset serves as a standard benchmark for evaluating and comparing the performance of object detection algorithms. Its rich annotations and diverse scenes make it a challenging dataset for training and testing object detection models.

#### **4.5.5 Integration with YOLO**

In our project, we leverage the YOLO algorithm, pre-trained on the COCO dataset, for object detection. By using a pre-trained YOLO model trained on the COCO dataset, we benefit from the robustness and generalization capabilities learned from the diverse set of objects and scenes present in the dataset. This allows our delta robot to accurately detect and localize fruits within its workspace, facilitating precise pick-and-place operations.

## 4.6 Coordinate Localization and Conversion

### 4.6.1 Coordinate Localization:

Coordinate localization refers to the process of determining the precise position of detected objects within an image or video frame. In our delta robot project, coordinate localization is essential for accurately identifying the location of fruits within the robot's workspace. Here's an overview of the coordinate localization process:

#### 1. Bounding Box Calculation:

- After detecting objects using the YOLO algorithm, bounding boxes are generated around each detected object. These bounding boxes define the spatial extent of the objects within the image frame.

#### 2. Center Coordinates Extraction:

- From each bounding box, the center coordinates (x, y) are extracted. These center coordinates represent the midpoint of the bounding box and serve as the reference point for the object's position within the image.

```
Apple detected at: center-based x=-43.10 mm, y=45.06 mm, width=66.61 mm, height=64.66 mm
Sending to Arduino: A,-43.10,45.06
Apple detected at: center-based x=-43.10 mm, y=45.06 mm, width=68.57 mm, height=66.61 mm
Sending to Arduino: A,-43.10,45.06
Apple detected at: center-based x=-43.59 mm, y=45.06 mm, width=67.59 mm, height=66.61 mm
Sending to Arduino: A,-43.59,45.06
Apple detected at: center-based x=-43.59 mm, y=44.57 mm, width=67.59 mm, height=65.63 mm
Sending to Arduino: A,-43.59,44.57
Apple detected at: center-based x=-43.59 mm, y=44.57 mm, width=67.59 mm, height=65.63 mm
Sending to Arduino: A,-43.59,44.57
□
```

Figure 63 Results & Outputs

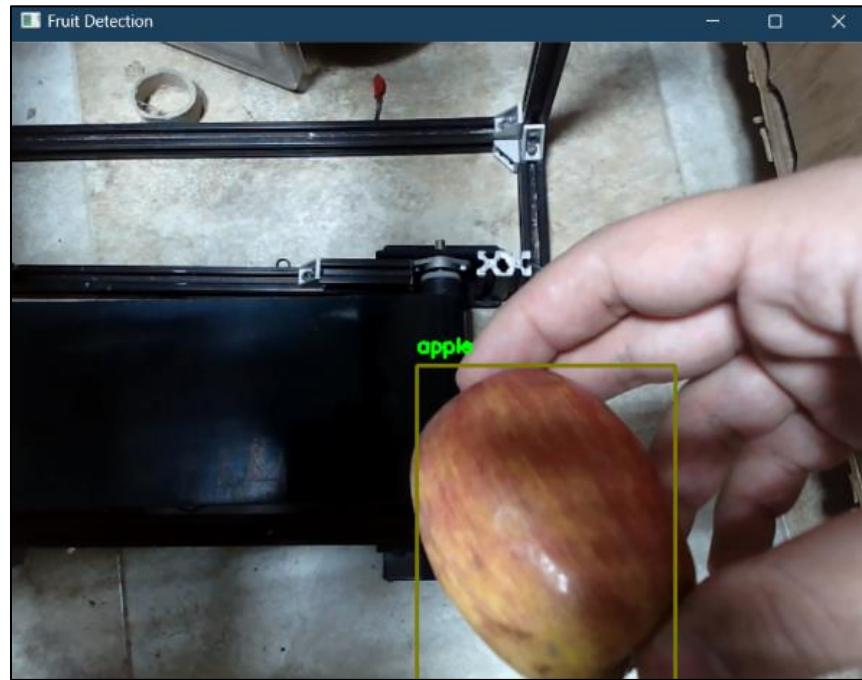


Figure 64 Bounding Box

#### 4.6.2 Coordinate Conversion

Coordinate conversion involves transforming the pixel coordinates obtained from the image frame into real-world coordinates, typically measured in millimeters. This conversion is crucial for enabling the robot to accurately position its gripper relative to the detected objects. Here's how coordinate conversion is achieved in our project:

##### 1. Scale Factor Calculation:

- A scale factor is calculated to convert pixel measurements to real-world millimeters. This involves capturing an image of a ruler with a known length and measuring its pixel length. The scale factor is then computed as the ratio of the real length to the pixel length.

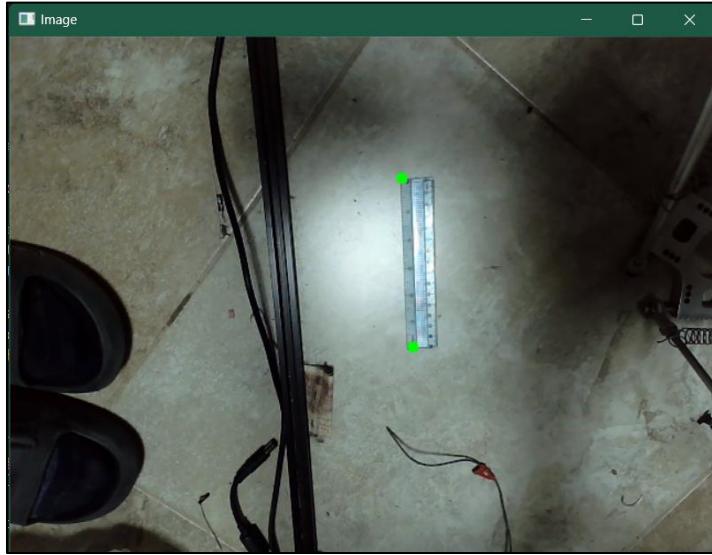


Figure 65 Getting ruler length in pixels

```
Distance between points in pixels: 153.33 pixels  
Scale factor: 0.9783 mm per pixel
```

Figure 66 Output of the scale factor code

## 2. Conversion Formula:

- Once the scale factor is determined, pixel coordinates ( $x_{pixel}$ ,  $y_{pixel}$ ) are converted to real-world coordinates ( $x_{mm}$ ,  $y_{mm}$ ) using the following formula:  $x_{mm} = x_{pixel} \times scale\ factor$  ,  $y_{mm} = y_{pixel} \times scale\ factor$

## 3. Applying Scale Factor:

- The scale factor is applied to each pixel coordinate ( $x_{pixel}$ ,  $y_{pixel}$ ) to obtain the corresponding real-world coordinates ( $x_{mm}$ ,  $y_{mm}$ ) in millimeters..

### 4.6.3 Benefits and Importance:

Coordinate localization and conversion are crucial components of our delta robot project, as they enable the robot to accurately perceive and interact with its environment. By localizing objects within the image frame and converting their coordinates to real-world measurements,

the robot can precisely position its gripper for pick-and-place operations, ensuring efficient and reliable fruit handling.

## 4.7 Communication with Arduino

In our delta robot project, communication with the Arduino controller is essential for coordinating the actions of the robot based on the information provided by the computer vision system. Here's an in-depth look at how communication with the Arduino is established and utilized:

### 4.7.1 Serial Communication Protocol

Serial communication is employed as the communication protocol between the computer vision system and the Arduino controller. Serial communication allows for the transmission of data between devices over a serial interface, typically using UART (Universal Asynchronous Receiver-Transmitter) communication.

#### Establishing Serial Connection:

##### 1. Port Selection:

- The computer vision system identifies and selects the appropriate serial port connected to the Arduino controller. This is typically done programmatically, with the system scanning available ports and allowing the user to select the desired port.

##### 2. Baud Rate Configuration:

- The baud rate for serial communication is configured to ensure consistent and reliable data transmission between the computer vision system and the Arduino. A common baud rate, such as 9600 baud, is often used for standard serial communication.

##### 3. Opening Serial Port:

- Once the port and baud rate are configured, the computer vision system opens the serial port to establish communication with the Arduino controller. This allows for bi-directional data exchange between the two devices.

#### **4.7.2 Data Transmission to Arduino**

##### **1. Data Formatting:**

- The data obtained from the computer vision system, such as the detected fruit type and its coordinates, is formatted into a string or byte array suitable for transmission over the serial connection.
- For example, the data may be formatted as follows:  
[fruit\_type,x\_coordinate,y\_coordinate]

##### **2. Data Encoding:**

- The formatted data is encoded into bytes, typically using UTF-8 or ASCII encoding, to ensure compatibility and proper transmission over the serial connection.

##### **3. Data Transmission:**

- The encoded data is sent to the Arduino controller via the open serial port. The Arduino receives and processes the incoming data, extracting relevant information to guide its actions.

#### **4.7.3 Utilizing Data in Arduino**

##### **1. Data Reception:**

- The Arduino controller continuously listens for incoming data on the serial port. Once data is received, it is read and decoded to extract the fruit type and its coordinates.

##### **2. Action Planning:**

- Based on the received data, the Arduino plans and executes the necessary actions, such as moving the robot's gripper to the specified coordinates for fruit handling.

#### Coordinate Localization Precision:

- The coordinate localization precision refers to the accuracy of determining the real-world coordinates of detected fruits within the image frame.
- By applying the scale factor obtained from the ruler calibration, we accurately converted pixel coordinates to millimeters, enabling precise localization of fruits within the robot's workspace.

#### Communication Reliability:

- The communication between the computer vision system and the Arduino controller was found to be reliable and robust. Serial communication ensured timely and accurate transmission of fruit type and coordinates to the robot.

## 4.8 Algorithm Description (Pseudocode)

1. Initialize webcam for video capture
  2. Load pre-trained YOLO model and COCO dataset
  3. Configure serial communication with Arduino
  4. Calibrate ruler to obtain scale factor for coordinate conversion
- // Main Loop //
5. while True:
    6. Capture frame from webcam
    7. Preprocess frame for object detection
    8. Detect objects using YOLO algorithm
    9. For each detected object:
      10. Extract bounding box coordinates
      11. Calculate center coordinates of the bounding box
      12. Convert pixel coordinates to real-world coordinates using scale factor
      13. Format fruit type and coordinates into a string
      14. Transmit data to Arduino via serial port
      15. Display detected objects and coordinates on the frame
      16. If 'q' is pressed, exit the loop
    17. Release webcam and close serial communication

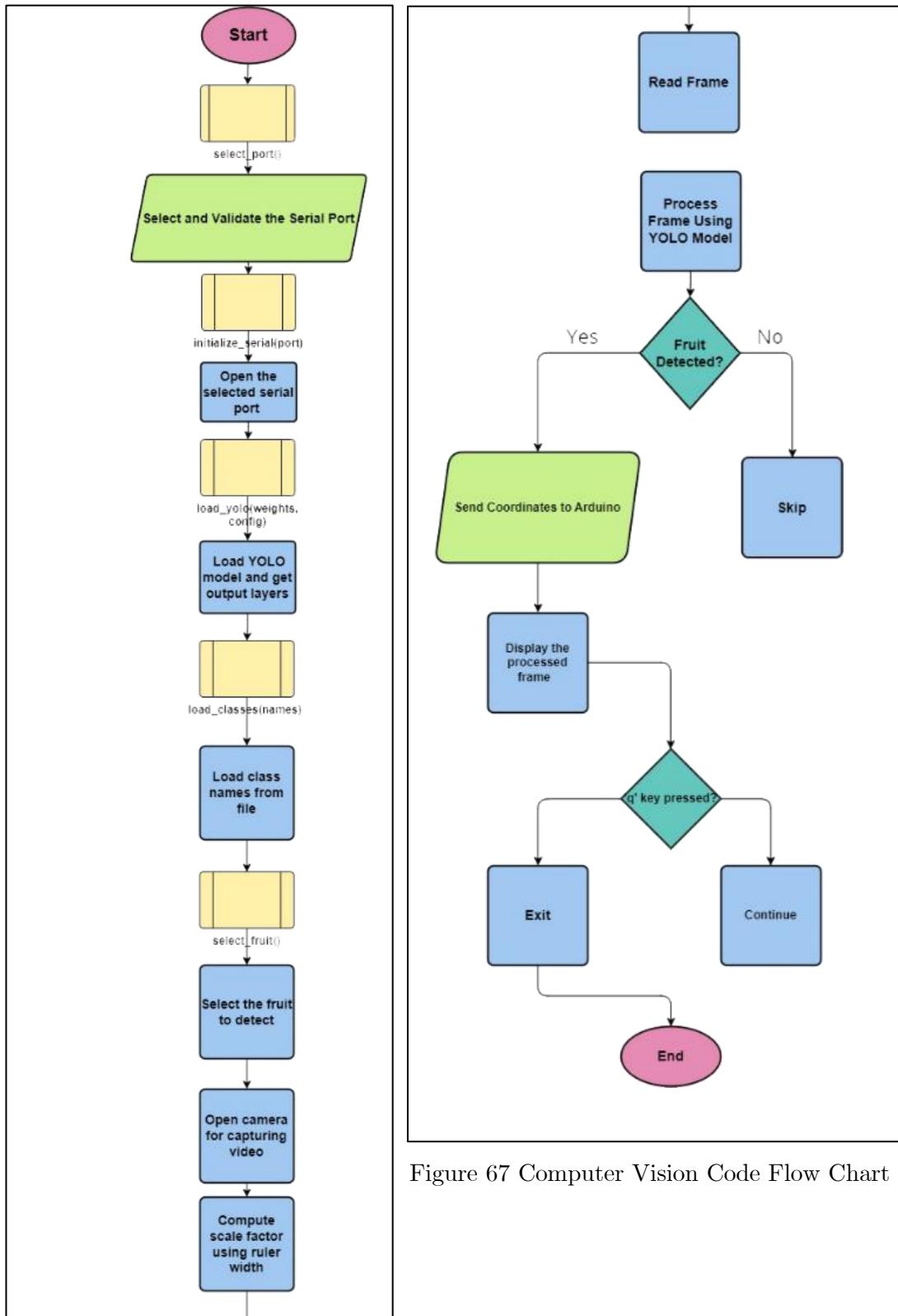


Figure 67 Computer Vision Code Flow Chart

## 4.9 Conclusion

Our delta robot project represents a successful integration of computer vision and robotics technologies to enable autonomous fruit handling. Throughout the development and implementation process, several key components and methodologies were employed, culminating in a functional system capable of detecting fruits, localizing their coordinates, and coordinating actions with the Arduino controller for precise fruit handling.

### 4.9.1 Performance and Results

- The system demonstrated high accuracy, precision, and reliability in fruit detection, localization, and robotic control tasks.
- Real-time performance was achieved, ensuring timely and responsive interaction with the environment.
- Extensive testing validated the robustness and effectiveness of the integrated vision-guided robotics approach, showcasing its potential for practical applications.

In conclusion, our delta robot project represents a significant milestone in the field of integrated vision-guided robotics, demonstrating the potential for autonomous fruit handling and showcasing the power of interdisciplinary collaboration between computer vision and robotics domains. By harnessing the capabilities of both technologies, we pave the way for innovative solutions in automation, agriculture, and beyond.

## **Chapter 5**

# **System Integration**

### **5.1 Embedded System Design**

This section discusses the embedded system design of our project. The embedded system in this project is designed to control and perform several functions, while coordinating between our hardware components.

#### **5.1.1 Functional Requirements**

This subsection discusses the functions our microcontroller must be able to perform, to ensure smooth operation and demonstration of the delta robot and the soft gripper.

##### **5.1.1.1 Delta robot control**

- Motion Control: To precisely place the end effector within its workspace, the system must regulate the movement of the Delta robot's arms. In order to ascertain the required joint angles for a specific end effector position we utilize inverse kinematics.
- Trajectory planning: To ensure smooth movement along a desired path with velocity profile control to ensure jerk free movement for the motors.

##### **5.1.1.2 Soft gripper control**

Gripper Actuation: Controlling the actuation of the soft gripper using a pneumatic control system to open and close the gripper at adequate time with adequate force.

##### **5.1.1.3 Communication with laptop GUI**

- Command Reception: Receive high-level commands from the laptop GUI, such as “start process,” “demonstrate,” and other operational commands.
- Coordination with Camera: Process coordinates sent by the camera to identify the position of objects to be picked, ensuring accurate pick-and-place operations

### **5.1.2 Performance Requirements**

This section states the performance requirements for our embedded systems design, which are crucial for the overall success of the project

#### **5.1.2.1 Response time**

- The system should have minimal response time from receiving a command to initiating its action to ensure real-time control. That's why delays in the code flow are completely forbidden and we use timers instead.

#### **5.1.2.2 Accuracy**

- Positional accuracy for the Delta robot's end effector should be within  $\pm 1$  millimeter. This level of precision is necessary for tasks that require high accuracy, such as picking small or precisely located objects.

#### **5.1.2.3 Reliability**

- The system should operate continuously for extended periods without failure. Time between failure should be high enough or non-existent to ensure proper demonstration.
- To minimize downtime, the system should have fault detection and recovery methods to handle and recover from mistakes autonomously crucial for the overall success of the project.

#### **5.1.2.4 Microcontroller**

We used an “Arduino Mega” as the microcontroller to fetch and execute commands sent by the GUI. The Arduino also acts as a communication medium between our computer vision and the delta robot.

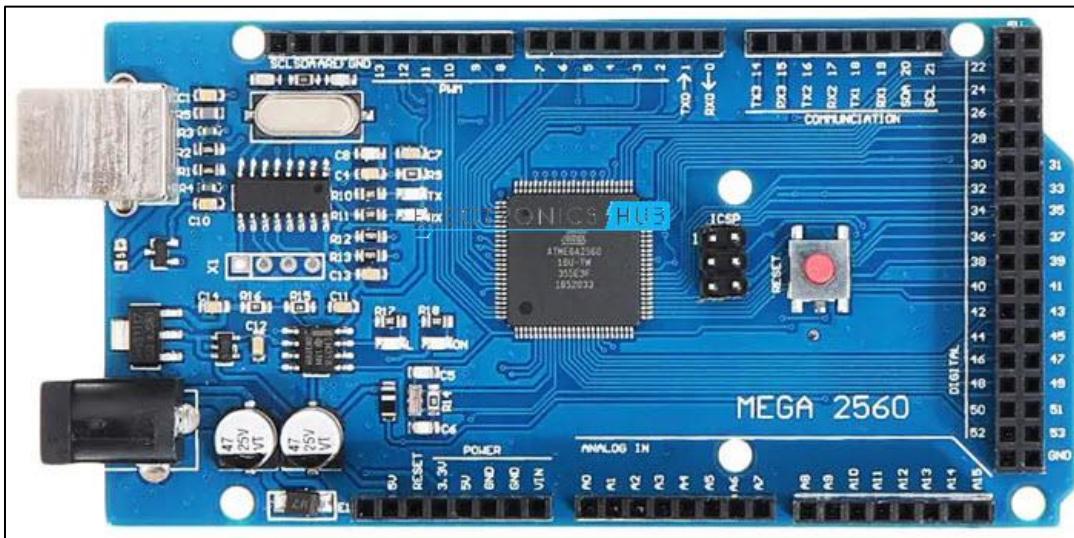


Figure 68 Arduino MEGA Development Board

We also used a laptop to handle the GUI which will be discussed below in this chapter and processing the computer vision algorithms. Computer vision algorithms are discussed in chapter 4 of this report.

### 5.1.3 Hardware Interface

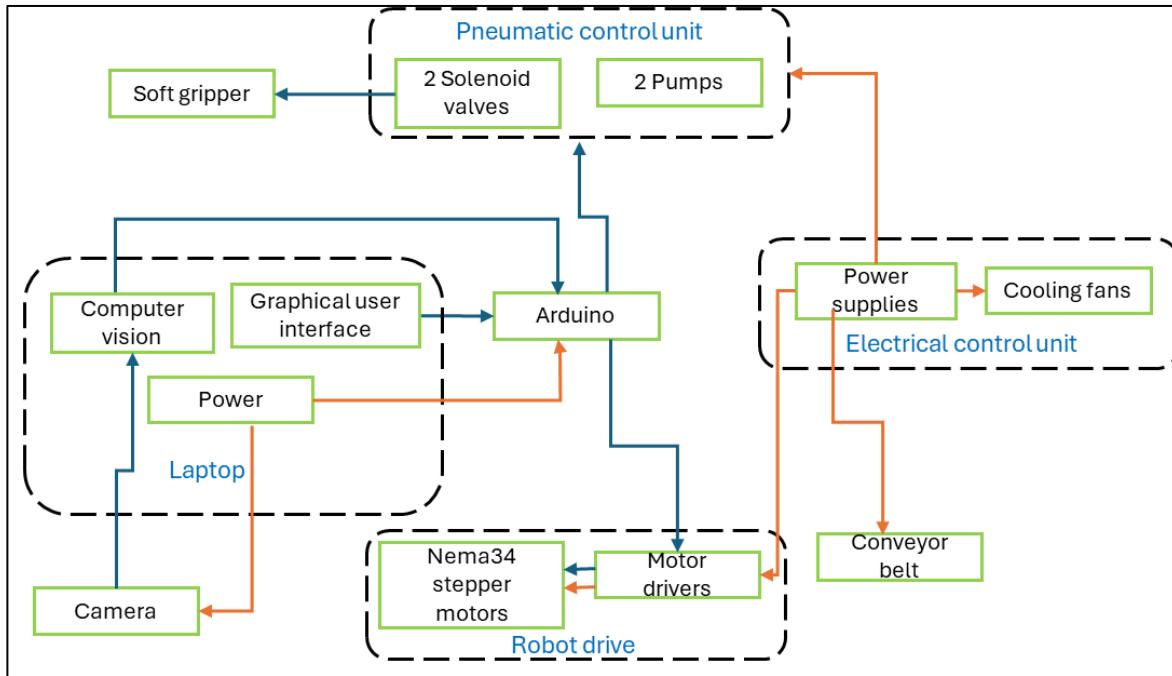


Figure 69 Embedded Hardware Architecture

The above block diagram shows the hierarchy and the hardware interfacing between project components. The Arduino is the prime controller powered by the laptop, it controls the robot drive, electrical and pneumatic control units. The Arduino takes strings from the laptop's computer vision and GUI. The nature of these strings is to be discussed later in the report.

## **5.1.4 Embedded Software**

### **5.1.4.1 Initialization & Calibration**

The initialization and calibration of the robot ensures that our system is completely intact and ready to go, it ensures that the delta robot and soft gripper are in a known state and position and ready for error-free operation. The following are key steps:

#### **1. System power on**

The Arduino, Delta robot, and sensors are among the connected components that must initially be initialized when the system is powered on.

#### **2. Robot homing**

The robot Delta travels to its designated "home" location. This is an established reference point that guarantees the accuracy of any moves made after it. When the robot is turned on while the end effector is on the ground, the three arms are lifted simultaneously by the three motors until the first limit switch is toggled. The arm corresponding to the toggled limit switch is lowered a known number of degrees corresponding to the known place of the limit switch. The same is carried for the two other arms.

#### **3. Camera calibration**

Calibrating the camera to correct any distortion and to ensure accurate object detection and position.

#### **4. Gripper initialization**

This soft gripper is set to a default state where it is deflated. This ensure the robot is always ready to pick up

#### **5. System ready**

System is initiated, calibrated and ready to execute commands

#### **5.1.4.2        Command Execution**

After the initialization and calibration of the system is completed the robot and conveyor belt stays in idle position until a command is sent from the user through GUI

#### **5.1.4.3        Operation Modes**

To guarantee the best way possible to demonstrate our work in a way that utilizes all our hardware capabilities, we created modes of operation.

The mode of operation is to be selected with the graphical user interface once the power is supplied to the robot. The modes of operation are summarized as follows:

- 1)     **Idle:** all operations are on Idle waiting for GUI command
- 2)     **Demonstration Mode:** Robot is to move in sequence of predefined positions
  - Corners: Moves the robot across the four corner boundaries in its workspace
  - Pick-place: Moves the robot in a predefined positions simulating a pick and place operation
- 3)     **Computer Vision Mode:** Robot synchronizes with computer vision's input coming from the camera. The robot moves to the pick position on the conveyor belt, picks the object by inflating the gripper, moves to the designated placing position then, place it by deflating the gripper. The process keeps on repeating until the stop command is sent through the GUI. Flowchart figure(62).

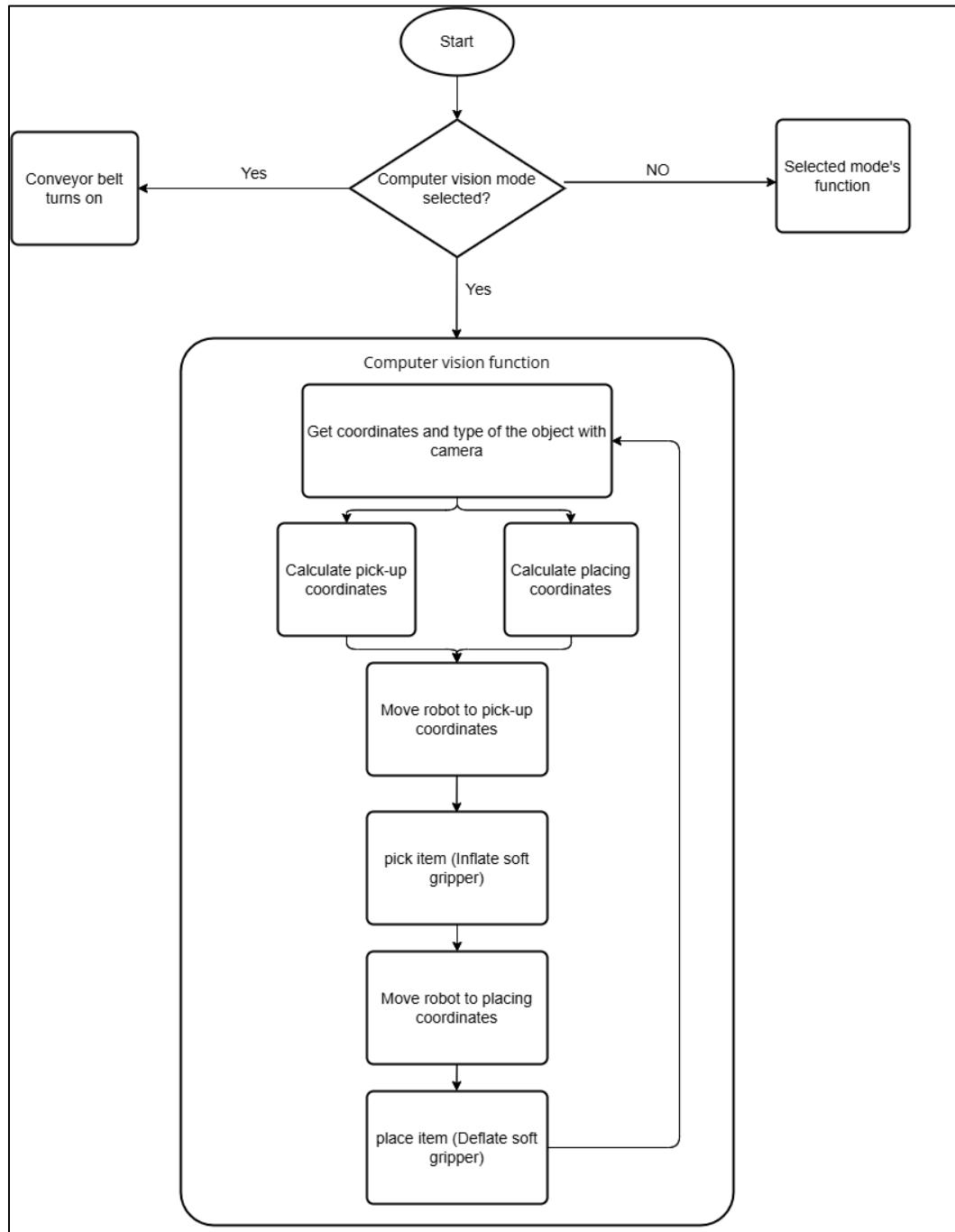


Figure 70 Computer Vision Controlled Mode Flowchart

## 5.2 Electrical Integration

The following diagram illustrates the connections between the components and the Arduino and how the signal is sent from one component to another. Most of the components are located at one enclosure “Control Box” to prevent any entanglement during the operation and to ease trouble shooting and maintenance.

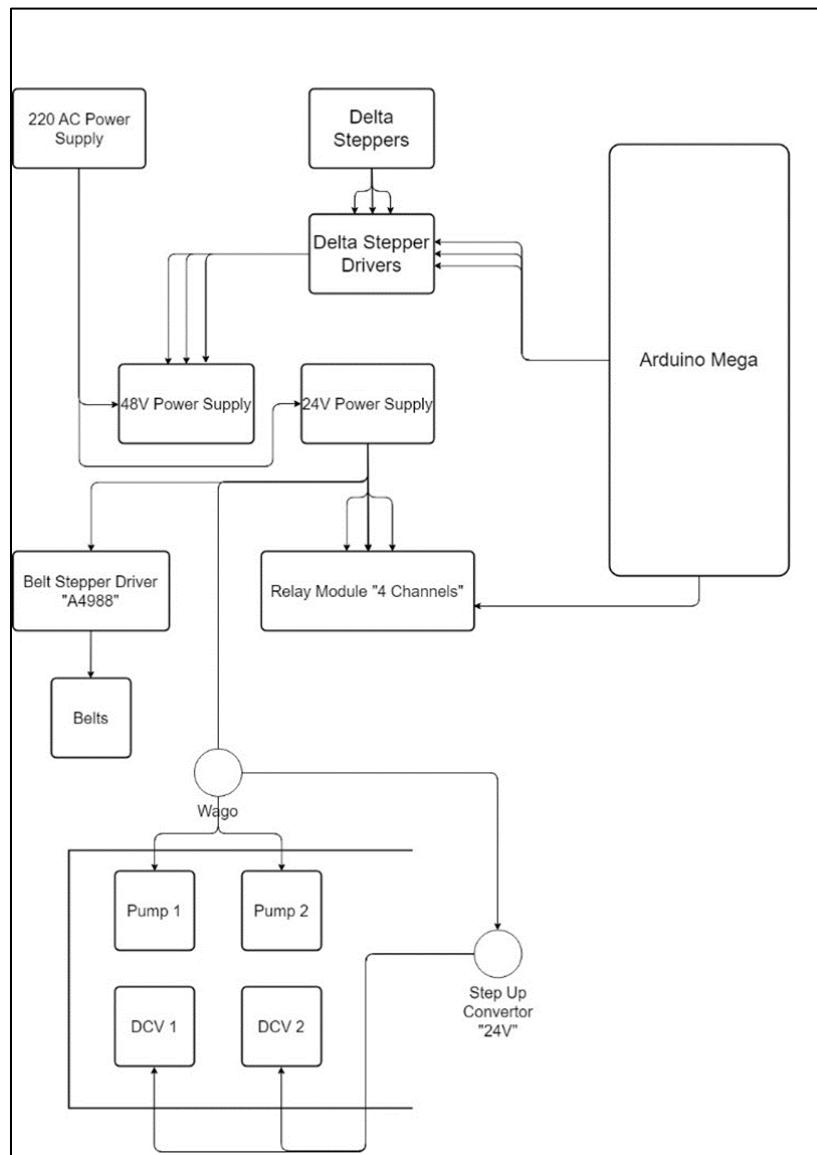


Figure 71 System Electrical Connections Schematics

### 5.3 Graphical User Interface

An essential part of the system is the Graphical User Interface (GUI), which gives users a way to communicate with and manipulate the Delta robot and the soft robotic gripper. The GUI's design, command interface, and user experience are covered in detail in this section.

The GUI's design prioritizes usability and simplicity to make it simple for users to operate the system and keep track of its state. Important components of the design include:

- Robot homing button
- Robot demonstration modes buttons (“cornering” or “pick and place simulation”) with the option of looping the demo nonstop or operating it for one cycle only
- Robot computer vision mode button
- A button that allows the user to select a type of object and quantity during the computer vision mode
- Conveyor belt open button
- Pneumatic unit open button

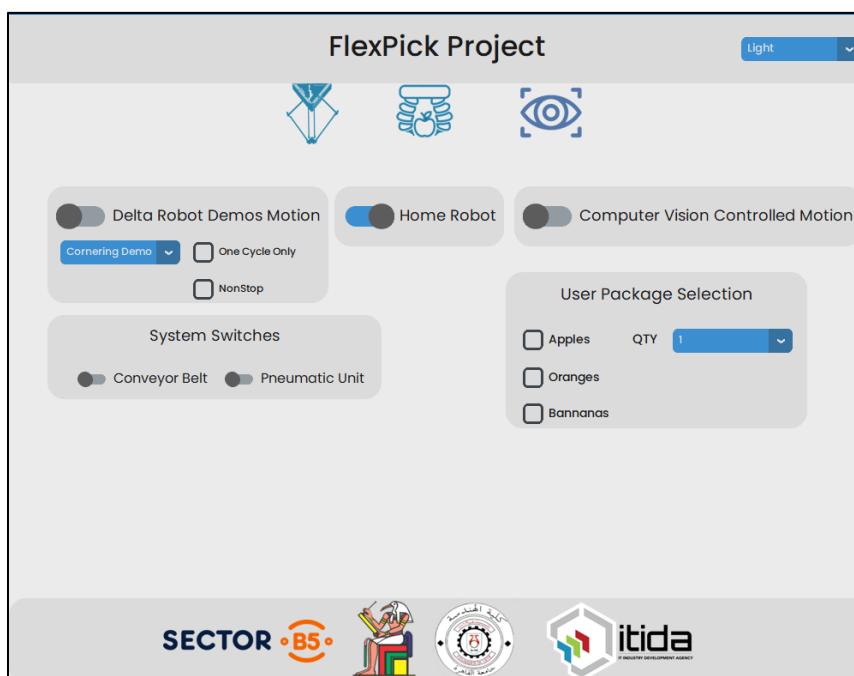


Figure 72 Screenshot from the System's Graphical User Interface

# **Chapter 6**

# **Conclusion**

## **6.1 Market Analysis**

In the field of industrial automation, delta robots with soft gripper end effectors offer a cutting-edge solution. These robots are changing a number of industries with their unmatched adaptability in handling a wide range of products, including fragile or oddly shaped ones. They are distinguished by their high speed, precision, and soft handling abilities. The objective of this market analysis is to examine the present state of the market for delta robots with soft gripper end effectors, including trends, opportunities, and obstacles.

### **6.1.1 Integrated System Commercialization**

#### **Market Overview**

The growing need for automation solutions across many industries is causing a paradigm shift in the worldwide robotics market. In this context, delta robots with soft gripper end effectors have become a leader, providing manufacturers and enterprises with an extremely flexible and effective way to automate processes like picking and placing, packing, sorting, and assembly. Due to considerations like the need for flexible automation, labor shortages, rising labor prices, and an increasing focus on product quality and safety, the market for these robots is expanding rapidly.

#### **Market Segmentation (Targeted Customer)**

Manufacturing, logistics and warehousing, food and beverage, and healthcare are some of the industrial verticals that may be used to segment the market for delta robots with soft gripper end effectors. Robotic automation has uses ranging from delicately manipulating medications in hospital settings to precisely handling electronic components in manufacturing. Each industry offers distinct opportunities and problems for its adoption.

## **Competitive Landscape**

A mixture of well-established robotics firms and quick-thinking startups compete for a bigger piece of the market. In order to obtain a competitive edge in the fiercely competitive market, businesses are concentrating on innovation, product differentiation, and customer-centric solutions. Major players make significant investments in R&D to improve gripper designs, boost robot performance, and incorporate cutting-edge technology like AI, ML, and collaborative features.

### **6.1.2 Delta Robot Commercialization**

#### **Market Overview**

Egypt's delta robot market is expanding steadily, propelled by the growing use of automation technologies in sectors like food and beverage, pharmaceuticals, electronics, and automotive. Because of its high-speed pick-and-place renown, delta robots are the perfect choice for applications that need accurate and timely product handling on assembly lines, packing lines, and material handling tasks.

#### **Market Segmentation (Targeted Customers)**

Delta robot configurations are segmented in Egypt based on a number of characteristics, namely industry verticals, application requirements, and customized setups for different types of operations. Delta robots are essential to several industries, including food & beverage, pharmaceutical, electronics, and automotive. These robots' fast processing speeds and accurate handling make them ideal for jobs including picking, packing, palletizing, and sorting in the food and beverage sector. Delta robots play a crucial role in the pharmaceutical industry by guaranteeing accuracy and regulatory compliance during the packing, labeling, and medicine dispensing processes. In the meantime, they provide accurate manipulation of sensitive parts during component installation, soldering, testing, and inspection in the electronics manufacturing process. Delta robots help in welding, painting, inspection, and part assembly for automotive applications, which lowers costs and improves quality.

Additionally, delta robot configurations are made to fit the needs of particular applications, like precise assembly jobs, high-speed pick-and-place operations, and cleanroom or sterile environment operations. In Egypt, 3-Axis delta robot will be compatible for most of the applications in Egypt.

### **Competitive Landscape**

The Egyptian delta robot business has several obstacles despite its bright future. High upfront investment prices, intricate technological integration and programming, and a lack of labor specialists with robotics and automation experience are a few of these. Furthermore, issues with upkeep, support, and dependability provide obstacles to market acceptance and penetration for small and medium-sized businesses.

#### **6.1.3 Soft Robotic Gripper Commercialization**

##### **Market Overview**

The need for automation solutions in a variety of industries is driving a paradigm shift in the worldwide robotics market. In this context, delta robots with soft gripper end effectors have become a leader, providing manufacturers and enterprises with an extremely flexible and effective way to automate processes like picking and placing, packing, sorting, and assembly. Due to considerations like the need for flexible automation, labor shortages, rising labor prices, and an increasing focus on product quality and safety, the market for these robots is expanding rapidly.

##### **Market Segmentation (Targeted Customers)**

Manufacturing, logistics and warehousing, food and beverage, and healthcare are some of the industrial verticals that may be used to segment the market for delta robots with soft gripper end effectors. Robotic automation has uses ranging from delicately manipulating medications in hospital settings to precisely handling electronic components in manufacturing. Each industry offers distinct opportunities and problems for its adoption.

In Egypt, the soft gripper consumers will be mostly using this product for filtering and packaging for fruits and vegetables warehouses.

## **Competitive Landscape**

There is currently no manufacturer in Egypt that produce this solution, all robotics manufacturers that deploy it are outside of Egypt. Although some of the available solutions (see table 1) include the integration of the soft gripper with a computer vision detection and could be imported to Egypt, they are quite expensive for the Egyptian market, which could help us have our first competitive edge by making such solution available at low prices for Egyptian food processing companies. A table of the available solutions in the market is included in the bottom.

## **6.2 Project Cost Analysis**

### **6.2.1 Introduction**

The financial aspects related to the creation and application of a delta robot with a soft grip mechanism are examined in this cost study. The goal is to present a comprehensive analysis of the costs related to developing, producing, and managing a robotic system of that kind. Stakeholders can make well-informed decisions about budget allocation, resource management, and investment by having a thorough grasp of the financial picture.

### **6.2.2 Scope**

This research covers all the costs associated with the lifecycle of the delta robot with a soft grip. It encompasses expenses related to R&D, prototyping, sourcing materials, assembling, testing, upkeep, and possible scalability. Moreover, an analysis will be conducted of the direct costs, which comprise labor, materials, and components, as well as the indirect costs, which comprise overhead, utilities, and operational expenses. Since our product consists of two major branches (Soft Grip and Delta robot), the analysis will allow to compare these three systems.

### **6.2.3 Methodology**

Data was obtained for this cost analysis from a few sources, including internal estimates, supplier quotations, industry benchmarking, and market research. Where appropriate, assumptions were made to close data gaps or take uncertainties into account. The costs related to each stage of the development and implementation of the delta robot are methodically broken down in this organized analysis.

Stakeholders will learn more about the financial viability of deploying a delta robot with a soft grip by doing this cost study. Decision-makers can use this analysis as a vital tool to

determine areas for cost-saving and optimization as well as to compare the cost-effectiveness of various solutions and the potential return on investment.

#### **6.2.4 Delta Robot**

Cost Components:

##### **1. Materials**

Aluminum Extrusions (2020x3, 2040x1): 1785

Stainless Steel Rods(x1): 370

Aluminum Sheets and Flanges (1x Sheet, 3x Flanges): 1830

Rod Ends (9x PHS6): 810

Aluminum frame Stiffeners(x4): 320

Aluminum Bracket corners(x50): 1000

**The total for Materials: 6115**

##### **2. Fasteners**

Bolts & Screws: 370

Snap Rings: 200

Nuts: 600

**The total Fasteners: 1230**

##### **3. Fabrication Costs**

Cutting: 1000

Drilling: 175

Threading: 1625

Coupling: 1200

Others: 675

**The total fabrication cost: 4675**

#### **4. Motors and Drivers**

Motors (3x Nema 34 12 N.m): 12654

Drivers (3x DM860H): 11970

Mounting Brackets: 900

The total Motor and Drivers: 25524

#### **5. Electrical Components**

Power Supply (48v): 2400

Power Supply (12v): 150

Electrical Switches(x2): 100

Wirings: 800

4-Channel Relay Module(1x): 185

Limit Switches(x3): 225

Electrical Box Enclosure: 600

The total electrical components: 4450

## **6. Webcam: 1500**

## **7. Arduino Controller**

Arduino Mega Development Board: 1600

Arduino Shield: 430

## **8. Conveyor Belt**

Stepper motor(1x NEMA 17): 800

Stepper motor driver(1x A4988): 100

Rollers: 650

The total for Conveyor Belt: 1550

**The total cost of the delta robot is approximately 47,000 Egyptian Pounds.**

### **6.2.5 Soft Robotic Gripper**

#### **1. The Mold of fabrication**

The mold is 3D printed, so it is done by Sector B5

#### **2. Soft Gripper Material**

EcoFlex 0030 (Smooth On) (3200)

#### **3. Pneumatic Control Unit**

Vacuum Pumps (2x TM40A-B0112V): 2100

Solenoid Valves (x2): 800

StepUp converter (x2): 100

Pressure Regulator: 250

Fittings: 150

Pneumatic Box Enclosure: Done by Sector B5, made from Acrylic

Connectors and hoses: 500

#### **4. Utilities (200)**

**The total cost of the delta robot is approximately 7,300 Egyptian Pounds**

**Note:** The prices are for 3-fingers fabrications with its connectors.

### **6.2.6 Total Integrated System**

The integrated, Delta robot with soft gripper, cost will be around 55,000 Egyptian pound, as there will be more wirings that must be added for integrating the two systems with each other. The cost of each component detailed is mentioned in the complete purchased parts list of this project that could be found in the appendix (D) of this report.

### **6.2.7 Market Comparison**

In this section we will compare the price of the Delta robot, Soft Gripper and the integrated system with other companies that produce products in the three industries.

### 6.2.7.1 Delta Robot

Our robot is 3-axis delta configuration with 500 mm working diameter and payload of 1 Kg.

The cost of our delta robot as mentioned above is 47,000 L.E, which is equal to approximately 1000 USD.

Table 11 Delta Robot Competition Price Comparison

Product name	Manufacture r Name	Price	Origin	Working Diameter	Payload
Hornet 565 4 axis industrial parallel robot	Basler Vision System	16,500 USD	China	565 mm	8 Kg
Igus 3-Axis Delta Robot	Igus GmbH	4100 USD	Germany	600 mm	5 Kg
Delta X	Delta X S V5	3700 USD	Vietnam	400 mm	2 Kg
Delta X Official	X	4500 USD	Vietnam	400 mm	2 Kg

### 6.2.7.2 Soft Robotic Gripper

Table 12 Soft Robotic Gripper Market Price Comparison

Product Name	Manufacturer Name	Price	Place of Manufacture	Positioning System Included?	Operating Pressures (Max)	Maximum Handled Weights	Handled Products
3 Finger Parallel Soft Gripper	SoftGripping GmbH	\$795	Germany	No, Gripper Only	100 KPa	450 g	Fruits & Vegetables
4 Finger Parallel Soft Gripper	SoftGripping GmbH	\$973	Germany	No, Gripper Only	100 KPa	600 g	Proteins (seafood and chicken) Bakeries
4 Finger I-EOAT Intelligent Grippers	SRT	\$1,100	Beijing, China	No, Gripper + Pneumatic Controller Only	100 KPa	628 g	Fruits & Vegetables Proteins (seafood and chicken) Bakeries
3 Finger I-EOAT Intelligent Grippers	SRT	\$890	Beijing, China	No, Gripper + Pneumatic Controller Only	100 KPa	870 g	Fruits & Vegetables
2 Finger I-EOAT Intelligent Grippers	SRT	\$660	Beijing, China	No, Gripper + Pneumatic Controller Only	100 KPa	450 g	Fruits & Vegetables
3 Finger Vegetable Gripper	Suzhou Rochu Robotics Co.,Ltd	N/A	Beijing, China	No, Gripper Only	N/A	N/A	Fruits & Vegetables

### **6.2.7.3 Integrated Delta Robot**

The total price of our robot is 55000 Egyptian Pounds, which is equal to approximately 1000 USD.

While the least price of Delta robot integrated with soft gripper is the Vietnamese Delta X V5 Robot (4700 USD) from Delta X with the German 3 Finger Parallel Soft Gripper (795 USD) from Soft Gripper GmbH. Concluding that the least price of integrated delta robot with soft gripper will be around 6500 USD. So, our robot is much lower in price which is our main goal, to be cost efficient.

### **6.2.8 Limitations & Conclusion**

In conclusion, the cost analysis of a delta robot equipped with a soft gripper reveals a promising avenue for industries seeking efficient and delicate object manipulation. The integration of low-cost compliant grippers with 3-DOF delta robots, utilizing off-the-shelf linear actuators and 3D-printed soft materials, presents a cost-effective solution. This innovative approach not only reduces the financial burden but also enhances the robot's dexterity and compliance, making it suitable for handling delicate items without causing damage.

The use of soft 3D-printed materials in the construction of the gripper allows for a reduction in manufacturing costs, while the modular design ensures ease of maintenance and adaptability to various tasks.

Overall, the delta robot with a soft gripper stands out as a cost-efficient, versatile, and innovative solution for modern automation challenges. It embodies the convergence of affordability and advanced technology, paving the way for broader adoption in sectors that require gentle yet precise handling capabilities.

The comparison with other companies is not accurate, as there are no fixed specifications for each company, so we tried to get the one where the specifications are relatively close to each other.

## **6.3 Future Work**

### **Delta Robot**

- For the vibration to be eliminated, a gear box must be implemented with constant reduction ratio. Vibration occurs at low speeds. So, when steps are reduced the delay becomes more noticeable which results in vibration.
- If loads increased, a slip might happen, therefore an encoder for each motor is required to take feedback of the current location to track where it is with respect to the desired location “Closed loop system”

### **Soft Robotic Gripper**

- The material used here is commercial material. For better results with special properties, it should be custom made for each application as desired.
- Force and/or pressure sensors are to be added to ensure that the product is grabbed properly. Force sensor will be implemented within the pneunet itself to sense the gripping force applied on the product. Pressure sensor is to be added to the line to take immediate actions upon measured pressures as it interfaces directly with the microcontroller.
- Relief valve is preferred to be added after the positive pump and before the gripper for more accurate results and to protect the pneunet from being excessively strained.

## Bibliography

- [1]. Design and Manufacturing of a Delta Robot. *2023 Graduation Project, Directory of Graduation Projects, Mechatronics Department, Faculty of Engineering, Cairo University.*
- [2]. Sebastian Castro. “*Trajectory Planning for Robot Manipulators*”. MathWorks Student Lounge. 2019. url: <https://blogs.mathworks.com/student-lounge/2019/11/06/robot-manipulator-trajectory/>
- [3]. Robert L. Williams II. “*The Delta Parallel Robot: Kinematics Solutions*”. In: (2016). url: <https://www.ohio.edu/mechanical-faculty/williams/html/PDF/DeltaKin.pdf>.
- [4]. AVR446: *Linear speed control of stepper motor*. (n.d.).
- [5]. McCormick, E., Wang, Y., & Lang, H. (2019). Optimization of a 3-RRR Delta Robot for a Desired Workspace with Real-Time Simulation in MATLAB. *2019 14th International Conference on Computer Science & Education (ICCSE)*, 935–941. <https://doi.org/10.1109/ICCSE.2019.8845388>
- [6]. Stoychitch, M. Y. (2013). AN ALGORITHM OF LINEAR SPEED CONTROL OF A STEPPER MOTOR IN REAL TIME. *International Journal Of Engineering*.
- [7]. Cianchetti, M., Laschi, C., Menciassi, A., & Dario, P. (2018). Biomedical applications of soft robotics. *Nature Reviews Materials*, 3(6), 143–153. <https://doi.org/10.1038/s41578-018-0022-y>
- [8]. Yasa, O., Toshimitsu, Y., Michelis, M. Y., Jones, L. S., Filippi, M., Buchner, T., & Katzschmann, R. K. (2023). An Overview of Soft Robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 6(1), 1–29. <https://doi.org/10.1146/annurev-control-062322-100607>
- [9]. F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides, “*Soft robotics for chemists*” *Angew. Chem.*, vol. 123, no. 8, pp. 1930–1935, Jan. 2011. DOI: 10.1002/anie.201006464

- [10]. Zhao, S., Li, D., & Xiang, J. (2020). Design and application of PneuNets bending actuator. *Aircraft Engineering and Aerospace Technology*, 92(10), 1539–1546. <https://doi.org/10.1108/AEAT-07-2019-0140>
- [11]. Hu, W. (n.d.). *Flexible Fluidic Actuators for Soft Robotic Applications*.
- [12]. Marechal, L., Balland, P., Lindenroth, L., Petrou, F., Kontovounisios, C., & Bello, F. (2021). Toward a Common Framework and Database of Materials for Soft Robotics. *Soft Robotics*, 8(3), 284–297. <https://doi.org/10.1089/soro.2019.0115>
- [13]. Xavier, M. S., Fleming, A. J., & Yong, Y. K. (2021). Finite Element Modeling of Soft Fluidic Actuators: Overview and Recent Developments. *Advanced Intelligent Systems*, 3(2), 2000187. <https://doi.org/10.1002/aisy.202000187>
- [14]. Patel, A. M. (n.d.). *Design and Analysis of Soft Actuator with Enhanced Stiffness with Granular Jamming*.
- [15]. EcoflexTM 00-30 Product Information,” Smooth-On, Inc. <https://www.smoothon.com/products/ecoflex-00-30/>
- [16]. Steck, D., Qu, J., Kordmahale, S. B., Tscharnuter, D., Muliana, A., & Kameoka, J. (2019). Mechanical responses of Ecoflex silicone rubber: Compressible and incompressible behaviors. *Journal of Applied Polymer Science*, 136(5), 47025. <https://doi.org/10.1002/app.47025>
- [17]. X. Peng, N. Zhang, L. Ge, G. Gu, in Proc. IEEE Int. Conf. on Soft Robotics, IEEE, Piscataway, NJ 2019, pp. 13–18.
- [18]. G. Agarwal, N. Besuchet, B. Audergon, J. Paik, *Sci. Rep.* 2016, 6,1.
- [19]. Dang, H. M., Vo, C. T., Trong, N. T., Nguyen, V. D., & Phung, V. B. (n.d.). *Design and development of the soft robotic gripper used for the food packaging system*.

- [20]. Mohamed, M. H., Wagdy, S. H., Atalla, M. A., Rehan Youssef, A., & Maged, S. A. (2020). A proposed soft pneumatic actuator control based on angle estimation from data-driven model. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 234(6), 612–625.  
<https://doi.org/10.1177/0954411920911277>

# Appendix

## Appendix B: EcoFlex 00-30 Material Data

### General Properties

<b>Product Type</b>	Silicone Rubber
<b>Appearance</b>	Translucent
<b>Recommended for</b>	Healthcare, Medical, Molds, etc.
<b>Agency Rating</b>	ISO 10993-10
<b>Key Features</b>	Low Viscosity, Low Shrinkage

### Physical Properties

Physical Property	Value
Mixed Viscosity	3000 cps
Specific Gravity	1.07 g/cm <sup>3</sup>
Pot Life	45 min
Linear Mold Shrinkage	<0.001 in/in
Cure Time	4 Hrs.
Specific Volume	26.0 cu.in./lb.
Mixing Ration	1A:1B

### Mechanical Properties

Mechanical Property	Value
Shore Hardness	00 – 30
Maximum Elongation	900%
Tensile Strength	200 PSI (1379 KPa)
Tensile Modulus	10 PSI
Maximum Service Temperature	-53 C to 232 C

Dielectric Strength	>350 volts/mil
Tear Strength	38 pli

## Appendix B: Purchased Parts List

Purchased Parts List				
Item	Quantity	Unit Price	Price	Vendor
EcoFlex 0030 Material	1	3200	3200	Smooth On
Stainless Steel Rods (6m)	1	370	370	Al-Fahd Steel
PHS6 Rod End Bearings	12	85	1020	Al-Hutaib
Snap Rings (40 pcs. 6mm shaft)	40	0	0	Free from el sheikh sobhi
M4 Bolts 12mm length	30	1	30	Sabteya
M4 Bolts 15mm length	30	1	30	Sabteya
Aluminum Extrusion 2020 (3m Rod)	3	375	1125	Al-Masreya for Aluminum Trading
Aluminum Extrusion 2040 (3m Rod)	1	660	660	Al-Masreya for Aluminum Trading
Aluminum Brackets Corners	50	20	1000	Al-Masreya for Aluminum Trading
Aluminum Brackets Nuts	100	5	500	Al-Masreya for Aluminum Trading
Aluminum Brackets Bolts	1	88	88	Al-Masreya for Aluminum Trading
Aluminum Sheet 10mm	8	210	1680	Al-Masreya for Aluminum Trading
NEMA34 Stepper Motor 12N.m	3	4218	12654	Top Laser CNC
DM860H Stepper Motor Drivers	3	3990	11970	Top Laser CNC
Active Arm Flange Raw Material	3	50	150	El-Nomrosy
NEMA34 Mounting Brackets	3	300	900	Ampere Electronics
Power Supply 48V/20A	1	2400	2400	MTM Electronics
M6 Bolts Hex Head 50mm + nuts	25	1.2	30	-
M6 Bolts Allen Head	1	30	30	-
M4 Set Screw 10mm (Flange)	20	0.5	10	-
M4 bolts Allen Head + Nuts 25mm	1	30	30	-
M5 Bolts 8mm (Aluminum Profile Brackets)	1	120	120	-
Arduino Mega	1	1600	1600	RAM Electronics
Vacuum Pump	1	1050	1050	Makers Electroincs
Solenoid Valves	2	400	800	Derhali Supplies

Arduino Mega Shield + Wire	1	430	430	RAM Electronics
Electrical Control Box Enclosure	1	600	600	-
Pneumatic Control Box Enclosure	1	0	0	Sector B5 (Funded)
NEMA17 Stepper Motor (Belt)	1	800	800	Ampere Electronics
A4988 Stepper Driver	1	100	100	Ampere Electronics
4 Channel Relay Module	1	185	185	RAM Electronics
DC-DC Step Up Converter	2	50	100	RAM Electronics
Limit Switches	3	75	225	MTM Electronics
Aluminum Frames Corners Stiffeners	80	4	320	-
Webcam	1	1500	1500	-
Electric Switches	2	50	100	-
Pneumatic Fittings	10	15	150	-
Pressure Regulator	1	250	250	-
Wirings	1	800	800	-

### Fabrication & Manufacturing Processes List

Item	Quantity	Unit Price	Price	Place
Aluminum Extrusions Cutting + Drilling	1	325	325	-
Passive Arms & Connecting Rods Threading	1	1200	1200	-
Base Platform & Active Arms Laser Cutting	1	550	550	-
Travelling Platform Laser Cutting	2	140	280	-
Travelling Platform Internal Threading	1	300	300	-
Motor Flanged Coupling	3	200	600	-
Travelling Platform Internal Threading	1	125	125	-
Conveyor Belt Rollers Manufacturing	1	650	650	-

## **Appendix C: Delta Robot Mechanical Working Drawings**