

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF MECHANICAL ENGINEERING

DEPARTMENT OF INSTRUMENTATION AND CONTROL  
ENGINEERING



Master's thesis

## Laboratory Model of Delta Robot

*Bc. Rodrigo Rafael Pinheiro Pereira*

Supervisor: doc. Ing. Martin Novák, Ph.D.

12th June 2019

## I. Personal and study details

Student's name: **Pinheiro Pereira Rodrigo Rafael** Personal ID number: **472431**  
Faculty / Institute: **Faculty of Mechanical Engineering**  
Department / Institute: **Department of Instrumentation and Control Engineering**  
Study program: **Mechanical Engineering**  
Branch of study: **Instrumentation and Control Engineering**

## II. Master's thesis details

Master's thesis title in English:

**Laboratory model of delta robot**

Master's thesis title in Czech:

**Laboratorní model delta robota**

Guidelines:

- 1) Design mechanically the robot as a laboratory model. The robot should be able to pick and move objects running on a conveyer belt
- 2) Connect the electronics, program the robot to perform desired moves
- 3) Install a web camera on the robot frame, on a PC with software such as OpenCV program an application for shape detection. The delta robot should then be able to sort various object shapes running on the conveyer belt into separate boxes (for example cubes, cylinders etc.)

Bibliography / sources:

Opl M. et al. (2011) DELTA - Robot with Parallel Kinematics. In: Jabłoński R., Březina T. (eds) Mechatronics. Springer, Berlin, Heidelberg

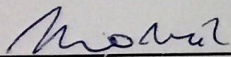
Name and workplace of master's thesis supervisor:

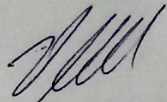
**doc. Ing. Martin Novák, Ph.D., Division of electrotechnics, FME**

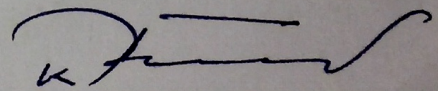
Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **26.04.2019** Deadline for master's thesis submission: **12.06.2019**

Assignment valid until: \_\_\_\_\_

  
\_\_\_\_\_  
doc. Ing. Martin Novák, Ph.D.  
Supervisor's signature

  
\_\_\_\_\_  
Head of department's signature

  
\_\_\_\_\_  
prof. Ing. Michael Valášek, DrSc.  
Dean's signature

## III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

23/04/2019  
Date of assignment receipt

Rodrigo Rafael P. Pereira  
Student's signature

---

## Acknowledgements

Firstly, I would like to thank God for helping me on this journey. Secondly, I would like to be thankful for the support given by my supervisor Martin Novák. Finally, I would like to thank my family and friends for support during writing this thesis.

---

## Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the “Work”), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a

way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In Prague on 12th June 2019

.....

Czech Technical University in Prague

Faculty of Information Technology

© 2019 Rodrigo Rafael Pinheiro Pereira. All rights reserved.

*This thesis is school work as defined by Copyright Act of the Czech Republic. It has been submitted at Czech Technical University in Prague, Faculty of Information Technology. The thesis is protected by the Copyright Act and its usage without author's permission is prohibited (with exceptions defined by the Copyright Act).*

### **Citation of this thesis**

Pinheiro Pereira, Rodrigo Rafael. *Laboratory Model of Delta Robot*. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2019.

---

# Abstract

This thesis had the aim to design a delta robot that is able to pick and place objects. The central controller used in this project was an Arduino Mega 2560; this controller is responsible for the movement of the motors. The motors used for this thesis were servo motors. Furthermore, Python and OpenCV were used in order to implement image recognition. With the help of a camera, image recognition was responsible for the shape detection and the position of the objects, where lately this position would be sent as an input to calculate the inverse kinematics. The objects used were circles and rectangles, and they had similar colour and sizes. The inverse kinematics were implemented using the Arduino IDE. As a result, the robot was able to pick and place objects, defining if there is a circle or rectangle and setting it in different boxes determined by the shapes.

**Keywords** Delta robot, inverse kinematics, image recognition, shape detection

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation and objectives . . . . .	2
1.2	Problem statements . . . . .	2
<b>2</b>	<b>Related Researches</b>	<b>5</b>
2.1	General Related Researches . . . . .	5
2.2	Delta Robots pick and place coloured objects (related researches)	6
2.3	Delta Robots pick and place moving parts (related researches)	9
<b>3</b>	<b>Theoretical Foundation</b>	<b>11</b>
3.1	Delta Robot . . . . .	11
3.2	Arduino . . . . .	14
3.3	Kinematics Theory . . . . .	15
3.4	Image recognition Theory . . . . .	21
<b>4</b>	<b>Methodology</b>	<b>24</b>
4.1	System Implementation . . . . .	24
4.2	Description of Materials . . . . .	26
4.3	Design of the Delta Robot . . . . .	28
4.4	Inverse Kinematics modelling . . . . .	33



4.5	Inverse Kinematics Limitations . . . . .	37
4.6	Inverse Kinematics Coordinates Identification . . . . .	39
4.7	Image recognition . . . . .	40
4.8	Camera calibration . . . . .	43
4.9	Image recognition limitations . . . . .	47
<b>5</b>	<b>Experiments and Results</b>	<b>49</b>
5.1	Range of the model applying the Inverse Kinematics . . . . .	49
5.2	Precision of Movement . . . . .	51
5.3	Quality of Image Recognition . . . . .	53
5.4	Communication between the Arduino and Camera . . . . .	56
<b>6</b>	<b>Conclusion</b>	<b>58</b>
	<b>Bibliography</b>	<b>60</b>
<b>A</b>	<b>Symbols and Acronyms</b>	<b>63</b>
<b>B</b>	<b>Contents of CD</b>	<b>64</b>

---

# List of Figures

2.1	Testing of a delta robot for domino pick and place . . . . .	7
2.2	Testing of a delta robot for pick and place based on colour selection	8
2.3	Working area of Delta robot for pick and place moving parts . . .	9
2.4	Real-time location piece identification . . . . .	10
3.1	ABB Flexible Automation's IRB 340 FlexPicker . . . . .	12
3.2	Delta Robotic with 3 translation DOF . . . . .	13
3.3	Arduino Hardware Example (Mega 2560) . . . . .	14
3.4	IDE Arduino example . . . . .	15
3.5	Fixed Base of the Delta . . . . .	17
3.6	Moving Base of the Delta . . . . .	18
3.7	Real data used in the Handwritten Digit Recognition with a Back- Propagation Network . . . . .	22
3.8	Shape detection application using OpenCV . . . . .	23
4.1	System Implementation Structure . . . . .	25
4.2	Fixed Base Upper View from Autodesk Inventor . . . . .	29
4.3	End-effector perspective view from Autodesk Inventor . . . . .	30
4.4	End-Effector 3D printed . . . . .	31
4.5	Arm perspective view from Autodesk Inventor . . . . .	31

4.6	Arm 3D printed . . . . .	32
4.7	Forearm Representation in the real structure . . . . .	33
4.8	Designed Structure of the Delta . . . . .	34
4.9	Example of collision due to angular movement in the real model . . . . .	38
4.10	Coordinates identification of a Delta . . . . .	39
4.11	Movement Cycle Representation . . . . .	41
4.12	Center Point Representation in an Object . . . . .	45
4.13	Graphical representation of camera calibration . . . . .	46
5.1	Graphical representation of camera calibration . . . . .	51
5.2	Graphical representation of range of motion . . . . .	52
5.3	Representation of Image and Mask in Natural Lighting . . . . .	55
5.4	Representation of Image and Mask in Dark Room . . . . .	56

---

## List of Tables

3.1	Relationship of sizes between arm an forearms . . . . .	13
3.2	Description of variables in Figure 3.5 and 3.6 . . . . .	17
4.1	Robot Structure Used Material . . . . .	26
4.2	List of Hardware . . . . .	27
4.3	List of Hardware Functionality . . . . .	27
4.4	List of Software . . . . .	28
4.5	Parameters value collected from real design . . . . .	36
4.6	Parameters value for HSV colour filtering . . . . .	44
4.7	Parameters necessary for camera calibration . . . . .	47
5.1	Range of motion depending on angle limitations . . . . .	50
5.2	constraints used in Matlab . . . . .	50



---

# Introduction

This chapter is designed to present a brief introduction of this thesis. In order to understand this project, it is necessary to determine the motivation and objectives.

On this project, the focus of the study is a Delta Robot. The Delta Robot is a parallel robot that consists of a movable platform and a fixed one [1]. Moreover, the degrees of freedom are based on the number of kinematic chains that is controlled by an actuator [2].

The advantages of a delta robot are due to its accuracy, speed and rigidity [3]. In the other hand, the delta robots are also limited in the workspace [4]. This robot has a simple structure and is mainly divided into arms, base, end-effector, and joints. Usually, to control this robot is used the direct or inverse kinematics.

Furthermore, in this research, it is used image recognition. Image recognition has the aim to identify an object and treat this image according to its target [5]. Nowadays, this technique has been used for many applications such as differentiate objects [6], road sign recognition [7], lip-detection [5]. However, in this research, image recognition is used to identify shapes and allow

them to calculate the coordinates of the objects.

### 1.1 Motivation and objectives

Firstly, it is essential to understand the reason to develop the research. The motivations of this research is described in the following items :

- Have research in a topic where it is related to the industrial field. So, using a delta robot that it is widely used in the industry, especially in the manipulation of material.
- Understanding the principle of the Inverse Kinematics and apply it in a real project. It is crucial to have the experience to use the kinematics in a real project and not just in a simulated environment.
- Understand the needs of a real robot in the industry. This gives the ability to have a better understanding of an actual field in the industry and what could be the challenges that will be faced.
- Development of an unique lab task, that can be studied for other students.

Although the motivations for this research are essential, it is necessary to point out the objectives. The objectives of this project are a major key to understand this research.

### 1.2 Problem statements

The aim of this topic can be divided into two categories. These categories are the main goals and the sub-goals. This division is vital in order to separate the project into many parts that could be developed separately.

Firstly, it is necessary to describe the main target of this project. The final result expected from this project is the design and implementation of a delta robot that is able to pick and place objects using image recognition. Image recognition is responsible for identifying the shapes of the collected objects and its coordinates. This robot will be used for laboratory purposes where students could do a further development on the same structure.

Secondly, understanding the primary goal of this project allow it to be divided into many topics and sub-topics. These new topics are mainly divided into hardware development and software development.

- Using CAD software, design a delta robot.
- Using the designed parts in the CAD software, it is necessary to build the structure of the robot (arms, base, end-effector, support for the structure)
- Implementation of inverse kinematics according to the designed delta robot
- Identify the working space of the robot. Understanding the range of movement of this robot.
- It is necessary to select an appropriate camera that will be responsible for collecting the images
- Calibration of the camera
- Identify the object coordinates
- Development of image recognition using Python + OpenCV
- Establish the communication between the camera and the robot using Python + OpenCV and Arduino



## 1. INTRODUCTION

---

- Prepare the Arduino code for pick and place objects according to the coordinates collected from the image recognition

It is possible to identify that this project is divided into topic and subtopics. This division allows a better division of the steps and also simultaneous topics can be developed without the entirely need of the physical structure of the others. As it is possible to identify the main topic is actually a brief description of the whole project and the sub-topics are the steps necessary to achieve this goal.

---

## Related Researches

This chapter concatenates some researches that are related to the project. The reason to point out this researches is to have a theoretical background that can be used for further studies.

Moreover, having access to this previous researches can allow a better understanding of the advantages and limitations of the problems. Usually, in the results presented here is possible to see the achievements of the study and also if there was any distortion between the project and the actual final product.

### 2.1 General Related Researches

This section is used to collect researches related with the use of Delta Robot. This researches might be use as background and orientation for future projects.

Firstly, Huang et al. in 2007 developed a 2-DOF transitional parallel robot for pick and place operations. The main idea of this research was using the torque and velocity find the minimum transversal time to collect an object [8]. In order to find this solution, it was used the path jerk limit.

Secondly, Nabat et al. in 2005, introduces a 4-DOF parallel robot to collect objects. The main focus on this research was the speed of object collection, where an adaptation in the architecture of a delta, H4 and I4 robots [9].

Another similar research uses a robot for apple harvesting. In [3], a device with manipulator and end-effector using a image system was used for collect apples. Using the vision-based module the prototype of this robot device was able to collect one apple in every 15 seconds [3].

Furthermore, there was developed a delta robot for educational purposes. In 2012, Kovar et al. used swarm optimization for image recognition in a delta robot. There are many tasks in this project such as design, control, image recognition and kinematics analysis.

Finally, [10] proposed a complete modelling of the Delta Robot. This research developed a complete solution for an Delta Robot model with all the necessary parameters such as direct, inverse kinematics and also, inverse statics and dynamics. Furthermore, in [10], it was presented some computation results according to the model that was proposed.

### **2.2 Delta Robots pick and place coloured objects (related researches)**

This section is destined to collect some related researches in the topic of Delta Robots where the use of pick and place and object identifications is used. This researches can be used as a base for the methodology or even to compare results with the current project.

Firstly, in 2015, a project named Design and Implementation of a New DELTA Parallel Robot in s Competitions has strong similarities with this project. This project had some parts that were 3D fabricated by a 3D printer.

## 2.2. Delta Robots pick and place coloured objects (related researches)

The main target of this project was to develop a delta robot with a low budget that could be able to pick and place objects based on image recognition [3].

Moreover, there were 3 case scenarios that the robot should be able to execute. Firstly, it was a case of a pick and place operation of some domino pieces that is demonstrated in Figure 2.1. Secondly, the robot should be able to write and draw according to some specifications. Finally, the third case is the most similar to this project that is picking and place an object according to its colour.

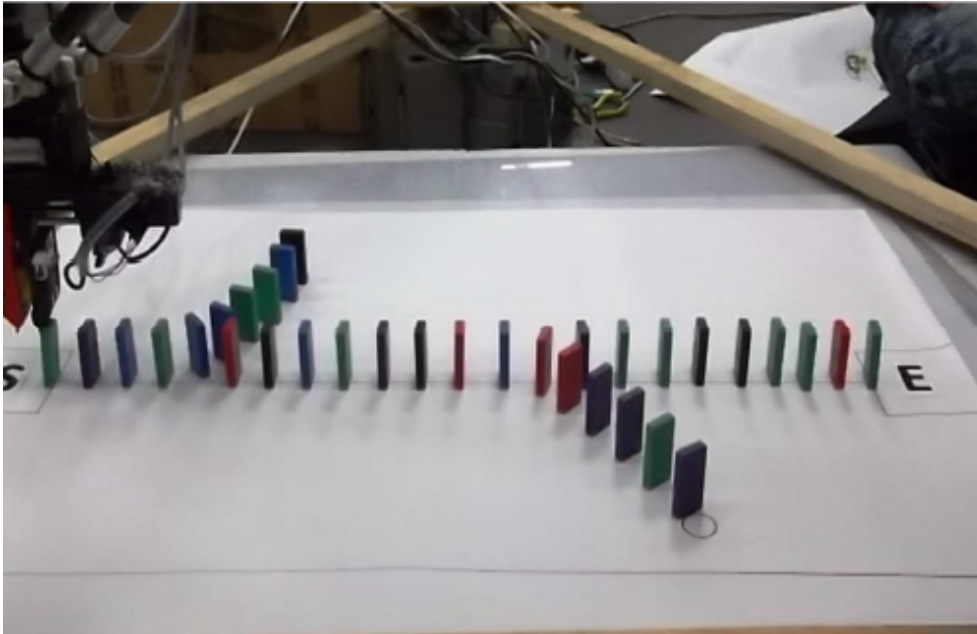


Figure 2.1: Testing of a delta robot for domino pick and place [3]

As we can see in Figure 2.2, the robot has to identify not just the correct object colour, but also the right hole to place. Furthermore, all objects are the same, so there is no need for shape recognition. The only requirement for this project is to work with RGB colours and place it correctly. According to [3], the accuracy of this test was approximately 92 per cent.

## 2. RELATED RESEARCHES

---

Beside the advantages, this project also shows us some limitations. In the research, it was pointed out that the literatures lack the detail about the correct position of the end-effector [3]. There was no explanation in precise details of how to eliminate this problem.

Furthermore, delta robot is based on design optimization. If there is not presented a plausible solution for the end-effector malfunction, there will be probably lower productivity and also the performance of the robot.

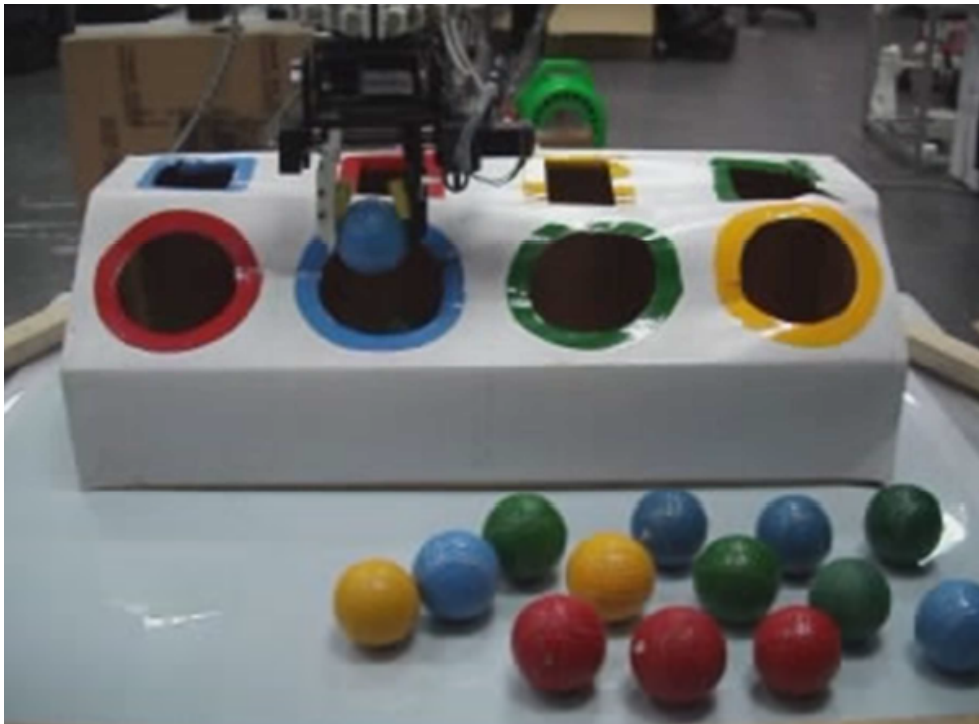


Figure 2.2: Testing of a delta robot for pick and place based on colour selection [3]

## 2.3 Delta Robots pick and place moving parts (related researches)

In 2016 Lin et al. created a delta robot to pick and place moving parts. The name of the project is Vision servo based Delta robot to pick-and-place moving parts. Furthermore, the main target was to have a delta robot able to pick and place objects in a moving conveyor [11].

Moreover, Lin et al. used forward and inverse kinematics to achieve the motion of the robot. According to their calculations, the robot should be able to move in working area of a cylinder with the dimension of 25mm x 300mm [11] (as shown in Figure 2.3).

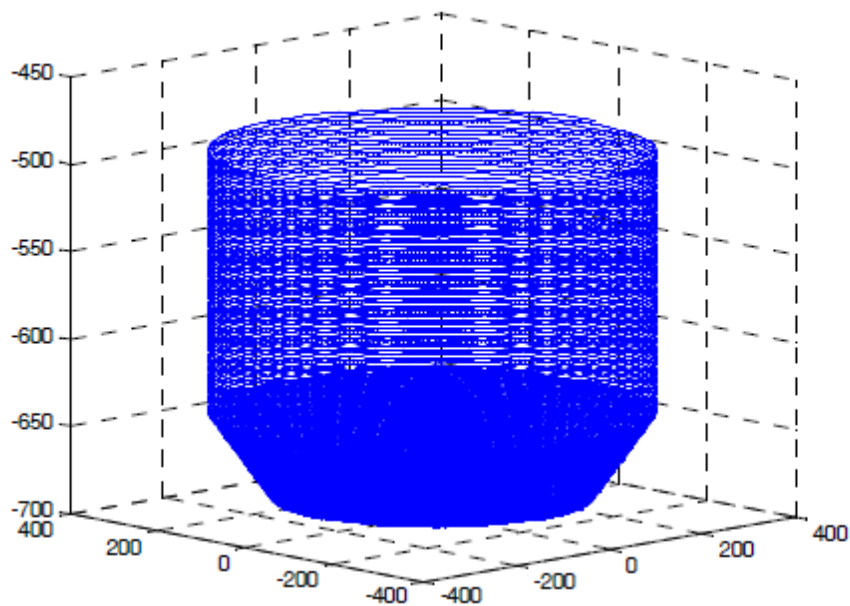


Figure 2.3: Working area of Delta robot for pick and place moving parts [11]

Furthermore, in this project, they developed image recognition. On their image recognition, they used two methodologies called Canny and Sobel for edge detection. In their conclusion, they achieve the result that Canny edge

## 2. RELATED RESEARCHES

---

detection is more suitable for edge detection [11].

This project also uses a real-time location. In the real-time location, it was used the visual servo system that the image is captured every time that a workpiece enters in a specific zone [11] (as shown in Figure 2.4).

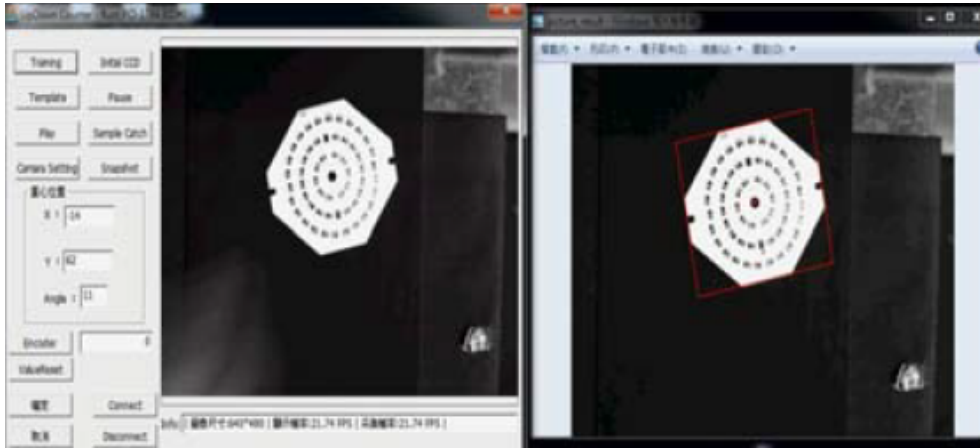


Figure 2.4: Real-time location piece identification [11]

---

# Theoretical Foundation

This chapter has the aim of introducing all the background necessary to understand the principles behind the project. Two parts composed this project: hardware and software, where hardware is all the physical components used to build the robot and the software are all the programming parts.

The development of the hardware is divided into the delta robot theory, servomotor and its sensors. The understanding of these parts is necessary in order to combine them correctly.

The software development is represented by the Arduino and also by image recognition. The main goal of the software is control the motion of the delta robot and also give it the ability to pick and place some object in a determined location.

In the following sections, further details will be given for each topic.

## 3.1 Delta Robot

A delta Robotic is a type of parallel mechanism that consists of three arms connected in a common base. According to [12], parallel robots have more



### 3. THEORETICAL FOUNDATION

---

accuracy because they do not have cumulative joint errors. Further advantages of this robot are low inertia and high stiffness. Furthermore, The delta robot is composed of four main parts such as, base, arm, forearms and end-effector these parts are described in Figure 3.2.



Figure 3.1: ABB Flexible Automation's IRB 340 FlexPicker [13]

The delta robot has 3 degrees of freedom (DOF). Each motor is responsible for one of the DOF's. In order to enhance control, the motors should be located at 120 degrees from each other, therefore, giving the same angle from each other, making in total 360 degrees (circle dimension).

Modelling a delta robot needs to be done by fulfilling three parameters in order to achieve the calibration of the model. Firstly, the model should be proportional, the angles, arm and forearms should have some logic proportion

[12].

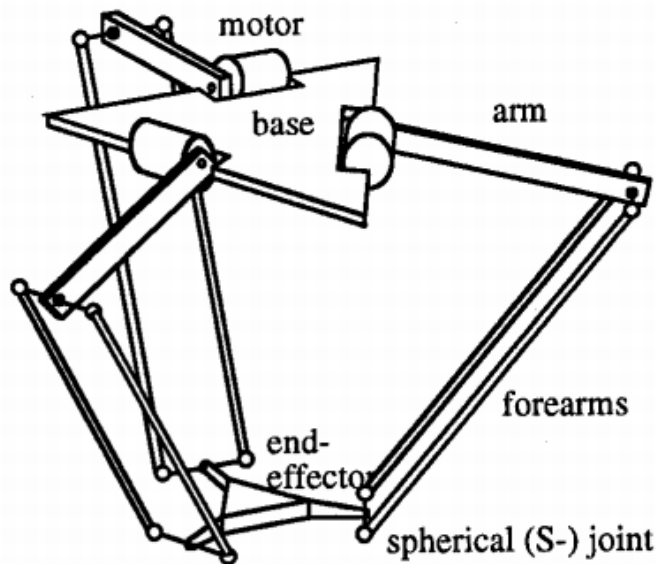


Figure 3.2: Delta with 3 translation DOF [12]

The choice of the size of arms and forearms depends on the requirements from the project. In order to have a more covered area where the robot could achieve it is necessary to use bigger arms and forearms. However, this relationship between forearms and arms can give you more or less torque, range of motion or speed. The Table 3.1 will show the difference according to the sizing of the robot.

Table 3.1: Relationship of sizes between arm and forearms

<b>Comparison between sizes of arm and forearm</b>	
<b>Forearm &gt; Arm</b>	<b>Forearm &lt; Arm</b>
Large range of motion	Lower range of motion
Lower speed	Higher speed
Higher torque	Lower torque

Many areas can require the use of the delta robot. On industries, this type of mechanisms is used for packing or working with electronic components. Furthermore, they can be used in the pharmaceutical and medical industries

with the ability to pick and place to handle drugs and chemicals [14].

## 3.2 Arduino

The Arduino is an open-source platform with the aim the connection between hardware and software. Firstly, the device presents a larger type of boards that can be selected according to the need of the project [15]. Some well-known boards for Arduino are Uno, Mega2560, Nano and Mini. One of the hardware of an Arduino is shown in Figure 3.3.

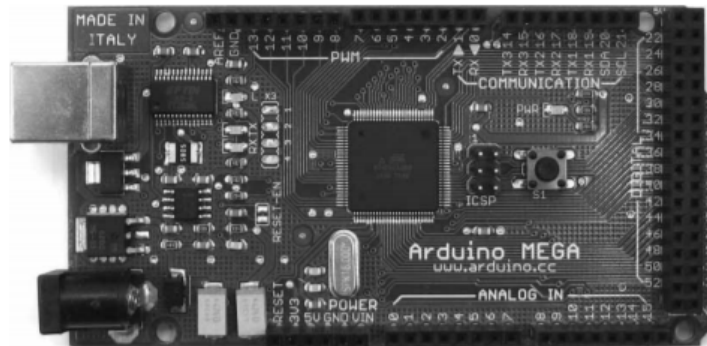


Figure 3.3: Arduino Hardware Example (Mega 2560) [15]

An open platform is used in order to program the Arduino hardware. The name of the platform is called Arduino IDE, and it is an open source platform that can be used to develop the projects from the users and apply directly into the board. One of the reasons for the use of this platform is due to its simplicity of programming when compared with other programming languages such as C and C++ [15]. One demonstration of an Arduino IDE can be found in Figure 3.4.

The Arduino process is based on inputs, processing and outputs. Firstly, the data can be processed via the sensors that send a signal to the board. The signal is processed via the board and also the coding into the IDE. Finally,

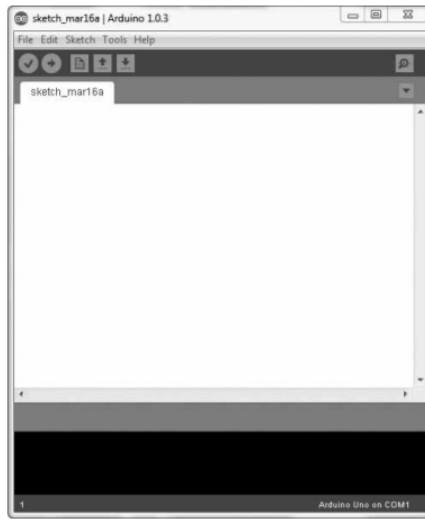


Figure 3.4: IDE Arduino example [15]

the processed signal generates an output that may give the desired response [16].

The Arduino is extensively used due to its advantages. First of all, the Arduino presents a low cost, due to the board's cost be around 50 dollars. Furthermore, the Arduino can be connected into a wide diversity of operating systems and also it is an "easy" programming language when compared with other languages [15]. Finally, it is an open platform that can be modified according to user needs. Therefore, the Arduino is a platform desired due to its low cost and full usability in the projects.

### 3.3 Kinematics Theory

This section will discuss the kinematics of the delta robot. The kinematics are divided into two categories that are inverse and forward kinematics. In this project, the focus will be on inverse kinematic because the robot that was developed need to identify some distance and via this distance calculate the

### 3. THEORETICAL FOUNDATION

---

proper angles to achieve the object.

First of all, it is necessary to understand the inverse kinematic principle. The inverse kinematics is based on finding the suitable combination of angles in order to achieve some coordinates in the axis X, Y and Z [17]. The methodology used to find the inverse kinematic can vary between algebraic methods, interactive methods and Jacobian inversion method, optimization based, cyclic coordinate, genetic programming and others. In this project, it was used the algebraic method due to the simple design of the delta robot.

In order to model the inverse kinematics of the delta robot, it is necessary to find the geometrical analysis for the model. In Figure 3.5 is possible to identify the fixed base of the structure, where B1, B2 and B3 are the positions of the motors. From Table 3.2 it is possible to identify the meaning of each variable used to describe the inverse kinematics of the delta robot.

Furthermore, the Equations 3.1, 3.2, 3.3 and 3.4 are demonstrating the relationship between the pairs  $w_B$  and  $s_B$ ,  $u_B$  and  $s_B$ ,  $w_P$  and  $s_P$ ,  $u_P$  and  $s_P$  [18].

$$w_B = \frac{\sqrt{3}s_B}{6} \quad (3.1)$$

$$u_B = \frac{\sqrt{3}s_B}{3} \quad (3.2)$$

$$w_P = \frac{\sqrt{3}s_P}{6} \quad (3.3)$$

$$u_P = \frac{\sqrt{3}s_P}{3} \quad (3.4)$$

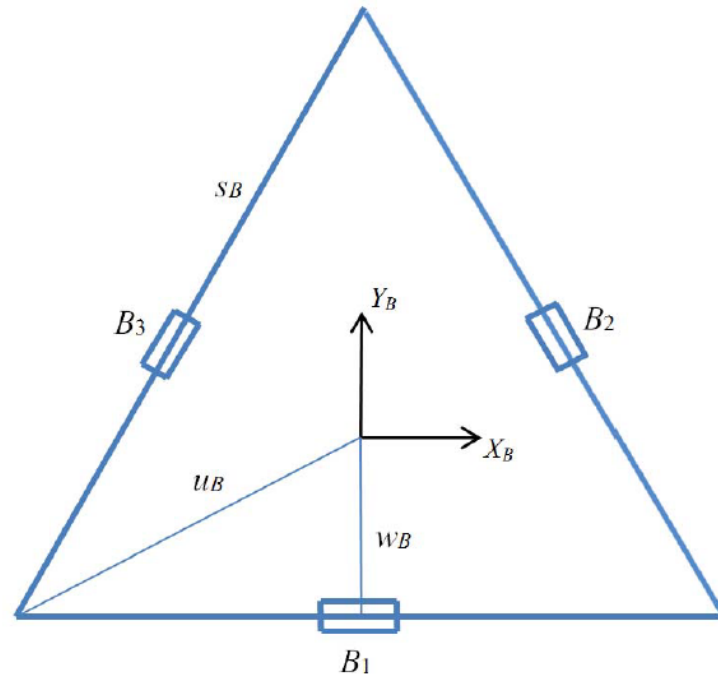


Figure 3.5: Fixed Base of the Delta [18]

Table 3.2: Description of variables in Figure 3.5 and 3.6

Parameter	Description
sB	Triangle side of the fixed base
sP	Triangle side of moved base
L	Length of arm
l	Length of forearm
wB	Distance from 0 to near side
uB	Distance from 0 to vertex
wP	Distance from 0 to near side
uP	Distance from 0 to vertex

In order to achieve the solution, it is necessary to find the appropriate angles in the matrix  $\theta$  according to the coordinates from matrix  $P$ . Where,

$$\theta = [\theta_1 \ \theta_2 \ \theta_3] \text{ and } P = [x \ y \ z]$$

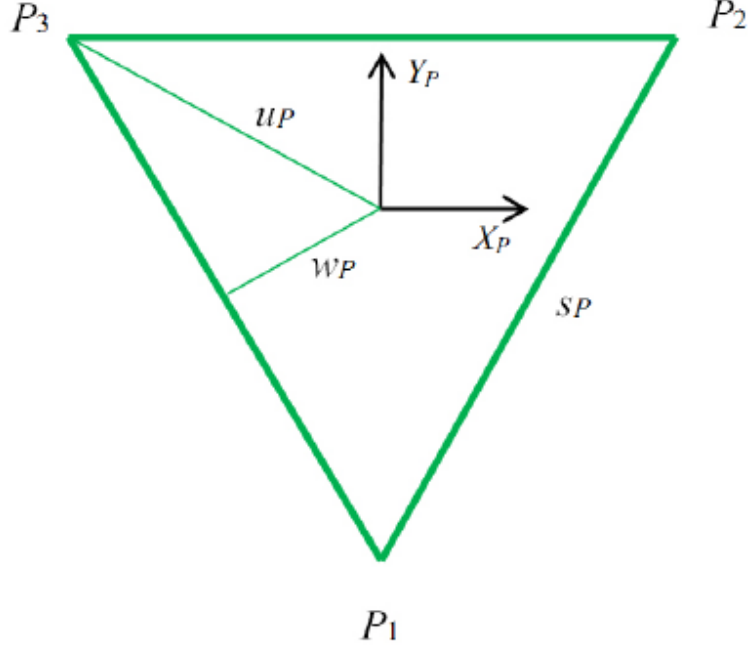


Figure 3.6: Moving Base of the Delta [18]

According to [18], there are three constraints for a delta robot and these constraints are represented by the Equations 3.5, 3.6 and 3.7.

$$2L(y + a)\cos\theta_1 + 2zL\sin\theta_1 + x^2 + y^2 + z^2 + a^2 + L^2 + 2ya - l^2 = 0 \quad (3.5)$$

$$-L(\sqrt{3}(x+b)+y+c)\cos\theta_2+2zL\sin\theta_2+x^2+y^2+z^2+b^2+c^2+L^2+2xb+2yc-l^2 = 0 \quad (3.6)$$

$$L(\sqrt{3}(x+b)-y-c)\cos\theta_2+2zL\sin\theta_2+x^2+y^2+z^2+b^2+c^2+L^2-2xb+2yc-l^2 = 0 \quad (3.7)$$

Where a, b and c are defined in the equations 3.8, 3.9 and 3.10.

$$a = wB - uP \quad (3.8)$$

$$b = \frac{sP - \sqrt{3} \times wB}{2} \quad (3.9)$$

$$c = wP - \frac{wB}{2} \quad (3.10)$$

The next step is apply the Inverse Kinematics according to the constraints founded in the equations 3.5, 3.6 and 3.7. In order, to achieve it it is necessary to apply the geometrically/trigonometrical solution that will generate equations according to Equation 3.11 [18].

$$E_i \cos \theta_i + F_i \sin \theta_i + G_i = 0 \quad (3.11)$$

Where, i= 1,2,3

The Equation 3.11 is solved by using the Tangent Half-Angle Substitution where it will simplify to the form according to equation 3.12 to find  $\theta$ .

$$\theta_i = 2 \arctan(t_i) \quad (3.12)$$

Where  $t_i$  is defined by the equation 3.13.

$$t_i = \frac{-F_i \sqrt{E_i^2 + F_i^2 - G_i^2}}{G_i - E_i} \quad (3.13)$$



### 3. THEORETICAL FOUNDATION

---

In total, the Equation 3.13 will give to solutions due to the  $\pm$  symbol presented in the equation. Therefore, it is necessary to choose the angles for the inner side that will represent the movement of the motor.

The last step is define the value of  $E_{1,2,3}$ ,  $F_{1,2,3}$  and  $G_{1,2,3}$ . The solution of this equations derive from the constraints founded previously. Therefore, the solution that is necessary are presented in the equations 3.14,3.15,3.16, 3.17,3.18,3.19, 3.20,3.21,3.22.

$$E1 = 2L(y + a) \quad (3.14)$$

$$F1 = 2zL \quad (3.15)$$

$$G1 = x^2 + y^2 + z^2 + a^2 + c^2 + L^2 + 2ya - l^2 \quad (3.16)$$

$$E2 = -L(\sqrt{3}(x + b) + y + c) \quad (3.17)$$

$$F2 = 2zL \quad (3.18)$$

$$G2 = x^2 + y^2 + z^2 + b^2 + c^2 + L^2 + 2xb + 2yc - l^2 \quad (3.19)$$

$$E3 = L(\sqrt{3}(x - b) - y - c) \quad (3.20)$$

$$F3 = 2zL \quad (3.21)$$

$$G3 = x^2 + y^2 + z^2 + b^2 + c^2 + L^2 - 2xb + 2yc - l^2 \quad (3.22)$$

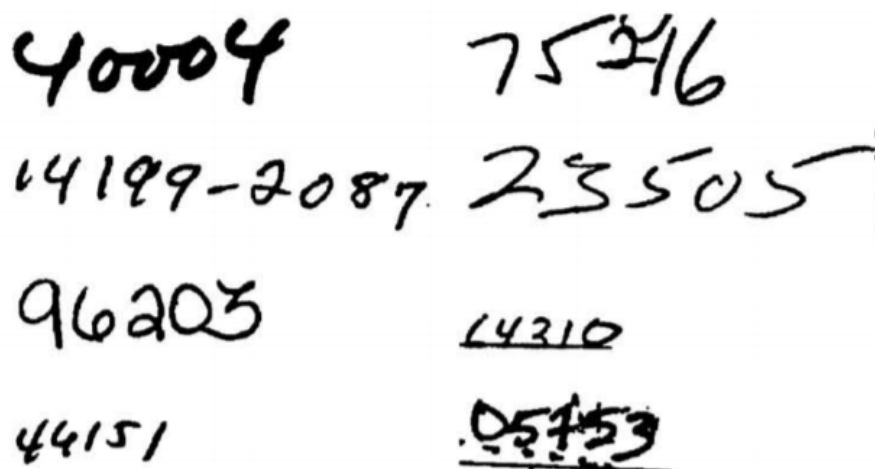
In conclusion, these are all the equations necessary to apply the Inverse Kinematics into the delta robot. In the next chapters, it will be described all the required values necessary to substitute for the constants to solve the equations.

### 3.4 Image recognition Theory

Image recognition has been a tool used all over the world. Scientists are developing every day more technologies that use image recognition in the field of automation. Some applications are implemented into cars and industrial machines where they can detect objects shapes or even faces. According to [19], image recognition is an instrument that enhances robot intelligence based on what they have ability to recognize.

An advantage of this technique is that it does not need to be a perfect image. In other words, the image does not need to be complete in order to be recognized if there is a large number of data collection to be trained. Even simple handwriting can be identified by this method due to its large amount of data collection [20].

Cun Et Al. in his article of Handwritten Digit Recognition with a Back-Propagation Network proved that even different handwriting could be recognized using different methods [20]. According to them, it is possible that the computer interprets different handwriting and give an output with high per cent of precision. Some examples of the images used for the article can be seen in Figure 3.7. Hence, it is possible to analyze one of the potential advantages of image recognition.



The image displays several examples of handwritten text, including digits and numbers, arranged in two columns. The left column contains: '40004', '14199-2087', '96203', and '44151'. The right column contains: '75216', '23505', '14310', and '05153'. The handwriting is in black ink on a white background.

Figure 3.7: Real data used in the Handwritten Digit Recognition with a Back-Propagation Network [20]

One of the software used for image recognition is OpenCV. Open CV is an open source computer vision [19] that is able to assist with the majority of vision problems. The main advantage of this software is its architecture and memory management. This software can communicate with other programming languages, such as Python. Another advantage of this program is that it is able to work with video (recorded or live) and also images depending on its need.

The OpenCV can be used in many image recognition solutions. This software can assist in image filtering, shape detection, identification of edges and corners and even colour detection [19]. Therefore, this program is helpful in the field of image recognition and is widely used depending on its application.

In Figure 3.8, it is possible to analyze one example of shape detection. As we can see, the table was detected with the line red and after debugged with green lines. On this image recognition, just the largest ellipse is identified [19].

In conclusion, it is possible to analyze that image recognition can be used

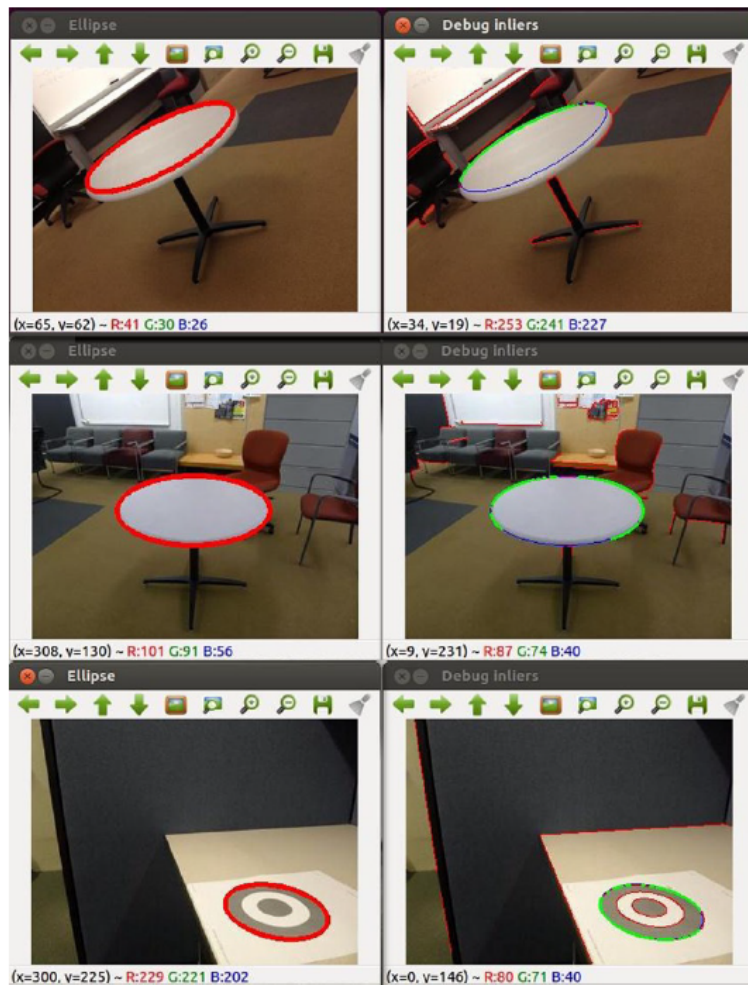


Figure 3.8: Shape detection application using OpenCV [19]

in many areas. Furthermore, there are many techniques that can be used such as shape detection. Furthermore, the OpenCV software is a tool used for applying the face recognition, due to make the communication between the camera and the computer.

---

# Methodology

This chapter aims to describe the methodology used in this project. Four main sections need to be discussed: the list of materials used, designing of the robot and the modelling of inverse kinematics and image recognition.

Therefore, this chapter should describe all the principal elements used in this project. This is necessary in order to correctly understanding of the project and also the allowance of further development of the experiment that uses the same criteria.

## 4.1 System Implementation

This section has the aim to present the system structure. In the system structure it should identify all the hierarchies and divisions of the main idea of the project. This structure is useful in order to have a clear visualization of the whole project.

First of all, it is possible to identify from Figure 4.1 that there are three main groups. These main groups are divided into mechanical and hardware development, system kinematics and software development. Each of this group has a specific roll in the project.

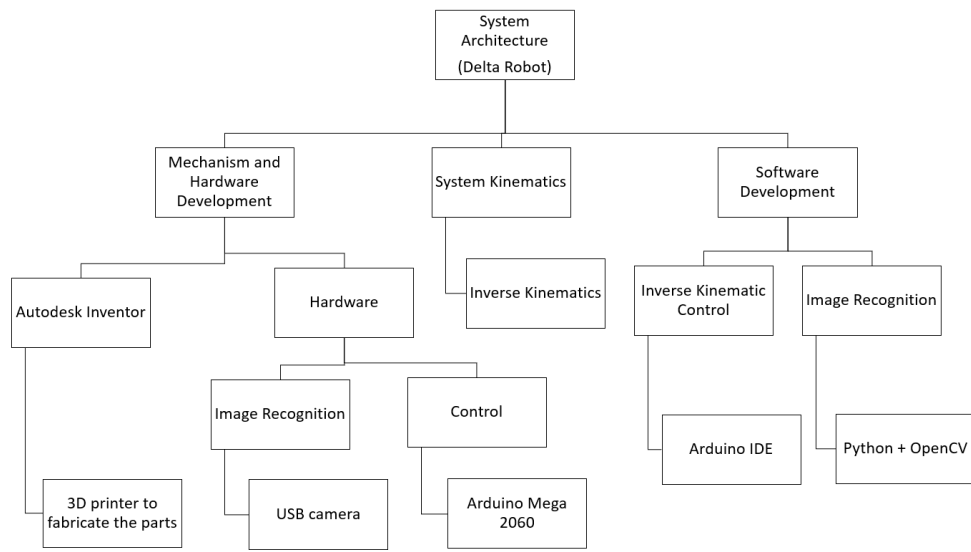


Figure 4.1: System Implementation Structure. Image adapted from [3]

As it is possible to identify from the Mechanism, the parts were 3D printed. All the plastic parts were modelled in the Autodesk Inventor software. After designed, those parts were fabricated in a 3D printer.

Furthermore, hardware development can be divided into two main groups that are image recognition and control part. Image recognition was done using a USB camera where it is necessary to be calibrated in order to fulfil its purpose. Moreover, the control was done by an Arduino Mega 2060 that should establish communication with the camera.

Another group is the system kinematics. The inverse kinematics's is where all the calculations necessary are done in order to calculate the angles for the Delta robot. All the parameters required for the calculations are taken from the design. Therefore, the inverse kinematic is entirely dependent on the dimensions of the robot.

The third group is software development. The software development is sub-

divided into two sub-groups that are the inverse kinematic control and image recognition implementation. The inverse kinematic calculations are done in the Arduino IDE, where the position of the servos will be sent through there. Image recognition uses the OpenCV library inside of the Python software. Furthermore, the python and openCV are responsible for collect the position of the object.

In conclusion, there are three main groups in these projects that are the development of the mechanical and hardware part, the kinematics development and implementation of the software. This tree groups are the central core of the project and combined are responsible for the movement of the structure.

## 4.2 Description of Materials

The description of materials is a fundamental part in other to describe the methodology. Without the proper description, it is not possible to analyze the integrity of the project. The materials are divided into two main parts that are the designed structure of the robot and the control part.

From Table 4.1 it is possible to acquire the physical material structure of the robot. The structure was constructed with carbon and metal rod and plastic (3D printed parts).

Table 4.1: Robot Structure Used Material

<b>Robot Part</b>	<b>Material</b>
<b>Fixed Base</b>	Plastic (3D printed)
<b>Movable Base</b>	Plastic (3D printed)
<b>Arms</b>	Plastic (3D printed)
<b>Forearms</b>	Steel rod 2mm
<b>Joints</b>	Ball joints
<b>Joint connections</b>	Carbon rod 5mm

In this project, it is possible to divide the control materials into two parts that are hardware and software. Firstly, the hardware would be all physical parts necessary to build the control part of the robot. Finally, the software was divided into the control part and image recognition part. The list of materials can be described in the Table 4.2 and 4.4.

Table 4.2: List of Hardware

<b>Equipment</b>	<b>Model</b>	<b>Quantity</b>
<b>Servo motor</b>	MG995 Tower Pro	3
<b>Arduino</b>	Mega2560	1
<b>Mega Servo Shield</b>	Keyes v2.0	1
<b>Power Supply</b>	5V	1
<b>USB camera</b>	i-Look 325T Genius	1

It is possible to analyze from the Table 4.2 the list of material used for the physical part of the control. For more details about the functionality of each equipment, the information can be founded in Table 4.3.

Table 4.3: List of Hardware Functionality

<b>Equipment</b>	<b>Functionality</b>
<b>Servo motor</b>	Movement of the arm
<b>Arduino</b>	Board Controller
<b>Mega Servo Shield</b>	PWM external board
<b>USB camera</b>	Image recognition

It is possible to identify from the Table 4.4 the three softwares used into this project. Firstly, Python was used in order to establish communication between the Arduino and OpenCV software. Secondly, the Arduino has the function of a controller, it is able to control the servo positions according to some inputs from the USB camera. Finally, the openCV has the aim of acquiring the axis coordinates of an object and send it to the Arduino.

Therefore, in this section it was possible to check the material utilized into



Table 4.4: List of Software

<b>Software</b>	<b>Functionality</b>
<b>Python</b>	Communication
<b>Arduino IDE</b>	Control of servos
<b>OpenCV</b>	Image recognition

this project. Furthermore, a brief description of each equipment was given in this section.

### 4.3 Design of the Delta Robot

This section will describe the designing of the delta robot. Firstly, it was necessary to determine the size of the fixed platform (base) and movable platform (end-effector), arm and forearms.

Firstly, in the designing of the platforms (base and end-effector) was used the Autodesk Inventor software version 2019 in order to visualize the robot parts and create models for 3D printing. Finally, the design of the arm and forearm followed the criteria of the Table 3.1. It was chosen that the forearm would be bigger than the arm.

The base is possible to be visualized from the Figure 4.2. From this figure, we can infer some observations such as:

- The servos are at angle of 120 degrees from each other.
- The rectangular holes presented in the structure allows the arms to move via the structure without colliding with it.
- The whole structure has a diameter of 200 mm.
- The rectangles walls are necessary for the attachment of the servos into the structure, giving the movement of up and down to the movable base.

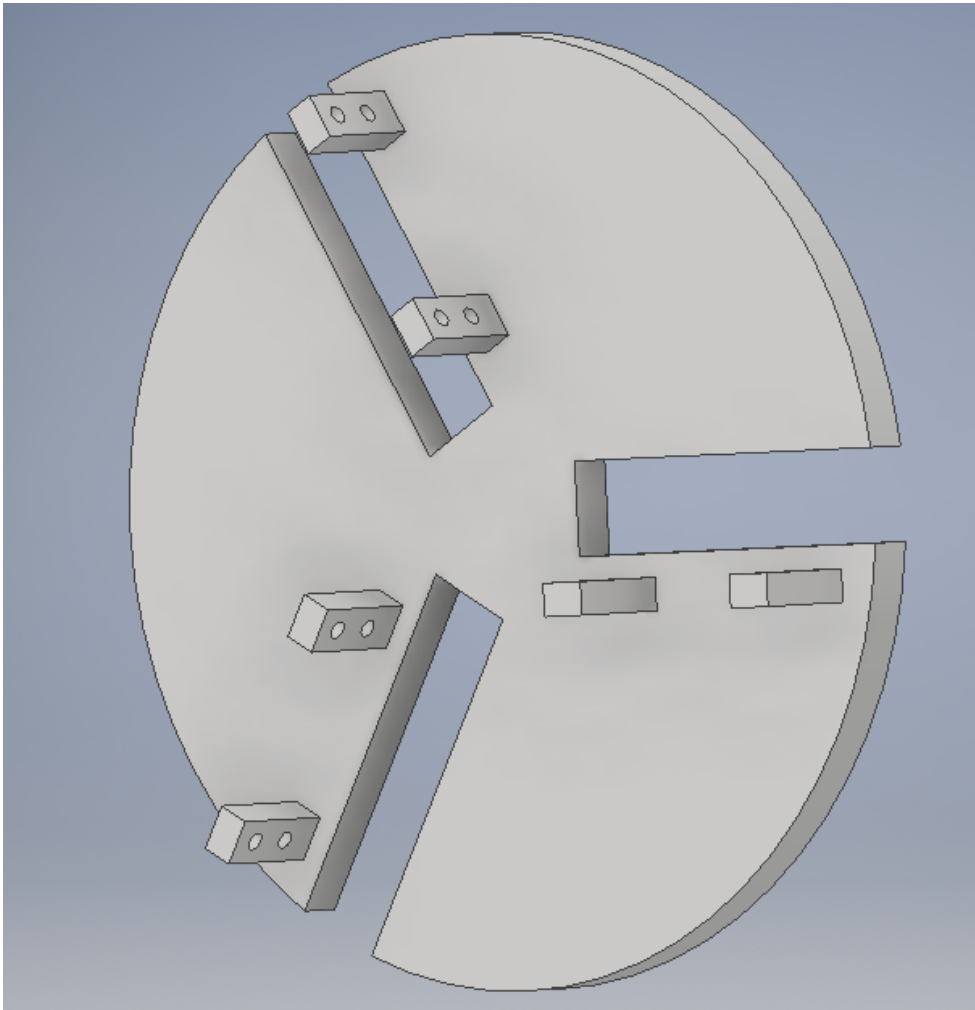


Figure 4.2: Fixed Base Upper View from Autodesk Inventor

The end-effector design can be described by Figure 4.3. In this figure the attachment of the arms into it is also in a space with 120 degrees. This 120 degrees distance from each forearm is necessary in order to create a proportion into the design. Furthermore, due to the low torque of the servo motors, the structure could not be heavy otherwise the servo would not be able to lift it, so a type of triangle shape was used in order to produce a lighter part.

Furthermore, it is possible to identify from Figure 4.4 some functionality

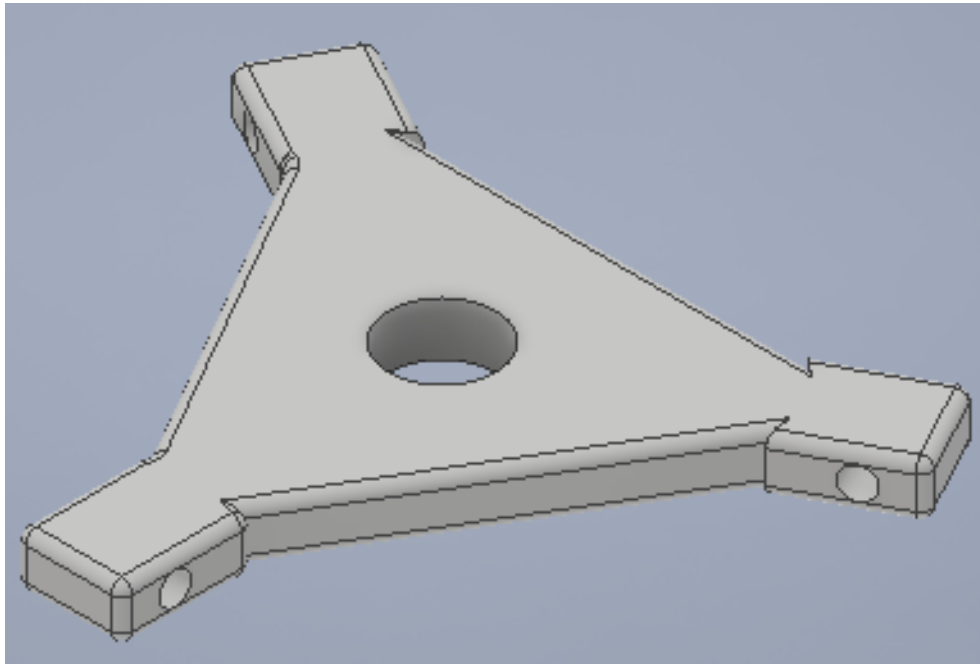


Figure 4.3: End-effector perspective view from Autodesk Inventor

applied into the real 3D printed part.

- From the centre of the right hole to the centre of the biggest left hole, the dimension is 150 mm.
- The central circle hole in the middle is the space for attaching the grabber.
- The carbon rob with 5mm will be enclosed in the lateral holes that is present in each corner.

The last piece designed in Autodesk Inventor for this project was the arms. The arm is attached directly into the servo motor structure with screws. The design of the arm can be observed in Figure 4.5.

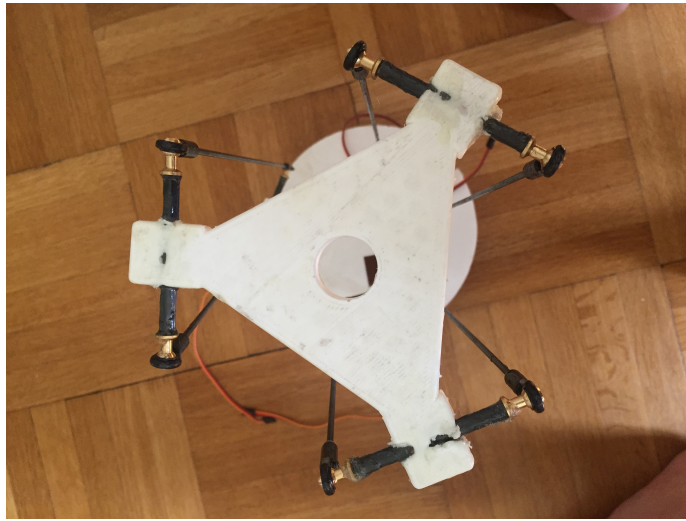


Figure 4.4: End-Effector 3D printed

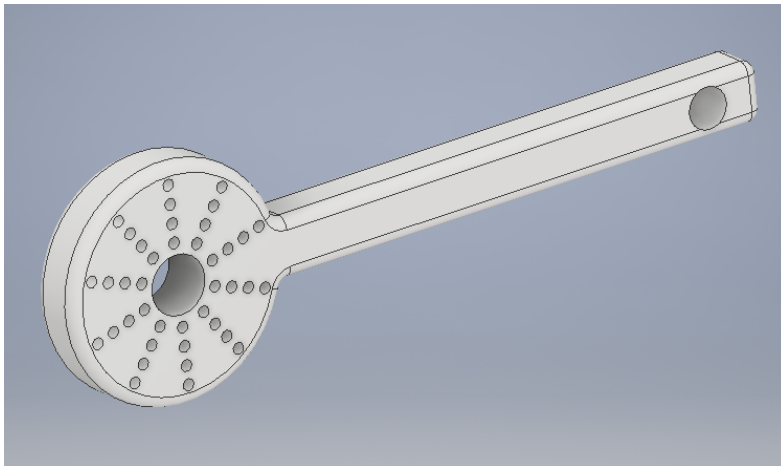


Figure 4.5: Arm perspective view from Autodesk Inventor

The arm design has the functionality to attach the servo motor and also the forearm into it. From the Figure 4.6 it is possible to identify that :

- The hole in the right side of the structure is used to attach the 5mm rod.

#### 4. METHODOLOGY

---

- The biggest hole from the left side is just a space left for the maintenance of the servomotors.
- The small holes from the left side is used to attach the arm to the servomotor piece.

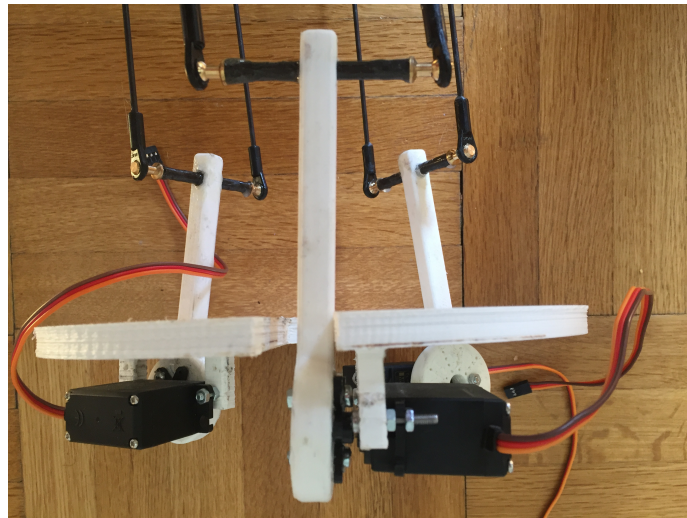


Figure 4.6: Arm 3D printed

The design of the forearm was done using a steel rod. The length of the structure is about three times bigger than the arm, and its actual measure is 350 mm. A critical feature of the forearm is that it can not bend meanwhile the structure is moving; otherwise it would compromise the inverse kinematics. It is possible to visualize the forearm structure from Figure 4.7.

The whole structure can be visualized in Figure 4.8. As we can see, there are four main parts in a delta robot: base, end-effector, arms and forearms.

Therefore, the design of a Delta is a fundamental part for control it. If a not suitable setup is designed, it is possible that the robot would move in a minimal area or even would not be able to lift a suitable material.



Figure 4.7: Forearm Representation in the real structure

## 4.4 Inverse Kinematics modelling

This section is destined to explain the implementation of the inverse Kinematics into the designed delta robot. In order to apply this technique into the design, it is necessary to define the sizing of the robot and apply the required calculations based on the equations from section 3.3.

First of all, it is necessary to define the parameters for the robot. In order to calculate the equilateral triangles from the fixed base (upper platform) and end-effector (free platform in the down part), it was necessary to create a triangle shape. Using the design created in the Autodesk Inventor Software, the sizes of the triangles of  $s_B$  and  $s_P$  were respectively 241 mm and 104 mm.

Besides the side of the triangle, determine the size of the arm and forearm are essential. Using the Table 3.1, it was chosen that the forearm should be bigger than the size of the arm. This sizing would allow a broad range of motion and also a higher torque to the model. The length of the arm ( $L$ ) and forearm ( $l$ ) are respectively 150 mm and 350 mm.

According to the primary parameters ( $s_B$ ,  $s_P$ ,  $L$  and  $l$ ), it is possible to

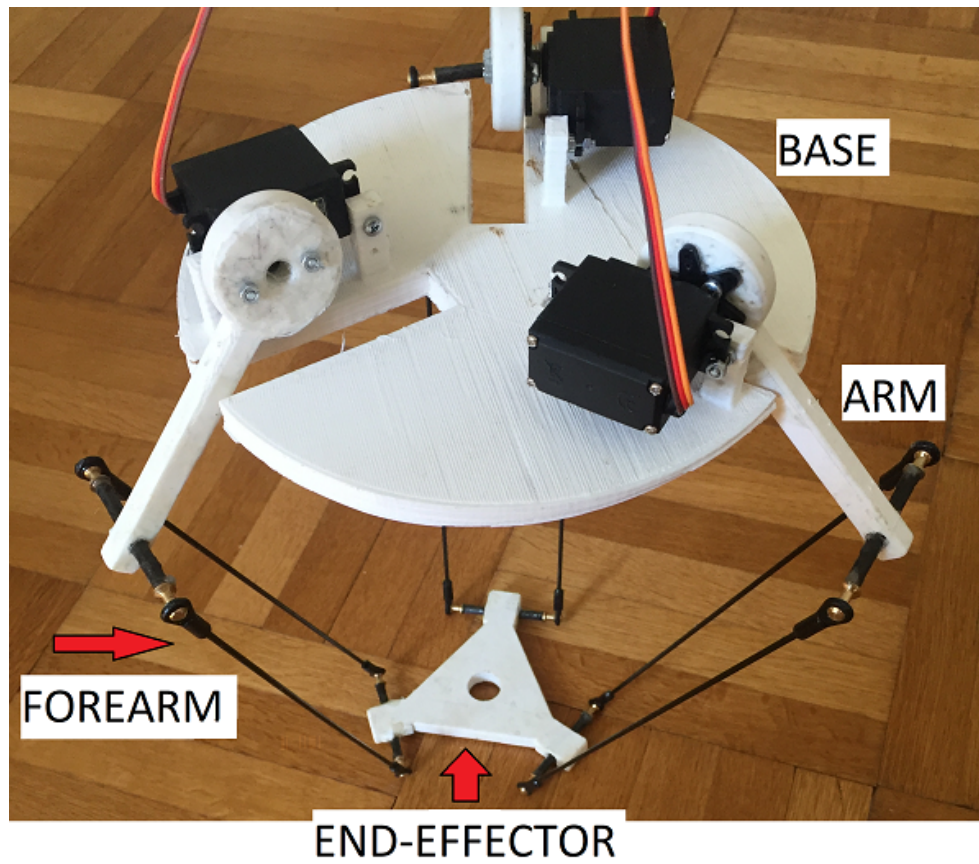


Figure 4.8: Designed Structure of the Delta

calculate the inverse kinematics. According to the axis X, Y and Z founded by the equations, it is possible to determine the angle values of  $\Theta_1$ ,  $\Theta_2$  and  $\Theta_3$ .

Therefore, we have from Equations 3.1 to 3.4 the values of  $w_B$ ,  $u_B$ ,  $w_P$ ,  $u_P$  from the following equations :

$$w_B = \frac{\sqrt{3} \times 0.241}{6} \quad (4.1)$$

$$u_B = \frac{\sqrt{3} \times 0.241}{3} \quad (4.2)$$

$$w_P = \frac{\sqrt{3} \times 0.104}{6} \quad (4.3)$$

$$u_P = \frac{\sqrt{3} \times 0.104}{3} \quad (4.4)$$

It is important to remember that the values applied for the equations are in meters. Furthermore, from the equations 3.8 to 3.10, it is possible to find the values of a, b and c using the calculated values of w<sub>B</sub>, u<sub>B</sub>, w<sub>P</sub>, u<sub>P</sub>.

$$a = 0.0696 - 0.06 \quad (4.5)$$

$$b = \frac{0.1040 - \sqrt{3} \times 0.0696}{2} \quad (4.6)$$

$$c = 0.03 - \frac{0.0696}{2} \quad (4.7)$$

All the necessary parameters for the next steps can be founded into Table 4.5. This table represents the values of s<sub>B</sub>, s<sub>P</sub>, L, l w<sub>B</sub>, u<sub>B</sub>, w<sub>P</sub>, u<sub>P</sub>, a, b and c. Looking into the values, we can conclude that the most critical parameters are the dimensions of the bases and also the length of the arm and forearm. If there is any modification into the value of these mains parameters, the inverse kinematic need to be re-calculated.



Table 4.5: Parameters value collected from real design

Parameter	Value	Description
sB	0.241 m	Triangle side of the fixed base
sP	0.104 m	Triangle side of moved base
L	0.15 m	Length of arm
l	0.35 m	Length of forearm
wB	0.0696 m	Distance from 0 to near side
uB	0.1391 m	Distance from 0 to vertex
wP	0.03 m	Distance from 0 to near side
uP	0.06 m	Distance from 0 to vertex
a	0.0095	Auxiliary variable
b	- 0.0082	Auxiliary variable
c	-0.0048	Auxiliary variable

---

After calculating the values of a, b and c, it is possible to calculate the final equations that will determine the angles for  $\Theta$  according to the following equations:

$$E1 = 2 \times 0.150 \times (y + 0.0095) \quad (4.8)$$

$$F1 = 2 \times z \times 0.150 \quad (4.9)$$

$$G1 = x^2 + y^2 + z^2 + 0.0095^2 + -0.0048^2 + 0.1^2 + 2 \times y \times 0.0095 - 0.350^2 \quad (4.10)$$

$$E2 = -0.150(\sqrt{3}(x + -0.0082) + y - 0.0048) \quad (4.11)$$

$$F2 = 2 \times z \times 0.150 \quad (4.12)$$

$$G2 = x^2 + y^2 + z^2 - (0.0082)^2 + (-0.0048)^2 + 0.1^2 - 2x(0.0082) - 2y(0.0048) - 0.350^2 \quad (4.13)$$

$$E3 = 0.150(\sqrt{3}(x + 0.0082) - y + 0.0048) \quad (4.14)$$

$$F3 = 2z(0.150) \quad (4.15)$$

$$G3 = x^2 + y^2 + z^2 - 0.0082^2 - 0.0048^2 + 0.1^2 + 2x(0.0082) - 2y(0.0048) - 0.350^2 \quad (4.16)$$

It is possible to conclude from the equations 4.10, 4.13 and 4.16 that all the necessary values to find the angles are the axis X, Y and Z.

Furthermore, inverse kinematics is related to image recognition. The coordinates of X, Y and Z are provided by the software used for image recognition (OpenCV), and after processed into the Arduino to achieve its angular calculations.

## 4.5 Inverse Kinematics Limitations

So far, the calculations were done based on the sizing of the designed structure. However, there are some constraints necessary to point out in order to appropriate application of the inverse mathematical modelling into this project.

Firstly, the usual motors used for the delta robot application are step motors. This preference occurs due to its capability of moving in an angular

#### 4. METHODOLOGY

---

range from 0 to 360 degrees. However, in this project, the chosen motors were servos that have the ability to rotate in a range of 0 (zero) to 180 degrees. Therefore, it is not possible to accept a calculated angle (from the equations) that will be lower than 0 (zero) or bigger than 180 degrees.

Besides the angle limitation due to the servo motors, there are angular limitations because of the structure. In the method that the structure was designed if the engines achieve the full range of 180 degrees, the forearms of the delta robot could collide. Therefore, in order to limit this collisions, the actual possible angular movement of the servos should have a minimum value of 5 degrees and a maximum value of 140 degrees. An example of collision can be observed in the real model represented by Figure 4.9.

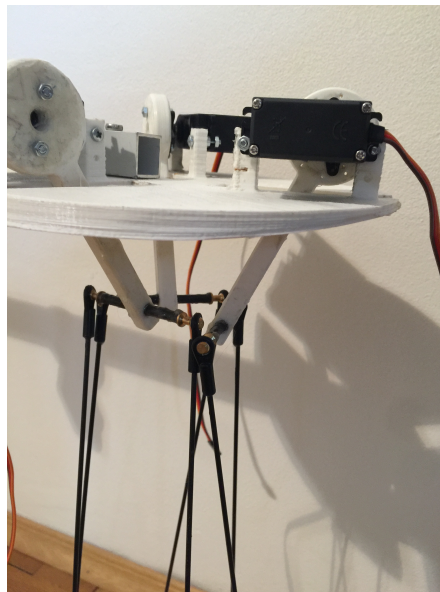


Figure 4.9: Example of collision due to angular movement in the real model

If there were a different size of the base, the collision presented would not exist. Due to the size of the base be relatively small for the length of the arms and forearms, the robot has this collision. One possible solution for this problem could simply increase the size of the base.

In order to map these limitations and check the real solution, a Matlab code was created. The methodology used to find its limitations was generating an algorithm in the Matlab software that is able to identify all the possible solutions that fit in the given constraints cited before. The result of this code is a graphical representation of a semi-sphere that can be used as a based to recognized possible X, Y and Z possibilities. The function used for it was called "boundary".

## 4.6 Inverse Kinematics Coordinates Identification

In order to find the angles, firstly is necessary to identify where in a space plane is located the axis and where are located the central points. This points can be modified if there will be implemented other type of inverse kinematics calculation. However, for this specific model, it is necessary to have these coordinates as a parameter. The coordinates can be inferred from Figure 4.10.

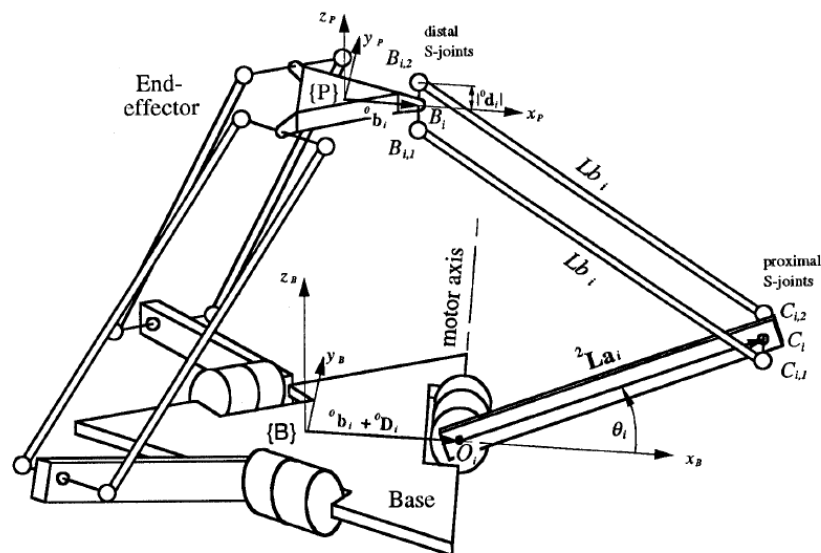


Figure 4.10: Coordinates identification of a Delta [12]

First of all, it is necessary to identify the angle position. As we can see from Figure 4.10, the base and  $\Theta$  are related. Therefore, if the position between the base and arm is parallel, it means that the angle  $\Theta$  is equal to zero. On the other hand, if the position between the arm and the base is perpendicular, there should be an angle bigger than zero.

Secondly, it is necessary to identify the coordinates for X, Y and Z. Also, from figure 4.10 it is possible to conclude that the axis coordinates are related to the end-effector and its distance from the centre of the base. Therefore, as higher as the end-effector is from the base as bigger should be the axis Y. The same principle is applied for the axis X and Z

### 4.7 Image recognition

Image recognition has a fundamental function in this project. The aim of it is to recognize the object and also determine its coordinates. Therefore, it is necessary to first understand the whole process before start to explain the steps used for apply image recognition.

According to the image 4.11, we can see the steps necessary until one full cycle of the movement is done. There are mainly ten steps that need to be followed in its order to achieve the required outcome. Image recognition starts from step 2 (camera goes on) and has its function until step 7 (send the coordinates to arduino).

In this project, Python IDE was used for image recognition. Even though python was used as the main programming language, the OpenCV library is responsible for the majority of the outcomes. The openCV is the library that contains all the functions already pre-programmed and ready to be used by the python or even other programming languages.

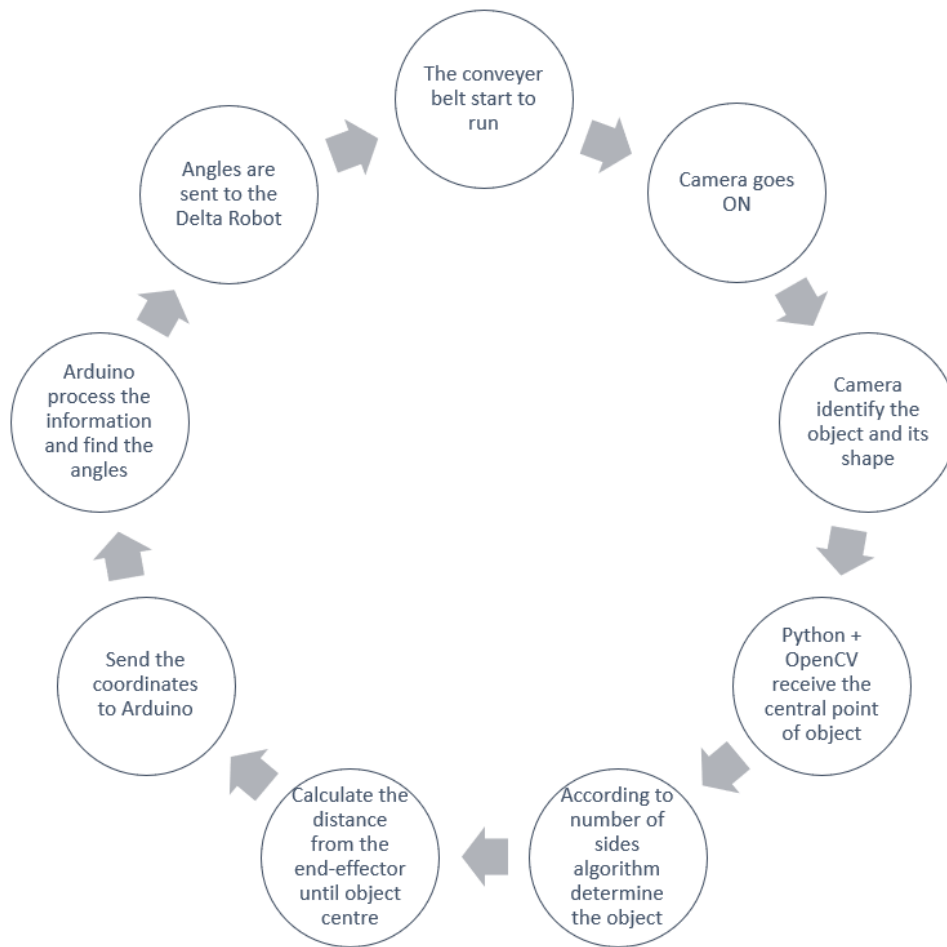


Figure 4.11: Movement Cycle Representation

Firstly, the camera needs to identify the object. The object identification in this case was done by the help of the color recognition. The technique using, in this case, is called colour space conversion, that converts colours from BGR to HSV. Where BGR stands for Blue Green Red and HSB means Hue Saturation Value.

In order to identify the objects, the conversion from BGR to HSV is essential. This technique filters the image according to its settings, where if properly set the only image identified will be related to a specific colour. Therefore,

this technique is responsible for filtering our colour target and removing the unwanted ones.

Secondly, the camera is responsible for detect the images and the Python and OpenCV for identifying the object shape. After the appropriate filtering of the colours, the camera with the help of Python and OpenCV is able to detect a format that will be used for detecting the objects. In this project just two shapes of objects where used, circle and rectangle.

In order to detect the shapes, it is necessary to differentiate between rectangles and triangles. In the openCV there is a function called findContours, where it approximates the form of an object according to the image presented. As filtering, if there is a contour that gives an approximation of four sides, it means that the camera is detecting a square or a rectangle. On the other hand, a circle has no edge defections or a high number of edge detection, so for this case, it was stipulated that the circles will be represented by a shape that contains a number higher or equal to ten edges.

Thirdly, it is necessary to identify the central point of the object. The openCV is able to locate the highest left corner of an object by its shape, however the program tries to define the shape as the smallest square possible where the object/ shape could fit inside. So, in order to identify the object center, it is necessary to divide the high and the width of this smallest rectangle possible by 2. Therefore, this approximation of the smallest rectangle is able to identify the object coordinates in the camera space.

One important characteristic is that the camera gives all the values in pixels. For an appropriated calculation of the angles for the inverse kinematics, the value of the coordinates has to be in real value (meters, centimeters or millimeters). Therefore, it is necessary to convert the pixels to another unit.

Finally, once the camera is calibrated and converted the coordinates from pixels to another unit, it is necessary to estimate the zero coordinates from the camera to the end-effector. The values of the centre of the object are based on the coordinates (0,0,0) of the camera, however, it is necessary to estimate the distance until the end-effector of the delta robot in order to achieve its destination and pick up the object. Therefore, the coordinates need to be the distance from the end-effector to the object.

The last step done by the image recognition software is send the coordinates  $x,y,z$  to the Arduino. After defining the precise coordinates from the object to the end-effector, the python + openCV needs to communicate with the Arduino and send the final coordinates allowing the Delta Robot to achieve its destination point.

## 4.8 Camera calibration

The camera calibration is vital in order to avoid errors. There are many types of errors that could be faced due to a non-appropriate calibration before starting the experiments. One of the most common problems is non-object detection or even objects that should not be detected due to the wrong conversion from BGR to HSV. Moreover, the object coordinates could be compromised due to an error in the camera calibration. Therefore, a proper calibration is fundamental in order to obtain the shape of the objects and also its coordinates.

First of all, it is necessary to setup the HSV filtering. The selected parameters for the HSV is done by choosing the lower and higher limit. Therefore, there is needed to choose six parameters in order to achieve the desired filtering.

HSB filtering was done manually. With the help of a track bar, it was pos-



sible to manually manipulate the higher and lower values of the HSV filtering and then finding the more suitable configuration according to the project's needs. The HSB parameters can be founded in the Table 4.6.

Table 4.6: Parameters value for HSV colour filtering

Parameter	Value
Lower H	0
Lower S	120
Lower V	174
Higher H	180
Higher S	255
Higher V	255

After setting the parameters for HSV, it is possible to calibrate the camera. Firstly, it is necessary to know the real size of the objects. In this project, the circles have a diameter of 2 cm, and the rectangles have the size of 2cm x 2cm. The python algorithm is able to measure the centre of the objects by using the method of the smallest square that will enclose the object in case (discussed in Section 4.7). Therefore, the output given by the code is the high ( $h_{object}$ ) and the width ( $w_{object}$ ) of the object that is:

$$h_{object} = 95 \text{ pixels}$$

$$w_{object} = 93 \text{ pixels}$$

Therefore, in order to find the centre of the object, it is necessary to divide the high and width by 2. An image representation of this can be founded in Figure 4.12. In this case, it could be any object (circle or rectangle), because the target was to calculate the difference between pixels and the real image in centimeters.

Once we have the  $h_{object}$  and  $w_{object}$  values, it is possible to compare the pixels values from the camera to centimeters. This is possible because the real value of the objects is known. Therefore, it is possible to equal the pixels

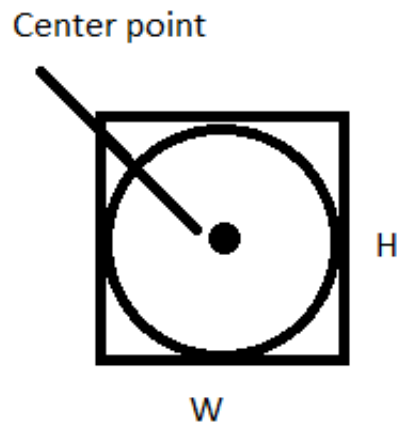


Figure 4.12: Center Point Representation in an Object

values founded with the actual values of the object.

$$h_{object} = 95 \text{ pixels} = 2 \text{ cm}$$

$$w_{object} = 93 \text{ pixels} = 2 \text{ cm}$$

Hence, in our estimation, we have that 1 centimeter represents 47.5 pixels for the high and 46.5 cm in width.

Moreover, it is also possible to estimate the total high ( $h_{total}$ ) and the total width ( $w_{total}$ ) (see Figure 4.13). Via the camera vision, it is possible to calculate in centimeters the  $h_{total}$  and  $w_{total}$  and it is respectively equal to 12 cm and 16 cm.

Using a proportion between the value for this camera of 1 cm in pixels, it can give the value in centimeters of the  $h_{total}$  and  $w_{total}$  in pixels. Therefore, we have that:

$$h_{total} = 47.5 \times 12 = 570 \text{ pixels}$$

$$w_{total} = 46.5 \times 16 = 744 \text{ pixels}$$

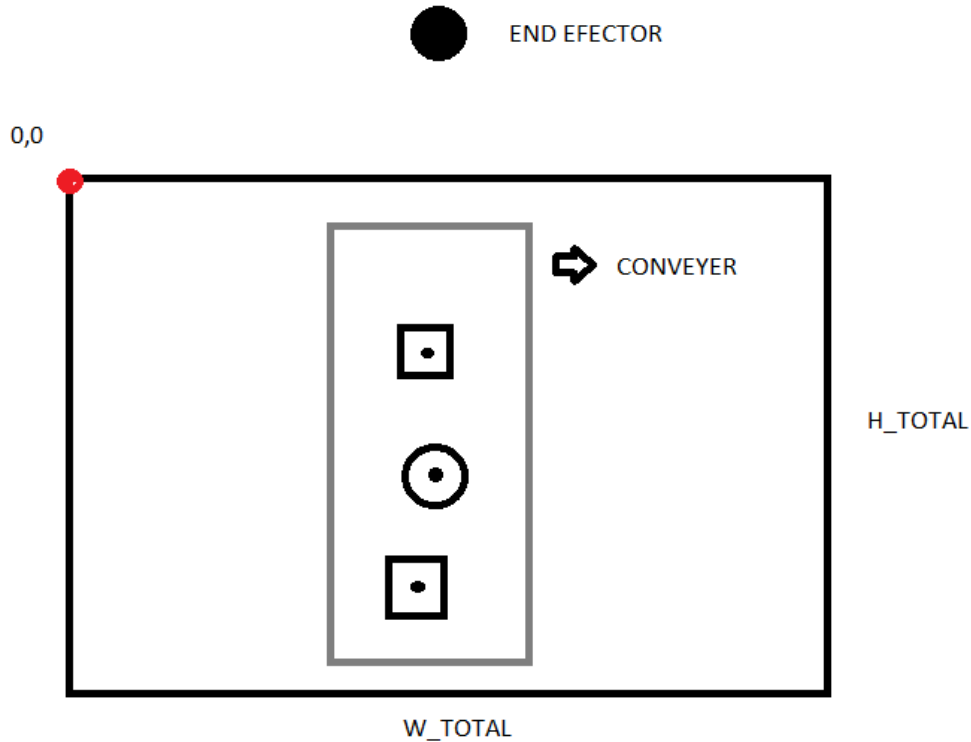


Figure 4.13: Graphical representation of camera calibration

According to the image 4.13, it is possible to conclude some considerations. Firstly, it is possible to see that the point  $(0,0)$  is the beginning point for the camera, so all the parameters for distance are given from this point. Secondly, we can identify that the end-effector of the Delta Robot is not in the start point; hence the calculate centre points for the objects needs to be manipulated.

The values founded for shifting the  $(x,y,z)$  for the same proportion of the beginning point of the camera were founded manually. The method calculated the distance from the end-effector to the beginning point  $(0,0,0)$  and add it to the calculated centre values given by the algorithm. Therefore it is necessary to shift  $x,y$  and  $z$  respectively into  $(-6.5, 15, -1)$  centimeters.

In order to concatenate all the calculated parameters, Table 4.7 was generated. Hence, all the computed values can be founded in this table in order to appropriate calibrate the coordinates of the camera.

Table 4.7: Parameters necessary for camera calibration

<b>Parameter</b>	<b>Value Pixels</b>	<b>Value Centimeters</b>
$h_{object}$	47.5 pixels	1 cm
$w_{object}$	46.5 pixels	1 cm
$h_{total}$	570 pixels	12 cm
$w_{total}$	744 pixels	16 cm

---

## 4.9 Image recognition limitations

This section will discuss image recognition limitations. Even though there are many methods of setup or calibration of the cameras, it is possible there will be still limitations. The restrictions can be presented by the camera, the lighting or even by the vibration.

In order to avoid these limitations, some of the constraints are necessary for this project. These constraints can be founded bellow.

- The objects should have a distance from each other at least 1 cm. If the distance of the objects is lower than 1 centimeter the shape recognition will detect everything as a circle.
- The room where the experiment is running should not be dark or too much light into the objects. If there is low luminosity the camera is not able to identify the colour of the objects. On the other hand, if there is too much luminosity on the objects, they will start to reflect, and the camera will not be able to identify its colour as good as in a stable environment.

#### 4. METHODOLOGY

---

- The colour filtering cannot be 100 per cent precise. There are still some dots that will be showing into the camera due to some error encountered by the colour filtering. However, there are other techniques that could improve this result.
- The camera cannot move. The camera should be stable at all time and also should not move in any circumstances. Otherwise, all the setup that was calculated before need to be re-calculated. Hence, these calculations can be used only for this specific camera in this particular location.

---

## Experiments and Results

This chapter is designed to demonstrate the experiments and results. In this project, the tests and results are shown into the topics of the inverse kinematics and also image recognition.

Firstly it will be presented the results of the inverse kinematic' calculations. It is necessary to show the range in the axis of  $(x,y,z)$  that the delta robot is able to achieve. Moreover, they express the precision of the movement.

Finally, it will be present the results of image recognition. Where it will be analyzing some error in the object detection position and also the quality of the image detected.

### 5.1 Range of the model applying the Inverse Kinematics

First of all, it is necessary to understand the range of motion of the designed delta robot. In order to analyze the envelope area for the movement is essential to remember the constraints of the robot in angles and size.

The angles that the robot is allowed to move is between 5 and 110 degrees.

## 5. EXPERIMENTS AND RESULTS

---

According to these angles, it permits the end-effector to fluctuate within a range of x, y and y maximum and minimum according to the Table 5.1.

Table 5.1: Range of motion depending on angle limitations

Axis	Minimum value	Maximum value
x	-340 mm	340 mm
y	-310 mm	380 mm
z	-490 mm	-120 mm

---

Therefore, it is possible to conclude that the robot has a specific range of motion. Moreover, it is possible that within the values of x,y and z will be some combination that will return a negative angle or even an imaginary number. In order to eliminate the problem, the programming in Arduino remove these numbers and do not move the end-effector to any position.

In the Figure 5.1, it is possible to verify the whole structure of the robot. In this structure there are the delta robot, the conveyor, and the pieces that the robot should pick up. Therefore, it is possible to have a better visualization of the structure used in this project.

A program in Matlab was done in order to map the number of combinations possible according to the axis x,y and z. The constraints that were used for it are described in Table 5.2.

Table 5.2: constraints used in Matlab

Parameter	Minimum value	Maximum value	Increment
x	-0.4 m	0.4 m	0.005
y	-0.4 m	0.4 m	0.01
z	-0.5 m	0 m	0.005
$\Theta$	5 degrees	110 degrees	NONE

---

As a result from the table 5.2 the Matlab software returned a possibility of 111,929.00 combinations. Therefore, it is possible to conclude that even with

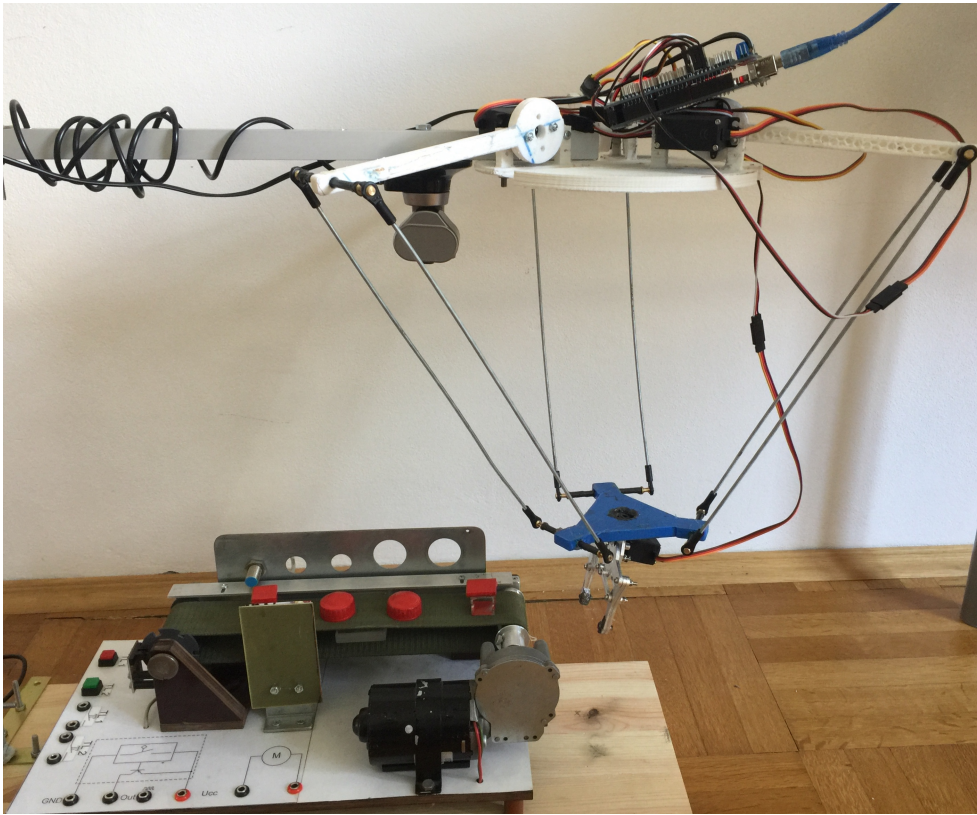


Figure 5.1: Graphical representation of camera calibration

the constraints proposed for this task, there are a large number of possibilities that this robot can achieve.

The Image 5.2 demonstrates a visual range of motion using the constraints established before. As it is possible to analyze, the range of motion represents almost a semi-sphere. This representation occurs due to the angle limitations because there are no negative numbers and also not bigger than 110 degrees.

## 5.2 Precision of Movement

In this section will be discussed about the precision of the movement applied in the delta robot. It is necessary to point out that the inverse kinematic has



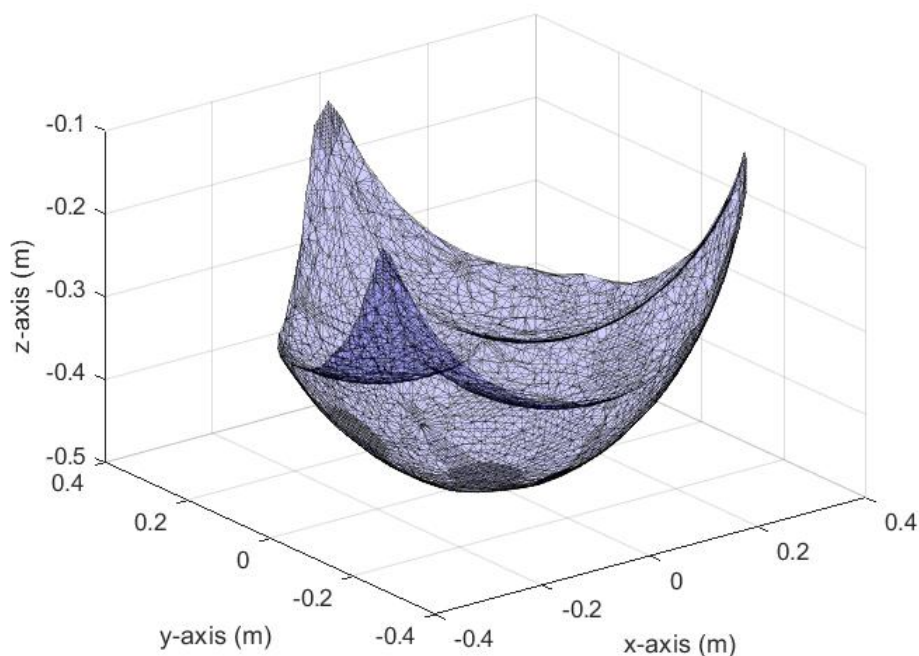


Figure 5.2: Graphical representation of range of motion

a correlation with image recognition. Therefore, it is not possible to analyze one without the other.

Firstly, it is necessary to analyze the precision of the camera. In this case, it was established that the camera has no error. There is an error in the camera, however, they were neglected in order to analyze the situation. So, all the error presented in the position was attributed to the inverse kinematics. Moreover, the inverse kinematics is responsible for the movement in itself and correcting this problem it would have a global impact on the project.

Secondly, an increasing error was identified. Having the objects in different positions, it was possible to recognize that there was an rising error according to the increasing position from the initial coordinates  $(0,0,0)$ .

In order to analyze the position error, it was used the linear regression technique. In this case, a range of positions was selected randomly and then compared with the position of the end-effector. Using the excel, the Equation 5.1 was given.

$$x_{real} = 1.19x + 0.035 \quad (5.1)$$

The linear regression was used just for the axis x. The reason for this is that in the conveyor the position of y can be constant. The main movement will occur for x direction (it is necessary to check the x and y of the project). Furthermore, the axis z is constant for all the movement, so it is not necessary any further modification for it.

In conclusion, in order to have a more suitable approximation to the real position and the calculated position it was used the linear regression. Even though there is a technique to eliminate the final position error, it is still an error depending on the location of the object.

### 5.3 Quality of Image Recognition

In this section will be discussed the quality of image recognition. The quality of image recognition depends on the colour of the object, quality of the camera and lightning of the environment. For this project, the most critical parameters are the object colour and illumination, because the camera was not possible to modify.

The image recognition layout is defined by img and mask. Img is an abbreviation for the image or real image, in this tab shows the vision as the camera can see. Furthermore, in the Img tab, the identified objects have a contour, and its central position is marked with a blue dot (See Figure 5.4) .

## 5. EXPERIMENTS AND RESULTS

---

Moreover, the tab mask is used to treat the image and define the shape and contours of the object.

In order to enhance the testing, all the objects need to have the same colour. In order to approximate as maximum as possible the colours of the objects, all the objects were painted with the same colour by a spray with intensive red colour. This was necessary to have the same filtering of HSV for both environments.

First of all, it is necessary to establish scenarios where the image will be analyzed. In this project, there was used a different shape of objects (circle and rectangle) with the same colour. These objects were identified in two types of environment where one of them had natural lighting, and the second one had a lamp in a dark room.

Firstly, it will be discussed the result of the environment with natural lightning. One problem faced using the natural lighting was the reflection. Even though there was a black background, the camera still were detecting some objects that did not exist. More details can be seen from the Image 5.3.

According to Image 5.3, it is possible to identify some errors. Firstly, the camera is identifying more objects. The blue dots represent the centre of an object, so the camera is identifying around seven objects and there were just four. Furthermore, the centre of the objects that should be identified was neglected by the camera. Using this image, it would not be possible to collect an object precisely.

Another analyze can be done using the same objects, however in a dark room with controlled lights. Using the controlled lights is a method of how to control the environment and therefore minimize the possible error on image recognition. Using this method, the results are presented in the Image 5.4

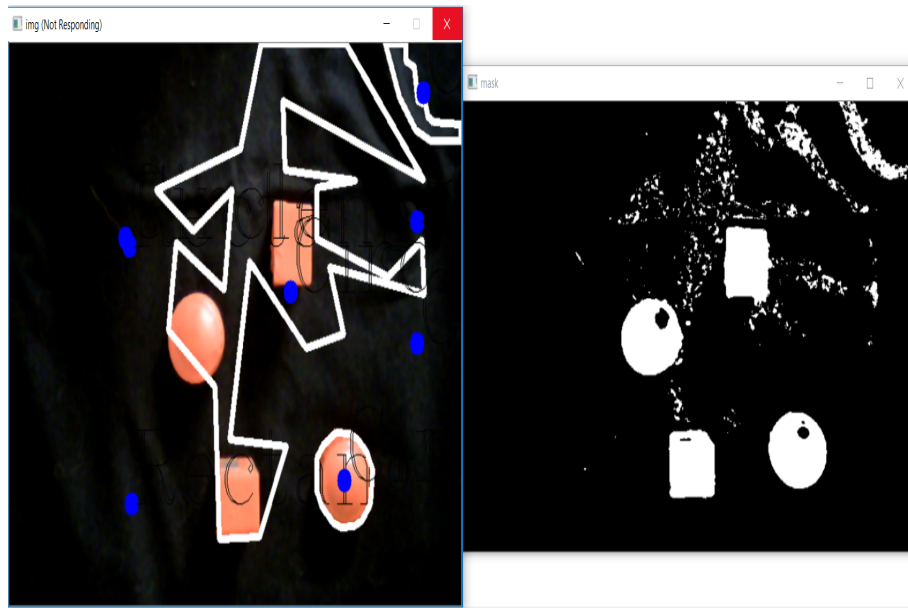


Figure 5.3: Representation of Image and Mask in Natural Lighting

were it is possible to analyze image recognition in the dark room.

Analyzing Figure 5.4, using a controlled environment, the results were precise. Firstly, all the objects exposed to the camera were identified correctly, not just its center, but also its type (circle or rectangle). Also, in a dark room, the reflection is low. Therefore, there are not identified objects that do not exist.

In conclusion, the controlled environment produces a better result for image identification than natural lighting. Those results are more suitable due to proper identification of the number of objects without error. Moreover, the dark room with controlled light identify the shape of the objects (circle or rectangle) correctly.

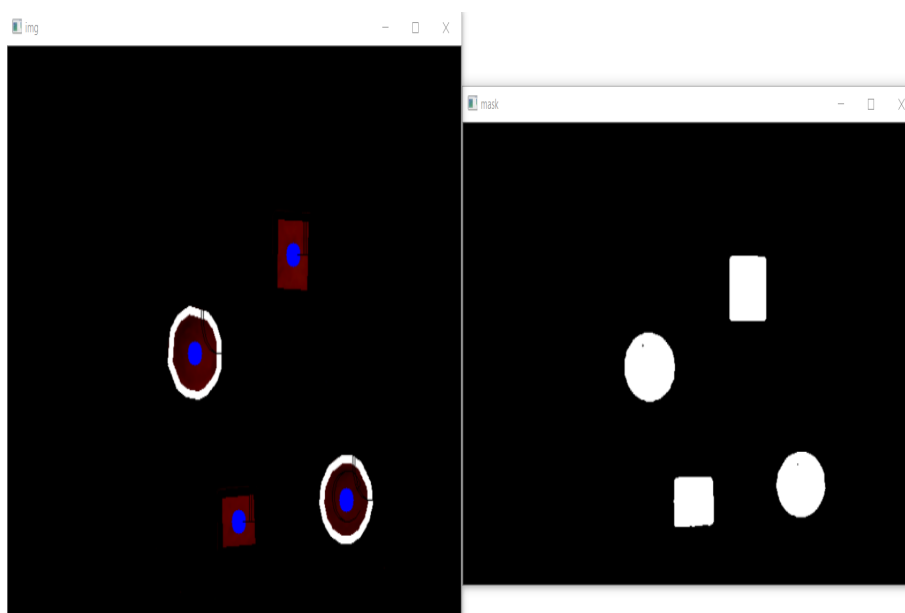


Figure 5.4: Representation of Image and Mask in Dark Room

## 5.4 Communication between the Arduino and Camera

This section will discuss the communication between the Arduino and the camera. Establishing this communication, it is possible to allow that the Arduino access the data from the camera and receive as an input the coordinates necessary to achieve the object.

In order to establish the connection between these two devices the user should always use the same USB ports. Using the same USB ports, there is no need for changing in the programming algorithm. Also, it is necessary to start the serial port a set in both algorithms with the same baud rates.

Firstly, there is two software to establish the communication that is python and Arduino IDE. The python is responsible for recognizing the object shape and also to identify its locations. On the other hand, the Arduino IDE receives

the location and send the end-effector to that position. In order to run the program, the first algorithm that should be running is the Arduino and then start the python.

The communication is fully established, however, there are still some issues in the results. One of the problems faced is that after restarting the program several times, it is necessary to unplug the camera from its USB port.

Moreover, the camera image does not have a constant update. It is necessary to complete the whole movement before the camera updates its image. Hence, it is just possible to know what the camera can visualize after the movement.

---

## Conclusion

In conclusion, this project had the main target of design a delta robot and apply the inverse kinematics associated with image recognition. First of all, it was necessary to design the robot according to the needs of the project. Secondly, according to the structure the inverse kinematics was designed for this robot. Finally, with the help of openCV and python it was possible to implement image recognition.

Firstly, the robot was designed using the software Autodesk Inventor to produce some of the parts. The manufactured parts were end-effector, arms and the fixed based. Those parts were fabricated by 3D printing in order to allow a low weight to the structure. Furthermore, the forearms were made by a steel rod of 3 mm.

Moreover, using the design of the robot, the inverse kinematic was applied in the project. It was possible to identify that there are four main elements that can be controlled in order to impact the inverse kinematics. The elements are size of arms, forearms, end-effector and fixed base. Also, the Arduino IDE was used to control the constraints of angles and positions as a result of the inverse kinematics.

---

Furthermore, image recognition is responsible for identifying the object shape and its location. Using the openCV and Python software, it was possible to approximate the shape of the objects and determine the object's central position. Calculating the central position of the object to the end-effector, allow to know the coordinates  $(x,y,z)$  that the algorithm should send to calculate the angles.

Although the project has succeeded, there are still some issues that should be discussed. Firstly, the quality of image recognition depends on the lighting, so the most suitable environment is the one that the illumination can be controlled. Furthermore, the inverse kinematic has a progressive error. Therefore, as further the end-effector goes to its initial position, higher is the error until the object.

Finally, in further researches, some improvements in the project could be made. The glue was used in some parts of the structure, avoiding the glue the robot would be able to be more stable. However, it is necessary to produce another design for it. Furthermore, instead of using a servo motor, it could be changed for a step motor. As a result of the motor changes, it would be possible to achieve a higher range of motion. Finally, further investigation could be done in order to increase the precision of picking the objects.



---

## Bibliography

- [1] Kong, X.; Gosselin, C. M. *Type synthesis of parallel mechanisms*, volume 33. Springer, 2007.
- [2] Rivas-Lalaleo, D. R.; Galarza-Zambrano, E. E.; Tumbaco-Mendoza, D. C.; et al. Delta robot controlled by robotic operating system. *Iteckne*, volume 12, no. 1, 2015: pp. 54–59.
- [3] Lin, J.; Luo, C.-H.; Lin, K.-H. Design and implementation of a new delta parallel robot in robotics competitions. *International Journal of Advanced Robotic Systems*, volume 12, no. 10, 2015: p. 153.
- [4] Guangfeng, C.; Linlin, Z.; Qingqing, H.; et al. Trajectory planning of Delta robot for fixed point pick and placement. In *2012 Fourth International Symposium on Information Science and Engineering*, IEEE, 2012, pp. 236–239.
- [5] Yang Pingxian; Guo Rong; Guo Peng; et al. Research on lip detection based on Opencv. In *Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*, Dec 2011, pp. 1465–1468, doi:10.1109/TMEE.2011.6199484.

- [6] Tarr, M. J.; Bühlhoff, H. H. Image-based object recognition in man, monkey and machine. *Cognition*, volume 67, no. 1-2, 1998: pp. 1–20.
- [7] Russell, M.; Fischhaber, S. OpenCV based road sign recognition on Zynq. In *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, July 2013, ISSN 1935-4576, pp. 596–601, doi:10.1109/INDIN.2013.6622951.
- [8] Huang, T.; Wang, P.; Mei, J.; et al. Time minimum trajectory planning of a 2-DOF translational parallel robot for pick-and-place operations. *CIRP annals*, volume 56, no. 1, 2007: pp. 365–368.
- [9] Nabat, V.; de la O RODRIGUEZ, M.; Company, O.; et al. Par4: very high speed parallel robot for pick-and-place. In *2005 IEEE/RSJ International Conference on intelligent robots and systems*, IEEE, 2005, pp. 553–558.
- [10] Pierrot, F.; Reynaud, C.; Fournier, A. DELTA: a simple and efficient parallel robot. *Robotica*, volume 8, no. 2, 1990: pp. 105–109.
- [11] Lin, C.-J.; Shaw, J.; Tsou, P.-C.; et al. Vision servo based delta robot to pick-and-place moving parts. In *2016 IEEE International Conference on Industrial Technology (ICIT)*, IEEE, 2016, pp. 1626–1631.
- [12] Vischer, P.; Clavel, R. Kinematic calibration of the parallel Delta robot. *Robotica*, volume 16, no. 2, 1998: pp. 207–218.
- [13] Bonev, I. Delta parallel robot—the story of success. *Newsletter*, available at <http://www.parallelmic.org>, 2001.
- [14] Patel, Y.; George, P. Parallel manipulators applications—a survey. *Modern Mechanical Engineering*, volume 2, no. 03, 2012: p. 57.
- [15] McRoberts, M. *Arduino básico*. São Paulo: Novatec, 2011.
- [16] DI RENNA, R. B.; BRASIL, R. D.; CUNHA, T. E.; et al. *Introdução ao kit de desenvolvimento Arduino*. 2013.

## BIBLIOGRAPHY

---

- [17] Buss, S. R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, volume 17, no. 1-19, 2004: p. 16.
- [18] Robot, A. F. D. The Delta Parallel Robot: Kinematics Solutions Robert L. Williams II, Ph. D., williar4@ohio.edu Mechanical Engineering, Ohio University, October 2016.
- [19] Brahmabhatt, S. *Practical OpenCV*. Apress, 2013.
- [20] LeCun, Y.; Boser, B. E.; Denker, J. S.; et al. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, 1990, pp. 396–404.

## Symbols and Acronyms

**IDE** Integrated Development Environment

**PWM** Pulse Width Modulation

**DOF** Degrees of freedom

$\Theta$  Angular position

**L** Length of the arm

**l** Length of the forearm

**BRG** Blue Red Green

**HSV** Hue Saturation Value

**mm** Millimeters

---

## Contents of CD

Visualize the contents of enclosed media. In the CD will contain the following elements:

```
readme.txt ..... the file with CD contents description
├── programming ..... the data files directory
│   ├── *.py ..... the Python files
│   ├── *.m ..... the Matlab files
│   └── *.ino ..... the Arduino files
├── text ..... the thesis text directory
│   └── thesis.pdf ..... the Diploma thesis in PDF format
└── Design ..... the design files directory
    └── *.ipt ..... the Inventor files
```