



Designing, Building and Controlling of
**Designing, Building and Controlling of
the 3-DoF Delta Robot**

By

Humam Abu-Alkebash

Omar Abu-Alfeilat

Sufyan Alsharawe

Supervisor:

Dr. Iyad Hashlamon

Submitted to the College of Engineering
in partial fulfillment of the requirements for the degree of
Bachelor degree in Mechatronics Engineering

Palestine Polytechnic University

May, 2015

29354



Abstract

Parallel robots have inherent advantages for many applications in the fields of robotics. A 3 DoF Delta parallel robot designed and built, which consists two platforms, one is fixed and the other is mobile. The movements are transmitted to the mobile platform through the closed chain loops. The used actuators are fixed to the base. The three kinematic chains consist of group of links connected to each other by ball bearing joints. Forward and inverse kinematic analysis is used to study the motion of this robot. A discrete tracker controller is implemented to achieve the accuracy, speed and acceleration.

Table of Contents:

Abstract	<i>iii</i>
List of Figures	<i>vi</i>
List of Tables	<i>viii</i>
1. Chapter 1:Introduction	1
1.1.Introduction.....	1
1.1.1. Parallel Robots.....	3
1.1.2. Serial Robots.....	5
1.1.3. Delta Robot.....	6
1.2.Previous Studies.....	7
1.3.Motivation.....	8
1.4.Problem Definition.....	8
1.5.Time Table.....	10
1.6.The Report Outline.....	11
2. Chapter 2 : Delta Robot	12
2.1.Delta Robot.....	12
2.2.Delta Robot as a commercial product.....	15
2.3.Applications.....	18
2.4.Conceptual Design.....	18
3. Chapter 3 : Components Selection	20
3.1.Introduction	20
3.2.Micro-Controller	20
3.3.Position sensor.....	21
3.4.Three axes accelerometer	23
3.5.Current sensor.....	23
3.6.DC servo actuator.....	24
3.7.Motor's Driver.....	26
3.8.Mechanical Design.....	27
3.9.Budget.....	31

4. Chapter 4 : Kinematics and Dynamics.....	32
4.1. Inverse Kinematics.....	33
4.2. Forward Kinematics	40
4.3. Velocity Kinematics.....	44
4.4. Forward and Inverse Singularity analysis.....	49
4.5. Dynamic model.....	50
4.5.1 Virtual Work Dynamics.....	51
4.5.2 Non-Rigid Body Effects.....	54
5. Chapter 5 : Controller Design	55
5.1 Introduction	55
5.2 System equation	55
5.3 Transfer function of the actuator.....	56
5.4 Design a Discrete Tracker controller.....	59
5.5 Real Results.....	65
6. Chapter 6 : Software Program	73
6.1. The program	73
7. Chapter 7 : Assembly	76
7.1 Introduction.....	76
7.2 Assembly for the mechanical parts.....	76
7.3 Assembly for the electrical parts.....	79
7.4 Schematic wire diagram for the electrical components.....	81
7.5 Workspace analysis.....	82
8. Chapter 8 : Conclusion and Future work	85
8.1 Conclusion.....	85
8.2 Future Work.....	85
References.....	86

List of Figures:

Figure 1.1:Robots used in industry.....	2
Figure 1.2: A parallel manipulator.....	3
Figure 1.3:Hexapod Parallel Robot.....	4
Figure 1.4: Serial Robot.....	5
Figure 1.5:CODIAN Robotics Parallel robot.....	6
Figure 1.6:Delta robot developed at the University of Western Australia.....	8
Figure 1.7: Delta Robot.....	9
Figure 2.1: The Clavel's Delta Robot.....	12
Figure 2.2: Delta Robot.....	14
Figure 2.3:(a) Rostock (delta robot 3D printer),(b) Customized MOST Delta RepRap.....	15
Figure 2.4: Demareux's Line-Placer installation for the packaging of pretzels in an industrial..	16
Figure 2.5:Two of the three Delta robot models offered by SIG Pack Systems,C33 and CE33..	16
Figure 2.6:Hitachi Seiki's Delta robots for pick-and-place and drilling.....	17
Figure 2.7:ABB Flexible Automation's IRB 340 FlexPicker.....	17
Figure 2.8:Delta robot used to place lenses into led array.....	18
Figure 2.9 : Block diagram of the system.....	19
Figure 3.1: Arduino MEGA ADK.....	20
Figure 3.2: Rotary encoder.....	21
Figure 3.3: Incremental and absolute rotary encoder.....	22
Figure 3.4: The Initial position of Delta Robot.....	22
Figure 3.5: ADXL345 accelerometer.....	23
Figure 3.6: ACS712current sensor.....	23
Figure 3.7: RH mini-series DC servo Actuators.....	24
Figure 3.8: I 298N Motor Driver Module.....	27
Figure 3.9: The base.....	27
Figure 3.10: The structure.....	28
Figure 3.11: Upper link.....	28
Figure 3.12: Lower link.....	29
Figure 3.13: End-effector.....	29
Figure 3.14: Ball joint bearing.....	30
Figure 3.15: Delta robot after assembly.....	30
Figure 4.1 : Delta Robot.....	32
Figure 4.2 : equilateral triangle on the fixed platform ,fixed and spherical joint in delta Robot.....	34
Figure 4.3 : Tow circles act in tow point.....	35
Figure 4.4 : Equilateral triangle on the Task space , position of $E1'$	36
Figure 4.5 : Shown the position of $J1'$, $J2'$ and $J3'$ after transition.....	40
Figure 4.6 : Three spheres will intersect in the center of task space.....	41
Figure 4.7 : Upper section shows the center of task space after transition and equilateral triangle.....	42

Figure 4.8: Projection of link i on x, z_i plane.....	45
Figure 4.9: y, z_i plane.....	45
Figure 5.1: The electric equivalent circuit of the armature and the free-body diagram of the rotor.....	56
Figure 5.2: State space representation of the plant.....	59
Figure 5.3: Plant with state-variable feedback.....	60
Figure 5.4: Discrete Tracker controller.....	61
Figure 5.5: Simulation for the position response before designing a controller.....	62
Figure 5.6 : Simulation position response of the end-effector.....	63
Figure 5.7 : Torque response of the actuator.....	63
Figure 5.8 : Simulation speed response.....	64
Figure 5.9: Real position response of the end-effector.....	65
Figure 5.10: Real response of the speed.....	66
Figure 5.11 : Real position response of the end-effector in x-axis.....	67
Figure 5.12 : Real speed response of the end-effector in x-axis.....	67
Figure 5.13 : Real position response of the end-effector in y-axis.....	68
Figure 5.14 : Real speed response of the end-effector in y-axis.....	68
Figure 5.15 : Real position response of the end-effector in z-axis.....	69
Figure 5.16 : Real speed response of the end-effector.....	69
Figure 5.17 : Real position response of the end-effector in x-axis.....	70
Figure 5.18 : Real position response of the end-effector in y-axis.....	70
Figure 5.19 : Real position response of the end-effector in x-axis.....	71
Figure 5.20 : Real position response of the end-effector in y-axis.....	71
Figure 5.21 : Real position response of the end-effector in z-axis.....	72
Figure 6.1: The communication between user and Delta robot.....	73
Figure 6.2: The interface window of the program.....	74
Figure 7.1: The base fixed on the steel structure.....	
Figure 7.2: The actuators are fixed on the base.....	76
Figure 7.3: The mechanical parts of the robot be assembled together.....	77
Figure 7.4 : The mechanical parts of delta robot be assembled with the actuators.....	77
Figure 7.5: The connection of the microcontroller and drivers with the actuators and encoders.....	78
Figure 7.6: The labels are added and the current sensors connection.....	79
Figure 7.7: The plastic cover of the electrical components.....	80
Figure 7.8: The wiring diagram for electrical components.....	81
Figure 7.9: Minimum distance along z-axis.....	82
Figure 7.10: Maximum distance along z-axis.....	82
Figure 7.11: The workspace of the Delta Robot.....	83

List of Tables:

Table 1.1: The advantages and disadvantages of the parallel robot.....	4
Table 1.2: the advantages and disadvantages of serial robots.....	5
Table 3.1: Technical specification of ADC Microcontroller.....	20
Table 3.2: Some technical data about the actuators.....	23
Table 3.3: Budget.....	30

CHAPTER 1

Introduction

Robots play a very important role in the industry. Robots are classified into Parallel and Serial Robots, Delta Robot is a parallel robot which contains three closed kinematic chains. A summary of the differences between serial and parallel robot structures, related work, then the motivation of the project and problem definition are explored in this chapter.

1.1 Introduction

Robots today are widely used in many areas such as factories, universities, schools, hospitals, homes, underwater, in the space and many other places. Robots play a very important role in numerous applications in the industrial field as shown in Figure 1.1 as an example. They are employed in metal production, packaging, transportation and electronics industries. The field of robotics is a significant part of modern engineering that interacts with other fields such as electric & electronics engineering, computer engineering, mathematical modeling and mechanism design.

Robots are designed to do repetitive tasks to increase the productivity, accuracy, efficiency, speed and acceleration. They save both time and cost, further robots take the people far away from the harm environments, Further, they visited the space for information recording and transformation[1, 2].



Figure 1.1: Robots used in industry.

According to the International Organization for Standardization (standard ISO 8373:2012), a definition of a robot is: "An actuated mechanism programmable in two or more axes with a degree of autonomy, moving within its environment, to perform intended tasks".

Hence the robot has the following basic elements:

- **Links:** Rigid elements connected with each other by joints.
- **Actuators:** Elements responsible for giving the capability of movement to the robot and driving the effector to the desired point. The actuator is just a part of a complete drive system. Several types of actuators are available like pistons and electric motors. The actuator can be connected to a drive system to transfer torque or force such as bully-belt and harmonic drive systems. The choice of actuator type in a design depends on :
 - a. Load process.
 - b. Available volume.
 - c. Whole construction.
 - d. Surrounding.
 - e. Costs.
- **Controller:** Component, hardware or software, which is used to supply the plant or actuator with the suitable command signal based on the desired and actual conditions of the process.
- **End Effector:** It refers to the device at the end of a robotic arm as seen in Figure 1.2. It can move in deferent ways, translation or rotary motion which are called its degrees of freedom (DOF).

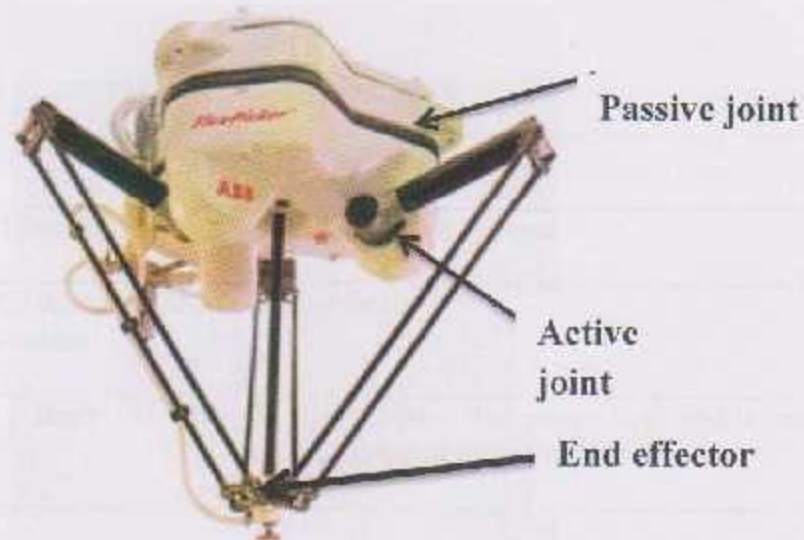


Figure 1.2: A parallel manipulator.

- **Sensors:** They are used to sense the change of a physical phenomenon.
- **Joints:** Connect two links or more together. They are two types: active joints which are driven by actuators and passive joints which are not connected to actuators [3].

Industrial robots play an important role in the industrial automation in terms of manufacturing, assembly and material handling which need a fast and agile robot with a light moving parts. The robotic systems are classified into two typical groups (serial and parallel robots).

1.1.1. Parallel Robots

Also called parallel manipulators, are closed-loop mechanisms, characterized by their high accuracy, velocity, rigidity and ability to move large loads. Due to these advantages (as in Table 1), they have been used in a large number of applications and are becoming increasingly popular in the field of machine-tool industry. Parallel robots are specifically designed for high-speed applications in packaging, manufacturing, assembly, and material handling[4].

An n degrees of freedom parallel robot consists of an end-effector and a fixed base. They are linked together by more than one independent kinematic chain. Normally, each kinematic chain has a series of links connected to each other by joints. Some of these joints are active (connected to an actuator) whereas the others are passive. Actuation takes place through n actuators. In parallel robots, the number of actuators are the same as the number of degrees of freedom of the end-effector. The actuators in parallel robots are fixed to the base, which makes the moving parts relatively light [4].

Table 1.1: The advantages and disadvantages of the parallel robot.

Advantages	Disadvantages
1-The parallel robot provides high rigidity.	1-Limited work volume.
2- High payload-to-weight ratio since the load is shared by several kinematic chains.	2-Low dexterity.
3-High accuracy because there is no accumulation of joint errors.	3- Singularities that occur both inside and on the envelope of the work volume.
4-Low inertia of moving parts.	
5-High agility.	

There are many variant designs of parallel robots. Stewart platform or Hexapod Parallel Robot shown in Figure 1.3 is an example of parallel robot, this robot is used in aircraft motion simulators. Further the Delta robot, which is used in packaging plants [2].

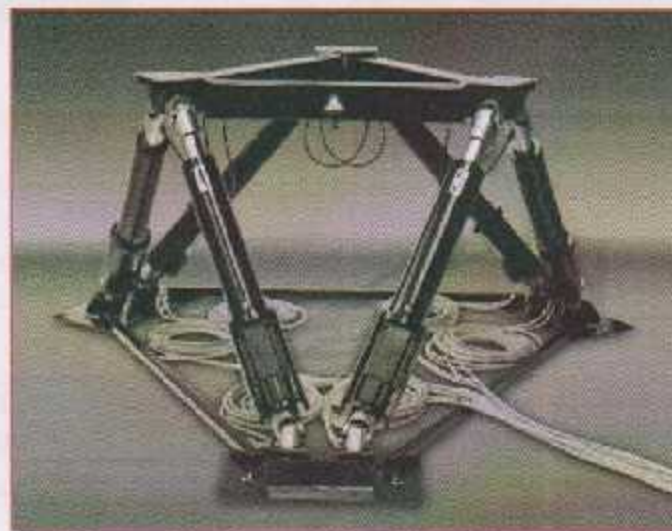


Figure 1.3: Hexapod Parallel Robot.

In general, for high speed and agility, the links of the parallel robot are made of lightweight materials. They have a robust construction and they can move bodies of considerable masses and dimensions with high speeds [5, 6].

1.1.2. Serial Robots

A serial robot has a single open loop kinematic chain, which consists of a group of links connected together in series by joints as shown in Figure 1.4 , each joint has one actuator or more, so all joints are considered as active joints. The end effector is connected to the base by a single kinematic chain [6].



Figure 1.4: Serial Robots.

Serial robots need to be agile enough in order to perform the requested task. The serial robot must be strong enough to perform its tasks hundreds of thousands of times. These robots are used in the industrial systems and have the advantages as in Table 1.2.

Table 1.2: the advantages and disadvantages of serial robots.

Advantages	Disadvantages
1-A large work volume (large workspace with respect to its size of the robot)	1-Low precision: stems from cumulative joint errors and deflections in the links.
2-High dexterity.	2-Poor force.
3-Occupy small floor space.	3-Exertion capability.
	4-The relatively low effective load that they can be manipulated.
	5-large number of moving parts which have to be carried and moved, leading to high inertia.

The low precision and payload-to-weight ratio lead to expensive serial robots utilizing extremely accurate gears and powerful motors. The high inertia disadvantage prevents the use of serial robots for applications requiring high accelerations and agility, such as flight simulation and very fast pick and place tasks[6].

1.1.3. Delta Robot

Delta robot is a parallel robot with three kinematic closed chains as in Figure 1.5. It has two platforms, one is called the base where the motors are attached to, the other one is called the end effector. These two platforms are connected with each other by three kinematic chains which have group of links connected to each other by joints. Delta configuration allows for a distribution of payload among their three closed-loop kinematic chains.



Figure 1.5: CODIAN Robotics Parallel robot.

Three motors will be used in the delta robot with the drivers. These motors will move the active joints in order to move the end effector to specific positions. Further, a controller will be used to ensure that the end effector will move in a specific required speed and acceleration, and to specific position.

1.2. Previous Studies

Kinematic and dynamic analysis of a Delta parallel robot and an inverse dynamic problem is solved by using the virtual powers method as represented in [7]. Discussion of a number of performance criteria dealing with workspace and a discussion of the performance analysis of delta robot are presented in [8]. Studying of a parallel delta robot controlled via iPhone accelerometer, from kinematics theory to a practical implementation is presented in [9]. Describing the kinematics and dynamics of Delta parallel robot with 3 DoF is presented in [10]. A brief review of the kinematic analysis, the singularity analysis are presented in [11]. Established some recursive matrix relations used for positional, kinematic and dynamic analysis for a three degree of freedom Delta robot as represented in [7]. Introduced the performance, the dimensioning of the 3 DOF DELTA parallel robot and presenting the performance evaluation results are presented in [12]. Introduce two different kinematic calibration models of 3-DoF Delta parallel robot is presented in [13]. Modelling and index analysis of a Delta-type mechanism, the forward and inverse kinematics are derived to be used for performance analysis which presented in [14].

The Delta robot design has attracted great interest not only in the industry but also in university laboratories. Academically, the importance of the studies related to these robots increased as well. Delta robot was built at the University of Maryland. Another, was constructed at the University of Genoa. Three linear-motor versions were also constructed at Ferdinand-von-Steinbeis Schule, ETH Zurich, and the University of Stuttgart. Another linear version and two conventional Delta robots were fabricated at the University of Michigan. Delta robot, as part of a decoupled 6-DOF haptic interface, was implemented at Tohoku University. And an even more original application of the Delta robot, as a vegetable transplanting device, was implemented at Dalhousie University[15]. Yet another highly-optimized version, NUWAR, was built at the University of Western Australia shown in the Figure 1.6. [5, 16].



Figure 1.6: Delta robot developed at the University of Western Australia.

1.3. Motivation

In assembly, packaging and handling materials process, a high speed and high accuracy motions are essential. In order to achieve this, parallel robots are designed with high agility. Delta robot is built with light weight links in order to achieve a higher load capacity due to the decrease in the mass of the overall system [2] .

Introducing this new technology in domestic market has a big role, in developing and improving the products, in order to increase the quality of these products in addition to the productivity.

Future research will be conducted for disturbance rejection and trajectory planning of the end-effector. Different controller can be built for controlling this robot by the students at the PPU university.

Workspace analysis is required in the design of the parallel robot. It is necessary to make sure that delta parallel robot has a good workspace volume. In addition, a PID controller will be designed to achieve the required specifications and to improve the response of the delta robot.

As the device introduces a new technology for parallel robot in Palestine, it should satisfy some features and requirements.

1. The device must be safe.
2. The device must include a simple user interface.
3. The device must be convenient for objects with different sizes.
4. Suitable for industrial environment.
5. The device has high speed and acceleration.
6. The device have to be robust.
7. The device must have high accuracy and good efficiency.

1.4. Problem Definition

In order to build the delta robot there is a need to solve two problems.

First one: Detecting the position of the end effector will be given by the x, y and z coordinates(for the desired positions of the end effector. For example, to move the end-effector to specific point with coordinates X,Y and Z) , we need to determine the corresponding angles, of each one of the motors, because the motors take θ_1 , θ_2 and θ_3 (motors' angles), to set the end effector in proper position for picking. The process is known as inverse kinematics.

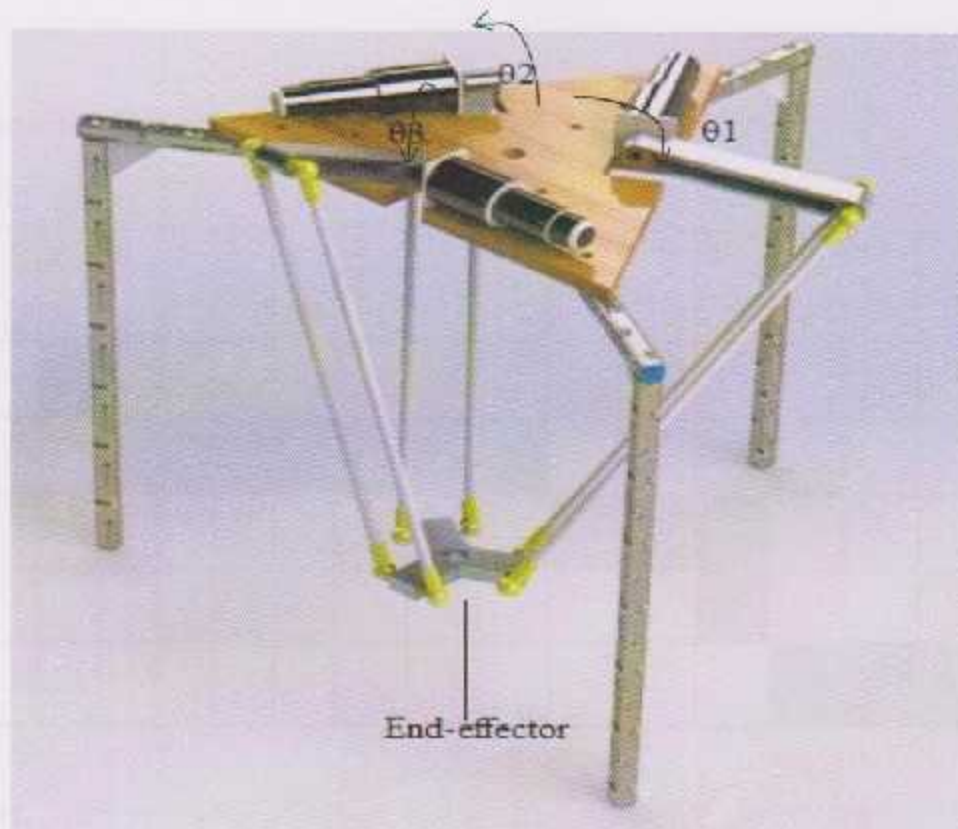


Figure 1.7: Delta Robot.

Secondly by knowing the angles of the motors' joints, the position of the end effector is to be determined, this is done by using forward kinematics. Here each motor can be actuated alone or all the three actuators be actuated, we can operate each actuator independently because here the angle of each actuator is entered as the user want. Both forward and inverse kinematic algorithms have been developed, which are essential for the motion planning and control of a parallel robot, see Figure 1.7.

1.5. Time Table

Week \ Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Task 1																
Task 2																
Task 3																
Task 4																
Task 5																

First semester:

Task 1: Choosing the idea of the project.

Task 2: Collecting the information that related to the project idea / Documentation.

Task 3: Design the robot structure (by SolidWorks).

Task 4: Building a rapid prototype (by wood).

Task 5: Deriving the kinematic equations.

Second semester:

Week \ Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Task 6																
Task 7																
Task 8																
Task 9																
Task 10																

Task 6: Selecting the components and ordering them.

Task 7: Writing the controlling code for microcontroller.

Task8: Designing a controller and simulation.

Task 9: Building the robot.

Task 10: Doing experiments and documentation.

1.6 The Report Outline

Chapter 2 presents the explanation of the Delta Robot, its work, components. Short introduction about a different designs of Delta Robot. It provides some applications of Delta Robot

Chapter 3 presents the components that are used in this project with the functional specifications for each subsystem of the device and the budget of the project.

Chapter 4 presents the Forward and Inverse Kinematics of the Delta Robot.

Chapter 5 presents the controller design for the system.

Chapter 6 presents the software (Graphical User Interface) used to communicate with microcontroller to operate the Delta robot.

Chapter 7 present the assembly operation for the mechanical and electrical components

CHAPTER 2

Delta Robot

This chapter introduces an explanation about the Delta Robot. There are different designs of Delta Robot, some are three DoF and other are four or six DoF. Also, The main components of Delta robot and how it works, it's usage in industry for packaging and manipulating small objects also is mentioned on this chapter, including some commercial Delta Robots.

2.1. Delta Robot

The delta robot (a parallel arm robot) was invented in the early 1980s by a research team led by professor Reymond Clavel at the École Polytechnique Fédérale de Lausanne (EPFL, Switzerland) as shown in Figure 2.1. The purpose of this new type of robot was to manipulate light and small objects at very high speed and acceleration, an industrial need at that time. The motors in parallel robots are fixed to the base, this makes the mobile part relatively light. This has made Delta robots the fastest robots that have been designed for years now [4, 7, 17].

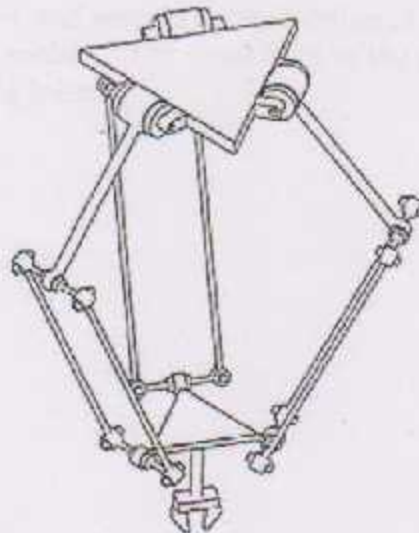


Figure 2.1: The Clavel's Delta Robot.

Delta robot can be high speed robot (dealing with lightweight (up to 1 Kg)) or robot to handle heavy objects. Both have low inertia structure because the motors are fixed to the base and the links are made of light weight material. Delta robot is suitable for high speed pick-and-place applications involving lightweight parts [5].

The main advantage of delta robots is its high speed. The only moving part of delta robot is its links, which are usually made of lightweight composite materials in order to have low inertia. Further, delta robots have high agility, accuracy and acceleration which make the delta robot widely used in industry[5].

Delta robot has two platforms as shown in Figure 2.2, the upper platform (the base) with three motors mounted on it, and the lower moving platform which is the end effector. The motors set the position of the arms and so the XYZ-position of the end effector to the desired position.

The Delta robot consists of three closed chain loops. Each chain consists of groups of links, connected together by revolute joints. A moving platform is connected to a fixed base through three parallel kinematic chains. Each chain contains ball bearing joints. Each chain is activated by an actuator fixed on the base platform. The three actuators are rigidly mounted on the base plate with 120° in between. The movements are transmitted to the mobile platform through the closed chain loops as shown in Figure 2.2.

The main design feature is the usage of parallelograms in the arms, which always maintains the orientation of the end effector to be parallel to the base [2]. Using three parallelograms make the orientation of the moving end effector to be only with purely translational degrees of freedom, and restrain the orientation of the lower end effector in order to keep it parallel to the working surface. The input links of the three parallelograms are mounted on rotating links via ball bearing joints.

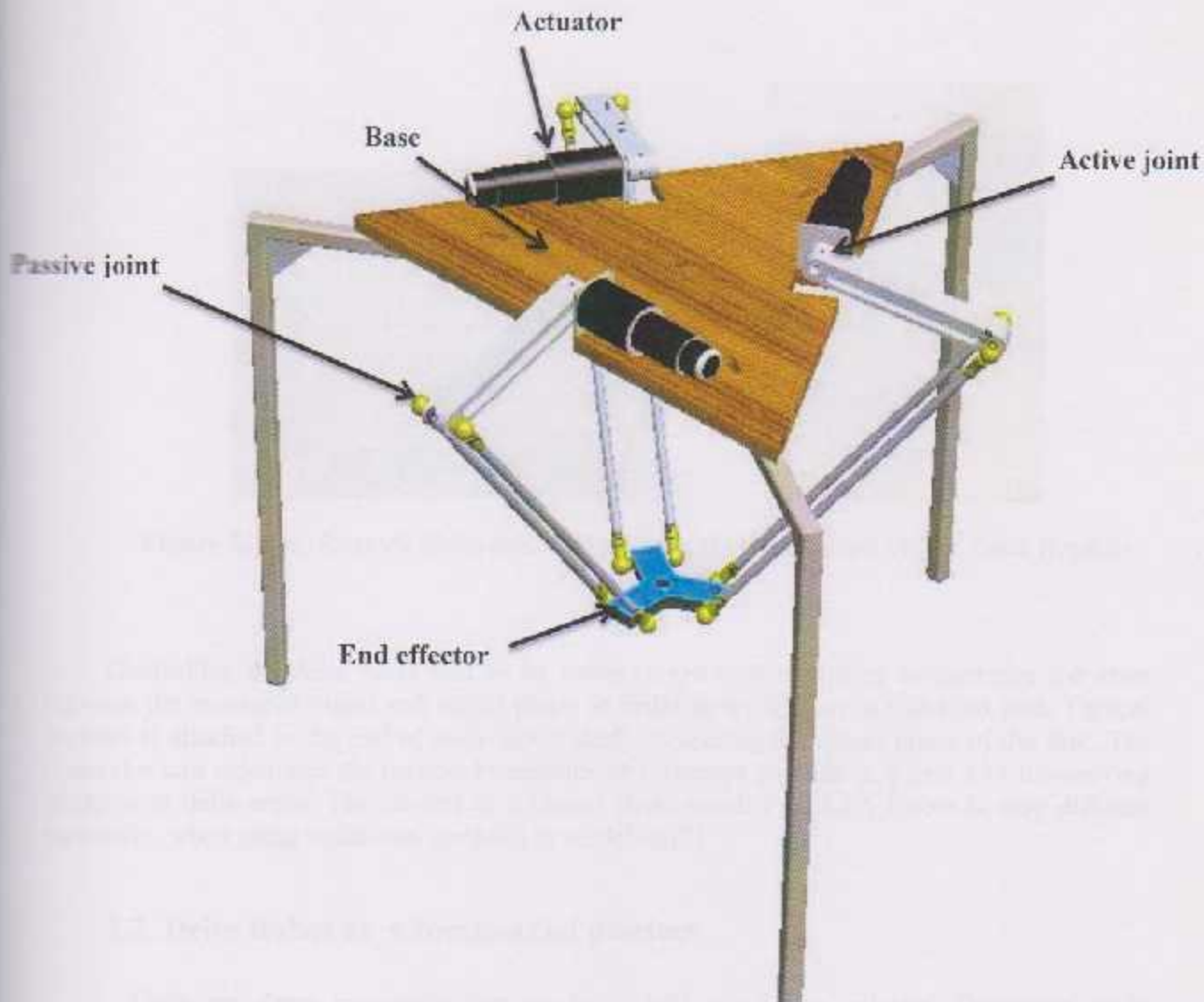


Figure 2.2: Delta Robot.

The input links of the three parallelograms are mounted on rotating levers via revolute joints. The revolute joints of the rotating levers are actuated in two different ways: with rotational (DC or AC servo) motors or with linear actuators. The majority of delta robots use rotary actuators. Recently, vertical linear actuators have been used (using a linear delta design) to produce a novel design of 3D printer as shown in Figure 2.3 [18].



Figure 2.3: (a) Rostock (delta robot 3D printer), (b) Customized MOST Delta RepRap.

Controlling the delta robot will be by using closed loop controller to minimize the error between the measured phase and actual phase in order to track such a specified path. Optical encoder is attached to the end of each motor shaft, measuring the actual phase of the link. The controller unit calculates the inverse kinematics of reference position x , y and z of the moving platform in delta robot. The control of a closed chain parallel DELTA Robot is very difficult especially, when using traditional methods in modeling[2].

2.2. Delta Robot as a commercial product

There are many companies that produce Delta robots for industry. Demarex made several versions under the names of *Pack-Placer*, *Line-Placer*, *Top-Placer*, and *Presto*, shown in Figure 2.4. Japanese company Hitachi Seiki manufactured Delta robots of small dimensions for packaging and for drilling (*PA35*), as shown in Figure 2.5. ABB Flexible Automation made its Delta robot under the name *IRB 340 FlexPicker*, as shown in Figure 2.6. Delta robot is offered by SIG Pack Systems, the *CE33*, as shown in Figure 2.7. while the *C23* and *C33* are manufactured by Demarex.

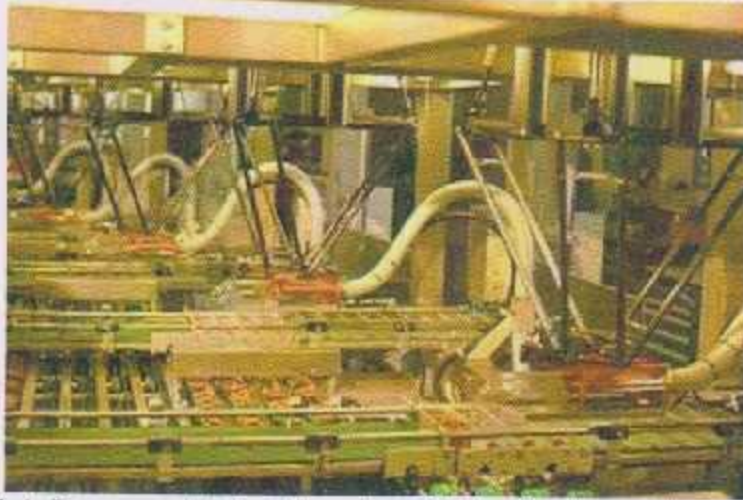


Figure 2.4: Demarex's Line-Placer installation for the packaging of pretzels in an industrial.



Figure 2.5: Two of the three Delta robot models offered by SIG Pack Systems, C33 and CE33.

2.3 Applications



Figure 2.6: Hitachi Seiki's Delta robots for pick-and-place and drilling.



Figure 2.7: ABB Flexible Automation's IRB 340 FlexPicker.

2.3. Applications

Industries take advantage of the Delta robot's high speed are the packaging, medical and pharmaceutical industries. Possible applications include assembly tasks or operation in a clean room. Also, they can be used for welding the electronic components.

Delta robot can be used in industry field for paint, draw, moving something, grasping and packaging in factories etc, because they are quite fast.

Delta robot can be found in different kinds of industries, like:

- Solar panels manufacturers where delta robot is used to place photovoltaic silicon wafers onto glass substrates[19].

- Energy-efficient lighting Manufacturers where Delta robots are used to place lenses into LED arrays as shown in Figure 2.8 [5].



Figure 2.8: Delta robot used to place lenses into led array.

2.4 Conceptual Design

Delta robot is composed of different parts and components connected with each other. A set of factors must be considered before building the robot. Some are related to the robot itself such as: safety, cost, design simplicity, workspace availability, volume occupied by the robot. Whereas the others are related to the user of the robot, who needs a simple program to deal with the robot.

It is desired to design, build and control three degrees of freedom delta robot. The mobile platform (end-effector) of the delta robot can move purely translation along three axes (x-axis, y-axis and z-axis). The motion of the end-effector starts when the user sends command to delta robot to move the end-effector to a desired position, in order to send the command to the robot, a Graphical User Interface program is needed.

The end-effector can move by using three actuators, these actuators are connected with the end-effector by using kinematic chains. Position sensors detect this motion and measure the change of the position for each actuator. These sensors are connected to a microcontroller which in turn can control the motion of the actuator which leads to control the motion of the end-effector.

In order to control the operation of the actuators, drivers will be used. These drivers are controlled by the microcontroller and also used to give the actuators the required power. The electrical actuators will be fixed on the base, so this required to use some kinds of mechanical components that are commonly used for this goal.

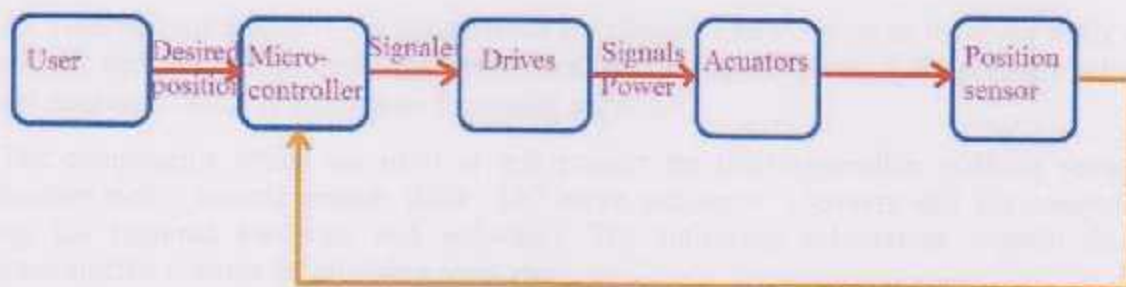


Figure 2.9 : Block diagram of the system.

The kinematic chains consist a parallelogram in order to keep the end-effector always moves purely translation (parallel to the workspace and the base). The connection between the parallelogram and the end-effector is by a kind of joints. Revolute joint can be used but the workspace be limited than the spherical joint workspace, so the ball bearing joint would be used for this robot.

A current sensor could be used to measure the consumed current for each actuator in order to make torque control, this sensor is connected to the power circuit. There is a need to measure the acceleration of the end-effector so a three axes accelerometer be used.

The mechanical structure of the robot could be made by different materials, but the aluminum is better because it is light and strong enough to achieve the goal of using this robot.

The chosen motion sensor must afford a high resolution for determining the accurate position. For this reason, three rotary encoders are used (one for each actuator), these sensors can read the change of position of the three actuators and send these readings of position to the microcontroller. The microcontroller will translate these readings into the coordinates of the position of the end-effector through C-code.

The programming code contains the forward and inverse kinematic equations. The user will be able to move the end-effector to the desired position by using inverse kinematic equation, or the user can operate any actuator(s) by using forward kinematic equations. Also if the user would to save some positions and then allow the robot to move between then , here the user can save these positions in the C-code.

CHAPTER 3

Components selection

3.1 Introduction

As a mechatronics project, the components are chosen depending on an involved study of the objectives and the requirements for mechanical and electrical parts, this approach called concurrent approach which is better than sequential approach.

The components which are used in this project are microcontroller, position sensor, 3-axis accelerometer, current sensor, three DC servo actuators, drivers and the computer (including the required hardware and software). The following subsections explain these components and the reasons for choosing each one.

3.2 Micro-Controller

The chosen micro-controller is Arduino MEGA ADK as shown in Figure 1. It has many advantages that makes it the preferred choice such as: open source codes which are available for many applications and can be downloaded from the company's website, the available shields for many components which allows the connection of modules easily. Other advantages are the upgradability, the modularity and the last important thing is that, it does not need a dedicated programmer as it can be programmed through the same cable that is used for the PC connection.

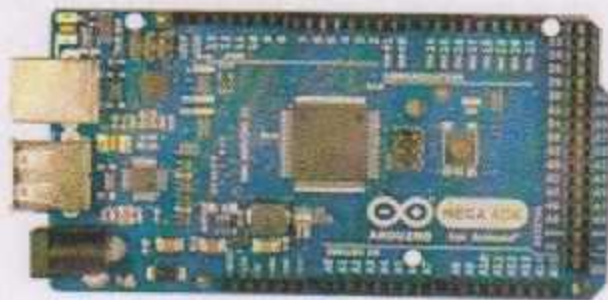


Figure 3.1: Arduino MEGA ADK.

The Arduino MEGA ADK can be powered via the USB connection or with an external power supply. The board can operate on an external supply of 5.5 to 16 volts. The recommended range is 7 to 12 volts. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The Arduino MEGA ADK can be programmed with the Arduino software which is available freely. Some of technical specification of ADC Microcontroller is shown in Table 3.1

Table 3.1: Technical specification of ADK Microcontroller.

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g

The Arduino MEGA ADK has a resettable polyfuse that protects the computer's USB ports from shorts and overcurrent. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

3.3 Position sensor

Position sensor is a device that is used for position measurement. It can be either an absolute position sensor or a relative one (displacement sensor) as shown in the figure2. Delta Robot has three angles to be measured, according to the three used actuators(θ_1 , θ_2 and θ_3) respectively.



Figure 3.2: Rotary encoder.

The most common angular position sensors are encoders. Encoders have two types: absolute encoders and incremental encoders as the one shown in Figure 3. The advantage of using the absolute encoder over the incremental one appears in case of power failure. An absolute encoder can continue from the position which it lies on even if it is changed during the power lost. While the incremental encoder starts counting from zero.



Figure 3.3: Incremental and absolute rotary encoder.

An incremental encoder can be used if there is a selected home position. This position relates the initial position of the device to the beginning of count from zero. The incremental encoder can be used instead of the absolute encoder because we detect an initial position for the Delta Robot. An initial position for the Delta Robot is chosen as shown in Figure 4, where the upper links are parallel to the base.



Figure 3.4: The Initial position of Delta Robot.

The required incremental encoder should satisfy some requirements such as: firstly, the precision (the number of pulses per revolution), but the clock speed of the used micro controller must be considered. This is due to the fact that the angular positions of the three motors must be read from the attached encoders in asynchronics fashion, therefore too high precision will limit the microcontroller reading performance. Secondly, the size because the encoder will be connected directly to the shaft of the actuator and these actuators will be fixed on the base. Thirdly, the encoder operating voltage.

3.4 Three axes accelerometer

The ADXL345 is a low-power, 3-axis MEMS accelerometer module with high resolution (13-bit) and measurement range up to ± 16 g. It is used for this project. Digital output data is formatted as 16-bit. The ADXL345 is supplied in a small, thin, .PCB board size: 31 (mm) \times 13 (mm), 14-lead, plastic package, as shown in Figure 5. This sensor consists of a micro-machined structure on a silicon wafer. The structure is suspended by polysilicon springs which allow it to deflect smoothly in any direction when subject to acceleration in the X, Y and/or Z axis. Deflection causes a change in capacitance between fixed plates and plates attached to the suspended structure. This change in capacitance on each axis is converted to an output voltage proportional to the acceleration on that axis.

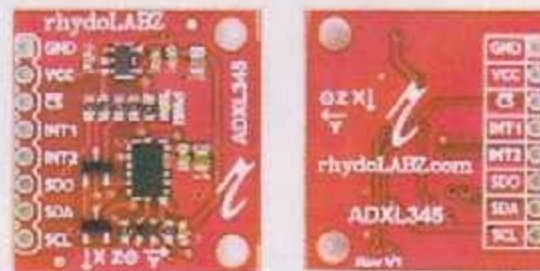


Figure 3.5: ADXL345 accelerometer.

3.5 Current sensor

ACS712, as shown in figure 6, provides economical and precise solutions for AC or DC current sensing in industrial systems. It can be used to make torque control for the robot. The device consists of a precise, low-offset, linear Hall sensor circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which is sensed by the integrated Hall IC and converted into a proportional voltage. This sensor has the following features:

- 1- Output voltage proportional to AC or DC currents.
- 2- The module can measure the positive and negative 5 amps.
- 3- The analog output is 185mV / A.
- 4- The module requires just 3 connections: +5 Vcc, ground and analog voltage out.
- 5- Nearly zero magnetic hysteresis.
- 6- Operating voltage is 5V DC.



Figure 3.6: ACS712current sensor.

3.6 DC servo actuator

The 3 DoF Delta Robot needs 3 actuators in order to move the end-effector to the specific position. The chosen actuator is DC servo actuator because it has good features, which make it suitable to be used for our project, like zero backlash, high positional accuracy and high stiffness. The RH actuators, as shown in figure 7, combining precision Harmonic Drive gear and DC servo motors offer unique features unsurpassed by conventionally geared drives.



Figure 3.7: RH mini-series DC servo Actuators.

RH mini-series DC servo Actuators are used in highly demanding industrial servo systems, they provide precision motion control and high torque capacity in very compact packages. The encoder is directly mounted onto the motor shaft. Since the gear has zero backlash, The Harmonic Drive gear is lubricated with a specially developed grease to ensure minimum maintenance requirements and long service life. The motor brush holders have seals to prevent dust transfer.

Table 3.2: Some technical data about the actuators:

Rated Output Power [W]	18.5
Rated Voltage [V]	24
Rated Current [A]	1.8
Rated Output Torque [Nm]	5.9
Rated Output Speed [rpm]	30
Peak Current [A]	4.1
Maximum Output Torque [Nm]	20
Maximum Output Speed [rpm]	50

Torque Constant [Nm/A]	5.76
Voltage Constant (B.E.M.F.) [V/rpm]	0.60
Inertia at Output Shaft [Kgm ²]	81.6*10 ⁻³
Mechanical Time Constant [ms]	7.0
Viscous Damping Constant [Nm/rpm]	1.5·10 ⁻¹
Gear Ratio	100
Motor Rated Output [W]	30
Motor Rated Speed [rpm]	3000
Armature Resistance [Ω]	2.7
Armature Inductance [mH]	1.1
Electrical Time Constant [ms]	0.41
Starting Current[A]	0.43
No-Load Running Current[A]	0.91

-Selection Procedure

Requirements for Preliminary Selection:

Load Torque T_L [Nm] < Rated Torque T_N [Nm]

Load Speed n_L [rpm] < Rated Output Speed n_N [rpm]

Load Inertia J_L [kgm²] < 3 J_s (Actuator Inertia) acceptable

Load Inertia J_L [kgm²] < J_A (Actuator Inertia) for best possible dynamic response.

Determination of the acceleration torque T_1 [Nm] :

$$T_1 = T_L + \frac{2\pi}{60} * \frac{(J_A + J_L) * n_L}{t_1}$$

- Acceleration Torque T_1 < Maximum Output Torque T_m .

Determination of the average torque T_A [Nm] :

$$T_A = \sqrt{\frac{T_1^2 * t_1 + T_2^2 * t_2 + T_3^2 * t_3}{t_1 + t_2 + t_3 + t_4}}$$

where:

T_1 : Acceleration Torque.

$T_2 = T_L$: Load Torque.

T_3 : Braking Torque

$T_3 = T_2 - (T_1 - T_2)$ (if $t_1 = t_3$).

t_1 : Acceleration Time.

t_2 : Constant Speed Time.

t_3 : Braking Time.

t_4 : Idle Time.

- Average Torque $T_A < \text{Rated Torque } T_N$ of the actuator.

Load and Operating Conditions :

Load Torque $T_L = 3 \text{ Nm}$.

Load Speed $n_L = 30 \text{ rpm}$.

Load Inertia $J_L = 0.25 \text{ kgm}^2$.

Acceleration Time $t_1 = 0.2 \text{ s}$.

Constant Speed Time $t_2 = 0.3 \text{ s}$.

Braking Time $t_3 = 0.2 \text{ s}$.

Idle Time $t_4 = 0.7 \text{ s}$.

Actuator's Data:

$T_N = 5.9 \text{ Nm}$

$n_N = 30 \text{ rpm}$

$J_A = 0.0816 \text{ kgm}^2$

$T_m = 20 \text{ Nm}$

$T_1 = 8.08 < 20 \text{ [N.m]}$, $T_A = 3.6976 < 5.9 \text{ [N.m]}$

- The above procedure leads to the following selection
Actuator **RH - 14D - 3002 - E100AL**

3.7 Motor's Driver

In order to control the direction and the operating of the motor, a driver is required. The L298 is an integrated monolithic circuit as shown in figure 3.8. It is a high voltage, high current dual full-bridge driver. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor.

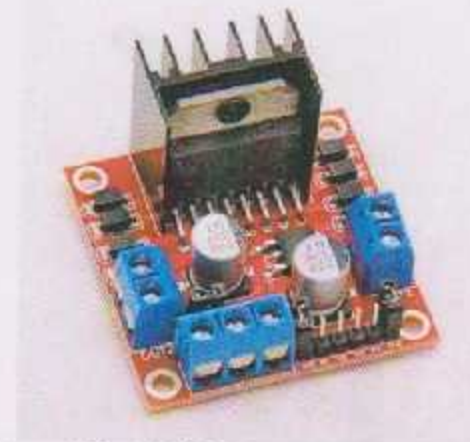


Figure 3.8: L298N Motor Driver Module.

L298N Motor Driver Module has the following characteristics:

- 1-Operating supply voltage up to 46V.
- 2-Total DC Current up to 4A.
- 3-Low saturation voltage.
- 4-Over temperature protection.

3.8 Mechanical Design

The Delta Robot consists two plate form , one is fixed and called the base and the other is mobile and called end-effector. Also it consists three kinematic chains, each one includes the upper link and parallelogram in addition to four bearing connecting the parallelogram sides.

The base made by wood as shown in figure 8, the shape of the base is triangle ,each side is 555mm and thickness 20mm. The base is fixed to the three links made by steel as shown in figure 3.9. The length of the steel structure is 300mm and the other piece is550mm. The cross section of the steel structure is square with dimension 20X20mm.

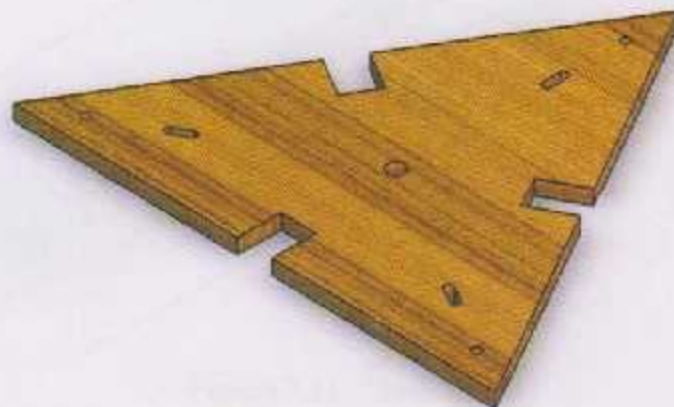


Figure 3.9: The base.



Figure 3.10:The structure.

The upper links, bearings, lower links and the end-effector all are made by Aluminum. The dimensions of the lower link are 410mm length and 8mm cross section radius as shown in figure 3.10. Further, the upper link is 180mm length, 30mm width and 15mm depth as shown in figure 3.11. The mass of upper and lower links are 190, 55g respectively. The end-effector also made by aluminum and has a mass of 196g as shown in figure 3.12. The ball joint bearing made by aluminum and has a mass of 24g for each piece as shown in figure 3.13.

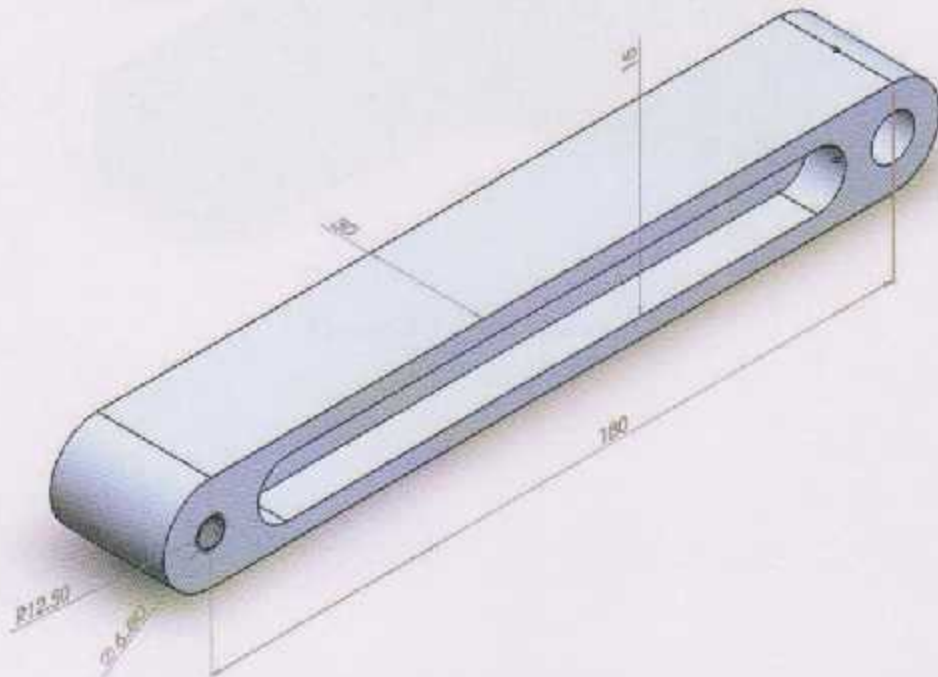


Figure 3.11: Upper link.

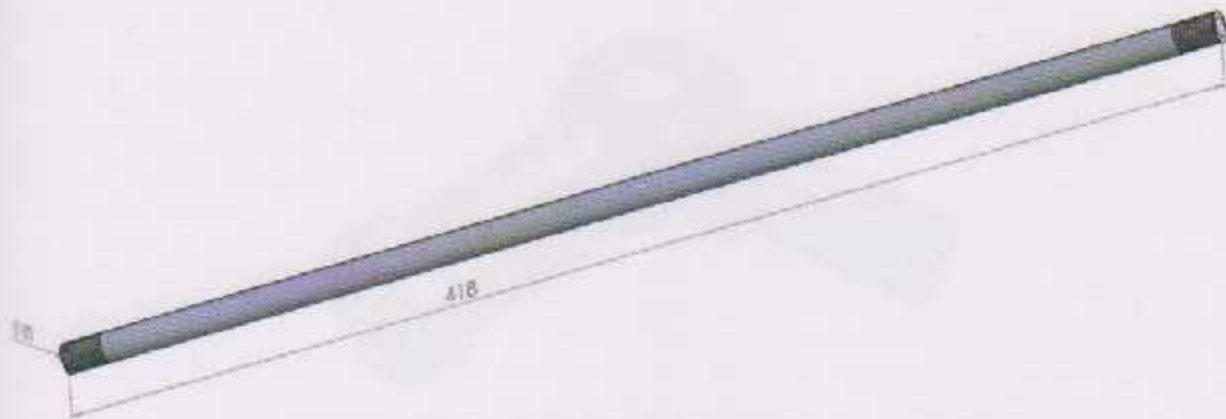


Figure 3.12: Lower link.

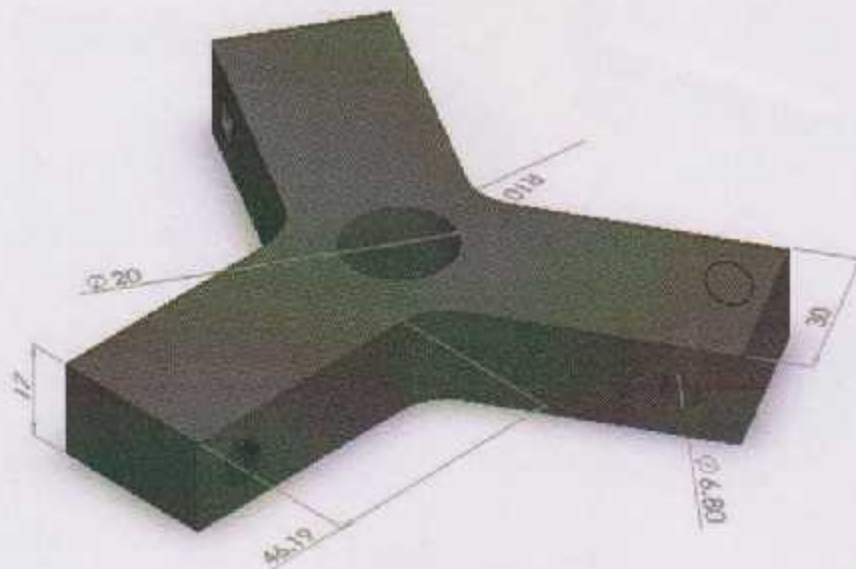


Figure 3.13: End-effector.



Figure 3.14: Ball joint bearing.



Figure 3.15: Delta robot after assembly.

3.9 Budget.

Table 3.3: Budget.

Components	Price (NIS)	Quantity	Total (NIS)
Microcontroller	300	1	300
Position sensor	25	3	75
Accelerometer	150	1	150
Current sensor	31	3	93
Actuators	530	3	1590
Mechanical structure	540	1	540
Driver	55	2	110
Bearing	10	12	120
Mechanical works for actuators	50	3	150
Wood pieces	85	1	85
Bolts, nuts and washers	27	1	27
Mechanical components	350	1	350
Plastic cover	190	1	190
Other parts	55	1	55
Printed labels	40	1	40
			3875

CHAPTER 4

Kinematics and dynamics

Delta robot consists of three closed-loop kinematic chains, each chain represent parallelogram and it consists fixed link and tow moving links as shown in Figure 4.1. The parallelograms ensure the constant orientation between the fixed platform and the mobile platform (task space), allowing only translation movements of the task space. Delta Robot can move in a three dimensional Cartesian coordinate system.

The combination of the constrained motion of the three arms connecting the task space to the fixed plate ensues in a resulting 3 translator degrees of freedom (DOF).

As mentioned in the previous chapter, there is two problem, one is forward kinematics and the other is inverse kinematics. In this chapter we will study the kinematics of three degree of freedom parallel delta robot.

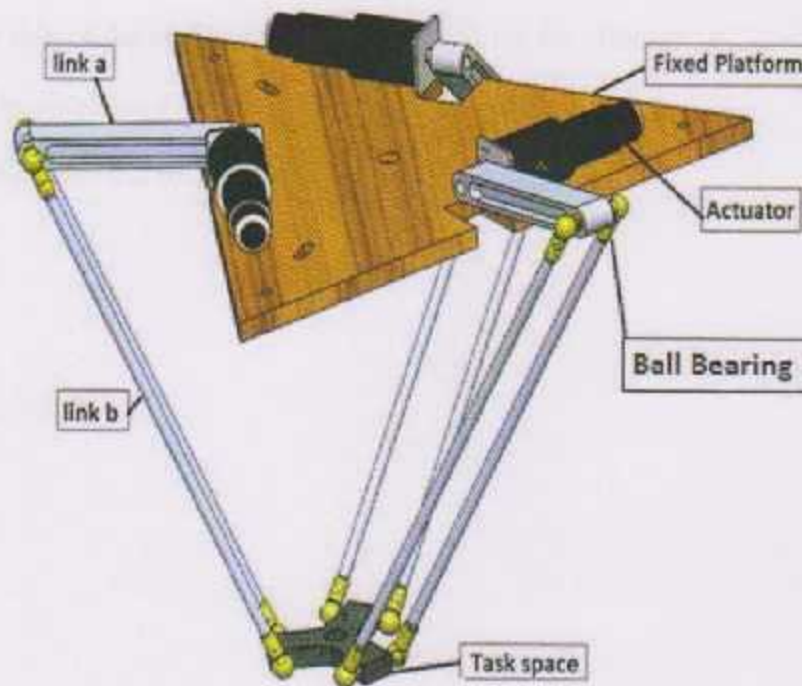


Figure 4.1: Delta Robot

4.1 Inverse Kinematics

Inverse kinematics solution used in order to find the joints space $(\theta_1, \theta_2, \theta_3)$, known the task space position (x, y, z) .

The purpose for determining the inverse kinematics of this parallel delta robot is to accurately model the angle produced for each joint space at a specific location of the task space. This is advantageous for two main reasons: firstly, that it is relatively simple to define any reasonable trajectory for the task space to traverse it and secondly, it can track different trajectories in a non-singular region [20].

Inverse kinematic is essential for the position control of parallel robots.

It should be noted that the parameters of the overall system are known, which include: the range of the desired angles for $(\theta_1, \theta_2, \theta_3)$, the overall length of the upper link ' L_a ' and the overall length of the lower link ' L_b '. The desired location of the task space (x, y, z) and the angles between the actuators.

We begin by extracting the kinematic model in which the parameters are as follows:

- Length of one side of the equilateral triangle that forms the framework: f
- Length of one side of the equilateral triangle that forms the effector: e
- Length from the fixed joint F1 until the spherical joint J1: L_a
- Length from the spherical joint J1 until the spherical joint E1: L_b

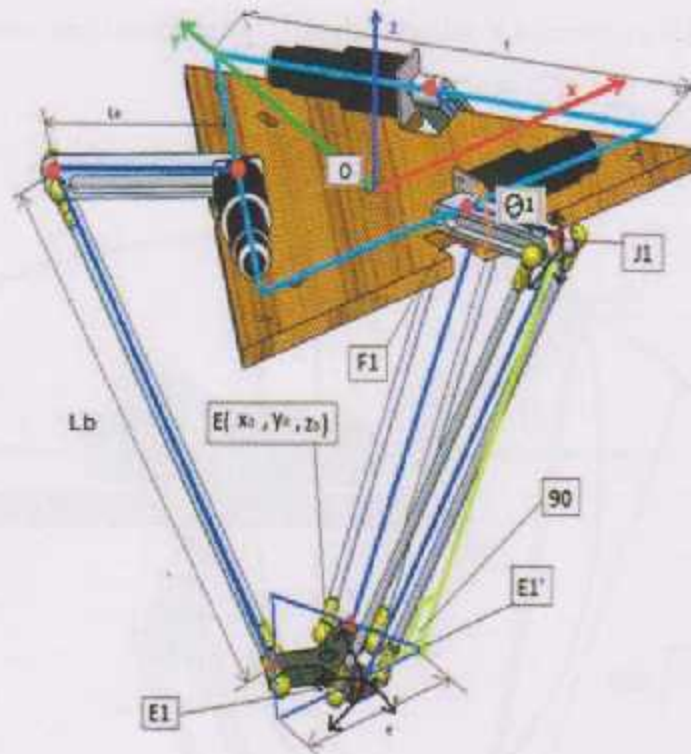


Figure4. 2: Equilateral triangle on the fixed platform, fixed and spherical joint in delta Robot.

The point J_1 can be found as the intersection of two circles. One with center in F_1 and radius L_b . Other with center in E_1 and radius: $\sqrt{L_b^2 - x_0^2}$.

We should choose only one intersection, the smaller Y coordinate solution, as shown in Figure 4.3.

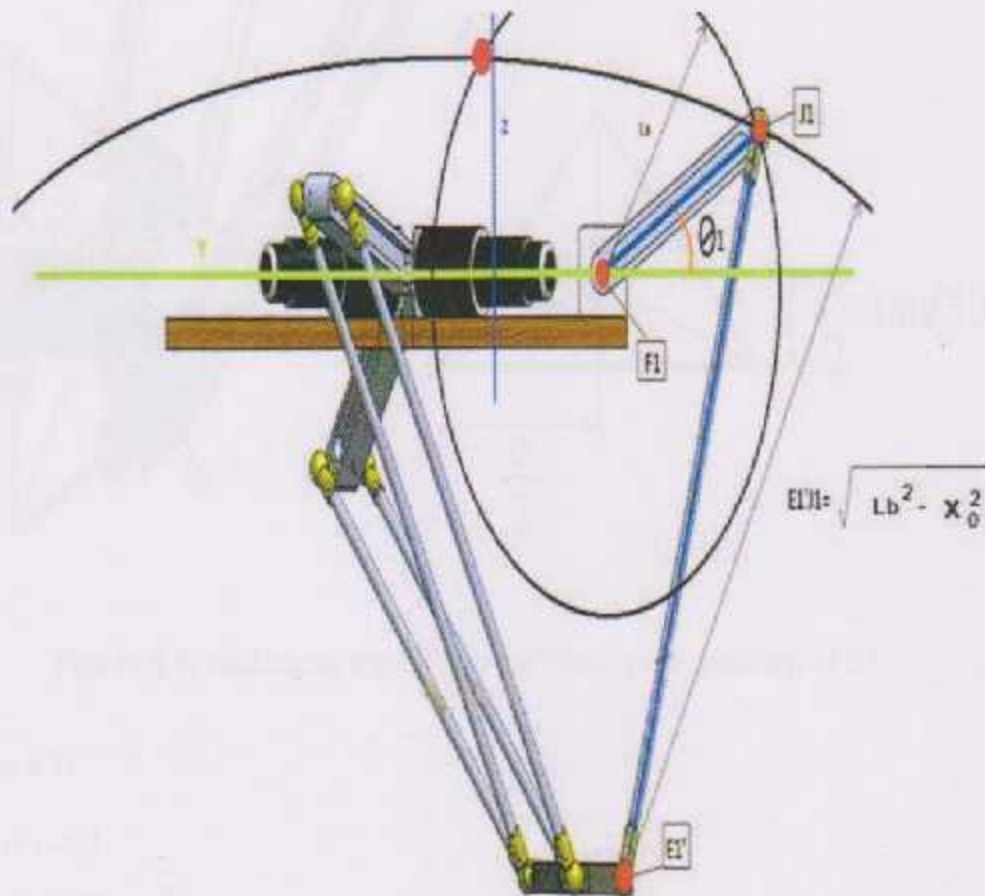


Figure 4.3: Two circles act in two point

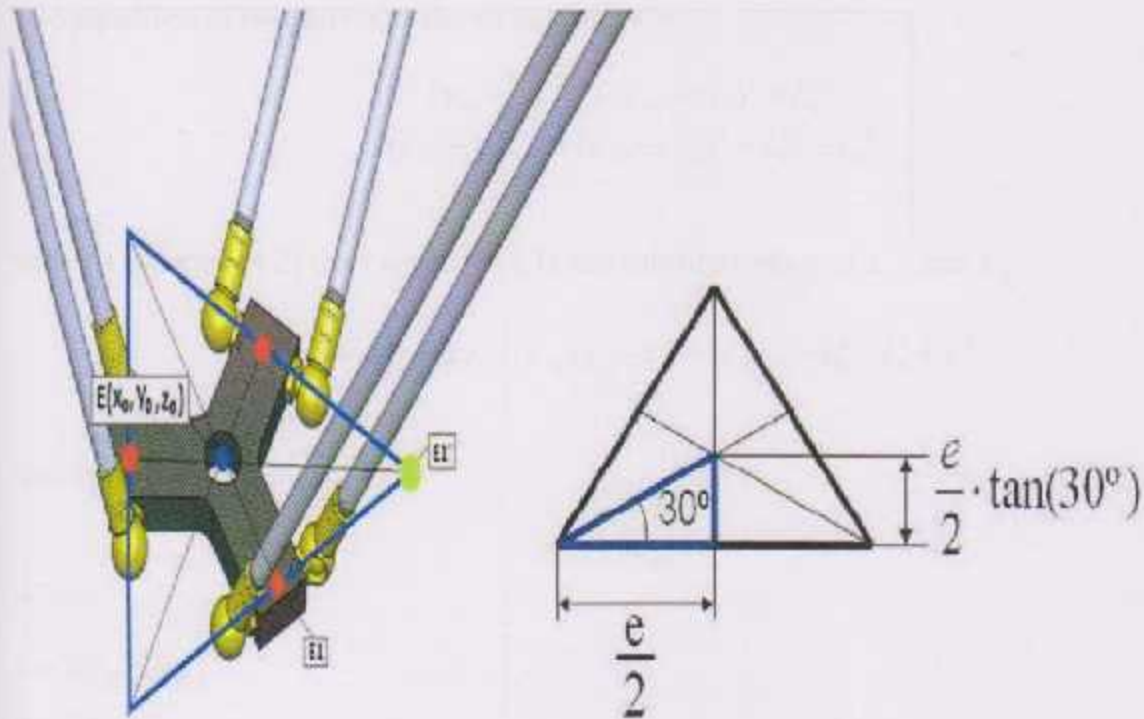


Figure 4.4: equilateral triangle on the Task space, position of E1'

Fro
m

the figure 4.4:

$$E = (x_0, y_0, z_0)$$

$$EE_1 = \frac{e}{2} \tan(30^\circ) = \frac{e}{2\sqrt{3}}$$

$$\left. \begin{aligned} E_1 &= \left\{ x_0, y_0 - \frac{e}{2\sqrt{3}}, z_0 \right\} \\ E_1' &= \left\{ 0, y_0 - \frac{e}{2\sqrt{3}}, z_0 \right\} \end{aligned} \right\} E_1 E_1' = x_0$$

$$F_1 = \left\{ 0, -\frac{e}{2\sqrt{3}}, 0 \right\}$$

$$E_1' J_1 = \sqrt{E_1 J_1^2 - E_1 F_1^2} = \sqrt{L_6^2 - x_0^2} \rightarrow \text{radius of the circle}$$

We can get the coordinates of J_1 in terms of the coordinates of E and the parameters of the machine.

Two equations of two circles as shown in figure 4.3:

$$(y_{j1} - y_{k1})^2 + (z_{j1} - z_{k1})^2 = L_a^2 \quad (4.1)$$

$$(y_{j1} - y_{k1})^2 + (z_{j1} - z_{k1})^2 = L_b^2 - x_0^2 \quad (4.2)$$

Subtract equation (4.2) from equation (4.1) and substitute value of z_{k1} and z_{j1} :

$$y_{j1}^2 - y_{k1}^2 + 2(y_{k1} - y_{j1})y_{j1} - z_0^2 + 2z_{j1}z_0 = L_a^2 - L_b^2 + x_0^2 \quad (4.3)$$

Rewrite equation (4.3) as:

$$ay_{j1} + bz_{j1} = d \quad (4.4)$$

Where:

$$a = 2(y_{k1} - y_{j1})$$

$$b = 2z_0$$

$$d = y_{k1}^2 - y_{j1}^2 + z_0^2 L_a^2 - L_b^2 + x_0^2$$

Rearrange equation(4.4):

$$z_{j1} = \frac{d - ay_{j1}}{b} \quad (4.5)$$

Substitute equation (4.5) in equation (4.1):

$$y_{j1}^2 + y_{k1}^2 - 2y_{k1}y_{j1} + \left(\frac{d - ay_{j1}}{b}\right)^2 - L_a^2 = 0 \quad (4.6)$$

Extract equation(4.6):

$$y_{j1}^2 + y_{k1}^2 - 2y_{k1}y_{j1} + G_1 y_{j1}^2 + G_2 y_{j1} + G_3 - L_a^2 = 0 \quad (4.7)$$

Where:

$$G_1 = \left(\frac{a}{b}\right)^2$$

$$G_2 = \frac{-2ad}{b^2}$$

$$G_3 = \left(\frac{d}{b}\right)^2$$

Rearrange equation (4.7) to be:

$$(1+G_1)y_h^2 + (G_2 - 2y_{R_1})y_h + (G_3 + y_{R_1}^2 - L_a^2) = 0 \quad (4.8)$$

Rewrite equation (4.8) as:

$$Hy_h^2 + Qy_h + L = 0 \quad (4.9)$$

Where:

$$H = (1+G_1)$$

$$Q = (G_2 - 2y_{R_1})$$

$$L = (G_3 + y_{R_1}^2 - L_a^2)$$

Solve equation(4.9):

$$y_h = \frac{-Q \pm \sqrt{Q^2 - 4HL}}{2H} \quad (4.10)$$

y_h is the smallest root.

To find z_{R_1} substitute y_h in equation (4.5).

Now from figure 4.3 and by help of equations (4.5) and (4.10) we can find the first angle:

$$\theta_1 = \tan^{-1} \left(\frac{-z_{R_1}}{y_{R_1} - y_h} \right) \quad (4.11)$$

In order to find θ_2 and θ_3 we know the axis are rotate about z-axis by 120° and 240° respectively. So we can write the equations as:

$$\begin{aligned}x_{0i} &= x_0 \cos(\alpha) - y_0 \sin(\alpha) \\y_{0i} &= x_0 \sin(\alpha) + y_0 \cos(\alpha) \\z_{0i} &= z_0\end{aligned}\tag{4.12}$$

Where $i = \{2, 3\}$ and $\alpha = \{120, 240\}$ for θ_2 and θ_3 respectively.

Now:

$$\theta_2 = \tan^{-1} \left(\frac{-z_{12}}{y_{12} - y_{11}} \right)\tag{4.13}$$

$$\theta_3 = \tan^{-1} \left(\frac{-z_{13}}{y_{13} - y_{11}} \right)\tag{4.14}$$

4.2 Forward Kinematics

Forward kinematics or direct kinematics determines the position of the task space (x, y, z) in the fixed frame. Given the actuators angles θ_i (Joints space), where i correspond to the number of loop ($i = 1, 2, 3$) as shown in Figure 4.2.

Direct kinematic is interest for the control of the position of the manipulator, but also for the velocity control of the end-effector.

To calculate the direct kinematics we move the center of the spheres from points J_1, J_2 and J_3 to the points J_1', J_2' and J_3' using the transition vectors $\overline{E_1E_0}, \overline{E_2E_0}, \overline{E_3E_0}$ respectively.

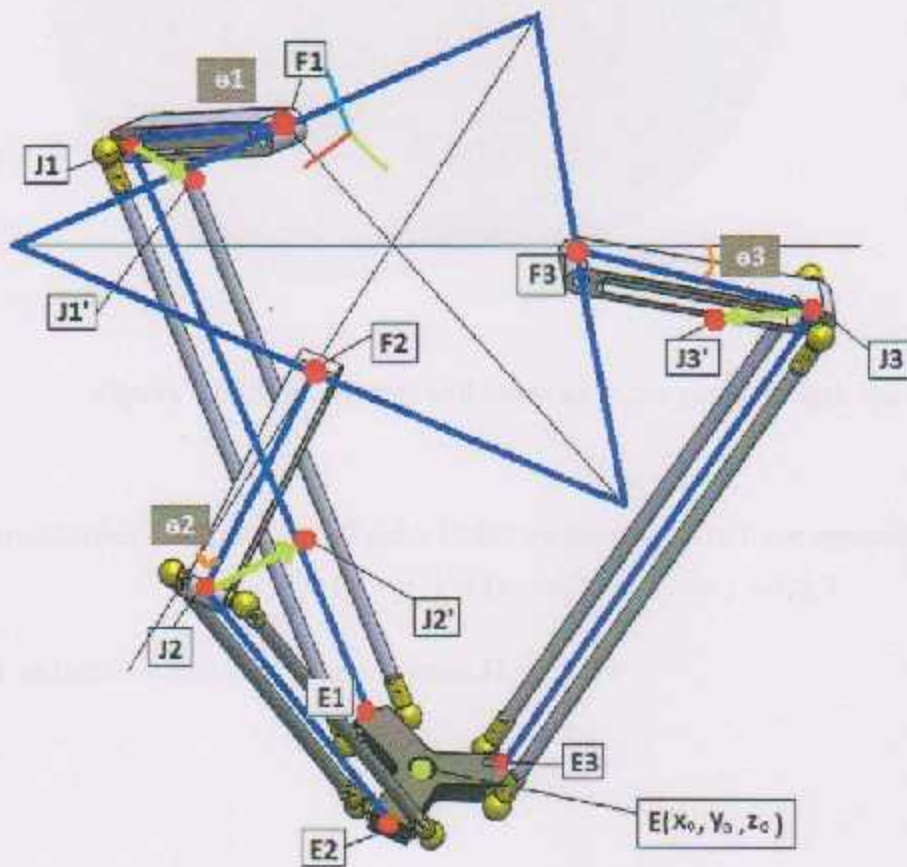


Figure 4.5: shown the position of J_1', J_2' and J_3' after transition.

After this transition the three spheres will intersect in the point: $E(x_e, y_e, z_e)$ as shown in Figure 4.5.

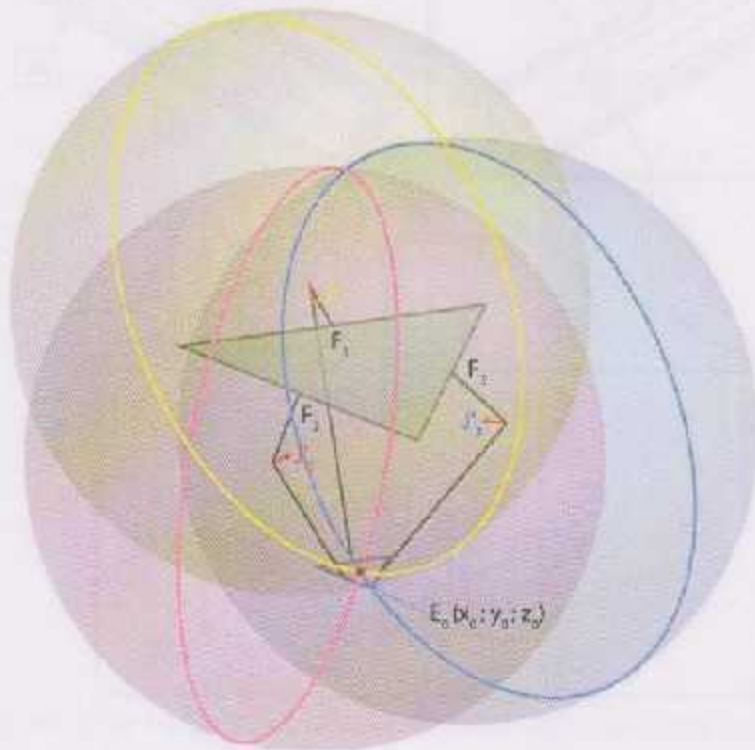


Figure 4.6: three spheres will intersect in the center of task space

To find coordinates $E(x_e, y_e, z_e)$ of point $E=E0$ we need to solve three equations like:

$$(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2 = L_j^2 \rightarrow j = 1, 2, 3$$

And (x, y, z) the coordinates of sphere centers $J1', J2', J3'$.

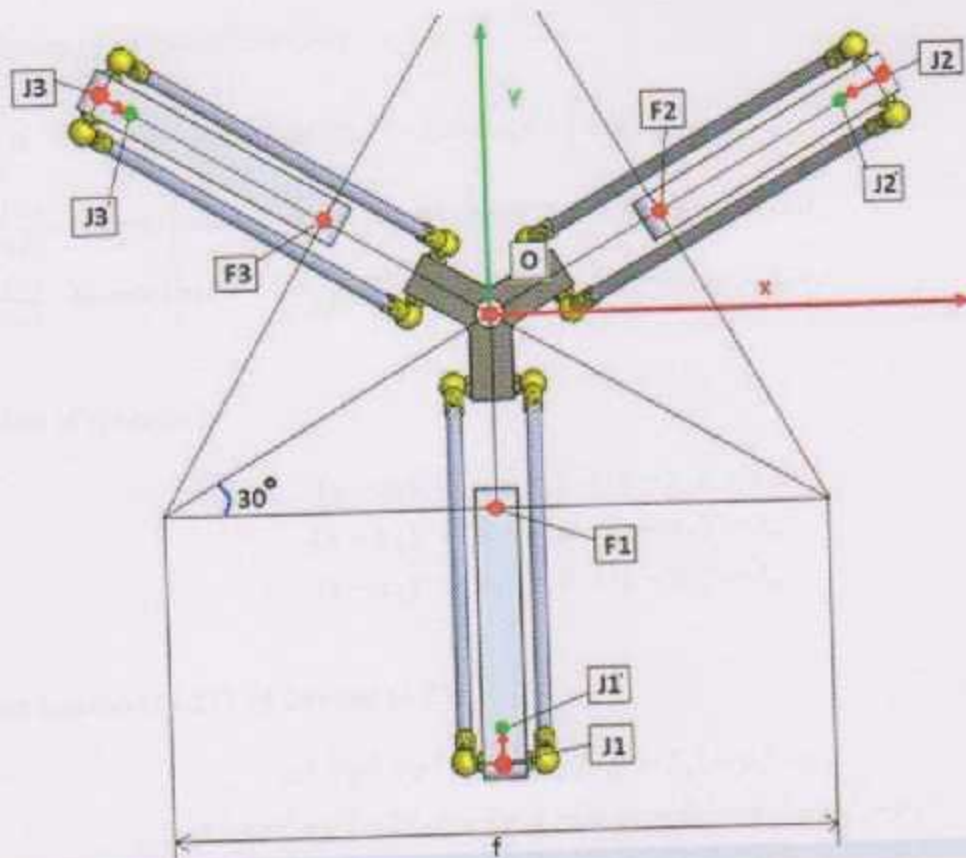


Figure 4. 7: Upper section shows the center of task space after transition and equilateral triangle.

From the figure 4.6:

$$OF1 = OF2 = OF3 = \frac{f}{2} \tan(30^\circ) = \frac{f}{2\sqrt{3}} \quad (\text{from equilateral triangle}) \quad (4.15)$$

$$J1J1' = J2J2' = J3J3' = \frac{f}{2} \tan(30^\circ) = \frac{e}{2\sqrt{3}} \quad (\text{from equilateral triangle}) \quad (4.16)$$

$$F1J1 = La \cdot \cos(\theta_1) \quad (4.17)$$

$$F2J2 = La \cdot \cos(\theta_2) \quad (4.18)$$

$$F3J3 = La \cdot \cos(\theta_3) \quad (4.19)$$

Coordinates of center spheres are:

$$J1' = \left(0 \quad \frac{-(f-e)}{2\sqrt{3}} - La \cdot \cos(\theta_1) \quad -La \cdot \sin(\theta_1) \right) = (x_1, y_1, z_1) \quad (4.20)$$

$$J2' = \left(\left[\frac{f-e}{2\sqrt{3}} + La \cdot \cos(\theta_1) \right] \cos(30^\circ) \quad \left[\frac{f-e}{2\sqrt{3}} + La \cdot \cos(\theta_1) \right] \sin(30^\circ) \quad -La \cdot \sin(\theta_1) \right) = (x_2, y_2, z_2) \quad (4.21)$$

$$J3' = \left(\left[\frac{f-e}{2\sqrt{3}} + La \cdot \cos(\theta_1) \right] \cos(30^\circ) \quad \left[\frac{f-e}{2\sqrt{3}} + La \cdot \cos(\theta_1) \right] \sin(30^\circ) \quad -La \cdot \sin(\theta_1) \right) = (x_3, y_3, z_3) \quad (4.22)$$

Equation of spheres:

$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = L_b^2 \quad (4.23)$$

$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = L_b^2 \quad (4.24)$$

$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = L_b^2 \quad (4.25)$$

Extract equation (4.23), (4.24) and (4.25):

$$x^2 + y^2 + z^2 - 2y_1y - 2z_1z = L_b^2 - y_1^2 - z_1^2 \quad (4.26)$$

$$x^2 + y^2 + z^2 - 2x_2x - 2y_2y - 2z_2z = L_b^2 - x_2^2 - y_2^2 - z_2^2 \quad (4.27)$$

$$x^2 + y^2 + z^2 - 2x_3x - 2y_3y - 2z_3z = L_b^2 - x_3^2 - y_3^2 - z_3^2 \quad (4.28)$$

Let,

$$w_1 = x_1^2 + y_1^2 + z_1^2 \quad (4.29)$$

Subtract equation (4.27) from equation (4.26):

$$x_2x + (y_1 - y_2)y + (z_1 - z_2)z = (w_1 - w_2) / 2 \quad (4.30)$$

Subtract equation (4.28) from equation (4.26):

$$x_3x + (y_1 - y_3)y + (z_1 - z_3)z = (w_1 - w_3) / 2 \quad (4.31)$$

Subtract equation (4.28) from equation (4.27):

$$(x_2 - x_3)x + (y_2 - y_3)y + (z_2 - z_3)z - (w_2 - w_3)/2 \quad (4.32)$$

From equation (4.30) and (4.31), we can find the solution of (x and y):

$$x = a_1 z + b_1 \quad (4.33)$$

$$y = a_2 z + b_2 \quad (4.34)$$

Where:

$$a_1 = \frac{1}{d} [(z_2 - z_3)(y_3 - y_1) - (z_3 - z_2)(y_2 - y_1)]$$

$$b_1 = \frac{1}{2d} [(w_2 - w_3)(y_3 - y_1) - (w_3 - w_2)(y_2 - y_1)]$$

$$d = (y_2 - y_1)x_3 - (y_3 - y_1)x_2$$

$$a_2 = \frac{-1}{d} [(z_2 - z_3)x_3 - (z_3 - z_2)x_2]$$

$$b_2 = \frac{1}{2d} [(w_2 - w_3)x_3 - (w_3 - w_2)x_2]$$

Substitute equation (4.33) and (4.34) in equation (4.26).

$$(a_1^2 + a_2^2 + 1)z^2 + 2(a_1 + a_2(b_2 - y) - z_1)z + (b_1^2 + (b_2 - y_1)^2 + z_1^2 - L_k^2) \quad (4.35)$$

The smallest negative equation root results $z = z_0$

(x_0, y_0) Are calculate with equations (4.33) and (4.34).

4.3 Velocity Kinematics

The most relevant loop should be picked up for the intended Jacobian analysis. Let $\vec{\theta}$ be the vector made up of actuated joint variables and \vec{P} is the position vector of the moving platform as shown in *Figure 4.8*.

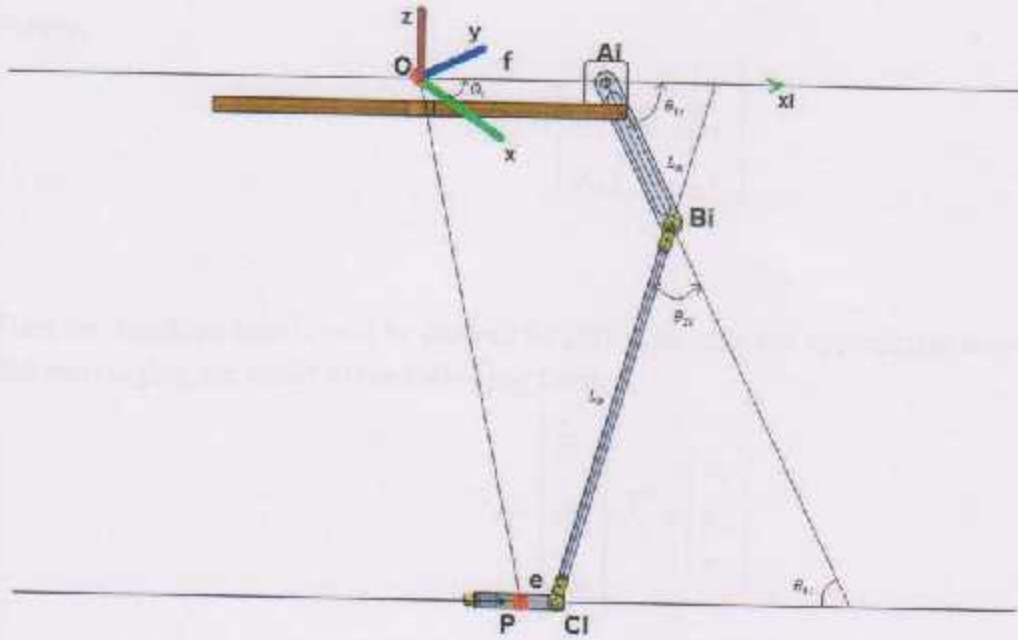


Figure 4.8: Projection of link i on x, z , plane.

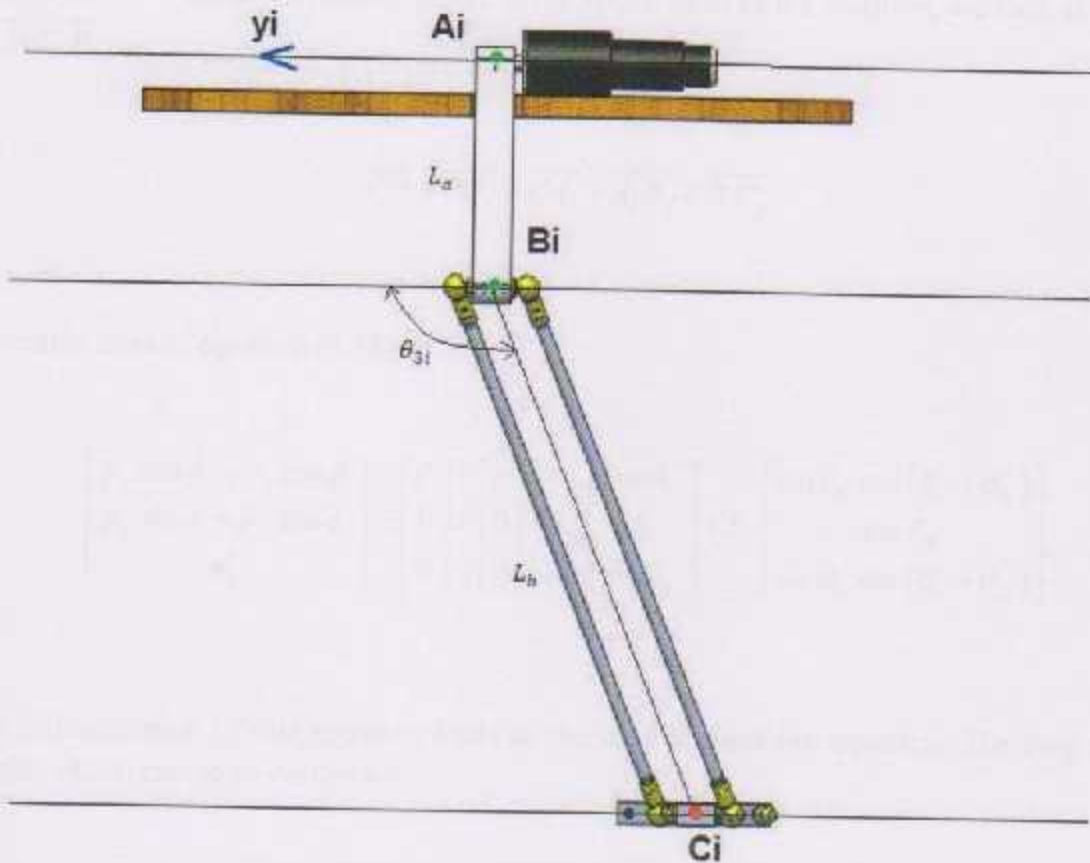


Figure 4.9: y, z , plane.

Where,

$$\bar{\theta}_{12} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix}, \bar{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4.36)$$

Then the Jacobian matrix will be derived by differentiating the appropriate loop closure equation and rearranging the result in the following form:

$$\bar{J}_\theta = \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix}, \bar{J}_P = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (4.37)$$

Where v_x, v_y and v_z are the x, y and z components of the velocity of the point \bar{P} on the task space in the xyz frame. In order to arrive at the above form of the equation, we look at the loop $OA_1B_1C_1P$. The corresponding closure equation in the $x_1y_1z_1$ frame is

$$\overline{OP} + \overline{PC_1} = \overline{OA_1} + \overline{A_1B_1} + \overline{B_1C_1} \quad (4.38)$$

The matrix form of equation (4.38) is:

$$\begin{bmatrix} p_x \cos \phi_1 - p_y \cos \phi_1 \\ p_x \sin \phi_1 - p_y \cos \phi_1 \\ p_z \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} e \\ 0 \\ 0 \end{bmatrix} + L_{11} \begin{bmatrix} \cos \theta_{11} \\ 0 \\ \sin \theta_{11} \end{bmatrix} + L_{12} \begin{bmatrix} \sin \theta_{12} \cos(\theta_{11} + \theta_{12}) \\ \sin \theta_{12} \\ \sin \theta_{12} \cos(\theta_{11} + \theta_{12}) \end{bmatrix} \quad (4.39)$$

Time differentiation of this equation leads to the desired Jacobian equation. The loop closure equation (4.38) can be re-written as:

$$(\bar{P} + \bar{e}) = \bar{f} + \bar{a}_1 + \bar{b}_1 \quad (4.40)$$

Where \overline{a}_i and \overline{b}_i represents vectors $\overline{A_i B_i}$ and $\overline{B_i C_i}$ respectively.

Differentiating equation (4.40) with respect to time and using the fact that \overline{f} is a vector characterizing the fixed platform, and \overline{e} is a vector characterizing the task space.

$$\dot{\overline{p}} = \dot{\overline{v}} - \dot{\overline{a}}_i + \dot{\overline{b}}_i \quad (4.41)$$

The linear velocities on the right hand side of equation (4.41) can be readily converted into the angular velocities by using the well-known identities.

Thus

$$\dot{\overline{v}} = \overline{w}_a \times \overline{a}_i + \overline{w}_b \times \overline{b}_i \quad (4.42)$$

\overline{w}_a and \overline{w}_b is the angular velocity of the link i . To eliminate \overline{w}_b , it is necessary to dot-multiply both sides of equation (4.42) by \overline{b}_i , therefore:

$$\overline{b}_i \cdot \dot{\overline{v}} = \overline{w}_a \cdot (\overline{a}_i \times \overline{b}_i) \quad (4.43)$$

Where,

$$\begin{aligned} \overline{a}_i &= L_a \begin{bmatrix} \cos \theta_{1i} \\ 0 \\ \sin \theta_{1i} \end{bmatrix}, \overline{b}_i = L_b \begin{bmatrix} \sin \theta_{2i} \cos(\theta_{1i} + \theta_{2i}) \\ \sin \theta_{2i} \\ \sin \theta_{2i} \cos(\theta_{1i} + \theta_{2i}) \end{bmatrix} \\ \overline{\omega}_i &= \begin{bmatrix} 0 \\ \dot{\theta}_{1i} \\ 0 \end{bmatrix}, v = \begin{bmatrix} v_x \cos \phi_i - v_y \cos \phi_i \\ v_x \sin \phi_i - v_y \cos \phi_i \\ v_z \end{bmatrix} \end{aligned} \quad (4.44)$$

Rewriting the vectors of equation (4.43) in the $x_i y_i z_i$ coordinate frame leads to,

$$j_{ix} v_x + j_{iy} v_y + j_{iz} v_z = L_a \sin \theta_{2i} \sin \theta_{3i} \dot{\theta}_{1i} \quad (4.45)$$

Where,

$$\begin{aligned} j_{ix} &= \cos(\theta_{1i} + \theta_{2i}) \sin \theta_{3i} \cos \phi_i - \cos \theta_{3i} \sin \phi_i \\ j_{iy} &= \cos(\theta_{1i} + \theta_{2i}) \sin \theta_{3i} \sin \phi_i - \cos \theta_{3i} \cos \phi_i \\ j_{iz} &= \sin(\theta_{1i} + \theta_{2i}) \sin \theta_{3i} \end{aligned} \quad (4.46)$$

Expanding equation (4.45) for $i = 1, 2$ and 3 yields three scalar equations which can be assembled into a matrix form as:

$$\vec{j}_z \vec{v} = \vec{j}_q \dot{\vec{q}} \quad (4.47)$$

Where,

$$\vec{j}_z = \begin{bmatrix} j_{1x} & j_{1y} & j_{1z} \\ j_{2x} & j_{2y} & j_{2z} \\ j_{3x} & j_{3y} & j_{3z} \end{bmatrix} \quad (4.48)$$

$$\vec{j}_q = L_a \begin{bmatrix} \sin \theta_{21} \sin_{31} & 0 & 0 \\ 0 & \sin \theta_{22} \sin_{32} & 0 \\ 0 & 0 & \sin \theta_{23} \sin_{33} \end{bmatrix} \quad (4.49)$$

$$\dot{\vec{q}} = \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} \quad (4.50)$$

After algebraic manipulations, it is possible to write,

$$\vec{v} = \vec{j} \dot{\vec{q}} \quad (4.50)$$

Where,

$$\bar{j} = \overline{j_3^{-1} j_4} = \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial z}{\partial \theta_3} \\ \frac{\partial x}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial z}{\partial \theta_3} \\ \frac{\partial x}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial z}{\partial \theta_3} \end{bmatrix} \quad (4.51)$$

4.4 Forward and Inverse Singularity analysis

From equation (4.47) it can be observed that singularity occurs:

1. When $\det(j_q) = 0$. This means that either $\theta_{2i} = 0$ or $\theta_{3i} = 0$ or π for $i=1,2$ and 3 .
2. When $\det(j_s) = 0$. This means that either $\theta_{1i} + \theta_{2i} = 0$ or π , or $\theta_{3i} = 0$ or π for $i=1,2$ and 3 .
3. When $\det(j_q) = 0$ and $\det(j_s) = 0$. This situation occurs when $\theta_{1i} = 0$ or π for $i=1,2$ and 3 .

In summary, singularity of the parallel manipulator occurs:

1. When all three pairs of the follower rods are parallel. Therefore, the moving platform has three degrees of freedom and moves along a spherical surface and rotates about the axis perpendicular to the moving platform.
2. When two pairs of the follower rods are parallel. The moving platform has one degree of freedom i.e. the moving platform moves in one direction only.
3. When two pairs of the follower rods are in the same plane or two parallel planes. The moving platform has one degree of freedom; i.e. the moving platform rotates about the horizontal axis only.



4.5 Dynamic model

Dynamics is the science of motion. It describes why and how a motion occurs when forces and moments are applied on massive bodies. The motion can be considered as evolution of the position, orientation and their time derivatives. In robotics, the dynamic equation of motion for manipulators is utilized to set up the fundamental equations for control. The links and arms in a robotic system are modeled as rigid bodies.

Therefore, the dynamic properties of the rigid body take a central place in robot dynamics. Since the arms of a robot may rotate or translate with respect to each other, translational and rotational equations of motion must be developed and described in body-attached coordinate frames $B_1, B_2, B_3 \dots$ or in the global reference frame G .

There are basically two problems in robot dynamics.

Problem 1. We want the links of a robot to move in a specified manner. What forces and moments are required to achieve the motion? The first Problem is called Inverse dynamics and is easier to solve when the equations of motion are in hand because it needs differentiating of kinematics equations. The first problem includes robots statics because the specified motion can be the rest of a robot. In this condition, the problem reduces finding forces such that no motion takes place when they act. However, there are many meaningful problems of the first type that involve robot motion rather than rest. An important example is that of finding the required forces that must act on a robot such that its end-effector moves on a given path and with a prescribed time history from the start configuration to the final configuration.

Problem 2. The applied forces and moments on a robot are completely specified. How will the robot move? The second problem is called direct dynamics and is more difficult to solve since it needs integration of equations of motion. However, the variety of the applied problems of the second type is interesting. Problem 2 is essentially a prediction since we wish to find the robot motion for all future times when the initial state of each link is given.

In this section, we will perform the inverse dynamic modeling of the parallel manipulator based upon the principle of virtual work. The inverse dynamics problem is to find the actuator torques and/or forces required to generate a desired trajectory of the manipulator.

It is often convenient to express the dynamic equations of a manipulator in a single equation that hides some of the details, but shows some of the structure of the equations. The state-space equation When the Newton-Euler equations are evaluated symbolically for any manipulator, they yield a dynamic equation that can be written in the form:

$$\tau = \mathbf{M}(\theta)\ddot{\theta} + \mathbf{V}(\theta, \dot{\theta})\dot{\theta} + \mathbf{G}(\theta) \quad (4.52)$$

Where $\mathbf{M}(\theta)$ is $n \times n$ Inertia matrix of the manipulator, $\mathbf{V}(\theta, \dot{\theta})$ is a $n \times 1$ vector of centrifugal and Coriolis terms, and $\mathbf{G}(\theta)$ is an $n \times 1$ vector of gravity terms. We use the term state-space equation because the term $\mathbf{V}(\theta, \dot{\theta})$ has both position and velocity dependence. Each element of $\mathbf{M}(\theta)$ and $\mathbf{G}(\theta)$ is a complex function that depends on θ , the position of all the joints of the manipulator. Each element of $\mathbf{V}(\theta, \dot{\theta})$ is a complex function of both θ and $\dot{\theta}$. We may separate the various types of terms appearing in the dynamic equations and form the mass matrix of the manipulator, the centrifugal and Coriolis vector, and the gravity vector [21].

4.5.1 Virtual Work Dynamics

In this section, we will perform the inverse dynamic modeling of the parallel manipulator based upon the principle of virtual work. The inverse dynamics problem is to find the actuator torques and/or forces required to generate a desired trajectory of the manipulator [22].

Without losing generality of model, we can simplify the dynamic problem by the following hypotheses:

The connecting rods of lower links can be built with light materials such as the aluminum alloy, so

- The lower links rotational inertias are neglected.
- the mass of each lower links, is divided evenly and concentrated at
- The two endpoints of the parallelogram.

Also it is supposed that:

- The friction forces in joints are neglected.
- No external forces suffered.

We consider that $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]$ and $\delta\theta = [\delta\theta_1, \delta\theta_2, \delta\theta_3]$ are the vector of actuator torques and vector of corresponding virtual angular displacements. Furthermore, $\delta p = [\delta x, \delta y, \delta z]$ represents the virtual linear displacements vector of the mobile platform. We can derive the following equations by applying the virtual work principle.

$$\boldsymbol{\tau}^T \delta\theta + M_{ca}^T \delta\theta + F_{on}^T \delta\theta - M_e^T \delta\theta - F_r^T \delta p = 0 \quad (4.53)$$

Where,

$$M_{Ca} = \left(\frac{1}{2} m_a + m_b\right) \cdot g \cdot L_a \cdot I \cdot \begin{bmatrix} \cos \theta_1 \\ \cos \theta_2 \\ \cos \theta_3 \end{bmatrix} \quad (4.54)$$

Is the upper links gravity torques vector, m_a and m_b are mass of upper link and each connecting rod of lower link, respectively. Here g denotes the gravity acceleration, and I represent the 3x3 identity matrix.

$$F_{Cp} = \begin{bmatrix} 0 \\ 0 \\ -(m_p + 3m_b)g \end{bmatrix} \quad (4.55)$$

Denotes the mobile platform gravity force vector, and m_p is mass of the mobile platform.

$$M_a = I_a \ddot{\theta} = I_a \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} \quad (4.56)$$

Where,

$$I_a = \left(\frac{1}{3} m_a L_a^2 + m_b I_b^2\right) \cdot I \quad (4.57)$$

Represents the upper links inertia torques vector and denotes the upper links inertial matrix with respect to the fixed frame $O\{x, y, z\}$ and,

$$F_p = M_p \ddot{p} = (m_p + 3m_b) \cdot I \cdot \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \quad (4.58)$$

Denote the mobile platform inertial forces vector. Equation (4.50) in section 4.3 can be rewritten to,

$$\ddot{p} = \ddot{J} \ddot{\theta} \quad (4.59)$$

Consequently,

$$\delta p = J \delta \theta \quad (4.60)$$

Substituting equation 4.59 into equation 4.53 results,

$$(\tau^T + M_{Ca}^T + F_{Cp}^T J - M_a^T - I_p^T J) \delta \theta = 0 \quad (4.61)$$

Equation (4.60) holds for any virtual displacements $\delta\theta$, so we have,

$$\tau = M_a + J^T F_p - M_{ca} - J^T F_{cp} \quad (4.62)$$

Substitute equation (4.55) and (4.56) into equation (4.61), allows the generation of,

$$\tau = I_a \ddot{\theta} + J^T M_p \ddot{P} - M_{ca} - J^T F_{cp} \quad (4.63)$$

Differentiating equation (5.59) with respect to time, yields,

$$\ddot{P} = J \ddot{\theta} + \dot{J} \dot{\theta} \quad (4.64)$$

Substituting equation (4.64) into equation (4.63), we can derive that,

$$\tau = M(\theta) \ddot{\theta} + V(\theta, \dot{\theta}) \dot{\theta} + G(\theta) \quad (4.65)$$

The previous equation described in equation (4.52) represents the dynamic model of parallel manipulator in joint space. Here, $\theta \in R^3$ is the controlled variables, and

$$M(\theta) = I_a + J^T M_p J \quad (4.66)$$

Denotes a symmetric positive definite inertial matrix, that $M(\theta) \in R^{3 \times 3}$.

$$V(\theta, \dot{\theta}) = J^T M_p \dot{J} \quad (4.67)$$

Where $V(\theta, \dot{\theta}) \in R^{3 \times 3}$ is the centrifugal and Coriolis forces matrix, and,

$$G(\theta) = -M_{ca} - J^T F_{cp} \quad (4.68)$$

Represents the vector of gravity forces, and $G(\theta) \in R^3$.

4.5.2 Non-Rigid Body Effects

It is important to realize that the dynamic equations we have derived do not encompass all the effects acting on a manipulator. They include only those forces which arise from rigid body mechanics. The most important source of forces that are not included is friction. All mechanisms are, of course, affected by frictional forces. In present-day manipulators, in which significant gearing is typical, the forces due to friction can actually be quite large perhaps equaling 25% of the torque required to move the manipulator in typical situations. In order to make dynamic equations reflect the reality of the physical device, it is important to model (at least approximately) these forces of friction. A very simple model for friction is viscous friction, in which the torque due to friction is proportional to the velocity of joint motion. Thus, we have

$$\tau_{friction} = \nu \dot{\theta} \quad (4.69)$$

Where ν is a viscous-friction constant. Another possible simple model for friction, Coulomb friction, is sometimes used. Coulomb friction is constant except for a sign dependence on the joint velocity and is given by

$$\tau_{friction} = c \sin(\dot{\theta}) \quad (4.70)$$

Where c is a Coulomb-friction constant. The value of c is often taken at one value when $\dot{\theta} = 0$ the static coefficient, but at a lower value, the dynamic coefficient, when $\dot{\theta} \neq 0$, whether a joint of a particular manipulator exhibits viscous or Coulomb friction is a complicated issue of lubrication and other effects. A reasonable model is to include both, because both effects are likely:

$$\tau_{friction} = \nu \dot{\theta} + c \sin(\dot{\theta}) \quad (4.71)$$

It turns out that, in many manipulator joints, friction also displays a dependence on the joint position. A major cause of this effect might be gears that are not perfectly round-their eccentricity would cause friction to change according to joint position. So a fairly complex friction model would have the form

$$\tau_{friction} = f(\theta, \dot{\theta}) \quad (4.72)$$

These friction models are then added to the other dynamic terms derived from the rigid-body model, yielding the more complete model

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta})\dot{\theta} + G(\theta) + F(\theta, \dot{\theta}) \quad (4.73)$$

This equation (4.73) known as Inverse dynamic equation.

Controller Design

5.1 Introduction

By using the inverse kinematics equations, we can calculate the actuator's angles for a desired position of the end-effector of the robot. By using a rotary encoders, the actual angles of the actuators will be measured then compared with the desired angles that be needed to move the end-effector to the desired position.

There are different controllers that can be designed to control the delta robot. A discrete tracker controller is designed for the position and the speed for each one of the used actuator.

5.2 System equation

For Delta robot mechanics, each joint is actuated with a DC servo Motors. In general, the torque generated by a DC motor is proportional to the armature current and the strength of the magnetic field. Here the magnetic field is constant and, therefore, the motor torque is proportional to only the armature current " i " by a constant factor K_t as shown in the equation below.

$$T = K_t i \quad (5.1)$$

The back emf, e , is proportional to the angular velocity of the shaft by a constant factor K_e .

$$e = K_e \dot{\theta} \quad (5.2)$$

In SI units, the motor torque and back electromagnetic field (emf) constants are equal, that is, $K_t = K_e$; therefore, we will use K to represent both the motor torque constant and the back emf constant.

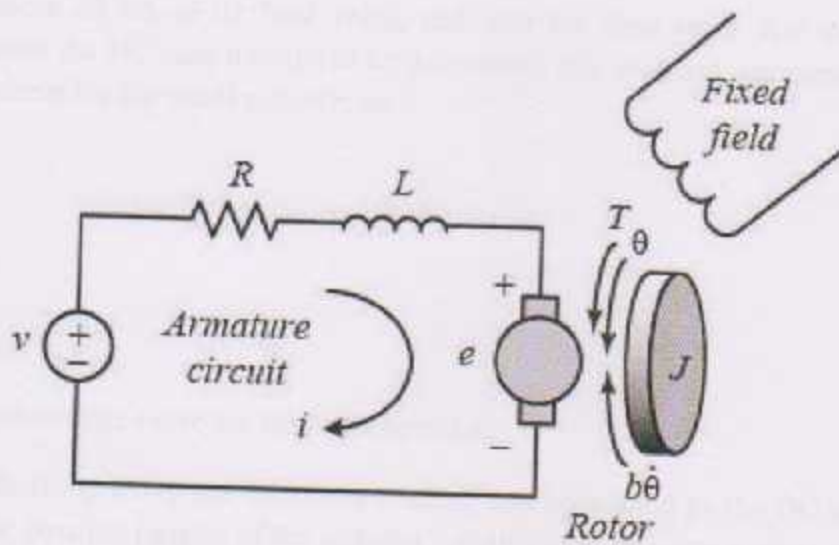


Figure 5.1: The electric equivalent circuit of the armature and the free-body diagram of the rotor.

From the figure above, we can derive the following equations based on Newton's 2nd law and Kirchhoff's voltage law.

$$J \ddot{\theta} + b \dot{\theta} = K i \quad (5.3)$$

$$L \frac{di}{dt} + R i = V - K \dot{\theta} \quad (5.4)$$

Where:

J: Moment of inertia of the rotor.

b: Motor viscous friction constant.

K: Motor torque constant.

R: Electric resistance.

L: Electric inductance.

5.3 Transfer function of the actuator

In order to design a discrete tracker controller for the DC actuator, the transfer function of this actuator should be known. There are different methods to get the transfer function of a DC actuator, all of these three methods done, but eventually the 3rd method is used for the controller.

First method: By making coupling between this DC actuator and another small DC actuator, which will work as a generator with known relationship between the applied voltage and the output speed. Then by applying known voltage to the DC actuator and see the response by using the oscilloscope, the value of (τ) can be known which represent the time needed by the

system to reach about 63.1% of its final value, and also the final value that we see on the oscilloscope represent the DC gain multiplied by a constant, this constant represent the slope of the speed-voltage curve for the small actuator, so

$$G(s) = \frac{DC\text{gain}}{\tau s + 1}$$

Where:

DCgain=final value * K.

K: slope of the speed-voltage curve for the small actuator.

Second method: by using the encoder's reading that connected to the DC actuator, this encoder measure the position (angles of the actuator's shaft), then by using a microcontroller we can get the speed and then plot the speed with the time curve and see the response by using an oscilloscope. From this response we can get the values of the constant (K & a) of the actuator's transfer function which in the form:

$$G(s) = \frac{K}{s + a}$$

Where : K=final value X a.

$$a = 1/\tau.$$

τ : The time needed by the system to reach 63.1% of it's final value.

-Here the final value that appear on the curve of the speed-time represent K/a.

Third method by applying the Laplace transform for the modeling equations 3 and 4, we get the following:

$$s(Js + b)\theta(s) = KI(s) \quad (5.5)$$

$$(Ls + R)I(s) = V(s) - Ks\theta(s) \quad (5.6)$$

We arrive at the following open-loop transfer function by eliminating $I(s)$ between the two above equations, where the rotational speed is considered the output and the armature voltage is considered the input.

$$P(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \quad \left[\frac{\text{rad/sec}}{V} \right] \quad (5.7)$$

The physical parameters for the used actuator are:

(J)	moment of inertia of the rotor	$81.6 \cdot 10^{-3}$	[kg.m ²]
(b)	motor viscous friction constant	1.433	[N.m.s]
(K _e)	electromotive force constant	5.73	[V/rad/sec]
(K _t)	motor torque constant	5.73	[N.m/Amp]
(R)	electric resistance	2.7	[Ohm]
(L)	electric inductance	$1.1 \cdot 10^{-3}$	[H]

Now, represent the above open-loop transfer function of the actuator in MATLAB by defining the parameters and transfer function as follows. Running this code in the command window produces the transfer function of the actuator as shown below.

```
>>J = 81.6*10^(-3);
>>b = 1.433;
>>K = 5.73;
>>R = 2.7;
>>L = 1.1*10^(-3);
>>s = tf('s');
>>TF_motor = K/((J*s+b)*(L*s+R)+K^2)
```

>>Transfer function:

5.73

 $8.976e-005 s^2 + 0.2219 s + 36.7$

The transfer function can be reduced to:

$$G(s) = \frac{25.8224}{s + 165.3898}$$

5.4 Design a Discrete Tracker controller

The goal of implementing any type of controller is to observe the output response it would generate based on the input conditions. In order to achieve this, it is necessary to solve for the control input (u) of the system [2].

Using of digital controller has the following advantages over analog control systems: reduced cost, flexibility in response to design changes and noise immunity.

The principle of feedback control can be expressed as: when the actual variable is smaller than the desired value, the controller will increase the control command. when the actual variable is larger than the desired value the controller will decrease the control command [2].

Controllability is the main issue in the analysis of a system before deciding the best control strategy to be applied, or even the system is possible to be controlled or stabilized or not. A control system that will be designed must always have some robustness property.

At any instant there can be a difference between the actual joint variables and the desired values. The difference is called the error and is measured by :

$$e = \theta_d - \theta_a.$$

where:

e : Error.

θ_d : Desire angle.

θ_a : Actual angle.

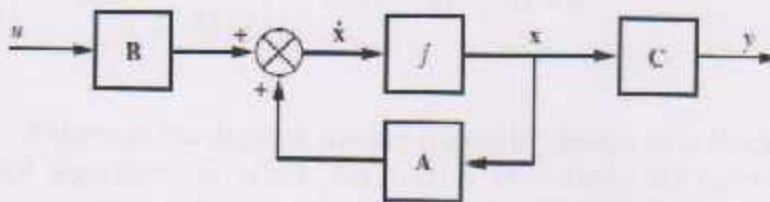


Figure 5.2: State space representation of the plant.

The process (shown in figure5.2):

$$\dot{x} = A \vec{x} + B \vec{u}$$

$$y = C \vec{x} + D \vec{u}$$

Note: \vec{x} and y are vectors.

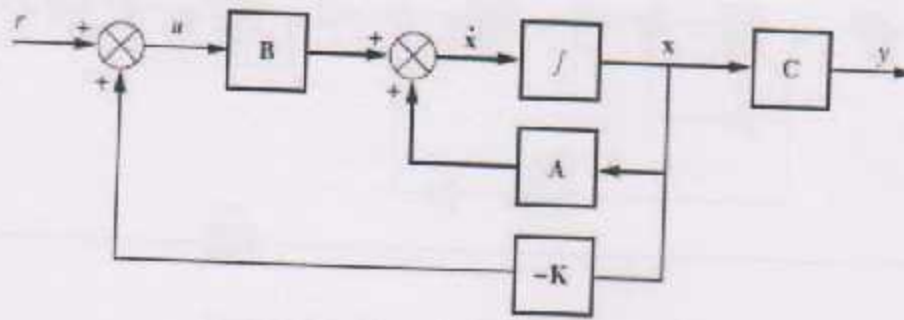


Figure 5.3: plant with state-variable feedback.

Discrete time process :

$$\begin{aligned}\bar{x}(k+1) &= A_d \bar{x}(k) + B_d \bar{u}(k) \\ y(k) &= C \bar{x}(k)\end{aligned}$$

Where

$$A_d = I + T A$$

$$B_d = T B$$

T: Sampling time.

Where A, B, C and D are the state space matrices. For our system, the matrices A, B, C and D are:

$$A = \begin{pmatrix} 0 & 1 \\ 0 & -165.38 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 25.8230 \end{pmatrix}, \quad C = [1 \quad 0], \quad D = 0$$

Figure 5.2 illustrates the discrete tracker controller design in a block diagram. This is a closed-loop control algorithm, in which the control commands are calculated based on the difference between actual and desired variables. Reading the actual variables and comparing with the desired values is called feedback, and because of that, the closed-loop control algorithm is also called a feedback control algorithm[2].


```

Ki=-Ke(3) % integral_gain
Fc=eig(Ad-Bd*Ke) % Controller_poles

```

The discrete tracker controller is designed for each actuator, but for controlling the Delta Robot it is needed to re-tune the Tracker controller parameters to get the optimal one approximately. Tracker controller parameters are shown in the next table.

Table 5.1: Tracker controller parameters.

Controller Parameters	Tuned Values
K1	15.8974
K2	-1.0579
KI	273.7128

The response before designing the controller shown in figure 5.3. The input is step input .

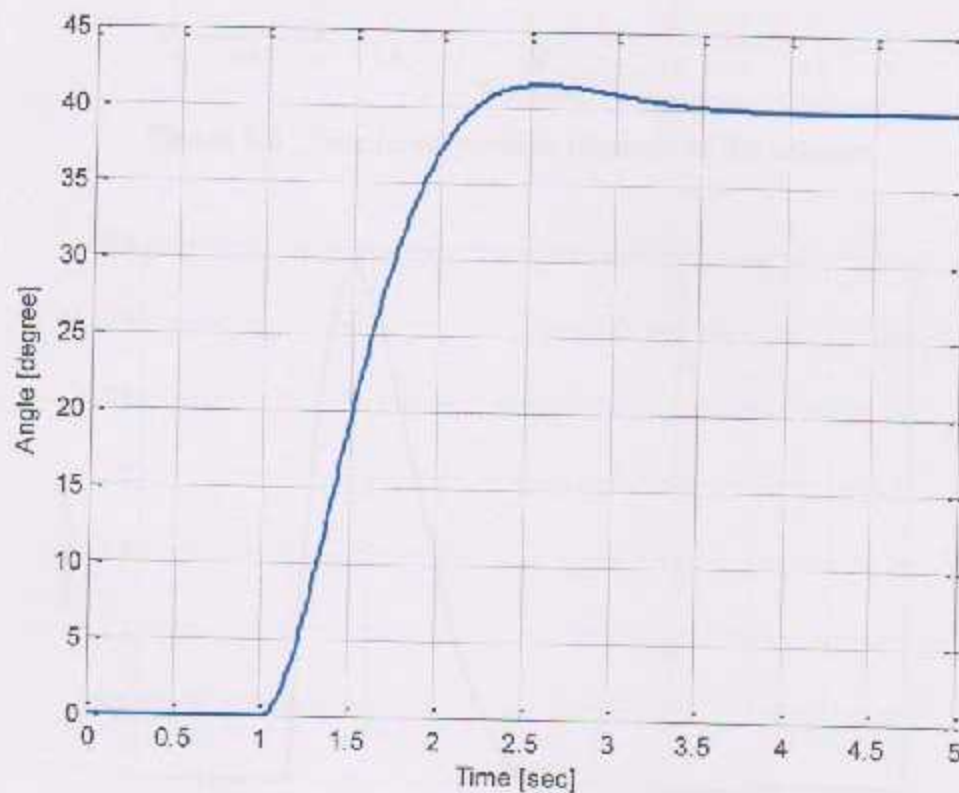


Figure 5.5: Simulation for the position response of the actuator before designing a controller.

Settling time = 2.5 sec, Overshoot=3.775%, peak time=1.59 sec.

After the discrete tracker controller is designed, the simulation response shown in figure 5.6, 5.7 and 5.8 .

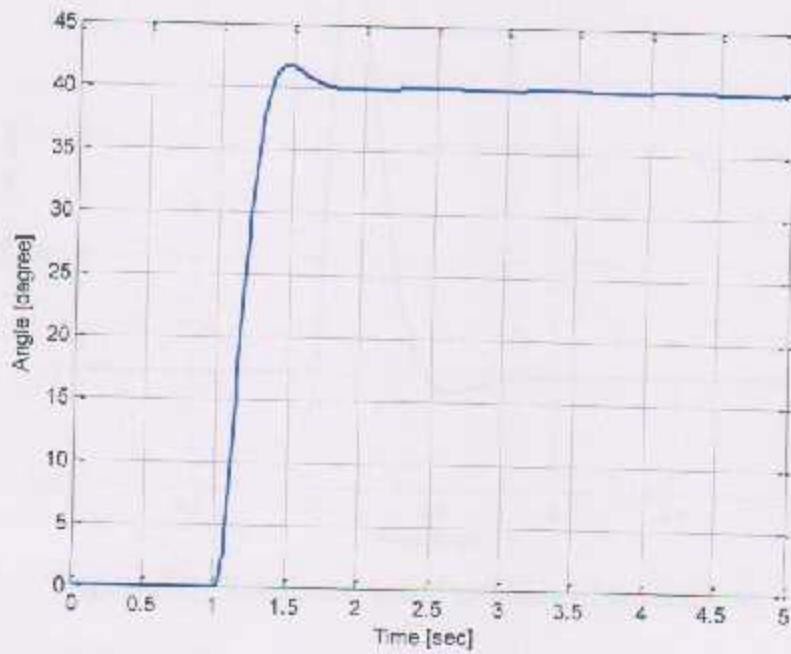


Figure 5.6 : Simulation position response of the actuator.

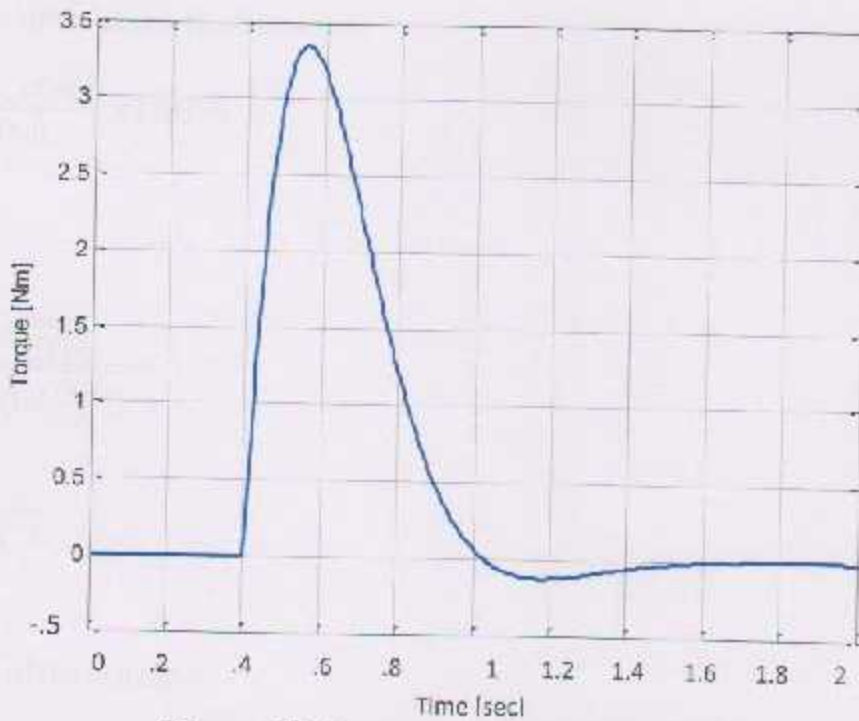


Figure 5.7 : Torque response of the actuator.

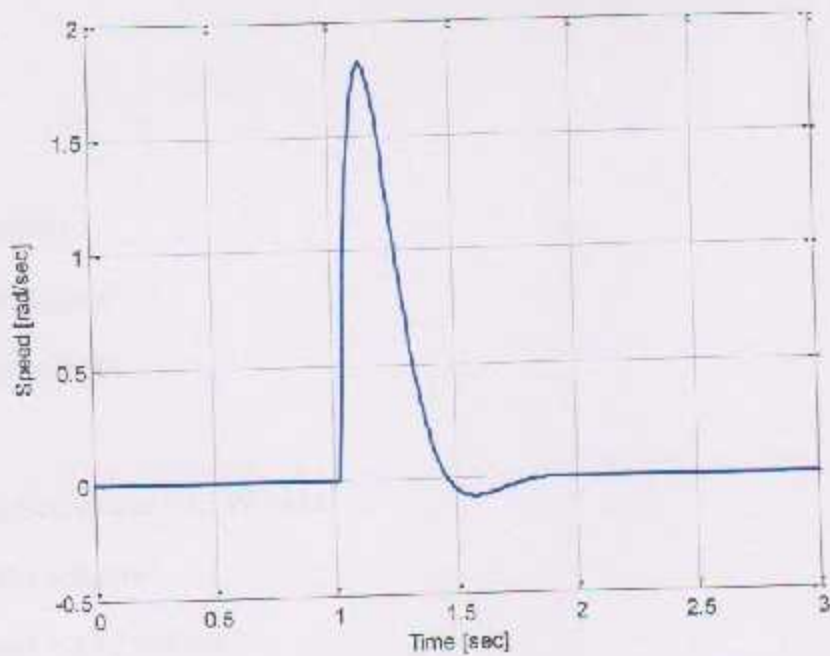


Figure 5.8 : Simulation speed response of the actuator.

The analysis of the simulation response:

$$\%OS = \frac{c_{max_cfinal}}{c_{final}} \times 100\%$$

$$T_s = \frac{4}{\omega\zeta}$$

$$\zeta = \frac{-\ln\left(\frac{\%OS}{100}\right)}{\sqrt{(\pi^2 + (\ln\left(\frac{\%OS}{100}\right))^2)}$$

$$T_p = \frac{\pi}{\omega\sqrt{1-\zeta^2}}$$

-For position of the actuator:

$$\%OS = \frac{0.417 - 0.4}{0.4} \times 100\% = 4.25\%$$

$T_s = 0.46$ sec.

$T_p = 0.26$ sec.

$\zeta = 0.7089$.

$\omega = 12.2652$ rad/sec.

final value = 40 degree.

steady state error = 0.0%

-For torque:

Maximum required torque = 3.349 N.m .

-For speed of the actuator:

Maximum speed = 1.82 rad/sec .

5.5 Real Results

Real response shown in figure 5.9 and 5.10.

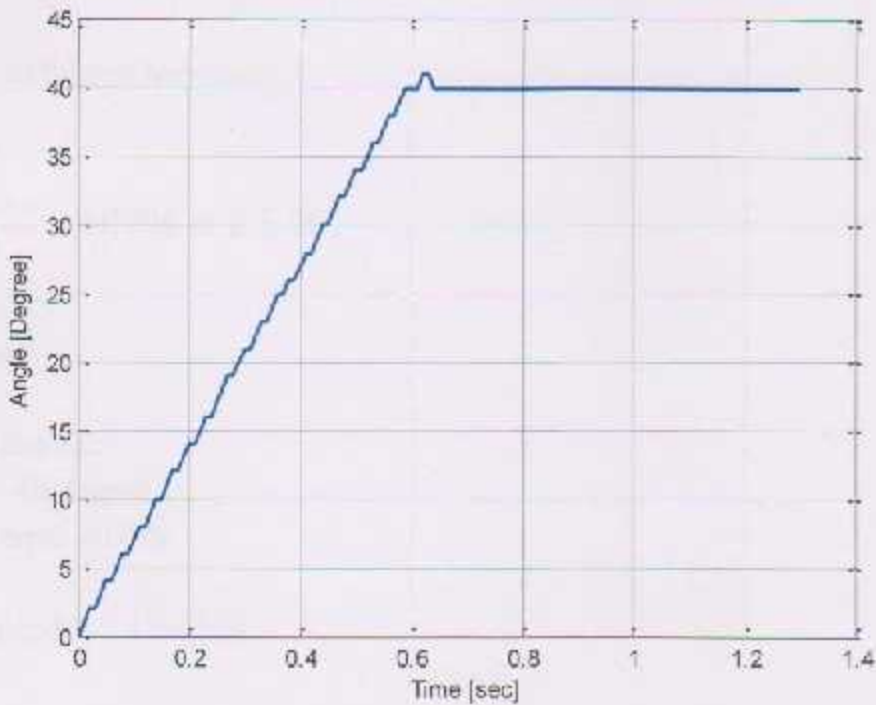


Figure 5.9: Real position response of the actuator.

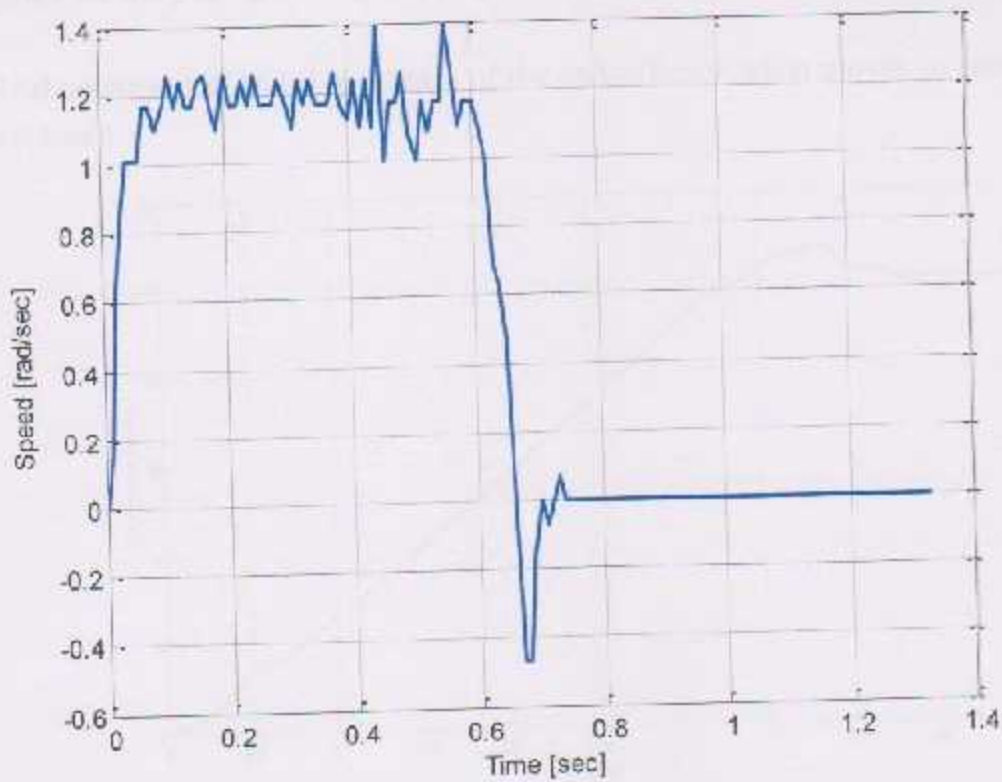


Figure 5.10: Real speed response of the actuator.

The analysis of the real response:

-For position:

$$\%OS = \frac{0.41 - 0.4}{0.4} \times 100\% = 2.5\%$$

$$T_s = 0.65 \text{ sec.}$$

$$T_p = 0.63 \text{ sec.}$$

$$\zeta = 0.7613$$

$$\omega = 8.0833 \text{ rad/sec.}$$

final value = 40 degree .

steady state error = 0.0%

-For speed:

Maximum speed = 1.4 rad/sec .

-Real results for the position and speed of the end-effector:

- Real position and speed responses of the end-effector when moves in one axis (x-axis):

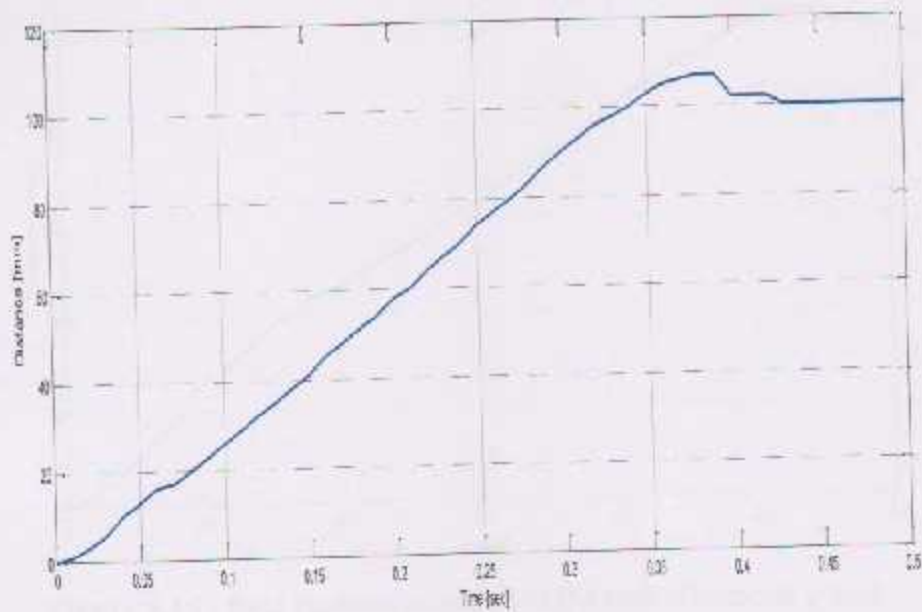


Figure 5.11 : Real position response of the end-effector in x-axis.

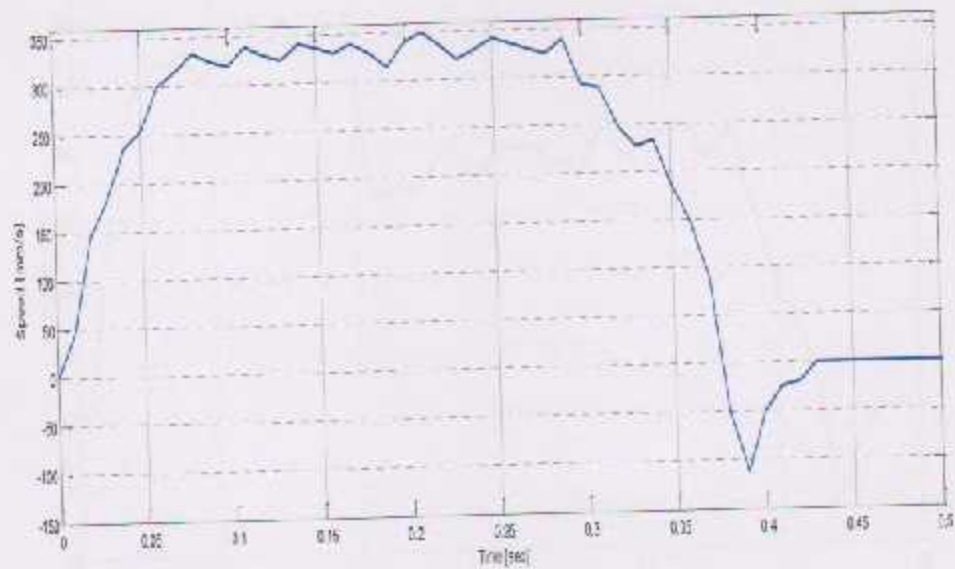


Figure 5.12 : Real speed response of the end-effector in x-axis.

- Real position and speed responses of the end-effector when moves in one axis (y-axis):

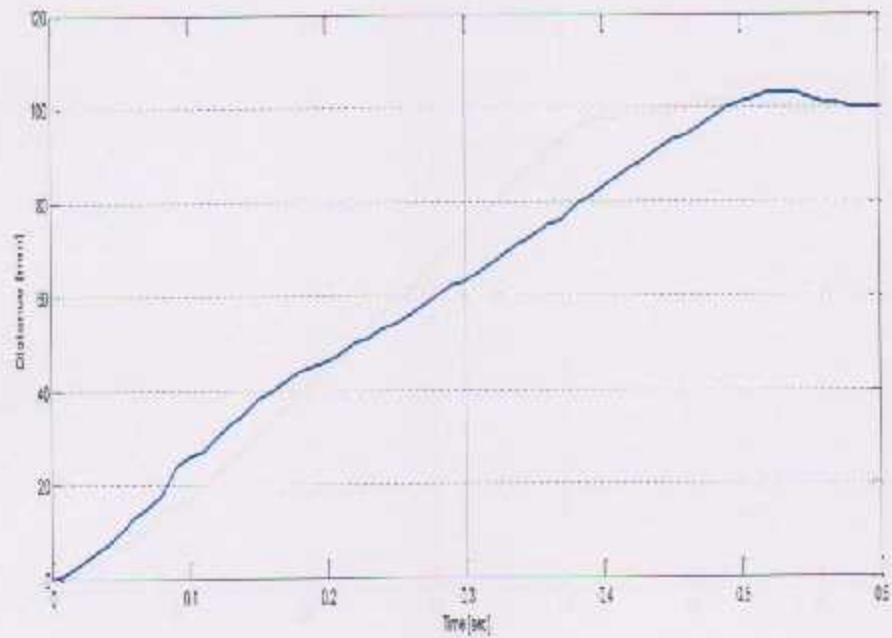


Figure 5.13 : Real position response of the end-effector in y-axis.

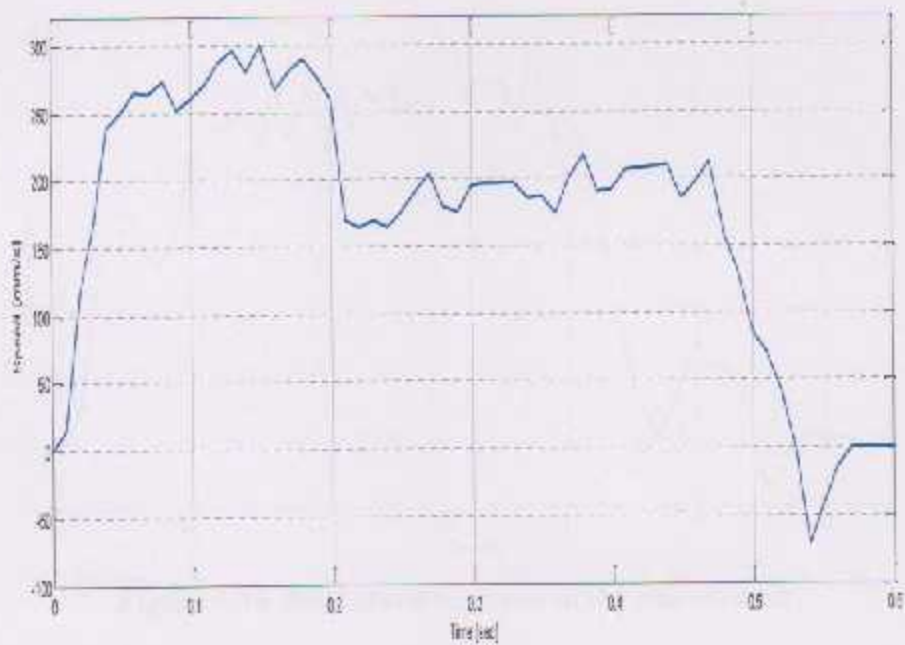


Figure 5.14 : Real speed response of the end-effector in y-axis.

- Real position and speed responses of the end-effector when moves in one axis (z-axis):

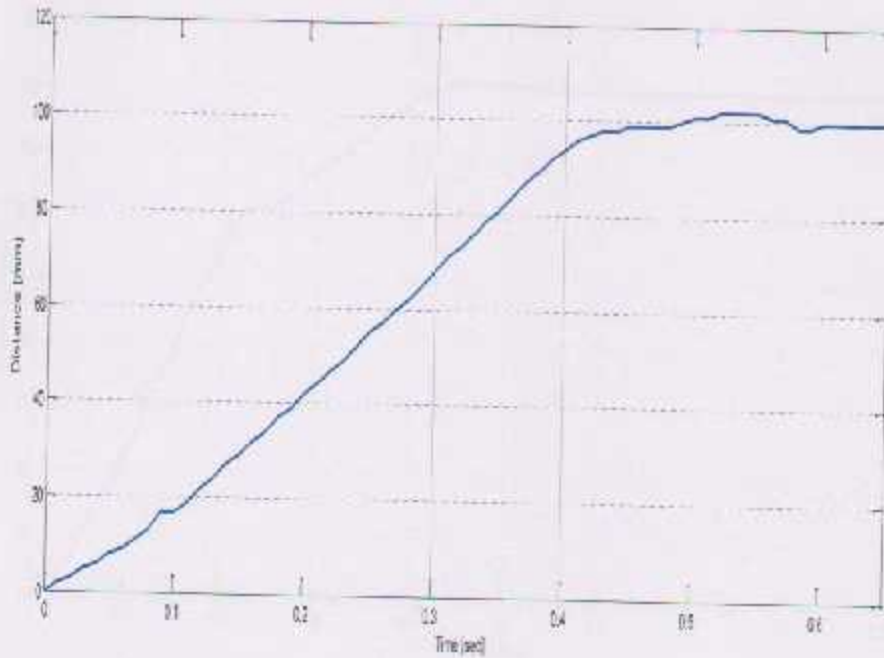


Figure 5.15 : Real position response of the end-effector in z-axis.

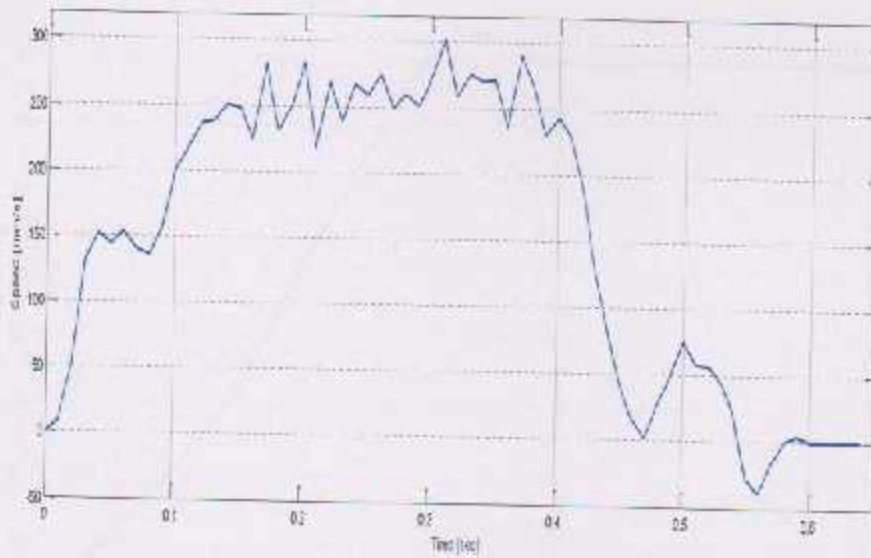


Figure 5.16 :Real speed response of the end-effector.

- Real position response of the end-effector when moves in two axes (x and y-axes):

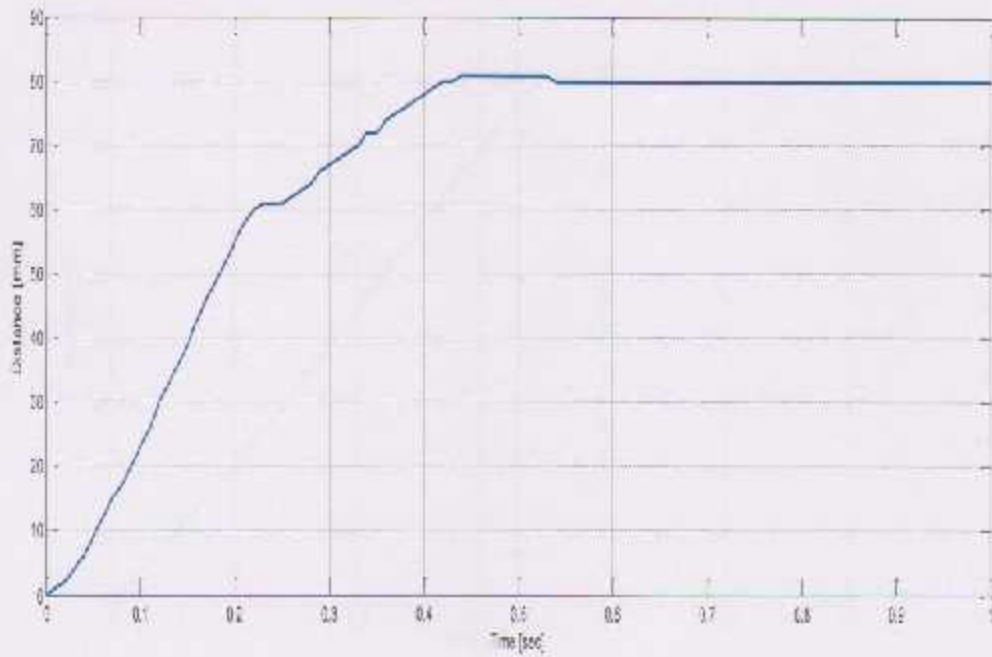


Figure 5.17 : Real position response of the end-effector in x-axis.

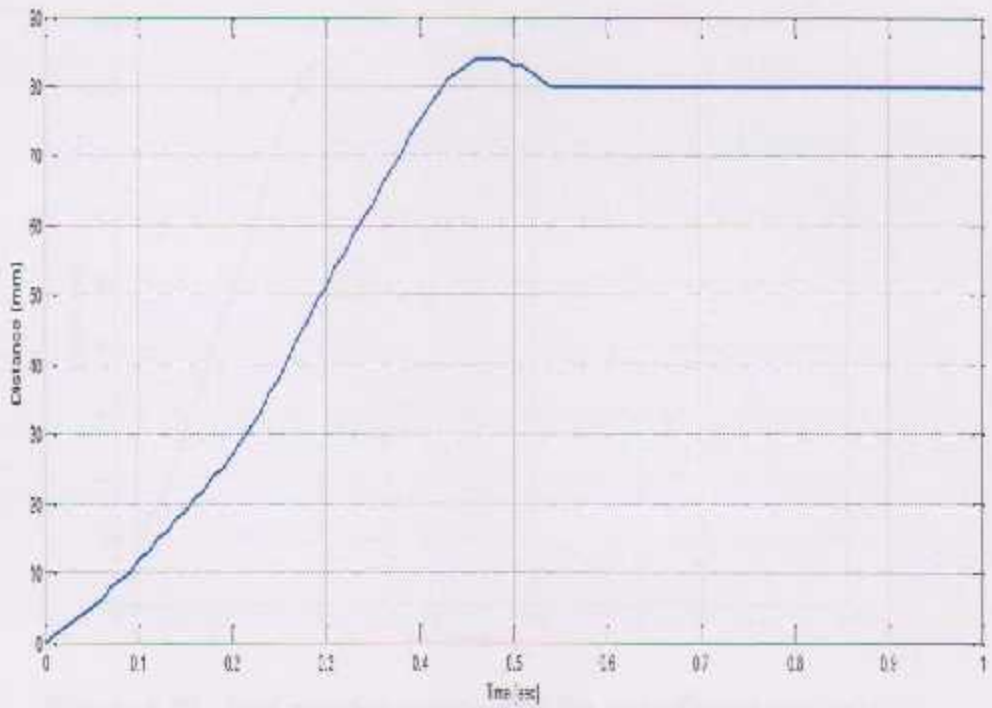


Figure 5.18 : Real position response of the end-effector in y-axis.

- Real position response of the end-effector when moves in three axes (x, y and z-axes) :

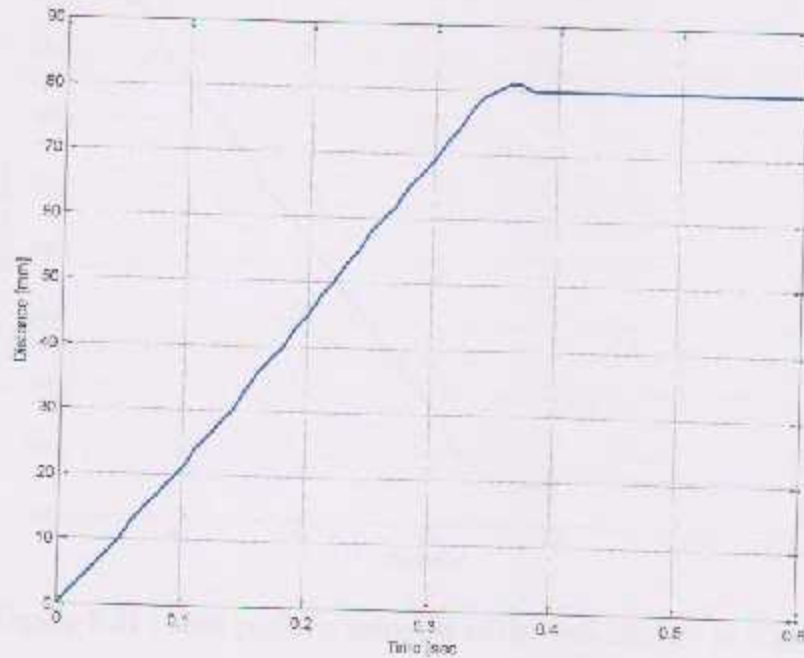


Figure 5.19 : Real position response of the end-effector in x-axis.

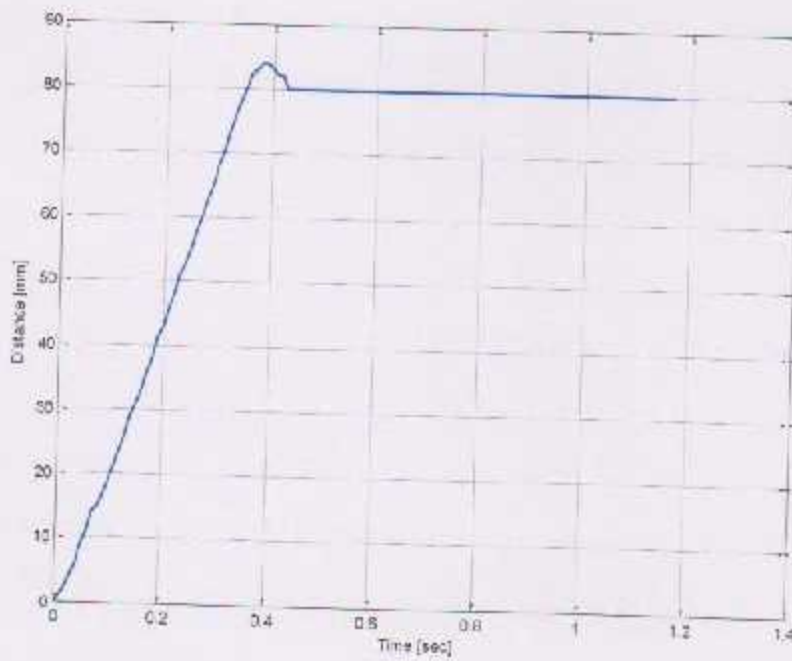


Figure 5.20 : Real position response of the end-effector in y-axis.

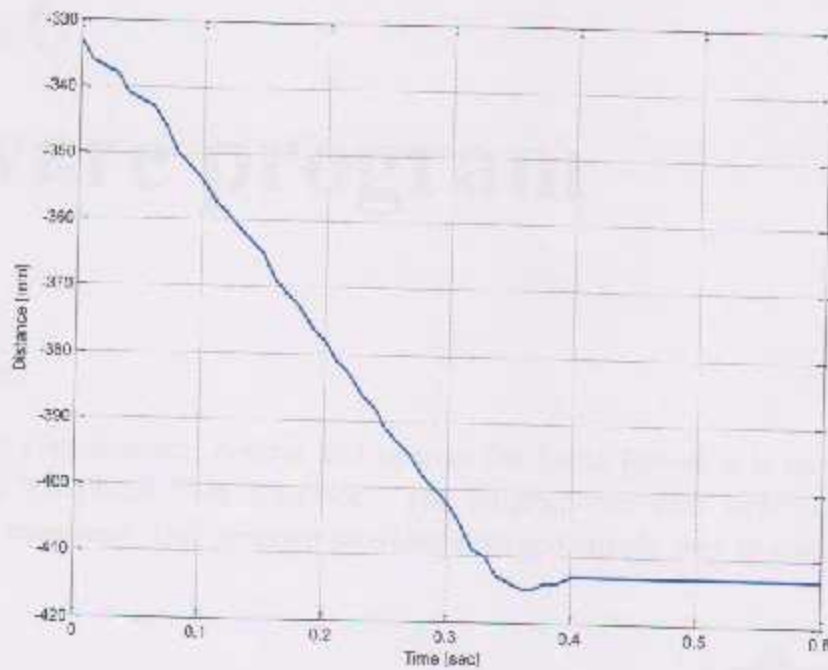


Figure 5.21 : Real position response of the end-effector in z-axis.

CHAPTER 6

Software program

6.1 Introduction

In order to communicate, control and operate the Delta Robot, it is required to use a software program "Graphical User Interface". The program can deal with the forward and inverse kinematic equations. This program provides easy and simple way to communicate with the Delta Robot.

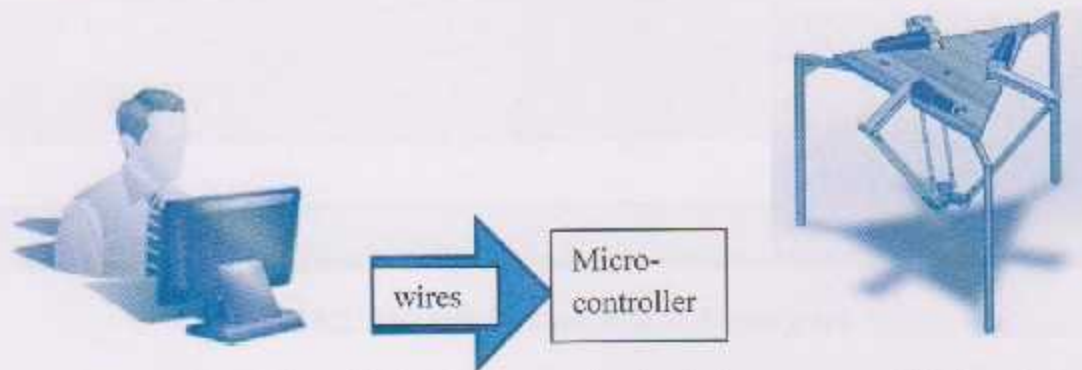


Figure 6.1: Schematic for the communication between user and Delta robot.

6.2 The program

This software program allow the user to write the commands, to move the end-effector to a desired position. For this system we have the commands which are the coordinates of the desired position if we want to move the end-effector by using the inverse kinematic equations. However, if we use the forward kinematic equation, the entered command will be the values of the angles for each actuator.

"Microsoft Visual Basic 2010 Express" is used for programming. This program can directly identify the connected Arduino microcontroller, connected port and can detect the baud rate of the microcontroller. The user needs to click on the button "Connect" in order to make the connection with Arduino microcontroller.

On the screen of the program as shown in figure 6.1, there is a command window, which appear on the button of the program's screen. In this command window the user can write the command that will be sent to the Arduino microcontroller, then the user must click on the "Send"

button to send the command to the microcontroller. Also, on the screen there is a window that show the position coordinates and the angles of the actuator of the Delta robot, for the current position.

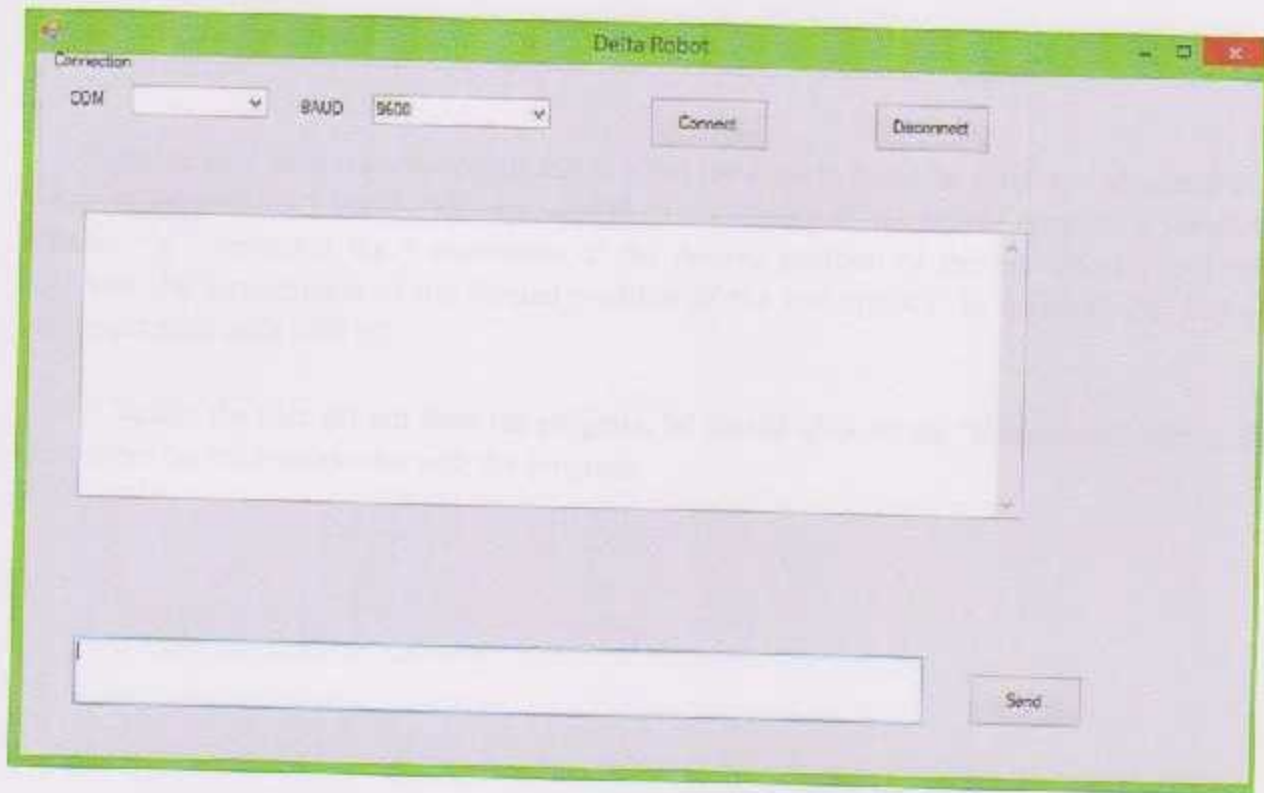


Figure 6.2: The interface window of the program.

The entered command has the following manners:

First, when controlling the Delta Robot by forward kinematic, a short message appear on the window in the program, this message is:

“ Forward Kinematics

Enter actuators angles such as:

= a10 b20 c30; ”

so the user must enter the command like this template in order the robot to understand the values of the actuator's angle. “a” represent the value of the angle the first actuator. “b” represent the value of the angle the second actuator and “c” represent the value of the angle the third actuator. The line of command must ends with “;”.

Second , when controlling the Delta Robot by inverse kinematic, a short message appear on the window in the program, this message is:

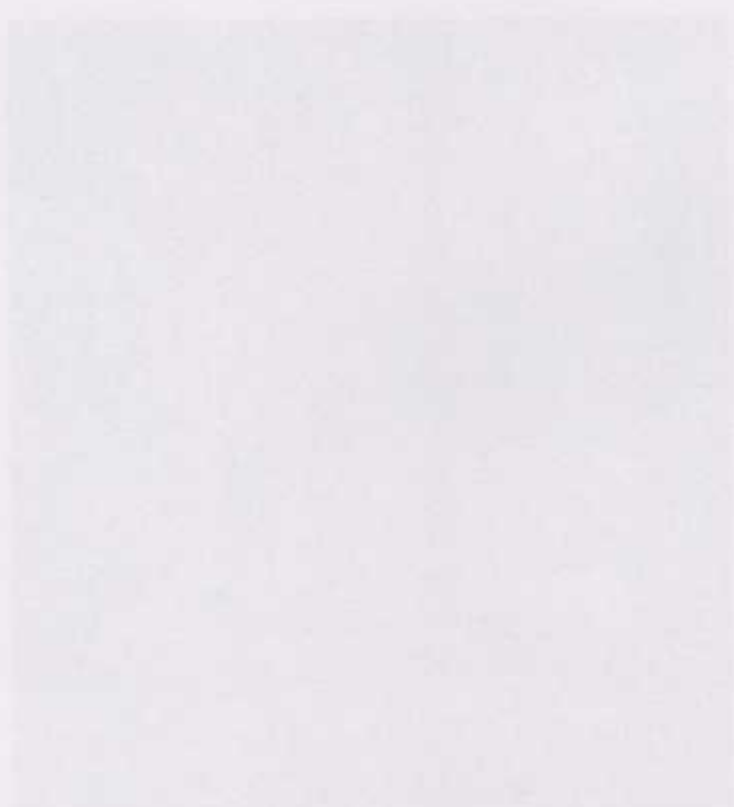
* Invers Kinematics

Enter Coordinates such as:

= x10 y20 z30; "

so the user must enter the command like this template in order the robot to understand the values of the actuator's angle. "x" represent the x-coordinate of the desired position of the end-effector, "y" represent the y-coordinate of the desired position of the end-effector and "z" represent the z-coordinate of the desired position of the end-effector. In addition, the line of command must ends with ";"

Before the user get out from the program, he should click on the "Disconnect" button, to disconnect the microcontroller with the program.



CHAPTER 7

Assembly

7.1 Introduction

Different components had used for building the Delta Robot. Mechanical components include the mechanical parts of the robot (upper links, lower links and end-effector) in addition to the fixed base and the steel structure which used to fixed the base on it. Electrical components include the DC servo actuators, drivers, encoders, current sensors and the microcontroller.

7.2 Assembly for the mechanical parts

Starting by the mechanical components. The base is fixed on the steel structure by using bolts nuts and washer, as shown in figure 7.1 .



Figure 7.1: The base fixed on the steel structure.

Then the actuators are fixed on the base by using bolts ,nuts and washers as shown in figure 7.2 .



Figure 7.2: The actuators are fixed on the base.

The mechanical parts of the robot (upper links, lower links and end-effector) are connected to each other as shown in figure 7.3, before be assembled with actuators as shown in figure 7.4 .

7.2 Assembly for the robot parts

Fig. 7.3 Mechanical parts of the robot



Figure 7.3: The mechanical parts of the robot be assembled together.



Figure 7.4 : The mechanical parts of delta robot be assembled with the actuators.

7.3 Assembly for the electrical parts

For the electrical parts, the encoders are coupled with the actuators. The drivers are connected to the actuators, power supply and the microcontroller. Also, the encoders are connected to the microcontroller. Figure 7.5 shows the connection of the microcontroller and drivers with the actuators and the encoders.



Figure 7.5: The connection of the microcontroller and drivers with the actuators and encoders.

Now, labels are added to each wire in order to facilitate the identification of the wires connection or direction. Also, the current sensors are connected with the microcontroller and the actuators as shown in figure 7.6 .



Figure 7.6: The labels are added and the current sensors connection.

A plastic cover be designed to cover the electrical components and wires. Also, to give protection for these components to be out of touched by the people. In addition to give the robot beautiful appearance, the plastic cover is fixed around the base and the electrical components as shown in figure7.7 .



Figure 7.7: The plastic cover of the electrical components.

7.4 Schematic wire diagram for the electrical components

Figure.8 shows the schematic wiring diagram for the electrical components that used for the Delta Robot.

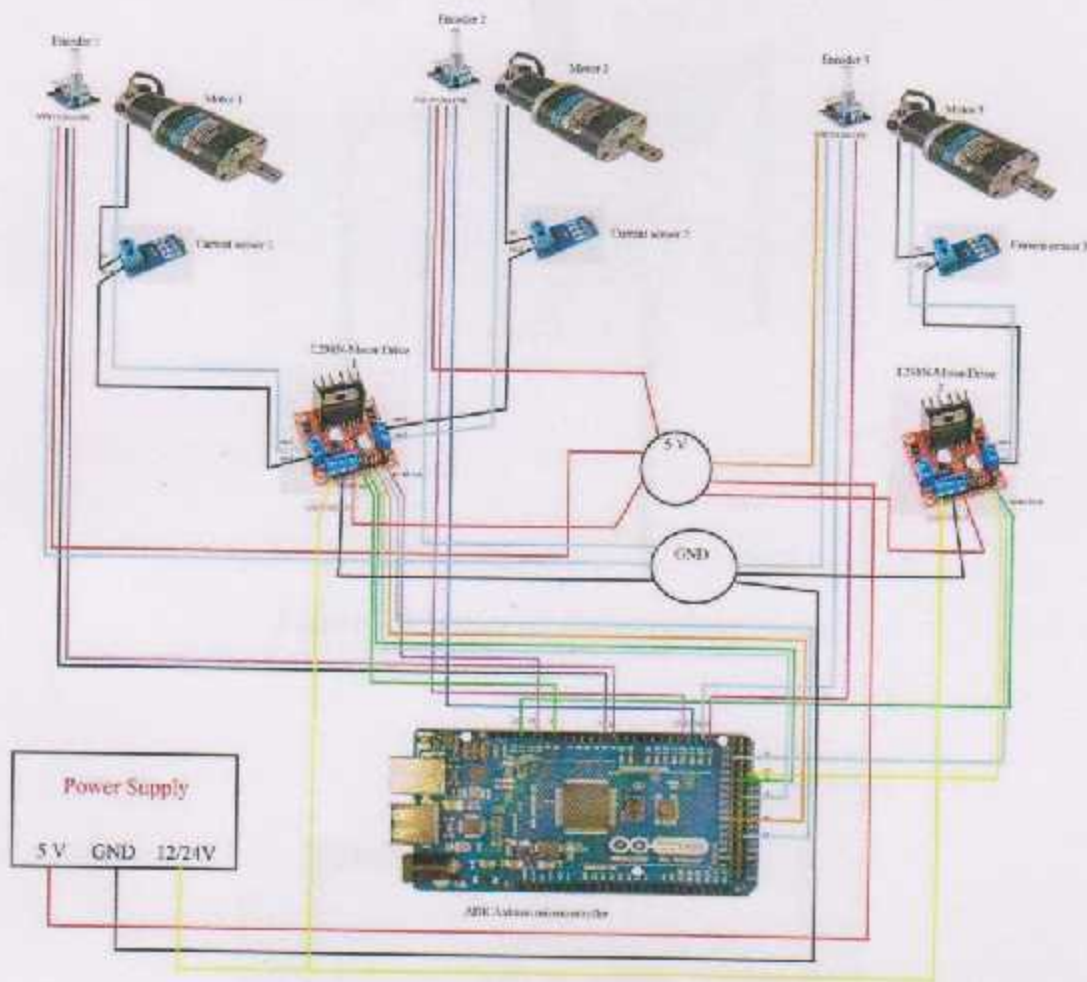


Figure 7.8: The wiring diagram for electrical components.

7.5 Workspace analysis

The workspace of the Delta Robot is limited due to the length of the links(upper and lower links). The target objects ,which will be dealing with, must be within the workspace of the Delta Robot, where the end-effector can move to deal with these objects.

The z-axis, which represent the vertical distance of that the end-effector can move along, he limited to (-255 to -530)mm, according to the axis which fixed on the base, as shown in figure 7.9 and figure 7.10 . The total distance that the end-effector can move along z-axis is 275mm.



Figure 7.9: Minimum distance along z-axis.

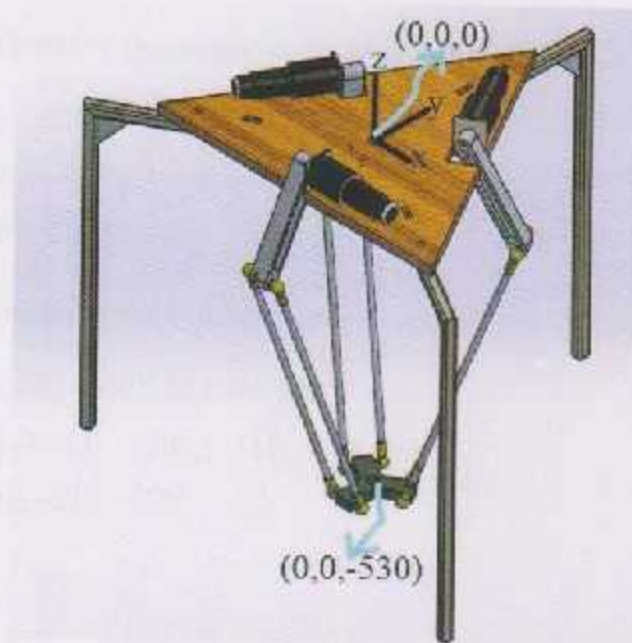


Figure 7.10: Maximum distance along z-axis.

For the x & y-axes the distance is symmetrical. The distance along x & y-axes is different depending on the distance along the z-axis. For the minimum distance along z-axis, the distance along (x & y-axes) is 10mm, as shown in figure 7.11 .

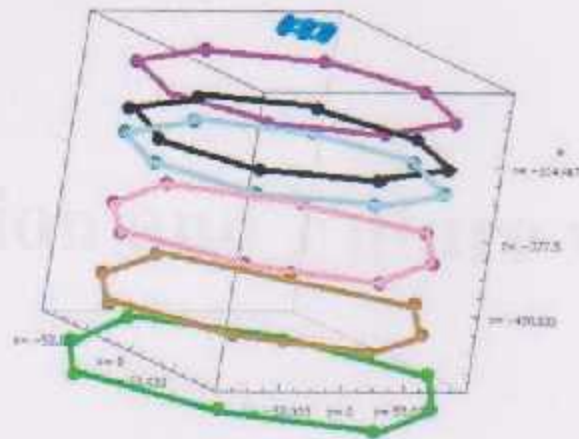


Figure 7.11: The workspace of the Delta Robot.

For $z = -255$, the coordinates of the points (for dark blue curve shown in figure 7.11) are:

$$\left\{ \begin{array}{l} \{20, 0, -255\}, \{-20, 0, -255\}, \{0, 20, -255\}, \\ \{0, -20, -255\}, \{10, 10, -255\}, \{-10, 10, -255\}, \\ \{10, -10, -255\}, \{-10, -10, -255\} \end{array} \right\}$$

For $z = -300$, the coordinates of the points (for purple curve shown in figure 7.11) are:

$$\left\{ \begin{array}{l} \{110, 0, -300\}, \{80, 80, -300\}, \{0, 110, -300\}, \\ \{-80, 80, -300\}, \{-110, 0, -300\}, \{-80, -80, -300\}, \\ \{0, 110, -300\}, \{80, -80, -300\} \end{array} \right\}$$

For $z = -530$, the coordinates of the points (for green curve shown in figure 7.11) are:

$$\left\{ \begin{array}{l} \{150, 0, -530\}, \{110, 110, -530\}, \{0, 150, -530\}, \\ \{-110, 110, -530\}, \{-150, 0, -530\}, \{-110, -110, -530\}, \\ \{0, -150, -530\}, \{110, -110, -530\} \end{array} \right\}$$

And so on.

CHAPTER 8

Conclusion and Future work

8.1 Conclusion

The purpose of this project was to build and control 3DOF parallel DELTA robot structure. A summary of differences between serial and parallel robot structure introduced the background of robotics, while the literature review provided a detailed account of the beginning of robotics.

The 3DOF parallel DELTA robot was built in the real life, and TRACKER controller was used to control of the DELTA robot. After many experiments on the robot by moving it in three coordinates (x, y and z) the results was showed that the robot move in accurate manner. And it can be follow the entered commands from the user. All of these was in joint space level.

Finally, the proposed model in this project will open the road to master students who wish to continue their graduate study in the field of parallel robots.

8.2 Future work

The work was in joint space level, but we can also work in task space level, by using accelerometer for example. Also modify the user interface to be by android OS.

We will put suction air valve on the end-effector in order to make the robot perform specific work.

References

- [1] P. P. S. Küçük, Serial and Parallel Robot Manipulators Kinematics, dynamics, control and optimization, (March, 2012),
- [2] E. H. A. H. Alashqar, *Modeling and High Precision Motion Control of 3 DOF Parallel Delta Robot Manipulator*, 2013.
- [3] L. C. Neves, *modeling and control of parallel manipulator robot*, September, 2009.
- [4] J.-P. MERLET, Parallel Robots (Second Edition), SOLID MECHANICS AND ITS APPLICATIONS, Volume 128 (2006 ,Springer),
- [5] V. Poppeová, J. Uriček, V. Bulej, and P. Šindler, Delta robots—robots for high speed manipulation, *Tehnički vjesnik*, 18 (2011),pp. 435-445.
- [6] N. Simaan, *Analysis and synthesis of parallel robots for medical applications*, Technion-Israel Institute of Technology, Faculty of Mechanical Engineering, 1999.
- [7] S. St. and C.-C. D. C., *DYNAMIC ANALYSIS OF CLAVEL' S DELTA PARALLEL ROBOT*, 2003.
- [8] S. Stan, M. Manic, C. Szep, and R. Balan, Performance analysis of 3 DOF Delta parallel robot, *Human System Interactions (HSI)*, 2011 4th International Conference on, 2011.
- [9] F. Sanfilippo and D. Prattichizzo, *STUDY OF A PARALLEL DELTA ROBOT CONTROLLED VIA IPHONE ACCELEROMETER*, (2010),
- [10] E. C. M López¹, G García^{a3}, and A Bashir⁴, Delta robot: inverse, direct, and intermediate Jacobians,
- [11] M. López, E. Castillo, G. García, and A. Bashir, Delta robot: inverse, direct, and intermediate Jacobians, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 220 (2006),pp. 103-109.
- [12] M. M. Sergiu-Dan Stan¹, Cristian Szep¹ and Radu Balan¹, Performance analysis of 3 DOF Delta parallel robot, (2011).
- [13] P. Vischer and R. Clavel, Kinematic calibration of the parallel Delta robot, *Robotica*, 16 (1998),pp. 207-218.

- [14] K. Hsu, M. Karkoub, M.-C. Tsai, and M. Her, Modelling and index analysis of a Delta-type mechanism, *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, 218 (2004),pp. 121-132.
- [15] I. Bonev, Delta parallel robot—the story of success, Newsletter, available at <http://www.parallelmic.org>, (2001).
- [16] K. Miller, Dynamics of the new UWA robot, *Proc. Australian Conference on Robotics and Automation*, 2001.
- [17] P. Zsombor-Murray, "An improved approach to the kinematics of Clavel's DELTA robot," Online, 2009.
- [18] J. L. Irwin, J. Pearce, G. Anzalone, and D. Oppliger, The RepRap 3-D Printer Revolution in STEM Education, 121st ASEE Annual Conference & Exposition, 2014.
- [19] N. Asadi and M. Jackson, Lightweight Robotic Material Handling in Photovoltaic Module Manufacturing-Silicon Wafer and Thin Film Technologies, *Dimension (mm)*, 125 (2012),p. 156x156.
- [20] A. Liadis, fuzzy logic control of a two degree of freedom parallel robot, (2010).
- [21] John J. Craig, "Introduction to robotics, Mechanics and Control, Third Edition, 2005.
- [22] Mohsen, Mahdi, Mersad," Dynamics and Control of a Novel 3-DoF Spatial Parallel " Robot", *International Conference on Robotics and Mechatronics*,2013.