Palestine Polytechnic University

College of Engineering & Technology

Mechanical Engineering Department

DeltaRobot

Design and control for the educational application

Produced by

Hamizeh Ali Awawdeh

Yousef Jamil Allan

AduyKhamis Al-Mohtaseb

Supervisor:

Dr. IyadHashlamon

Hebron – Palestine

May, 2016

ABSTRACT

Delta robots are parallel robots that are composed of three parallel kinematic chains connected at the base of the robot and at the end effector.

The design features of the robot maintains the orientation of the end effector. This robot is characterized by the high acceleration, accuracy and precision.

Therefore, they are used heavily in the industry. This project aims to develop an educational delta robot where the students can apply the principles of multi input multi output control algorithms, use the forward and inverse kinematics for control and trajectory planning, and apply intelligent control approaches including sensor fusion.

Furthermore, the delta robot is equipped by electronic items to implement the network control approach and wireless control

And as an educational robot, the safety of both the human being and the robot itself has been studied intensively and the required electrical and electronic components are added.

TABLE OF CONTENTS

LIST OF FIGURES

Figure	Page

Figure	Page

1

Chapter one

Introduction

1.1 Introduction

High-speed parallel robots with low movement of inertia andhigh acceleration that's is delta robot in brief.
This chapter introduces a summary of the differences between serial and parallel robot structures, related work, then the motivation of the project and problem definition .

1.2 Background

There are essentially two types of robot manipulators: serial and parallel. Serial manipulators consist of a number of links connected in series to one another to form a kinematic chain. Each joint of the kinematic chain is usually actuated. This type of structure is known as an open chained mechanism as shown in Figure 1.1 [8].

Figure 1.1: Serial Robots.

Parallel Robotsalso called parallel manipulators, are closed-loop mechanisms, characterized by their high accuracy, velocity, rigidity and ability to move large loads. Due to these advantages 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 [1].

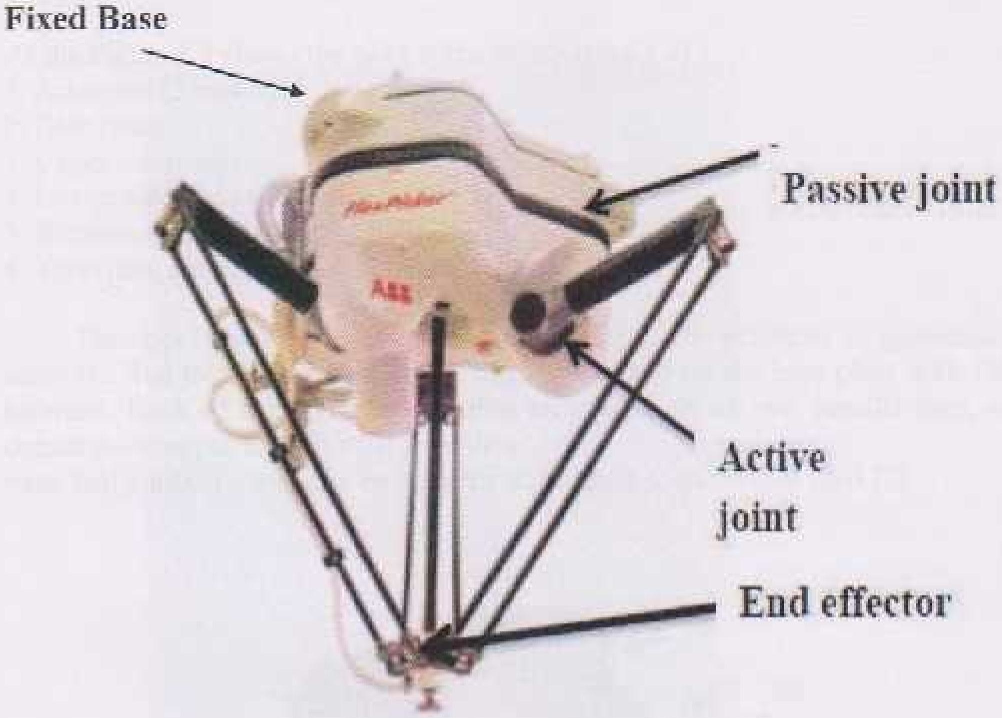


Figure 1.2: A parallel manipulator.

An n degrees of freedom parallel robot consists of an end-effector and a fixed base as seen in Figure 1.2. 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 . The actuators in parallel robots are fixed to the base, which makes the moving parts relatively light[1,3].

1.2.1Delta Robot

Delta robot transports products with high precision from the acceptance location to the target location. It can move products in a three dimensional Cartesian coordinate system. The combination of the constrained motion of the three arms connecting the traveling plate to the base plate ensues in a resulting 3translator degrees of freedom (DOF). As an option, with a rotating axis at the Tool Center Point (TCP), 4DOF are possible [1,3,4].

As the Figure 1.3 shows the delta robot which consist of :

1. Actuators (3 motors)
2. Base plate
3. Upper robot arm
4. Lower robot arm (Forearm)
5. Rotation arm (optional, 4-DOF)
6. Travelling plate, TCP (end effector)

The upper robot arms are mounted directly to the actuators to guarantee high stability. And the three actuators are rigidly mounted on the base plate with 120° in between. Each of the three lower robot arms consists of two parallel bars, which connects the upper arm with the travelling plate ball joints. To measure each motor shaft angle a encoder is used [2].

Figure 1.3: Delta robot

1.3 Previous Studies

Modeling and control of a Closed Chain Parallel DELTA Robot is very difficult especially, when using traditional methods in modeling.

- YangminLi, QingsongXu [9] proposed the simplified dynamic equations derived via the virtual work principle on 3-TRC translational parallel kinematic machine.

- André Olsson [2] describes the virtual work principle mathematical modeling of a Delta-3 robot actuated by motors and drive units. Experiments with comparison between the Simulink model and the real robot are done.

- Yangmin Li and QingsongXu [7] performed inverse dynamic modeling based upon the principle of virtual work for medical Delta Robot. The dynamic control uses computed torque method.

- Mohsen, Mahdi, Mersad[6] describes the Dynamics modeling and trajectory tracking control of a new structure of spatial parallel robots from Delta robots family. This paper compared implementation of computed torque (C-T) method using adaptive Neuro-fuzzy controller and conventional PD controller.

- Angelo Liadis[5] proposed Lagrangian principle for modeling 2 DOF parallel robot, and introduced eight controllers , fuzzy and non-fuzzy controllers. Experiments with comparison between the Simulink model and the real robot are done.

1.4Problem Definition

The optimal control problem can be stated as: find a closed loop optimal controller that minimizes error between the measured phase and actual phase wanted to track specified path. Optical encoder is attached in the end of each Servo motor shaft, measuring the actual phase of the link and from that; we can calculate speed and acceleration. Controlling of Delta Parallel robot wants true modeling for its dynamics, so, by using the Prototype, we can model the robot easily and test its motion in Simulink Matlab tool.

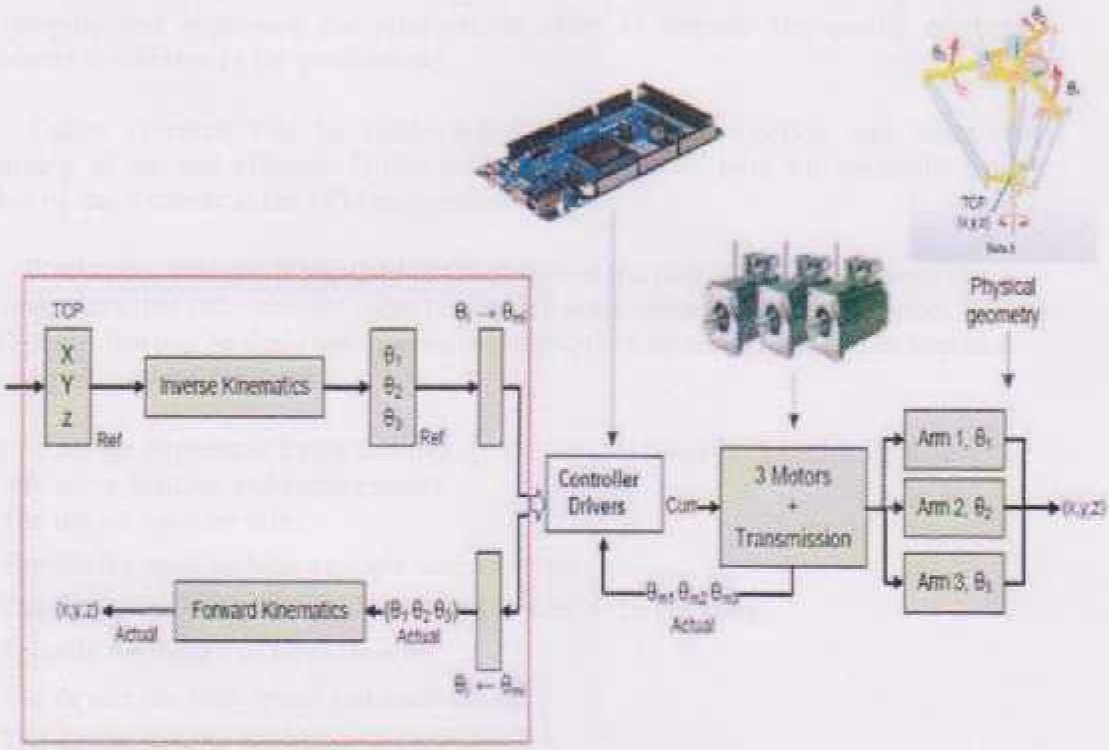


Figure 1.4: DELTA Robot Control System.

The motors that are used to actuate the three arms of the Delta-3. From Figure1.4, Controller unit (microcontroller) calculates inverse kinematics of reference position x,y,z of moving platform in delta robot , actuators drive servo motors under the effect of controller leading the end effector to the target point in minimum time and no error as possible .

1.5 Motivation

Modeling and Digital control of Dynamic system was our target of our project; modeling of mechanical systems. No need to extract the dynamic equations of robot. Simulink tool of Matlab can simulate the robot as equations built in Prototype located. This study Thesis treats the modeling of the Parallel DELTA robot actuated with Servo DC motors and drive units. Also the kinematics for a Delta-3 robot is implemented to be able to see where the traveling plate has for position.

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 controllers 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.

1.6 Time Table

First semester:

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															■	■

Task 1: Choosing the idea of the project.

Task2: Collecting the information that related to the project idea / Documentation.

Task 3: Deriving the kinematic and dynamic equations.

Task4: Designing a controller and simulation.

Task 5: Selecting the components,structure and ordering them.

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: Connecting the parts together

Task 7: Writing the controlling code for microcontroller.

Task 8: Run the project

Task 9: Doing experiments and documentation.

Task 10: Find applications for the project

2

Chapter two

Delta Robot, The MECHATRONICS SYSTEM_ DESIGN Approach

2.1 Introduction

2.2Recognition of need

2.3 Conceptual Design

2.4 Sensor And Actuator Selection

2.5 Mathematical modeling

2.6 Simulation

2.1 Introduction

The mechatronics design methodology is based on a concurrent approach instead of sequential to discipline design , resulting in products with more synergy.

This chapter introduces an explanation about the Delta Robot as a mechatronics project we will provide an overview of the mechatronics design process and a general description of the technologies employed in our project as the mechatronics approach.

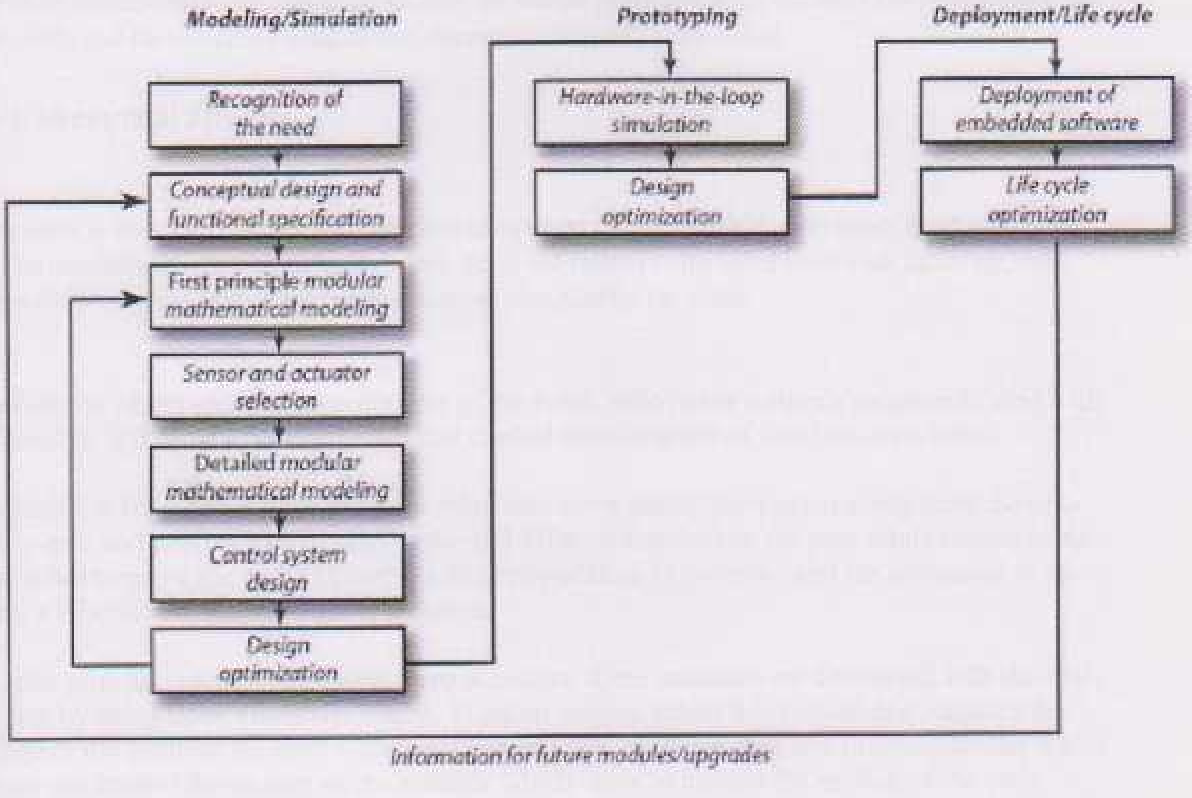


Figure 2.1 : Mechatronics design process

The mechatronics design process as shown in figure 2.1 consists of three phases : modeling and simulation ,prototyping , and deployment .

2.2 Recognition of need

The recognition of need of our project which is to develop an educational delta robot where the students can apply the principles of multi input multi output control algorithms, use the forward and inverse kinematics for control and trajectory planning, and apply intelligent control approaches including sensor fusion .

Furthermore, the delta robot is equipped by electronic items to implement the network control approach and wireless control

And as an educational robot, the safety of both the human being and the robot itself has been studied intensively and the required electrical and electronic components are added.

2.3 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 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 is required.

The end-effector can move by using three actuators, these actuators are connected with the end-effector by using three 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 supply the actuators by the required power. The electrical actuators will be fixed on the base.

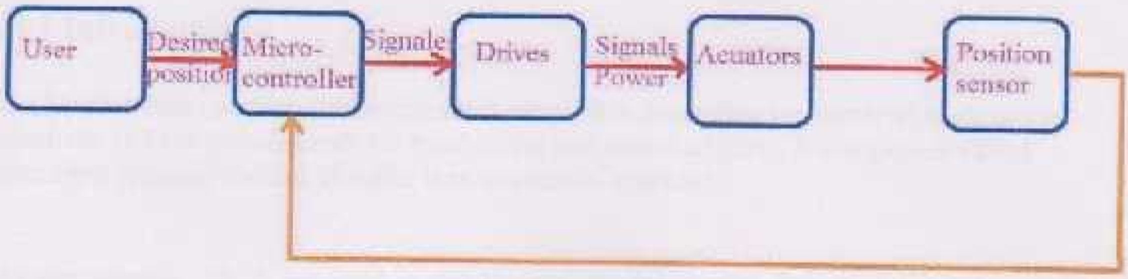


Figure 2.2: Block diagram of the system.

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 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. Also 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 as show in figure 2.2.

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 them , here the user can save these positions in the C-code.

2.4 Sensor and Actuator Selection

2.4.1 Introduction

As a Mechatronics project, the components are chosen depending on 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, Arduino Ethernet Shield, Display 16x4 LCD, Black 4x4 Keypad, 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.

2.4.2 Micro-Controller

The chosen micro-controller is Arduino as shown in Figure2.3.The Arduino is the first Arduino board based on a 32-bit RAM core microcontroller. With 54 digital input/output pins, 12 analog inputs, it is the perfect board for powerful larger scale Arduino projects.

The board contains is ready to be used by connecting it to a computer with a micro-USB cable or power it with a AC-to-DC adapter or battery to get started. The specifications of this Arduino are in table 2.1.

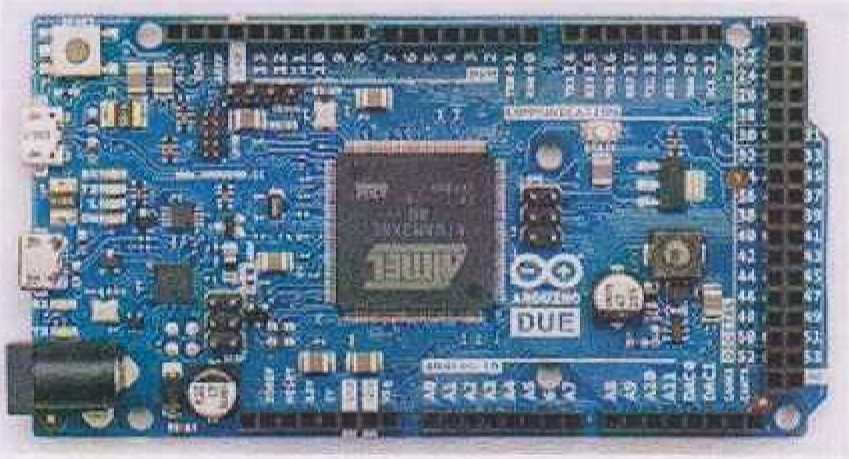


Figure 2.3: Arduino Due

Table 2.1: Technical specification of Due Microcontroller.

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-16V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Output Pins	2 (DAC)
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB all available for the user applications
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g

The Arduino Due can be powered via the USB connector or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

2.4.3 Arduino Ethernet Shield

The Arduino Ethernet shield as shown in Figure 2.4, allows an Arduino board to connect to the internet using the Ethernet library to read and write an SD card using the SD library.

The chosen Arduino Ethernet Shield is to control the system through the network.

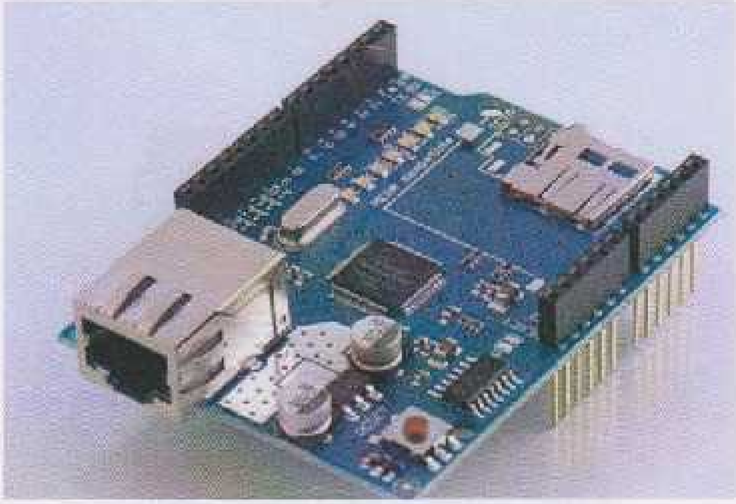


Figure 2.4:Arduino Ethernet Shield

2.4.4 Display 16x4 LCD

The chosen Display 16x4 LCD as shown in Figure 2.5. is to show the three measured angles, according to the three used actuators ($\theta_1, \theta_2 and\ \theta_3$) and the three direction axis X , Y and Z.

Figure 2.5: Display 16x4 LCD

2.4.5 Black 4x4 Keypad

The chosen Black 4x4 Keypadas shown in Figure 2.6, is used for data entry (the three angles ($\theta_1, \theta_2 and\ \theta_3$),and the three direction X, Y and Z) to Micro-Controller .

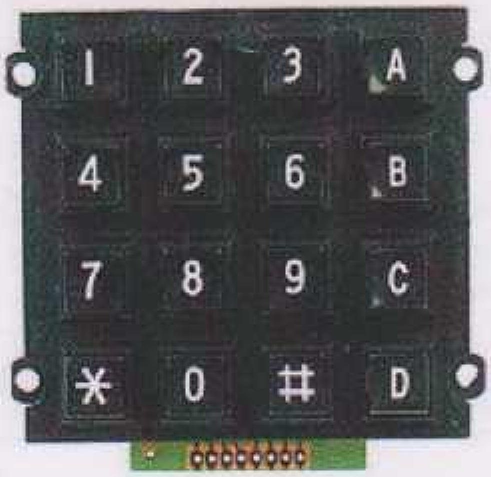


Figure 2.6: Black 4x4 Keypad

2.4.6 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 Figure 2.7. Delta Robot has three angles to be measured, according to the three used actuators($\theta_1, \theta_2 and\ \theta_3$), respectively.

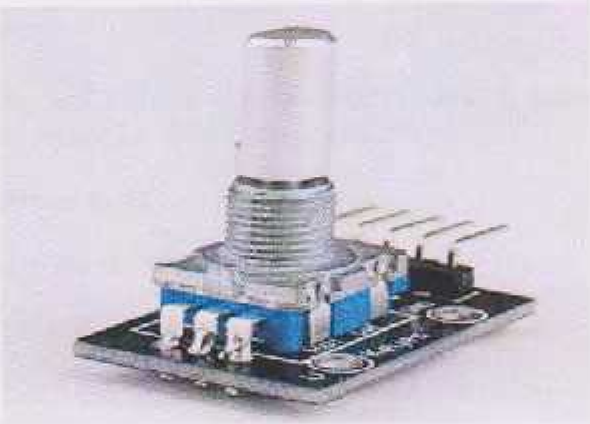


Figure 2.7:Rotary encoder.

The most common angular position sensors are encoders. Encoders have two types: absolute encoders and incremental encoders.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.

The incremental encoder can be used instead of the absolute encoder because we detect an initial position for the Delta Robot.

2.4.7 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) x13 (mm), 14-lead, plastic package. as shown in Figure 2.8.

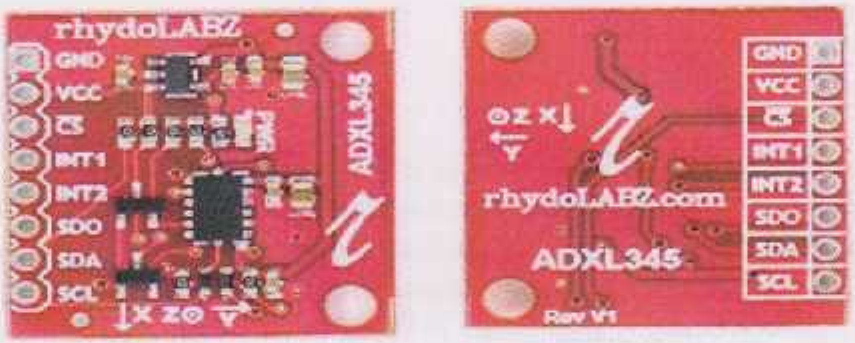


Figure 2.8:ADXL345 accelerometer.

2.4.8 Current sensor

ACS712, as shown in figure 2.9, is a precise current sensor. It can be used to make torque control for the robot. 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.

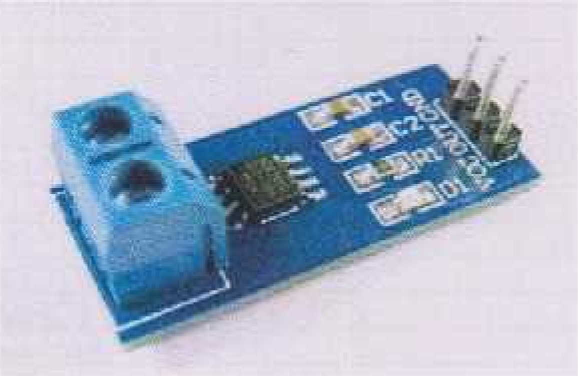


Figure 2.9:ACS712current sensor.

2.4.9 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 2.10, combining precision Harmonic Drive gear and DC servo motors offer unique features unsurpassed by conventionally geared drives.

Figure 2.10: RH mini-series DC servo Actuators.

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] Electrical Time Constant [ms]	1.1 0.41
Starting Current[A]	0.43
No-Load Running Current[A]	0.91

2.4.10 Motor's Driver

In order to control the direction and the operation of the motor, a driver is required. The L298 is an integrated monolithic circuit as shown in Figure 2.11. 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.

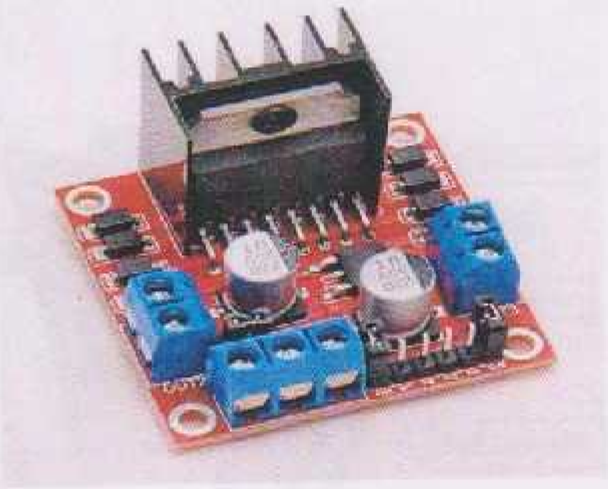


Figure 2.11: L298N Motor Driver Module

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.

2.5 Mathematical modeling

2.5.1 Introduction

Delta robot consists of three closed-loop kinematic chains, each chain represent parallelogram and it consists fixed link and two moving links as shown in Figure 2.12.In this section, we will study the kinematics and Dynamics of 3 DOF Parallel DELTA robot.

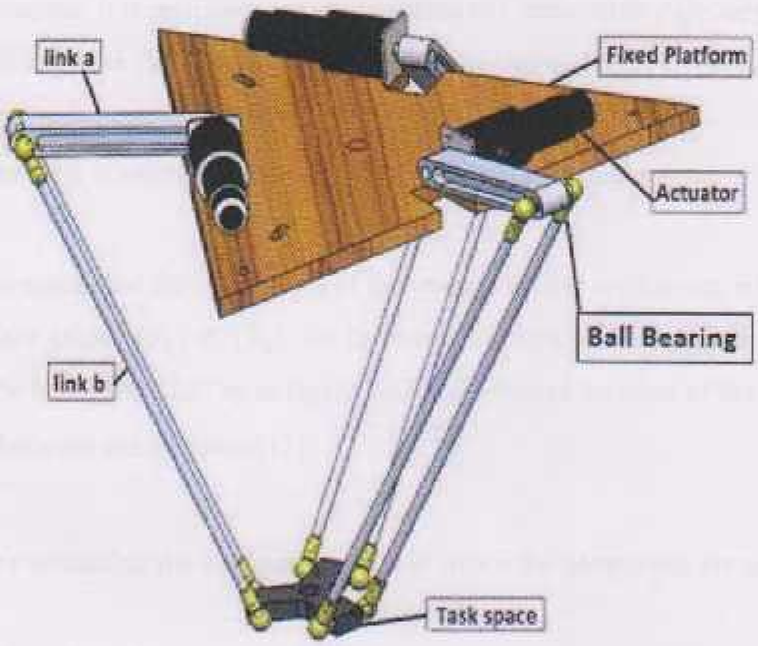


Figure 2.12: Delta Robot

2.5.2 Inverse Kinematics

Inverse kinematics solution used in order to find the joints space (θ_1 , θ_2 , θ_3), for known task space position (*x, y, z*).

The purpose of determining the inverse kinematics of this parallel robot is to accurately model the angle produced at each joint at a specific location of the effector. This is advantageous for two main reasons; it is relatively simple to define any reasonable trajectory for the end effector to travel and it can track different trajectories in a non-singular region [10,5].

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 (θ_1 , θ_2 , θ_3), for the overall length of the upper link 'La' and the overall length of the lower link 'Lb' as in figure 2.13. The desired location of the task space (*x, y, z*) and the angles between the actuators[11].

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: *La*
- Length from the spherical joint J1 until the spherical joint E1: *Lb*

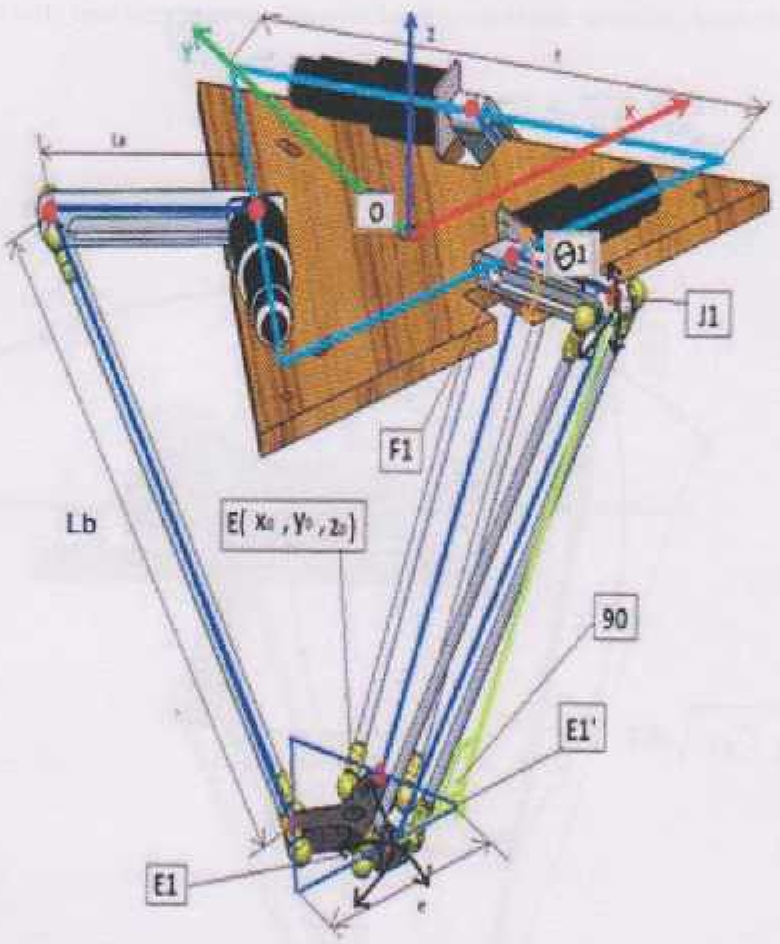


Figure2.13: Equilateral triangle on the fixed platform, fixed and spherical joint in delta Robot.

The point J1 can be found as the intersection of two circles. One with center in F1 and radius *Lb*. Other with center in E1' and radius: $\sqrt{L_b^2 - X_0^2}$.

We should choose only one intersection, the smaller Y coordinate solution, as shown in Figure 2.14.

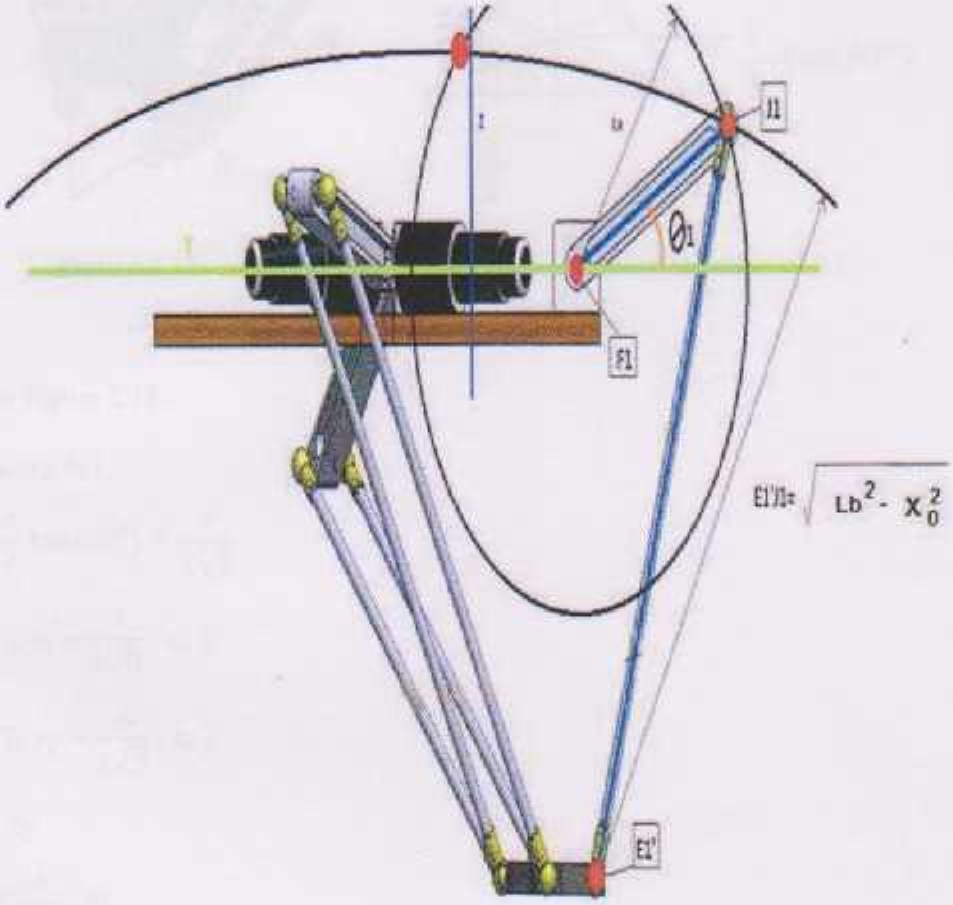


Figure 2.14: Two circles act in two point

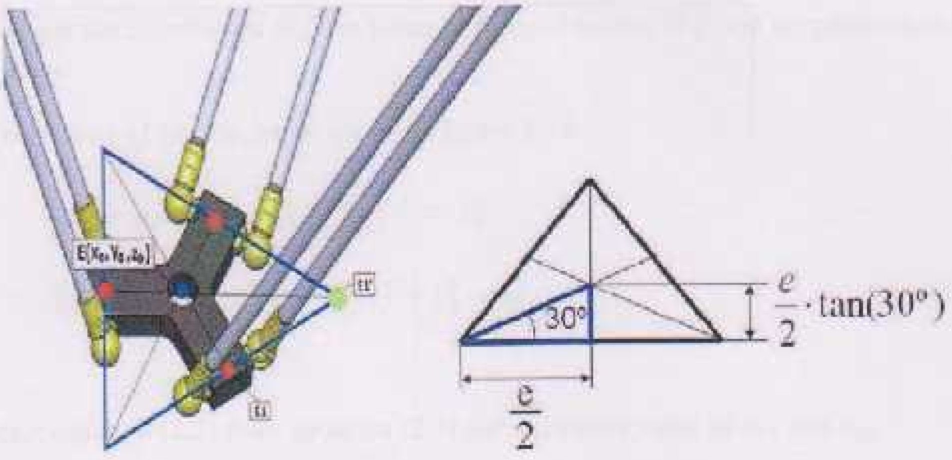


Figure 2.15: equilateral triangle on the Task space, position of E1'

From the Figure 2.15:

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

$$EE_1 = \frac{e}{2}\tan(30^0) = \frac{e}{2\sqrt{3}}$$

$$E_1 = \{x_0, y_0 - \frac{e}{2\sqrt{3}}, z_0\}$$

$$E1' = \{0, y_0 - \frac{e}{2\sqrt{3}}, z_0\}$$

$$E_1E_1' = x_0$$

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

$$E_1'J_1 = \sqrt{E_1J_1^2 - E_1J_1'^2} = \sqrt{L_b^2 - x_0^2}$$ → Radius of the circle

We can get the coordinates of J1 in terms of the coordinates of E and the parameters of the robot as follows.

Two equations of two circles as shown in figure 2.14:

$$(y_{j1} - y_{F1})^2 + (z_{j1} - z_{F1})^2 = L_a^2 \quad (2.1)$$

$$(y_{j1} - y_{E1})^2 + (z_{j1} - y_{E1}) = L_b^2 - x_0^2 \quad (2.2)$$

Subtract equation (2.2) from equation (2.1) and substitute value of z_{F1} and z_{E1} :

$$y_{F1}^2 - y_{E1}^2 + 2(y_{E1} - y_{F1})y_{j1} - z_0^2 + 2z_{j1}z_0 = L_a^2 - L_b^2 + x_0^2 \quad (2.3)$$

Rewrite equation (2.3) as:

$$a\,y_{j1} + b\,z_{j1} = d \quad (2.4)$$

Where:

$a = 2(y_{E1} - y_{F1})$

$b = 2z_0$

$d = y_{F1}^2 - y_{F1}^2 + z_0^2 L_a^2 - L_b^2 + x_0^2$

Rearrange equation (2.4):

$$z_{j1} = \frac{d - a\,y_{j1}}{b} \quad (2.5)$$

Substitute equation (2.5) in equation (2.1):

$$y_{j1}^2 + y_{F1}^2 - 2y_{j1}y_{F1} + \left(\frac{d - a\,y_{j1}}{b}\right)^2 - L_a^2 = 0 \quad (2.6)$$

Extract equation (2.6):

$$y_{j1}^2 + y_{F1}^2 - 2y_{j1}y_{F1} + G_1 y_{j1}^2 + G_2 y_{j1} + G_3 - L_a^2 = 0 \tag{2.7}$$

Where:

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

$$G_2 = \frac{-2a\,d}{b^2}$$

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

Rearrange equation (2.7) to be:

$$(1 + G_1)y_{j1}^2 + (G_2 - 2y_{F1})y_{j1} + (G_3 + y_{F1}^2 - L_a^2) = 0 \tag{2.8}$$

Rewrite equation (2.8) as:

$$H\,y_{j1}^2 + Q\,y_{j1} + L = 0 \tag{2.9}$$

Where:

$H = (1 + G_1)$
$Q = (G_2 - 2y_{F1})$

$L = (G_3 + y_{F1}^2 - L_a^2)$

Solve equation (2.9):

$$y_{j1} = \frac{-Q \pm \sqrt{Q^2 - 4HL}}{2H} \tag{2.10}$$

y_{j1} Is the smallest root.

To find z_{j1} substitute y_{j1} in equation (2.5)

Now from figure 2.2 and by help of equations (2.5) and (2.10) we can find the first angle:

$$\boxed{\theta_1 = \tan^{-1}\left(\frac{-z_{j1}}{y_{F1} - y_{j1}}\right)} \tag{2.11}$$

In order to find θ_2 and θ_3 we know the axis are rotate about z-axis by 120^0 and 240^0 respectively. So we can write the equations as:

$$x_{0i} = x_0 \cos\alpha - y_0 \sin\alpha$$

$$y_{0i} = x_0 \sin\alpha + y_0 \cos\alpha \tag{2.12}$$

$$z_{0i} = z_0$$

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

Now:

$$\theta_2 = \tan^{-1}\left(\frac{-z_{j2}}{y_{F3} - y_{j3}}\right) \tag{2.13}$$

$$\theta_3 = \tan^{-1}\left(\frac{-z_{j3}}{y_{F3} - y_{j3}}\right) \tag{2.14}$$

2.5.3 Forward Kinematics

Forward kinematics or direct kinematics determines the position of the task space(x, y, and 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 2.16.

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 J1, J2 and J3 to the points J1', J2'and J3' using the transition vectors $\overline{E1E0}, \overline{E2E0}, \overline{E3E0}$ respectively.

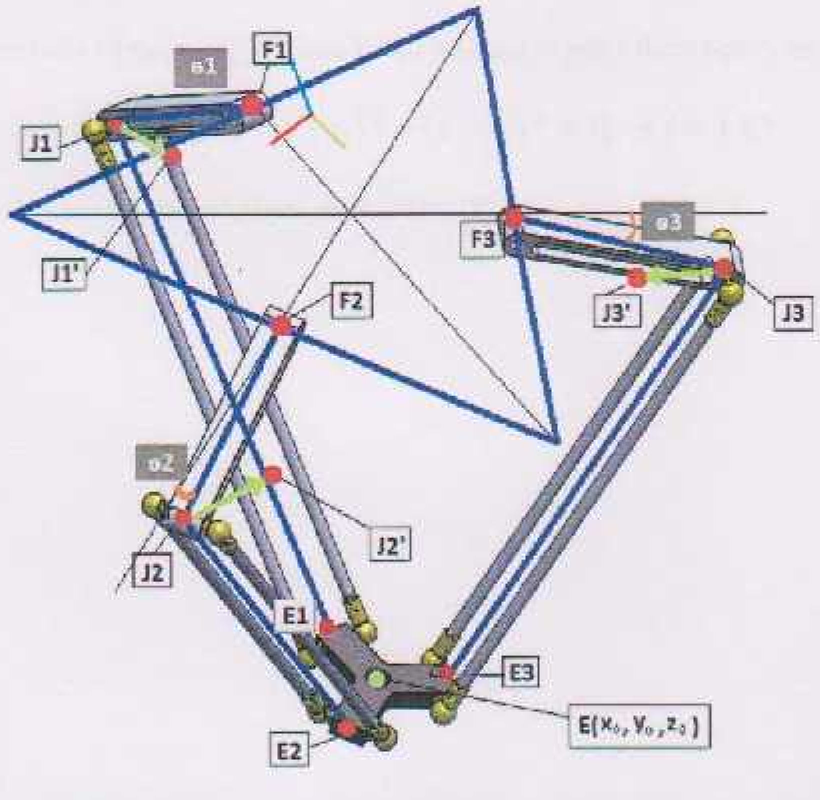


Figure 2.16: shown the position of J1', J2' and J3' after transition

After this transition, the three spheres will intersect in the point: $E(x_0, y_0, z_0)$ as shown in Figure 2.17.

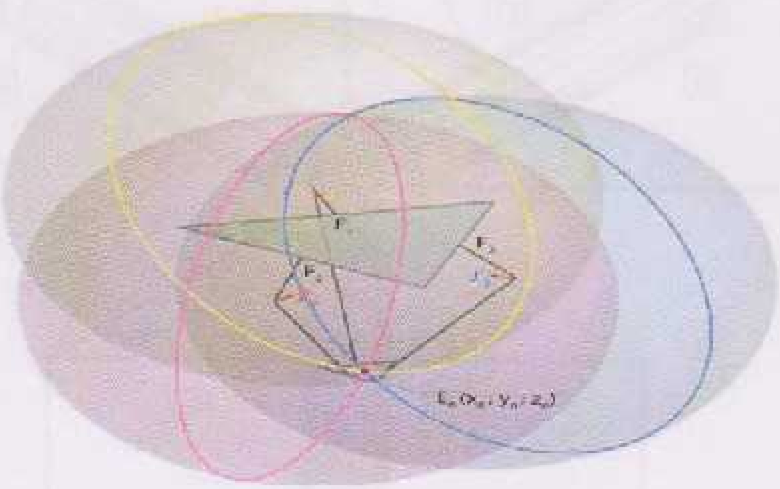


Figure 2.17: three spheres will intersect in the center of task space

To find coordinates $E(x_0, y_0, z_0)$ of point E=E0 we need to solve three equations like:

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

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

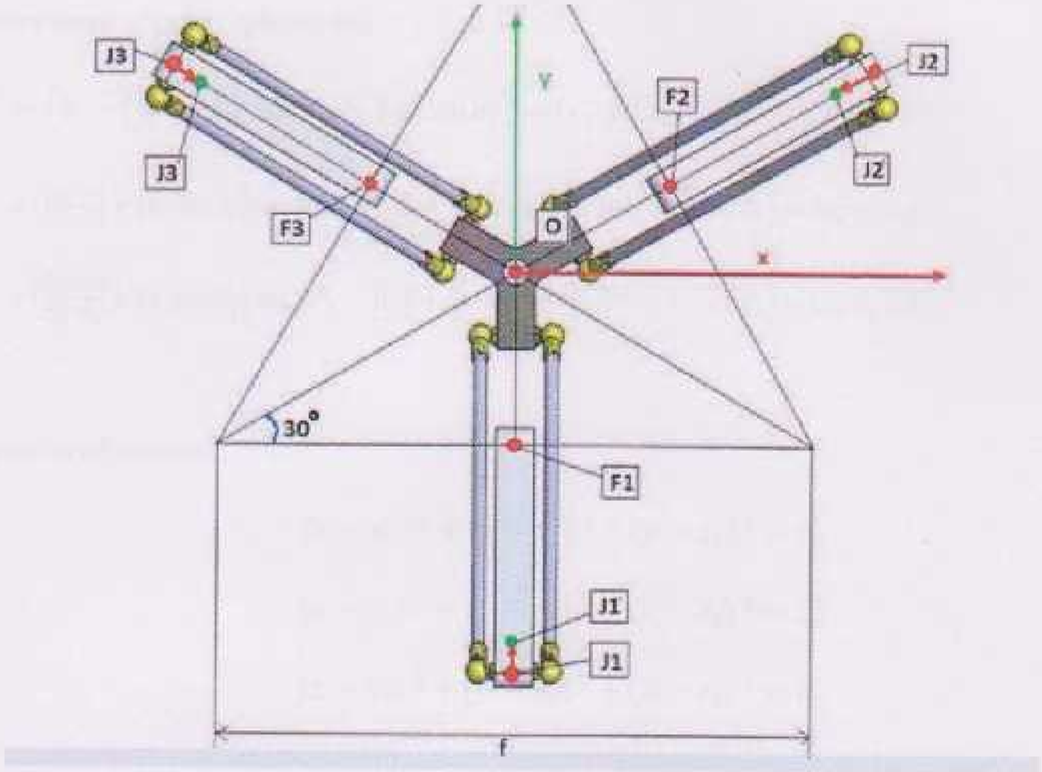


Figure 2.18: Upper section shows the center of task space after transition and equilateral triangle.

From the Figure 2.18:

$$OF1 = OF2 = OF3 = \frac{f}{2}\tan 30^0 = \frac{f}{2\sqrt{3}} \text{ (From equilateral triangle)} \quad (2.15)$$

$$J1J1' = J2J2' = J3J3' = \frac{f}{2}\tan 30^0 = \frac{e}{2\sqrt{3}} \text{ (From equilateral triangle)} \quad (2.16)$$

$$F1J1 = La.\cos\theta_1 \quad (2.17)$$

$$F2J2 = La.\cos\theta_2 \quad (2.18)$$

$$F3J3 = La.\cos\theta_3 \quad (2.19)$$

Coordinates of center spheres are:

$$J1' = \left(0 \quad \frac{-(f-e)}{2\sqrt{3}} - La\,.\cos\theta_1 \quad - La\,.\sin\theta_1\right) = (x_1, y_1, z_1) \tag{2.20}$$

$$J2' = \left(\left[\left[\frac{f-e}{2\sqrt{3}}\right] + La\,.\cos\theta_2\right]\cos 30^0 \quad \left[\frac{f-e}{2\sqrt{3}} + La.\cos\theta_2\right]\sin 30^0 - La.\sin\theta_2\right) = (x_2, y_2, z_2) \tag{2.21}$$

$$J3' = \left(\left[\left[\frac{f-e}{2\sqrt{3}}\right] + La\,.\cos\theta_3\right]\cos 30^0 \quad \left[\frac{f-e}{2\sqrt{3}} + La.\cos\theta_3\right]\sin 30^0 - La.\sin\theta_3\right) = (x_3, y_3, z_3) \tag{2.22}$$

Equation of spheres:

$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = L_b^2 \tag{2.23}$$

$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = L_b^2 \tag{2.24}$$

$$(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = L_b^2 \tag{2.25}$$

Extract equation (2.23), (2.24) and (2.25):

$$x^2 + y^2 + z^2 - 2y_1y - 2z_1z = L_b^2 - y_1^2 - z_1^2 \tag{2.26}$$

$$x^2 + y^2 + z^2 - 2x_2x - 2y_2z - 2z_2z = L_b^2 - x_2^2 - y_2^2 - z_2^2 \tag{2.27}$$

$$x^2 + y^2 + z^2 - 2x_3x - 2y_3z - 2z_3z = L_b^2 - x_3^2 - y_3^2 - z_3^2 \tag{2.28}$$

Let

$$w_i = x_i^2 + y_i^2 + z_i^2 \tag{2.29}$$

Subtract equation (2.27) from equation (2.26):

$$x_2x + (y_1 - y_2)y + (z_1 - z_2)z = (w_1 - w_2)/2 \tag{2.30}$$

Subtract equation (2.28) from equation (2.26):

$$x_3x + (y_1 - y_3)y + (z_1 - z_3)z = (w_1 - w_3)/2 \tag{2.31}$$

Subtract equation (2.28) from equation (2.27):

$$(x_2 - x_3)x + (y_2 - y_3)y + (z_2 - z_3)z = (w_2 - w_3)/2 \tag{2.32}$$

From equation (2.30) and (2.31), we can find the solution of (x and y):

$$\boxed{x = a_1z + b_1} \tag{2.33}$$

$$\boxed{y = a_2z + b_2} \tag{2.34}$$

Where:

$$a_1 = \frac{1}{d}[(z_2 - z_1)(y_3 - y_1) - (z_3 - z_2)(y_2 - y_1)]$$

$$b_1 = \frac{1}{2d}[(w_2 - w_1)(y_3 - y_1) - (w_3 - w_2)(y_2 - y_1)]$$

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

$$a_2 = \frac{-1}{d}[(z_2 - z_1)x_3 - (z_3 - z_2)x_2]$$

$$b_2 = \frac{1}{2d}[(w_2 - w_1)x_3 - (w_3 - w_1)x_2]$$

Substitute equation (2.33) and (2.34) in equation (2.26).

$$(a_1^2 + a_2^2 + 1)z^2 + 2(a_1 + a_2(b_2 - y) - z_1)z + (b_1^2 + (b_2 - y_1)^2 + z_1^2 - L_b^2) \quad (2.35)$$

The smallest negative equation root results $z = z_0$

(x_0, y_0) Are calculate with equations (2.33) and (2.34).

2.5.4 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 $\overline{P}$ is the position vector of the moving platform as shown in Figure 2.19 and figure 2.20 [12].

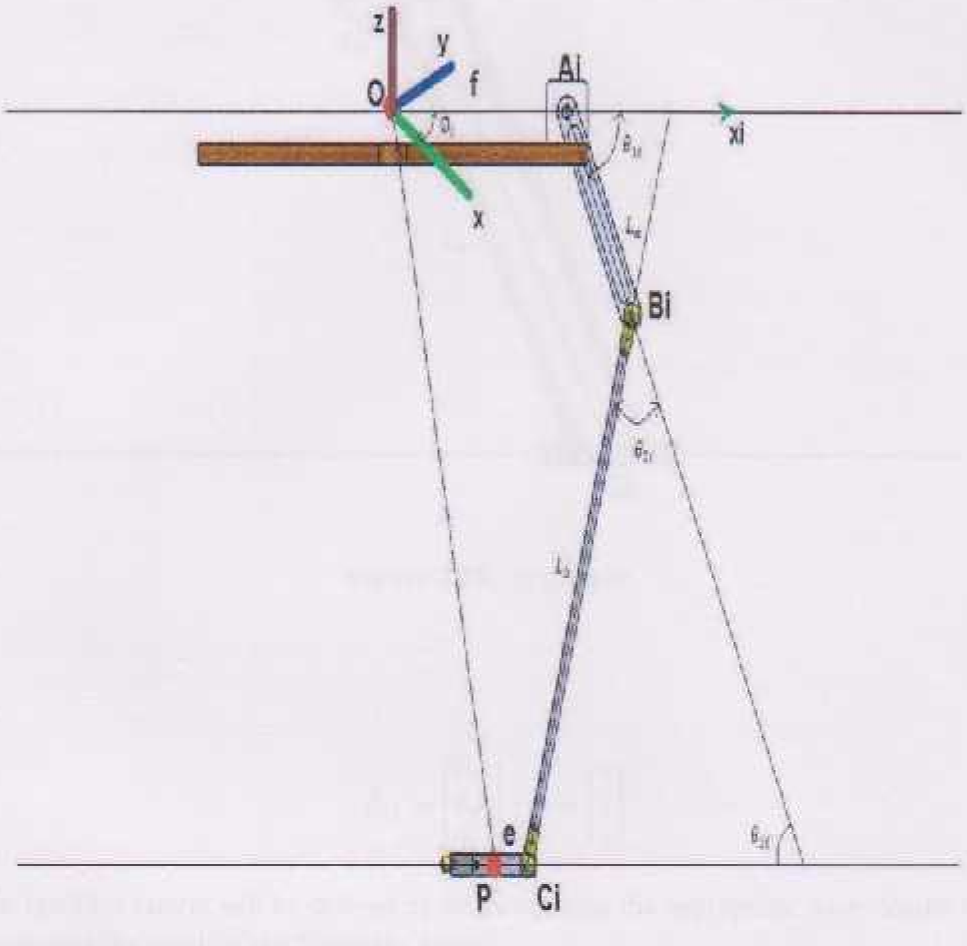


Figure 2.19: Projection of link i on $x_i z_i$ plane.

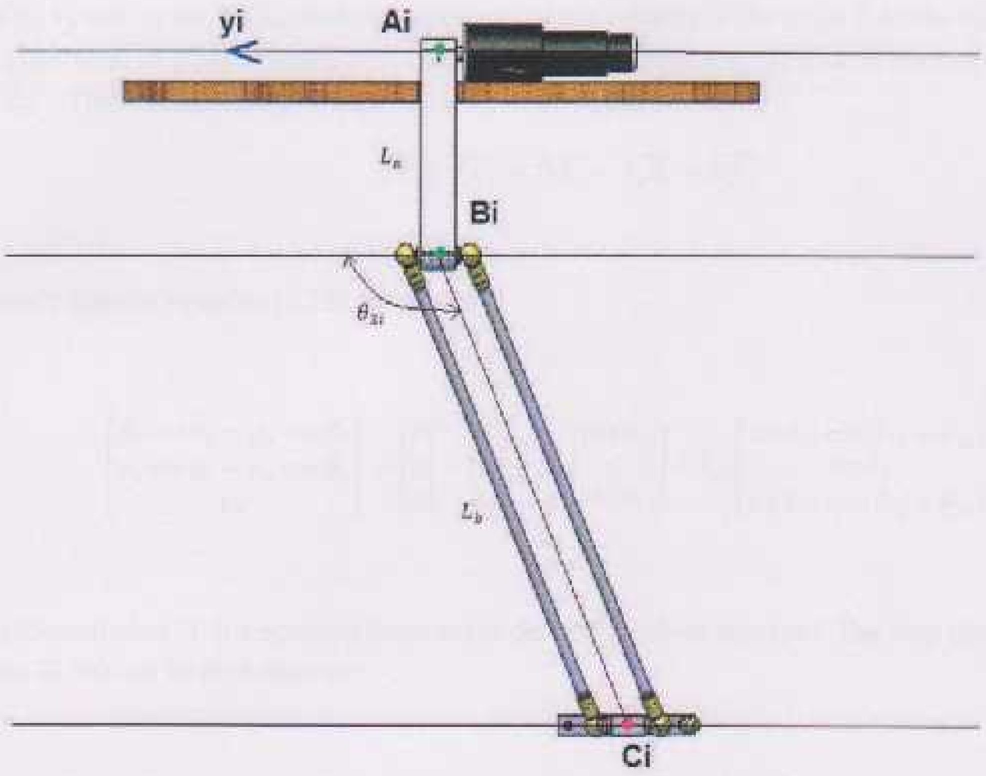


Figure 2.20: $y_i z_i$ plane

Where,

$$\vec{\theta}_{1i} = \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix}, \vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.36}$$

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

$$\vec{j}_\theta = \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix}, \vec{j}_p = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{2.37}$$

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

$$\overrightarrow{OP} + \overrightarrow{PC_i'} = \overrightarrow{OA_i} + \overrightarrow{A_iB_i} + \overrightarrow{B_iC_i'} \tag{2.38}$$

The matrix form of equation (2.38) is:

$$\begin{bmatrix} p_x \cos\phi_i - p_y \cos\phi_i \\ p_x \sin\phi_i - p_y \cos\phi_i \\ p_z \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} e \\ 0 \\ 0 \end{bmatrix} + L_a \begin{bmatrix} \cos\theta_{1i} \\ 0 \\ \sin\theta_{1i} \end{bmatrix} + L_b \begin{bmatrix} \sin\theta_{3i}\cos(\theta_{1i} + \theta_{2i}) \\ \sin\theta_{3i} \\ \sin\theta_{3i}\cos(\theta_{1i} + \theta_{2i}) \end{bmatrix} \tag{2.39}$$

Time differentiation of this equation leads to the desired Jacobian equation. The loop closure equation (2.38) can be re-written as:

$$(\vec{P} + \vec{e}) = \vec{f} + \vec{a_i'} + \vec{b_i} \tag{2.40}$$

Where $\vec{a_i}$ and $\vec{b_i}$ represents vectors $\overrightarrow{A_iB_i}$ and $\overrightarrow{B_iC_i}$ respectively.
Differentiating equation (2.40) with respect to time and using the fact that $\vec{f}$ is a vector characterizing the fixed platform, and $\vec{e}$ is a vector characterizing the task space.

$$\dot{\vec{P}} = \vec{v} = \dot{\vec{a_i}} + \dot{\vec{b_i}} \tag{2.41}$$

The linear velocities on the right hand side of equation (2.41) can be readily converted into the angular velocities by using the well-known identities.

Thus

$$\vec{v} = \overrightarrow{w_{ai}} \times \vec{a_i} + \overrightarrow{w_{bi}} \times \vec{b_i'} \tag{2.42}$$

$\overrightarrow{w_{ai}}$ and $\overrightarrow{w_{bi}}$ is the angular velocity of the link i . To eliminate $\overrightarrow{w_{bi}}$, it is necessary to dot-multiply both sides of equation (2.42) by $\vec{b_i}$, therefore:

$$\vec{b_i}\cdot\vec{v} = \vec{w_{ai}}\cdot(\vec{a_i}\times\vec{b_i}) \tag{2.43}$$

Where:

$$\vec{a_i} = L_a\begin{bmatrix}\cos\theta_{1i}\\ 0\\ \sin\theta_{1i}\end{bmatrix}, \vec{b_i} = L_b\begin{bmatrix}\sin\theta_{3i}\cos(\theta_{1i}+\theta_{2i})\\ \sin\theta_{3i}\\ \sin\theta_{3i}\cos(\theta_{1i}+\theta_{2i})\end{bmatrix}$$

$$\vec{\omega_i} = \begin{bmatrix}0\\ -\dot{\theta}_{1i}\\ 0\end{bmatrix}, v = \begin{bmatrix}v_x\cos\emptyset_i - v_y\cos\emptyset_i\\ v_x\sin\emptyset_i - v_y\cos\emptyset_i\\ v_z\end{bmatrix} \tag{2.44}$$

Rewriting the vectors of equation (2.43) in the $x_i y_i z_i$ coordinate frame leads to,

$$j_{ix}v_x + j_{iy}v_y + j_{iz}v_z = L_a\sin\theta_{2i}\sin\theta_{3i}\,\dot{\theta}_{1i} \tag{2.45}$$

Where:

$$j_{ix} = \cos(\theta_{1i}+\theta_{2i})\sin\theta_{3i}\cos\emptyset_i - \cos\theta_{3i}\sin\emptyset_i$$

$$j_{iy} = \cos(\theta_{1i}+\theta_{2i})\sin\theta_{3i}\sin\emptyset_i - \cos\theta_{3i}\cos\emptyset_i$$

$$j_{ix} = \sin(\theta_{1i}+\theta_{2i})\sin\theta_{3i} \tag{2.46}$$

Expanding equation (2.45) for $i = 1,2$ *and* 3 yields three scalar equations which can be assembled into a matrix form as:

$$\vec{J_x}\vec{v} = \vec{J_q}\vec{\dot{q}} \tag{2.47}$$

Where:

$$\vec{J_x} = \begin{bmatrix}j_{1x} & j_{1y} & j_{1z}\\ j_{2x} & j_{2y} & j_{2z}\\ j_{3x} & j_{3y} & j_{3z}\end{bmatrix} \tag{2.48}$$

$$\vec{J_q} = L_a \begin{bmatrix} \sin\theta_{21}\sin\theta_{31} & 0 & 0 \\ 0 & \sin\theta_{22}\sin\theta_{32} & 0 \\ 0 & 0 & \sin\theta_{23}\sin\theta_{33} \end{bmatrix} \tag{2.49}$$

$$\vec{\dot{q}} = \begin{bmatrix} \dot{\theta_{11}} \\ \dot{\theta_{12}} \\ \dot{\theta_{13}} \end{bmatrix} \tag{2.50}$$

After algebraic manipulations, it is possible to write:

$$\vec{v} = \vec{J}\vec{\dot{q}} \tag{2.51}$$

Where:

$$\vec{J} = \overrightarrow{J_x^{-1}}\overrightarrow{J_q} = \begin{bmatrix} \frac{\delta x}{\delta\theta_1} & \frac{\partial y}{\partial\theta_2} & \frac{\delta z}{\delta\theta_3} \\ \frac{\delta x}{\delta\theta_1} & \frac{\partial y}{\partial\theta_2} & \frac{\delta z}{\delta\theta_3} \\ \frac{\delta x}{\delta\theta_1} & \frac{\partial y}{\partial\theta_2} & \frac{\delta z}{\delta\theta_3} \end{bmatrix} \tag{2.51}$$

2.5.5 Dynamic model

Dynamics 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.

$$\tau = A(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} - G(\theta) \tag{2.52}$$

Where :

$A(\theta)$: n x n is the inertia matrix of the manipulator

$C(\theta,\dot{\theta})$: is a n x 1 vector of centrifugal and Coriolis terms

$G(\theta)$: is an n x 1 vector of gravity terms.

We use the term state-space equation because the term $C(\theta,\dot{\theta})$ has both position and velocity dependence. Each element of $A(\theta)$ and $G(\theta)$ is a complex function that depends on θ, the position of all the joints of the manipulator. Each element of . $C(\theta,\dot{\theta})$ 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 [8] .

2.5.5.1 VIRTUAL WORK PRINCIPLE

Virtual work on a system is the work resulting from either virtual forces acting through a real displacement or real forces acting through a virtual displacement. The term displacement may refer to a translation or to a rotation and the term force to a force or a moment. The principle does not refer to any geometric or inertial parameters and may therefore be considered as a fundamental connection between dynamics and kinematics [6].

Since any generalized forces system can be used, the equality of the virtual works associated to the two coordinates systems of interest in robotics yields:

$$\tau^T.\delta\theta = \tau_n^T.\delta X_n \qquad (2.53)$$

Where :

τ : is the force/torque vector corresponding to joint space virtual displacement

$\delta\theta$: is the force acting on the traveling plate corresponding to the virtual displacement

δXn : is the virtual displacement in Cartesian space.

τn: is torque acting on the traveling plate corresponding to the virtual displacement

The relationship between joint velocity and Cartesian velocity

$$\dot{v} = J\dot{\theta} \qquad (2.54)$$

Jacobian matrix can be used to transform the force/torques acting in Cartesian space to joint space as $\tau^T = \tau_n^T.J$ which is equal to $\tau = \tau_n.J^T$

2.5.5.2 CALCULATION OF DYNAMIC MODEL BASED ON VIRTUAL WORK PRINCIPLE

Two kinds of forces are acting on the traveling plate. The gravity force Gn and the inertial force Fn. These two is given by:

$$G_n = m_{nt}(0 \quad 0 \quad -g)^T \tag{2.55}$$

$$F_n = m_{nt}\ddot{X}_n \tag{2.56}$$

$$\ddot{X}_n = \dot{J}\dot{\theta} + J\ddot{\theta} \tag{2.57}$$

Where :

mnt : total contribution mass of traveling plate

$\ddot{X}_n$: the relationship between the Cartesian acceleration and the acceleration in joint space

The contribution of these two forces in joint space, can then be calculated with the transpose of the Jacobian matrix as described in section 2.5.1

$$\tau_n = J^T F_n = J^T m_{nt}\ddot{X}_n \tag{2.58}$$

$$\tau_{Gn} = J^T G_n = J^T m_{nt}(0 \quad 0 \quad -g)^T \tag{2.59}$$

Where :

τGb : The torque produced by the gravitational force of each upper arm

τb : the torque τb produced from the inertial force acting on each upper arm

The gravitational contribution can be calculated as the force that is acting perpendicular to the upper arm through the mass centre point as shown in Figure 2.21

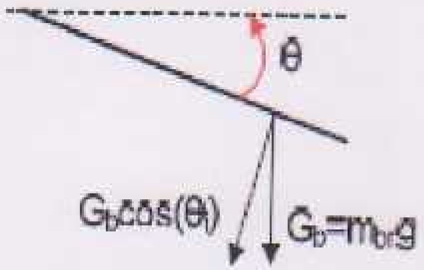


Figure 2.21: gravitational force acting on the upper arm of a Delta-3 robot.

This gives

$$\tau_{Gb} = r_{Gb}G_b(\cos\theta_1 \quad \cos\theta_2 \quad \cos\theta_3)^T \tag{2.60}$$

$$\tau_b = I_b\ddot{\theta} \tag{2.61}$$

$$I_b = \begin{bmatrix} I_{b1} & 0 & 0 \\ 0 & I_{b2} & 0 \\ 0 & 0 & I_{b3} \end{bmatrix} \tag{2.62}$$

where :

Gb: is the gravitational force acting on the mass centre point of each upper arm

Ib: is the inertia matrix of the arms in joint space

According to d'Alembert's principle the contribution of all the inertial forces must equal the contribution of all the non-inertial forces[13,14] , this applied at the joint level leads to:

$$\tau + \tau_{Gn} + \tau_{Gb} = \tau_b + \tau_n \tag{2.63}$$

This gives

$$\tau = (I_b + m_{nt}J^TJ)\ddot{\theta} + (J^Tm_{nt}\dot{J})\dot{\theta} - (\tau_{Gn} + \tau_{Gb}) \tag{2.64}$$

2.6 Control System Design

2.6.1 Controller Techniques

Control is the science of desired motion. It relates the dynamics and kinematics of a robot to a prescribed motion. It includes best solving to determine torque so that the system will behave ideally.

The inverse kinematics of a parallel manipulator determine the desired angle of each actuated joint given the position of the end effect of a robot. Replacement of the joint kinematics in the equations of motion provides the actuator commands.

2.6.2Control Technique

In feedback linearization technique, we define a control law to obtain a linear differential equation for the error command, and then use the linear control design techniques. The feedback linearization technique can be applied to robots successfully; however, it does not guarantee robustness according to parameter uncertainty or disturbances. This technique is a model-based control method.[16]

A robot is a mechanism with an actuator at each joint (i) to apply a force or torque to derive the link (i). The robot is instrumented with positionsensors to measure the joint variables kinematics. The measured values are usually kinematics information of the frame B_i, show Figure 2.8,attached to the link(i). Proportional to the frame B_{i-1} or B_0.To cause each joint of the robot to follow a desired motion, we must supply the required torque command.

The robot arm dynamics are

$$\tau = M(q)\ddot{q} + C(q,\dot{q}) + G(q) \quad (2.65)$$

Where q is the vector of joint variables, and τ is the torques applied at joints and $M(q)$ is the inertia matrix, $C(q,\dot{q})$the Coriolis or centripetal vector, and $G(q)$the gravity vector.

Let

$$N(q,\dot{q}) = C(q,\dot{q}) + G(q) \quad (2.66)$$

We can re-write Eq. 2.65too:

$$\tau = M(q)\ddot{q} + N(q,\dot{q}) \quad (2.67)$$

This is an open-loop control algorithm, that the control commands are calculated based on a known desired path and the equations of motion. Then, the control commands are fed to the system to generate the desired path, however, there is no mechanism to compensate any possible error.

Therefore, it is important to be able to find the desired joint space trajectory given the desired q_d . This is accomplished using the inverse kinematics. Suppose that a desired q_d has been selected for the arm motion. To ensure trajectory tracking by the joint variable, define an output or tracking error e as:

$$e = q_d - q \quad (2.68)$$

To show the influence of the input on the tracking error, differentiate the error e as:

$$\dot{e} = \dot{q}_d - \dot{q} \quad (2.69)$$

$$\ddot{e} = \ddot{q}_d - \ddot{q} \quad (2.70)$$

Solve for $\ddot{q}$ inEq.2.65:

$$\ddot{q} = M^{-1}(N - \tau) \quad (2.71)$$

Now substitute 2.71 into 2.70:

$$\ddot{e} = \ddot{q}_d - M^{-1}(N - \tau) \quad (2.77)$$

The control input:

$$\ddot{e} = u \quad (2.78)$$

Define a state x(t) by:

$$x(t) = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \quad (2.79)$$

Write the tracking error dynamics as:

$$\begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{pmatrix} 0 & I \\ 0 & 0 \end{pmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u \quad (2.80)$$

It is driven by the control input u (t). Notice that this deviation is a particular instance of the general feedback linearization procedure. The feedback linearization transformation may be reversed to give:

$$\tau = M(\ddot{q} - u) + N \quad (2.81)$$

We call this the computed-torque control law. Substituting Eq. 2.81 into Eq2.66 yields

$$M(q)\ddot{q} + N(q, \dot{q}) = M(\ddot{q} - u) + N \quad (2.82)$$

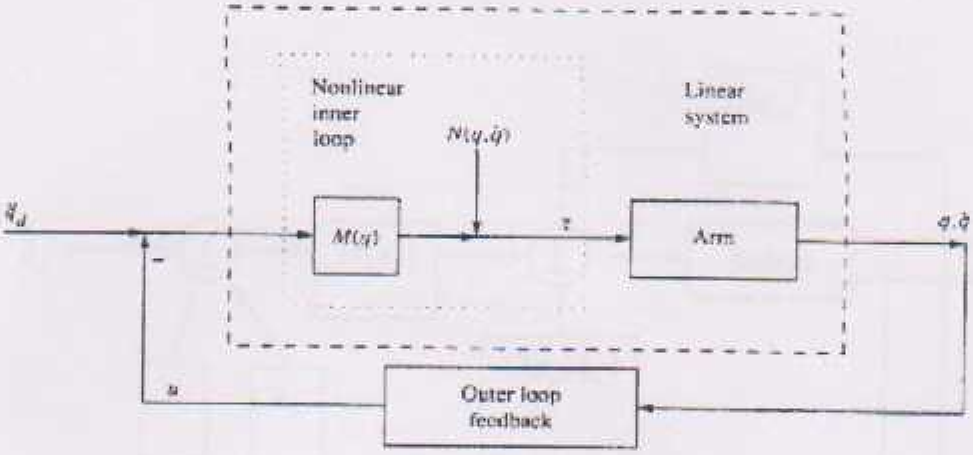


Figure 2.22: Computed-torque control system

The resulting control system appears in Figure 2.22. It is important to note that it consists of an inner nonlinear loop plus an outer control signal u (t). Since u (t) will depend on q and $\dot{q}$ the outer loop will be a feedback loop

2.6.3 PD Outer-Loop Designs

One way to select the auxiliary control signal u(t) is as the proportional-plus-derivative (PD) feedback

$$u = K_d e \; - K_v \dot{e} \quad (2.83)$$

Then the overall robot arm input becomes

$$\tau = M(q)(\ddot{q}_d + k_v \dot{e} + k_p e) + N(q,\dot{q}) \quad (2.84)$$

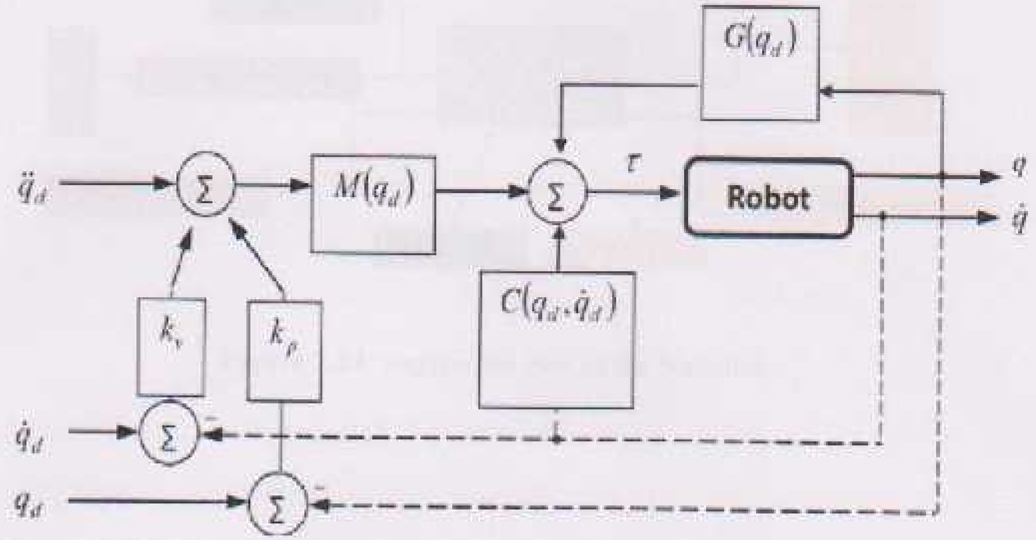


Figure 2.23: Block diagram of computed torque control.

2.6.4 Simulation

This section contains the simulation results for different parts such as robot kinematics, robot dynamics asShown in Figure 3.3.A Simulink block is created where the data from MatlabSimulink is imported into the generation block in the Simulink model. The output from this block, the desired arm angles are not recalculated this is achieved with an implemented forward kinematic block and inverse kinematic block, which takes the three end-effecter direction components x, y and z as input and gives out the three upper arm angles.

In figure 3.3,the inverse kinematics is used to transform the position (x,y,z) into desired angle for the three arms that are forwarded into respectively delta robot dynamic every time cycle. The delta robot dynamic every also returns an actual value of the three arm angles every time cycle. These returned angle values are only used to calculate how big the tracking error is. The forward kinematics is also used to calculate the initial position so the motion from the initial position to the start position of the motion can be calculated. Then during each time cycle the tracking error is calculated with help of the forward kinematics.

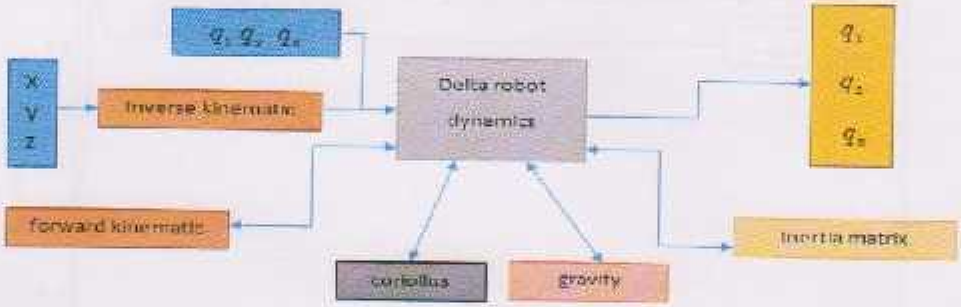
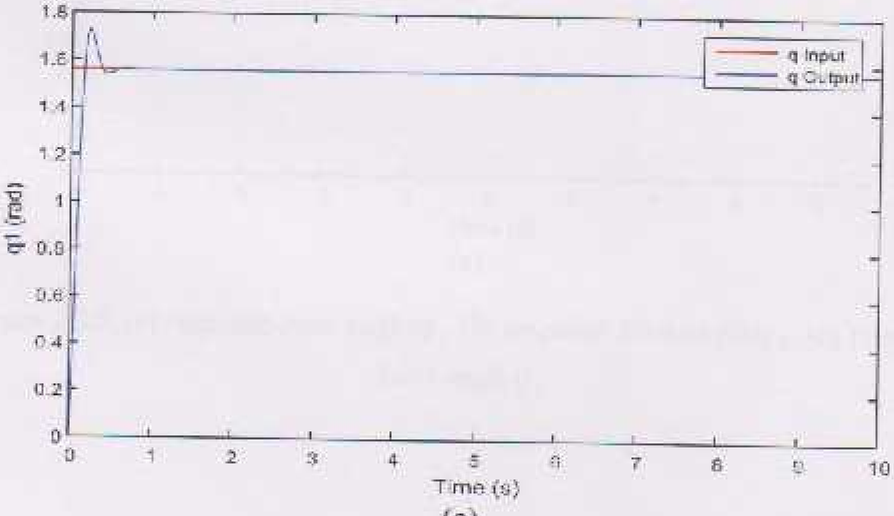


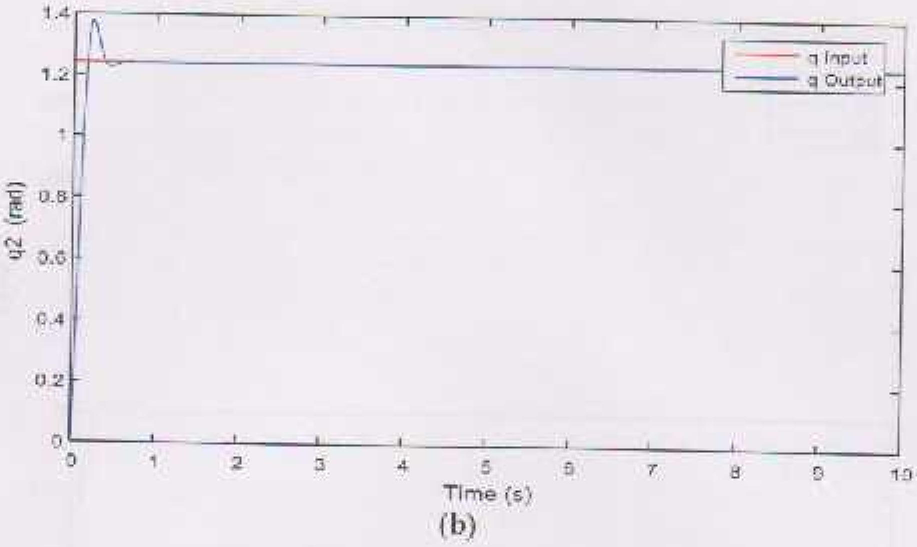
Figure 2.24: explain the part of the Simulink

2.6.5 Simulation Results

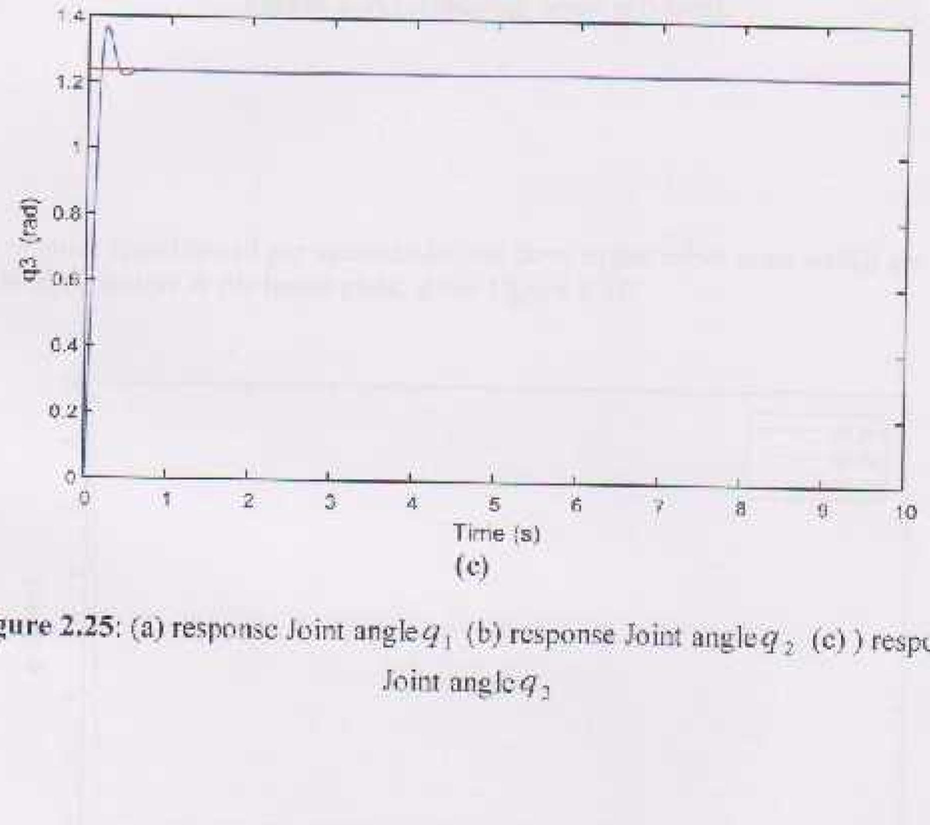
We simulated the PD computed-torque controller for a delta robot arm. First, the tracking error e(t) and its derivative are computed. Then $M(q)$ and $N(q,\dot{q})$ are computed. Shown figure2.25.In is dynamic that the differencebetween the desired angles q_d and the angles q be as small as possible by tuning K_d, K_p ,asshowing figure 3.4 in order for the end effecter error to be minimized to reach zero shows figure2.25 .



(a)



(b)



(c)

Figure 2.25: (a) response Joint angle q_1 (b) response Joint angle q_2 (c)) response Joint angle q_3

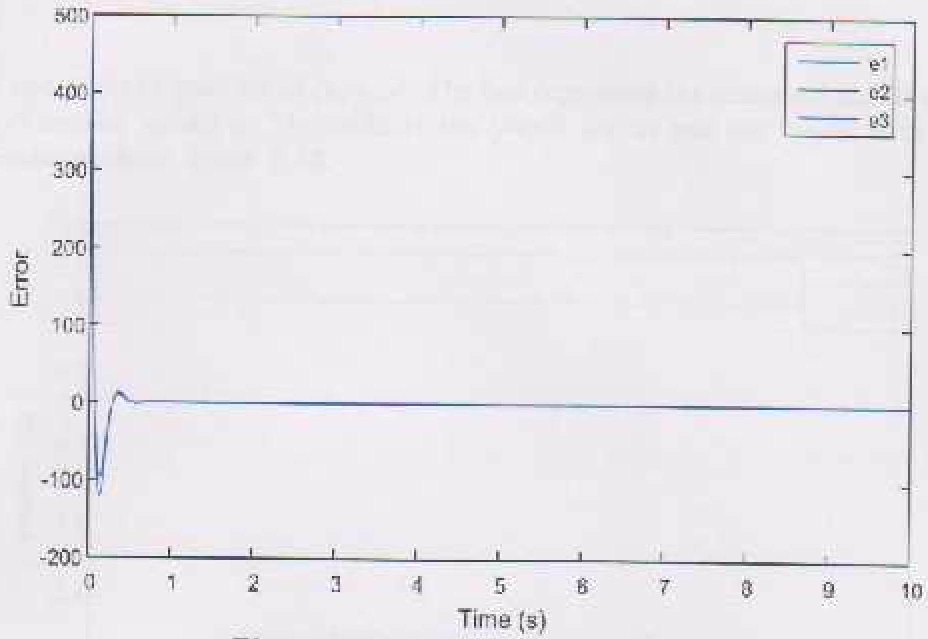


Figure 2.26: Tracking error e(t) (red).

The angular speed in rad per seconds for the three upper robot arms which are linked to the three motors at the based plate, show Figure 2.27.

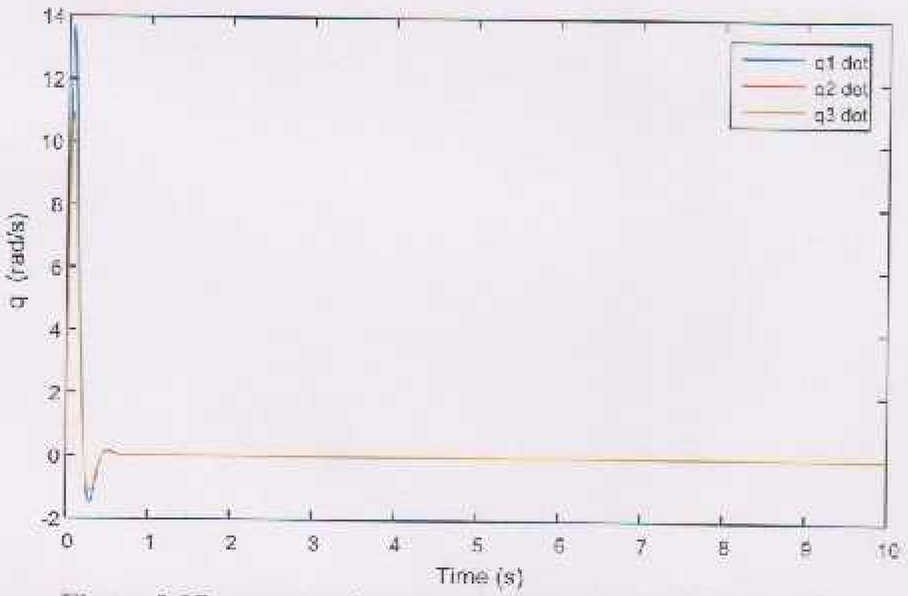


Figure 2.27: represent three angular velocity with PD *controller*

The component's position of (x, y, z). The line represents the measured actual value in the -Simulink model a. The units at the y-axis are m and the x-axis units are in seconds as show figure 2.28.

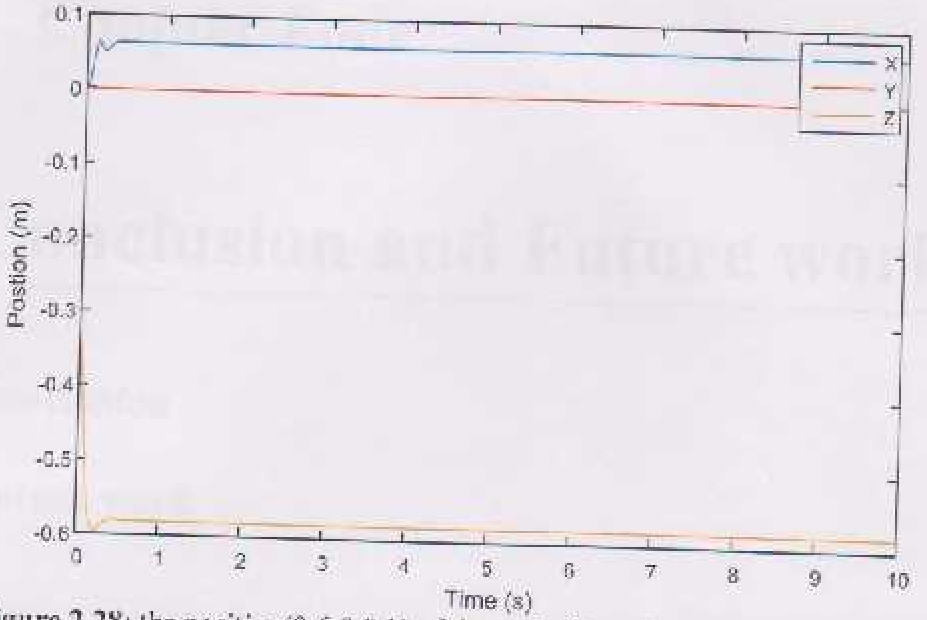


Figure 2.28: the position(0.6,0,0.1) of the end effect after applied PD controller

4

Chapter Four

Conclusion and Future work

4.1 Conclusion

4.2 Future work

4.1 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.

4.2 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] J.-P. MERLET, Parallel Robots (Second Edition). SOLID MECHANICS AND ITS APPLICATIONS, Volume 128 (2006 ,Springer).

[2] André Olsson, Modeling and control of a Delta-3 robot, master thesis, Department of Automatic Control, Lund University, 2009.

[3] S. St. and C.-C. D. C., DYNAMIC ANALYSIS OF CLAVEL' S DELTA PARALLEL ROBOT, 2003.

[4] P. Zsombor-Murray, "An improved approach to the kinematics of Clavel's DELTA robot," Online, 2009.

[5] V. Poppeová, J. Uríček, V. Bulej, and P. Šindler, Delta robots–robots for high speed manipulation, Tehnički vjesnik, 18 (2011),pp. 435-445.

[6]Mohsen, Mahdi, Mersad," Dynamics and Control of a Novel 3-DoF Spatial Parallel Robot", International Conference on Robotics and Mechatronics,2013.

[7]Yangmin Li and Qingsong Xu ,"Dynamic Analysis of a Modified DELTA Parallel Robot for Cardiopulmonary Resuscitation", Department of Electromechanical Engineering, Faculty of Science and Technology University of Macau, 2004.

[8]John J. Craig, "Introduction to robotics, Mechanics and Control, ThirdEdition, 2005.

[9]YangminLi, Qingsong Xu," Dynamic modeling and robust control of a 3 PRC translational parallel kinematic machine", Robotics and Computer-Integrated Manufacturing journal, Science Direct, 2009.

[10] A. Liadis, fuzzy logic control of a two degree of freedom parallel robot, (2010).

[11] Craig, J. John. Introduction to Robotics Mechanics & Control :Addison-Wesley Publishing Company Inc, 1986.

[12] Codourey, Alain. Dynamic Modeling of Parallel Robots for Computed-Torque Control Implementation. *The International Journal of Robotics Research*. 1998.

[13] Gross, Hauger, Schröder, Wall. Technische Mechanik 3. Berlin : Springer-Verlag, 2008.

[14] Sciavicco, L. and Siciliano, B. *Modelling and Control of Robot Manipulators*. London : Springer-Verlag London Ltd, 2000.