*Journal of Autonomous Intelligence* (2022) Volume 5 Issue 1.
doi: 10.32629/jai.v5i1.505

## Original Research Article

# Trajectory Planning and Execution of Delta Robot

***Diego Édison Flórez Vergara, Fabián Camilo Castro Riveros, Ricardo Andrés Castillo Estepa****

*Universidad Militar Nueva Granada, Ingeniería en Mecatrónica, Bogotá, Colombia.*

## ABSTRACT

The main purpose of this study is to plan and execute effective trajectory on the Delta robot through graphical user interface (GUI) and appropriate electronic circuits, and optimize the integration with the mechanical prototype designed and implemented by the robot laboratory of Nueva Granada Military University, while considering and respecting its different characteristics, such as mechanical structure, workspace, kinematics, dynamics and motion singularity allow the correct development and implementation of algorithms that can effectively describe and control the planned trajectory.

***Keywords:*** *Algorithm; Direct Kinematics; Inverse Kinematics; Cartesian Space; Joint Space; Workspace; Jacobian; Third-order Polynomial; Robot Delta; Singular Point*

**ARTICLE INFO**
Received: Jan 17, 2022
Accepted: Mar 6, 2022
Available online: Mar 12, 2022

***CORRESPONDING AUTHOR**
Ricardo Andrés Castillo Estepa
E-mail: ricardo.castillo@unimilitar.edu.co

**CITATION**
Flórez Vergara DE, Castro Riveros FC, Castillo Estepa RA. Trajectory Planning and Execution of Delta Robot. Journal of Autonomous Intelligence 2022; 5(1): 43-52. doi: 10.32629/jai.v5i1.505



## 1. Introduction

From its early model to the most influential model at present, people have produced various explanations for its being an industrial robot. Therefore, the following are the two most important explanations[1].

Robot Industries Association (RIA): The reprogrammable multifunctional manipulator can move materials, parts, tools or special devices according to different tracks, and program to perform different tasks.

International Organization for Standardization (ISO): Three axis or more manipulators with automatic control, programmable, multiplication, mobile or non-mobile functions are used in industrial automation applications. It includes manipulator (mechanical system and actuator) and control system (software and hardware of control and power supply).

Different industrial robots can be divided into the following[2].

1. Temporary, i.e. according to your production date.
2. Because of your intelligence level.
3. Because of its function.
4. For the control types, we classify the robots according to the French Association of Industrial Robots (FAIR), and divide them into four categories according to the control types used.

Type A or remote control: With manual control or remote control.

Type B or sequence: Automatic, i.e. with pre-adjusted cycle, adjusted by stroke or buffer, controlled by PLC and driven by pneumatic, electric or hydraulic pressure.

Type C or controllable track: This is a programmable continuous path or point-to-point, but lacks knowledge of its environment.

Type D or adaptive: It can retrieve data from the environment and readjust tasks based on the data.

5. Through its geometry or structure.

Industrial robots.

Mobile industrial robots.

Delta robot (**Figure 1**) is a parallel robot, which is composed of three closed PR (Ps) configuration motion chains, where (Ps) represents a four-bar spatial parallelogram with four spherical joints, and where PR represents that each motion chain is composed of passive prismatic joints and active rotating joints, connecting the fixed base to the mobile base[3].

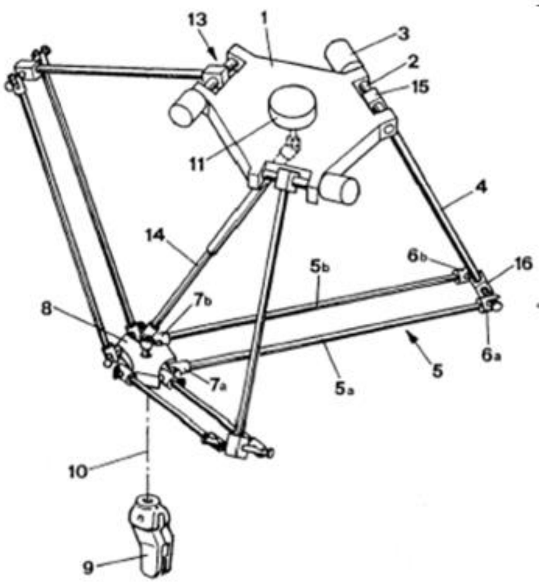


**Figure 1.** R. Carnation Delta robot. (U.S. Patent No. 4976582).

Some of the main advantages and disavantages of Delta robots and many other parallel platforms are as follows.

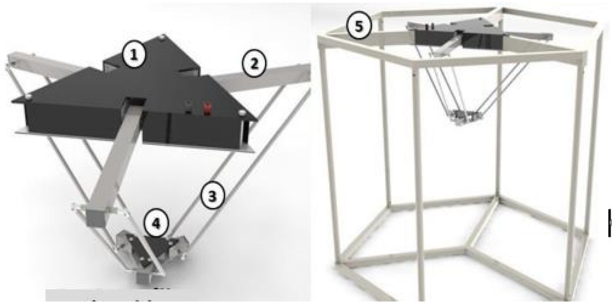
| Advantages | Disadvantages |
|---|---|
| Rigid building | Complex kinematics |
| Higher load/weight ratio | Reduce workspace |
| Light robot | Complex calibration |
| high-precision | Complex singular position |
| | |
| High speed and acceleration | Undeveloped Technology |

Like most industrial robots designed, built and implemented at present, the main goal of Delta robot is to improve the productivity and quality of human work in various complex and/or dangerous tasks. In these tasks, the robot can better perform these tasks, which require the coordinated and accurate movement of the robot. Therefore, it is very important to understand the kinematic and dynamic characteristics of robots, so as to understand and analyze their projection motion ability in planning and generating the trajectory of any activity.

With the continuous development of robot technology in industry, it is more and more necessary to have a deeper understanding of various robots and their applications. These robots account for a large proportion in today's world industry, including parallel robots.

Therefore, the main purpose of this paper is to enable it to simulate and execute the trajectory in Cartesian space and joint space through the graphical user interface and real-time interaction with the mechanical prototype, and focus on the trajectory planning and execution on the Delta robot mechanical prototype designed and implemented at the Nueva Granada Military University (**Figure 2**), which is similar to the actual tasks implemented by the current industry.

**Figure 2.** Prototype of robot Delta, robot laboratory UMNG



1. Fixed base
2. Link 1
3. Link 2
4. Mobile base
5. Supporting structure

# 2. Robot Delta

## 2.1.Working space

The workspace is defined by a set of points, which can reach around the robot according to the structure of the robot, the size of the connecting rod and wrist joint. The workspace shape of each robot is unique because it depends on its design[9].

Workspaces can be subdivided into two categories, as shown below[11].

Achievable workspace: It consists of a set of points that the manipulator can reach.

Right workspace: It consists of a set of points that the manipulator can reach in any direction of its end effector.

In order to make proper trajectory planning in Delta robot and any other industrial robot, we must be able to know the workspace where the robot will perform its various tasks, from simple motion to coordinated motion with other robots in some industrial processes.

The workspace of Delta robot (**Table 1**) is used to design different trajectories, and its main characteristics are as follows[4].

**Table 1.** Maximum and minimum range of mobile platform

| Coordinate | Minimum range | Maximum range |
|---|---|---|
| X | -3428015 mm | 3424004 mm |
| Y | -2969611 mm | 2969611 mm |
| Z | 1057695 mm | 6770132 mm |

## 2.2 Inverse kinematics

The main purpose of inverse kinematics is to find the angle of the joint by understanding the configuration (position and direction) of the robot end effector[5]. This analysis is of great significance to trajectory planning and execution, because it is based on the nonlinearity of the equation by finding multiple solutions. Therefore, the process of finding the inverse kinematics depends on the configuration of the robot, and can be solved geometrically on series and parallel robots.

Using the symmetry of Delta robot, the inverse kinematics of each kinematic chain is analyzed respectively, and the mathematical model describing each joint is obtained[4].

## 2.3.Direct kinematics

The main purpose of direct kinematics is to find the configuration (position and direction) where the manipulator end effector is located, because the joint variables have been defined in this analysis[5].

The direct kinematics analysis of Delta robot starts with a set of nonlinear equations, which connects the position of the mobile platform with the angle of its connecting rod to generate a mathematical model describing each Cartesian coordinate[4].

# 3. Planning Trajectory

## 3.1.Cartesian space

The motion description in Cartesian space considers the path generation method, in which the shape of the path is specified according to the function of calculating the Cartesian position and direction. As a time function (**Figure 3**), it is usually preferable in motion planning. In motion planning, the environment does not include obstacles, and there is no greater correlation between robot dynamics[6,7].

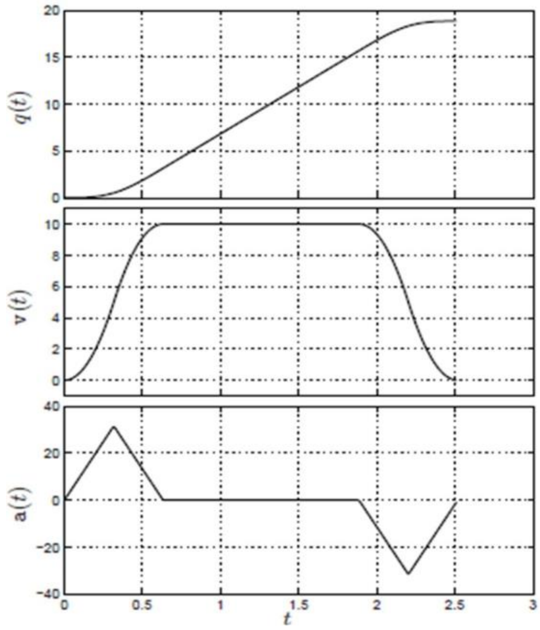



**Figure 3.** Position, velocity and acceleration in Cartesian space[8].

The trajectory planning in Cartesian space will be realized mainly through the Jacobian matrix or Jacobian matrix of the manipulator that realizes these trajectories. The matrix is a geometric representation of the elements constituting a time function mechanism that allows the differential motion or velocity of a single joint to be converted into the differential motion or velocity of the point of interest, such as the end effector[9].

Jacobiano or Jacobiana matrix is one of the most important tools to describe the characteristics of manipulator, because it is very useful for problems like the following[10].

Find the singularity.

Redundancy analysis.

Determine the inverse kinematics algorithm.

Describe the mapping between the force applied to the end effector and the torque generated in the joint.

The Jacobian matrix relates the velocity of the joint to the velocity of the Cartesian reference system

$$v = J\dot{q} \quad (1)$$

Therefore, the Jacobian matrix or Jacobian matrix of Delta robot is as follows[4].

$$J_x = \frac{\partial F}{\partial x} \quad y \quad J_\theta = \frac{\partial F}{\partial \theta} \quad (2)$$

Of which:

$Jx =$ Direct motion correlation matrix

$J\theta =$ Inverse kinematics correlation matrix

The following equations are proposed.

$$\begin{bmatrix} j_{x11} & j_{x12} & j_{x13} \\ j_{x21} & j_{x22} & j_{x23} \\ j_{x31} & j_{x32} & j_{x33} \end{bmatrix} \begin{bmatrix} V_{pX} \\ j_{pY} \\ V_{pZ} \end{bmatrix} = \begin{bmatrix} j_{\theta 1} & 0 & 0 \\ 0 & j_{\theta 2} & 0 \\ 0 & 0 & j_{\theta 3} \end{bmatrix} \begin{bmatrix} \theta_{11} \\ \theta_{12} \\ \theta_{13} \end{bmatrix} \quad (3)$$

It will generate a clear structure for the robot's Jacobian matrix or Delta Jacobian matrix.

$$J = (J_\theta)^{-1} J_x = \left( \begin{bmatrix} j_{\theta 1} & 0 & 0 \\ 0 & j_{\theta 2} & 0 \\ 0 & 0 & j_{\theta 3} \end{bmatrix} \right)^{-1} \begin{bmatrix} j_{x11} & j_{x12} & j_{x13} \\ j_{x21} & j_{x22} & j_{x23} \\ j_{x31} & j_{x32} & j_{x33} \end{bmatrix} \quad (4)$$

As the above matrix is singular, the closed-loop mechanism can have a singular direct kinematics configuration, a singular inverse kinematics configuration, or both.

Trajectory planning is developed through the inverse kinematics algorithm, which can be used to obtain the mathematical model representing differential motion, as shown in equation (4) and (5).

$$q(k+1) = q(k) + (jq(k))^{-1} Jx * v(k) * \Delta t \quad (5)$$

This equation can be easily developed by a computer because it is an iterative process that calculates the output q(k) in each sample.

## 3.2.Joint space

The description of motion in joint space considers the method of generating paths, where the shapes of these paths (in space and time) are specified according to the joint angle function (**Figure 4**)[6].

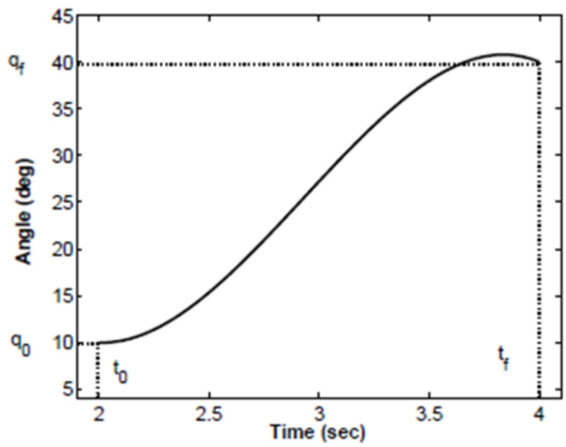



**Figure 4.** Typical representation of joint space[11].

The time required for each section of the route shall be the same for each joint so that all relevant joints reach the track point at the same time[6].

Inverse kinematics allows you to calculate a set of joint angles corresponding to the target configuration (position and direction), and know the start and end points, so as to find a function $t_o$ to determine the initial value of the configuration $t_f$ and the final value of the configuration.

In order to create a unified function, at least four constraints $\theta(t)$ are obvious, two of which are the constraints of initial and final position values, and the other two constraints will correspond to the conditions of initial and final velocity.

$$\theta(0) = \theta_0$$

$$\theta(tf) = \theta_f$$

$$\dot{\theta}(0) = 0$$

$$\dot{\theta}(tf) = 0$$

Therefore, these constraints satisfy a minimum cubic polynomial (cubic polynomial), which will describe its position, velocity and acceleration respectively.

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{6}$$

$$\hat{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{7}$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t \tag{8}$$

Therefore, by combining constraints with equations (6, 7 and 8), the following constants $a_0, a_1, a_2, a_3$ can be easily found.

$$a_0 = \theta_0$$
$$a_1 = 0$$
$$a_2 = \frac{3}{t^2}\left(\theta_f - \theta_0\right)$$
$$a_3 = -\frac{2}{t^3}\left(\theta_f - \theta_0\right)$$

Once all constants of the polynomial are found, any joint angle can be connected from the initial position to the desired end. This example is applicable to zero initial and final velocity, but I know whether it is necessary to increase the number of constraints and the order of the polynomial in order to have greater control when generating the trajectory, such as constraining velocity and acceleration at each point of the analysis.

In order to improve the computational performance, it is best to transform the equation into vector matrix form.

$$[\theta] = [M] * [C]$$

$$[C] = [M]^{-1}[\theta]$$

Of which:

$$[C] = \text{ Matriz de coeficientes } \left[a_{n\ldots}a_0\right]^t$$

$$[M] = \text{Time power coefficient matrix}$$

$$\left[b_n t^n \ldots b_0 t^o\right]^t$$

$$[\theta] = \text{Constraint matrix, velocity and acceleration}$$

As a final expression:

$$\begin{bmatrix} \theta(\text{ to }) \\ \theta(tf) \\ \dot{\theta}(to) \\ \theta(tf) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & t & t^2 & t^3 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2t & 3t^2 \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{9}$$

## 4. Execution Track

This section is devoted to analyzing and executing the trajectory realized on the Delta robot. The purpose is to test the behavior of the robot on the point-to-point trajectory relative to the inverse kinematics algorithm.

### 4.1.Execution algorithm

The algorithm for planning and executing trajectories has the following development structure.

1. Calculate the inverse kinematics on the point orbit.
2. Calculate the velocity between orbital points.
3. Calculation of Jacobiana matrix.
4. Implementation of equation (4) and (5).
5. Send the angle provided in step 4 to the electronic circuit.
6. The direct kinematic angle provided in step 4.

### 4.2.Track 1

The first implementation path is designed as an equilateral triangle. The path is planned on the XY plane, maintains a constant value at the Z coordinate, and is designed through a total of five points, three of which are the track points and two are the start and end points of the track. The coordinates of the points are shown in **Table 2**.

**Table 2.** The points that make up track 1

| Spot | X coordinate | Y coordinate | Z coordinate |
|---|---|---|---|
| Initial | 0 | 0 | 30 |
| Track | 15 | 0 | 30 |
| Track | 0 | 15 | 30 |
| Track | -15 | 0 | 30 |
| End | 0 | 0 | 30 |

The behavior of Delta robot in the joint space describing trajectory 1 is as follows (**Figure 5–7**).

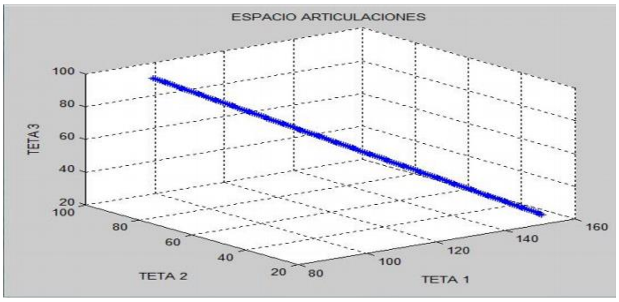



**Figure 5.** Behavior of joint space point trace 1 in MATLAB.

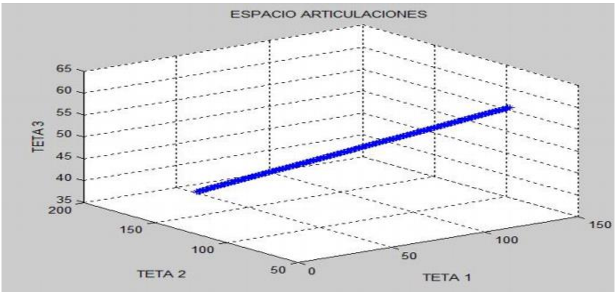



**Figure 6.** Behavior of point trace 2 in space joint in MATLAB environment.

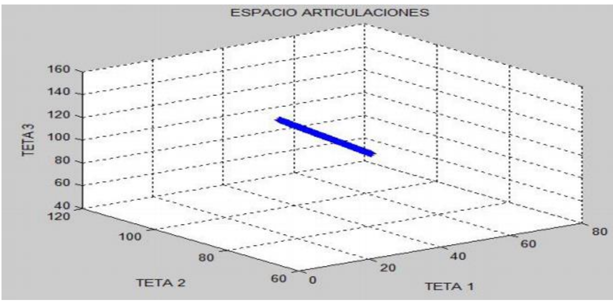



**Figure 7.** Behavior of point trace 3 in space joint in MATLAB environment.

The behavior of Delta robot in Cartesian space describing trajectory 1 is as follows (**Figure 8–10**).

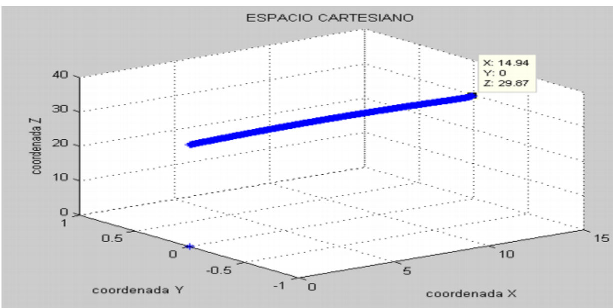



**Figure 8.** Behavior of point trace 1 in Cartesian space in MATLAB.

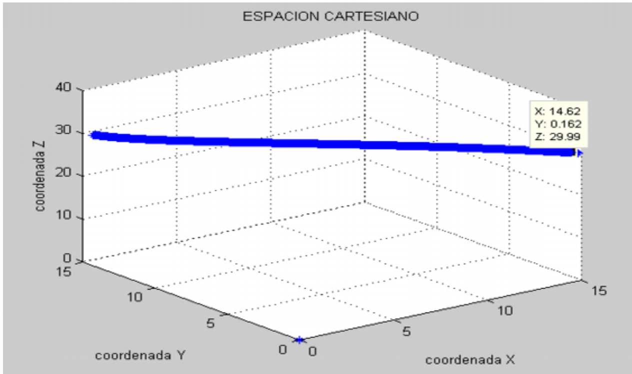



**Figure 9.** Behavior of point trace 2 in Cartesian space in MATLAB.

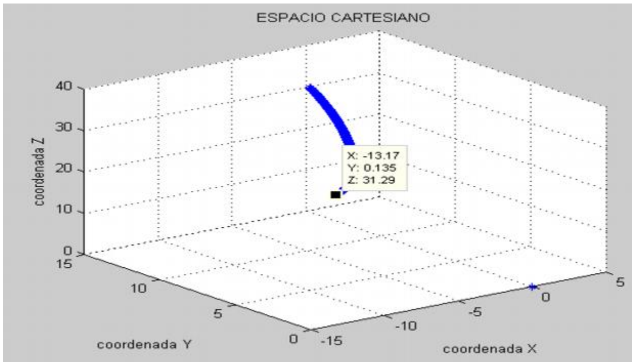



**Figure 10.** Behavior of point trace 3 in Cartesian space in MATLAB.

In this trajectory, a constant speed is selected between the trajectory points to ensure that there is a straight line between these points, which can be seen from the graph describing the joint space trajectory of the inverse kinematics algorithm.

In order to make the Delta robot track the trajectory correctly in the development process, 480 iterations are carried out to better understand the straight line constituting the three actuators.

Therefore, it can be seen that the angle is always between (0 to 180), which ensures the nonsingularity in the Delta robot. In this case, if the angle is greater than 180 or less than 0, singularity will be generated in the robot, resulting in a point in the workspace that the robot cannot reach.

Then, in the XY plane (**Figure 11**) and the Cartesian space on each coordinate axis (**Figure 12**), the overall behavior of the Delta robot during the execution of the whole trajectory 1 can be observed.

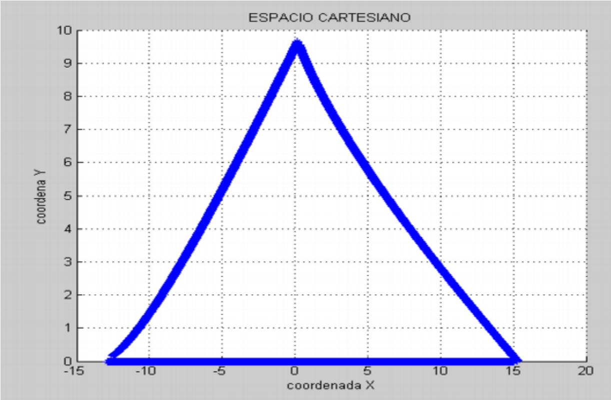



**Figure 11.** The trajectory behavior of the XY plane.

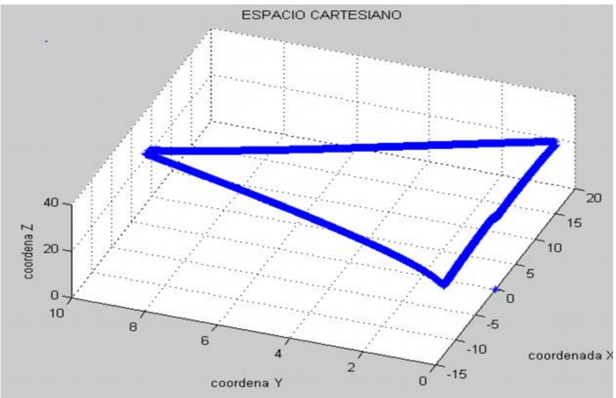



**Figure 12.** XYZ plane behavior trajectory.

The third-order characteristic polynomial and its respective position and speed limits are also analyzed to confirm that the actuator has reached the required position at each trajectory point.

Point track 1

$$39 + 0.2135t^2 - 0.0014t^3$$
$$39 + 0.2118t^2 - 0.0014t^3$$
$$39 + 0.2118t^2 - 0.0014t^3$$

Point track 2

$$59 + 0.2118t^1 + 01729t^2 - 0.0034t^3$$
$$59 + 0.2118t^1 + 01729t^2 - 0.0034t^3$$
$$59 + 0.2118t^1 + 01729t^2 - 0.0034t^3$$

Point track 3

$$75 + 2.6471t^1 + 0.0706t^2 - 0.0135t^3$$
$$75 + 2.6471t^1 + 0.0706t^2 - 0.0135t^3$$
$$75 + 2.6471t^1 + 0.0706t^2 + 0.0135t^3$$

Then, the behavior of each actuator (servo motor) can be evaluated by tracking the position generated by the above polynomial in different point trajectories constituting the trajectory (**Figure 13–21**).

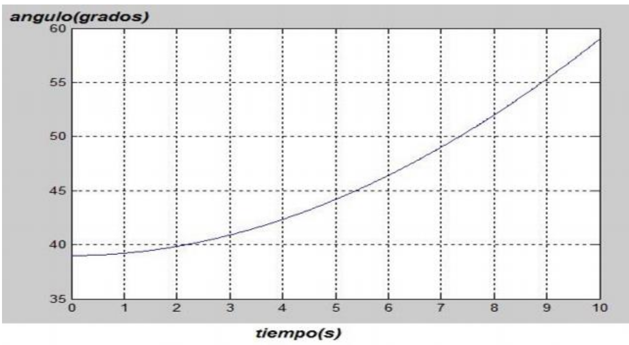



**Figure 13.** Behavior of actuator 1 on point trajectory 1 in MATLAB.

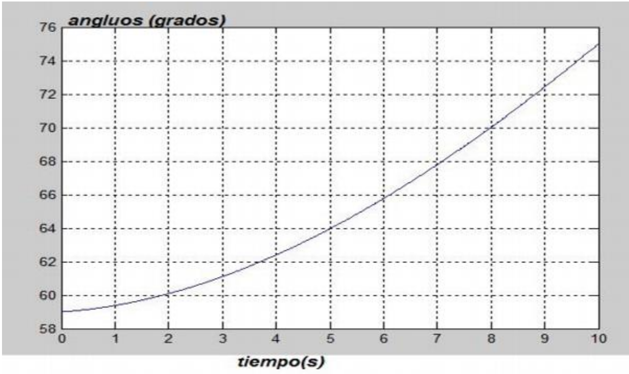



**Figure 14.** Behavior of actuator 1 on point trajectory 2 in MATLAB.

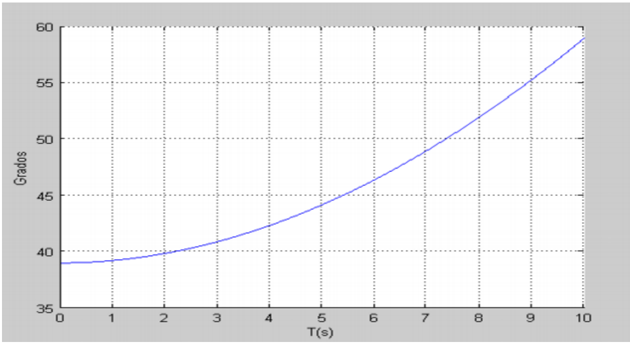



**Figure 15.** Behavior of actuator 1 on point trajectory 3 in MATLAB.

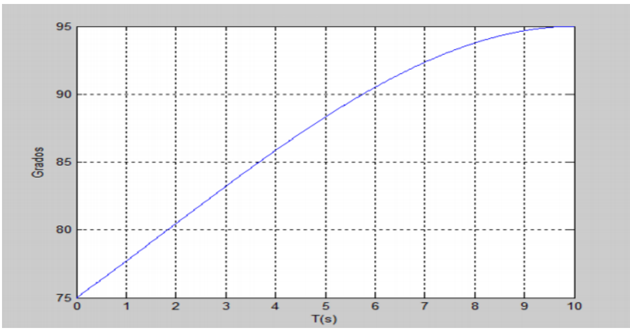



**Figure 16.** Behavior of actuator 2 on point trajectory 1 in MATLAB.

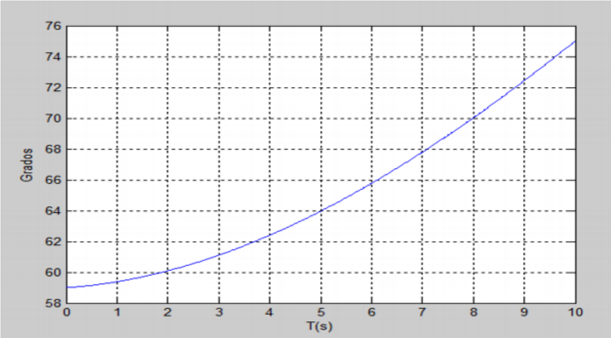



**Figure 17.** Behavior of actuator 2 on point trajectory 2 in MATLAB.

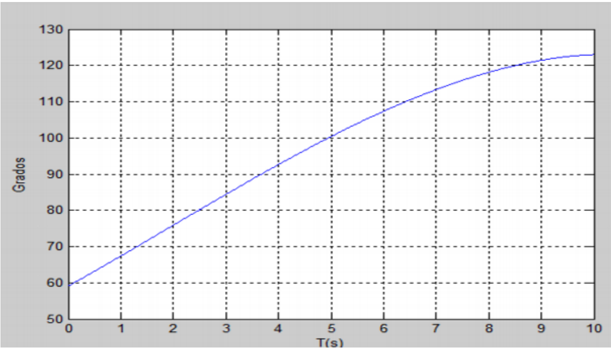



**Figure 18.** Behavior of actuator 2 on point trajectory 3 in MATLAB.

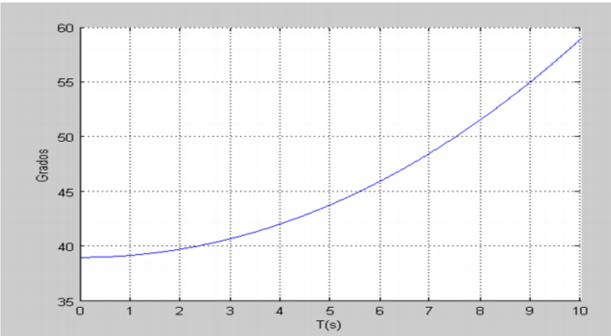



**Figure 19.** Behavior of actuator 3 on point trajectory 1 in MATLAB.

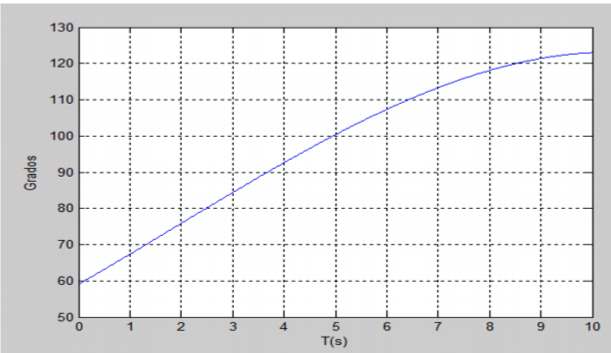



**Figure 20.** Behavior of actuator 3 on point trajectory 2 in MATLAB.

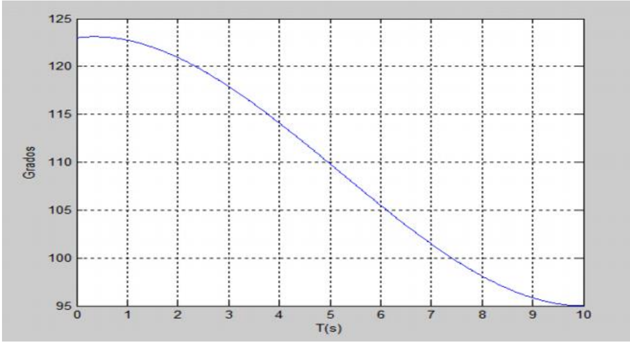



**Figure 21.** Behavior of actuator 3 on point trajectory 3 in MATLAB.

In the figure, you can see the continuous motion between the point trajectory difference of the inverse kinematics algorithm, so as to ensure that the end effector passes through each point trajectory accurately, which can greatly improve the time accuracy.

Finally, a simulation is generated in the three-dimensional environment of a mechanical system called MATLAB SimMechanics™. In this environment, the connecting rod behavior in the execution of the previously generated track can be verified by visualizing the positions of the initial point and the first point track (**Figure 22**) reached by the Delta robot along the straight line.

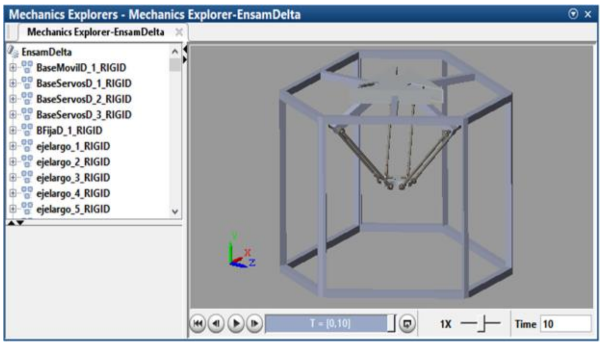



**Figure 22.** Simulation of Delta robot in MATLAB.

## 4.3. Track 2

The second path to be realized is realized by changing the speed in the algorithm, so that a semicircle centered on the origin between the track points can be formed in Cartesian space, where the coordinates of the points can be observed (**Table 6**).

**Table 6.** The points that make up track 2

| Spot | X coordinate | Y coordinate | Z coordinate |
|---|---|---|---|
| Initial | 0 | 0 | 30 |
| Track 1 | 15 | 0 | 30 |
| Track 2 | 0 | -15 | 30 |
| Track 3 | 0 | 0 | 30 |

The behavior of Delta robot in joint space and Cartesian space (**Figure 23–25**) describing trajectory 2 is as follows.

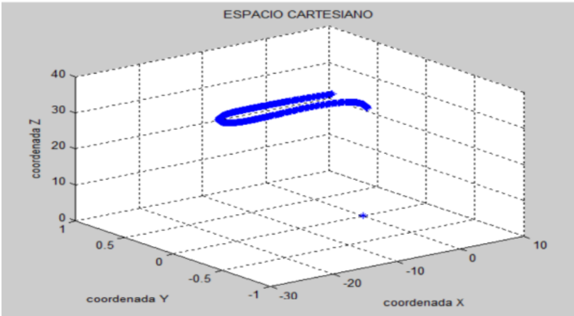



**Figure 23.** Behavior of point trace 1 in Cartesian space.

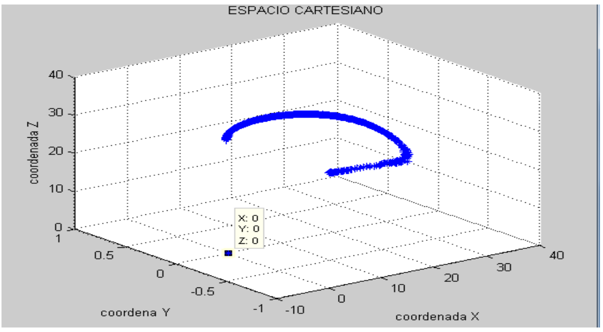



**Figure 24.** Behavior of point trace 2 in Cartesian space.

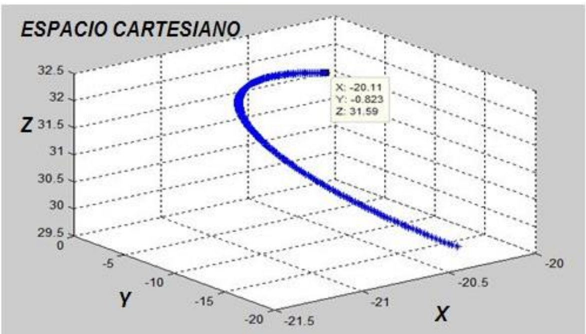



**Figure 25.** Behavior of point trace 3 in Cartesian space.

A fourth display of the trajectory is generated by changing the number of points (**Figure 26**).

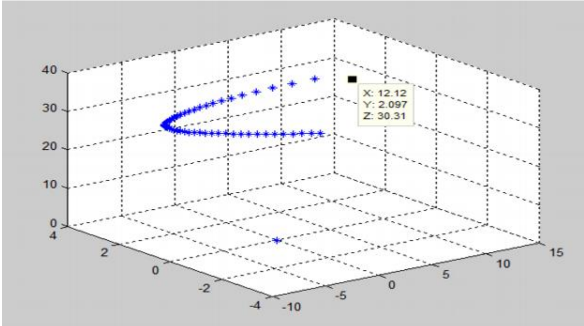



**Figure 26.** Behavior of point trace 4 in Cartesian space.

Finally, a simulation is generated in a three-dimensional environment of a mechanical system called MATLAB SimMechanics$^{TM}$. In this environment, the connecting rod behavior during the execution of the previously generated trajectory can be verified by visualizing the positions of the starting point and ending point (**Figure 27, 28**) of the trajectory reached by the Delta robot along the previously planned semicircular trajectory.

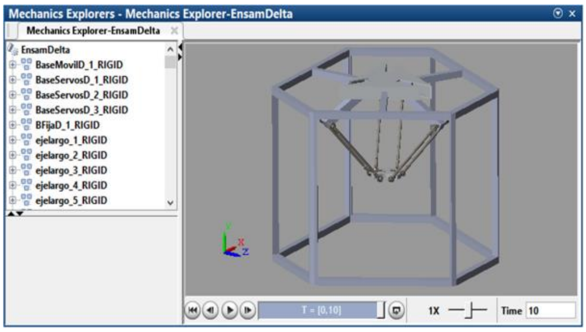



**Figure 27.** Robot Delta trajectory simulation 1.

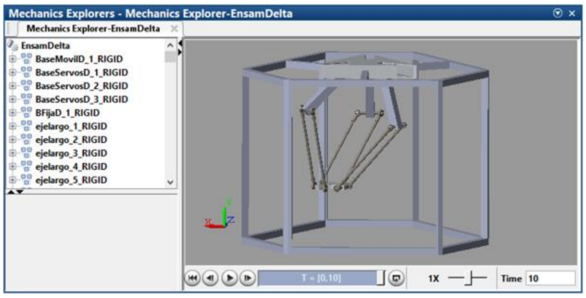



**Figure 28.** Robot Delta trajectory simulation 2.

## 5. Conclusion

The behavior of Delta robot is supported by the inverse kinematics trajectory algorithm. The algorithm needs to develop Jacobiano matrix or Jacobiana matrix, and simulate a cubic polynomial, which can compare the different computational advantages of the two algorithms.

Practice shows that the polynomial method has less computation than the iterative inverse kinematics algorithm, but its main advantage is that it can directly predict the singularity of the robot Delta together with the Jacobiano method.

These trajectories work according to the basic constant speed standard of linear motion. Therefore, with the change of speed vector, different trajectories will be generated. In some cases, the trajectories will be transferred from the interpolation process to the approximate process of position error.

## Conflict of interest

The authors declare that they have no conflict of interest.

## References

1. Barintos A, Peñin L, Balaguer C, *et al*. 2007. Fundamentos de Robótica [Fundamentals of robotics]. 2nd ed. New York: Mc Graw Hill; 2007.
2. Martin MA. Robótica [Robotics]. 1st ed. Valencia: Polytechnic University of Valencia; 2011.
3. Liu XJ, Wang J. Parallel kinematics: Type, kinematics and optimal design. Springer Tracts in Mechanical Engineering 2014. doi: 10.1007/978-3-642-36929-2.
4. Uriah, Lucas. Medina, Sergio. Diseño e Implementación de una Plataforma Robotica Tipo Delta [Design and implementation of Delta robot platform]. Bogotá: Nueva Grenada Military University; 2012.
5. Subir Kumar S. Introducción a la Robótica [Introduction to robotics] 1st ed. New York: Mc Graw Hill; 2010.
6. Craig, JJ. 2006. Robótica [Robotics]. 3rd ed. London: Pearson; 2004.
7. Nof SY. Handbook of industrial robotics. 2nd ed. San Francisco: Wiley; 1999.
8. Biagiotti L, Melchiorri C. 2008. Trajectory Planning for Automatic Machines and Robots. London: Springer Nature; 2008.
9. Niku SB. Introduction to robotics: Analysis, control, application. 2nd ed. San Francisco: Wiley; 2010.
10. Sicily B, Lorenzo C, Luigi V, *et al*. Robotics: Modelling, planning and control. IEEE Robotics & Automation Magazine 2009; 16(4): 101. doi: 10.1109/MRA.2009.934833.
11. Sponge MW, Hutchinson S, Vidyasagar M. Robot modeling and control. 1st ed. San Francisco: Wiley; 2005.