



Scientia Et Technica

ISSN: 0122-1701

scientia@utp.edu.co

Universidad Tecnológica de Pereira  
Colombia

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

Planeación y ejecución de trayectorias en un robot Delta

Scientia Et Technica, vol. 22, núm. 2, junio, 2017, pp. 183-192

Universidad Tecnológica de Pereira  
Pereira, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=84953103010>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# planeación y ejecución de trayectorias en un robot Delta

## Planning and implementation of paths in a robot Delta

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

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

u1801272@unimilitar.edu.co

u1801359@unimilitar.edu.co

ricardo.castillo@unimilitar.edu.co

**Resumen -** El principal objetivo de este trabajo de investigación es el planear y ejecutar trayectorias eficientes en un robot delta, a través de una interfaz gráfica (GUI), y un adecuado circuito electrónico que se integre óptimamente con el prototipo mecánico ya diseñado e implementado en el laboratorio de Robótica de la Universidad Militar Nueva Granada, teniendo en cuenta y respetando sus diferentes propiedades como; estructura mecánica, espacio de trabajo, cinemática, dinámica y singularidades de movimiento, que permitan el correcto desarrollo e implementación de un algoritmo que logre describir y controlar eficientemente las trayectorias planeadas.

**Palabras Clave-** Algoritmo, cinemática directa, cinemática inversa, espacio cartesiano, espacio de articulación, espacio de trabajo, Jacobiano, polinomio tercer orden, Robot Delta, singularidades.

**Abstract -** This article will show a mathematical development in order to plan and implement efficient paths on a Delta robot through a graphical user interface (GUI) that is integrated optimally with the mechanical prototype and designed and implemented in the laboratory of Robotics Universidad Militar Nueva Granada, taking into account and respecting their different properties; Dynamic mechanical kinematic structure, workspace, and singularity of motion, which allow the correct development and implementation of an algorithm that achieves the description and efficiently manage the planned trajectories.

**Key Words-** Algorithm, Cartesian space, Delta robot, forward inverse kinematics, Jacobian, joint space, kinematics, third-order polynomial, singularities, workspace.

## I. INTRODUCCIÓN

Desde sus primeros modelos hasta los más influyentes en la actualidad se han generado diversas interpretaciones sobre que es un robot industrial, por lo tanto, a continuación se presentan dos de las más significativas [1].

- **RIA (Robotic Industries Association):** Manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas o dispositivos

especiales, según trayectorias variables, programadas para realizar tareas diversas

- **ISO (International Organization for Standardization):** Manipulador de tres ejes o más, con control automático, reprogramable, multiplicación, móvil o no, destinado a ser utilizado en aplicaciones de automatización industrial. Incluye al manipulador (sistema mecánico y accionadores) y al sistema de control (software y hardware de control y potencia).

Los diferentes robots industriales se pueden clasificar así: [2].

1. Temporalmente, es decir de acuerdo a su fecha de fabricación.
2. Por su grado de inteligencia.
3. Por su funcionalidad.
4. Por el tipo de control, que de acuerdo con la Asociación Francesa de Robótica Industrial (AFRI) realiza una clasificación de los robots según el tipo de control utilizado en cuatro categorías:

- Tipo A o tele-manipulado: Que posee un control manual o telemundo.
- Tipo B o secuencial: Automático es decir, que posee ciclos pre-ajustados, regulación mediante fines de carrera o topes, control por PLC y accionamientos ya sean neumáticos, eléctricos o hidráulicos.
- Tipo C o de trayectoria controlable: Que es programable ya sea con trayectorias continuas o punto a punto, pero que carece de conocimientos sobre su entorno.
- Tipo D o adaptativo: Que es capaz de adquirir datos de su entorno, readaptando así sus tareas en función de dichos datos.

5. Por su geometría o configuración

- Robots industriales manipuladores.
- Robots industriales móviles.

El robot Delta (Figura 1) es un manipulador paralelo, el cual consta de tres cadenas cinemáticas cerradas de configuración PR(Ps), donde (Ps) representa el espacio paralelogramo de cuatro barras con cuatro articulaciones esféricas, y donde PR representa que cada una de las cadenas cinemáticas se compone de una articulación prismática pasiva y una articulación de rotación activa que conectan una base fija a una base móvil [3].

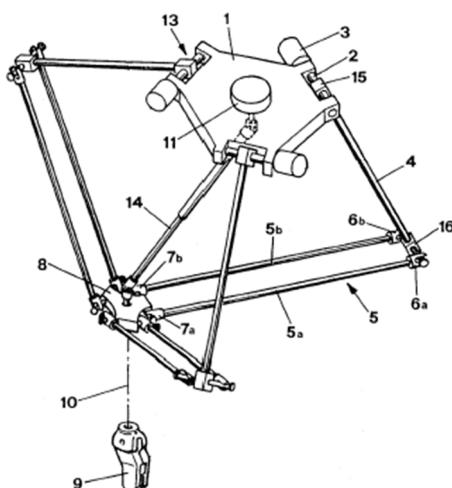


Figura 1. Robot Delta de R. Clavel. (US patent No. 4976582).

Algunas de las principales ventajas y desventajas del robot Delta como también de muchas otras plataformas paralelas son:

VENTAJAS	DESVENTAJAS
Arquitectura más rígida	Cinemática compleja
Relación carga/peso muy superior	Espacio de trabajo reducido
Robot ligero	Calibración compleja
Alta precisión	Posiciones singulares complejas

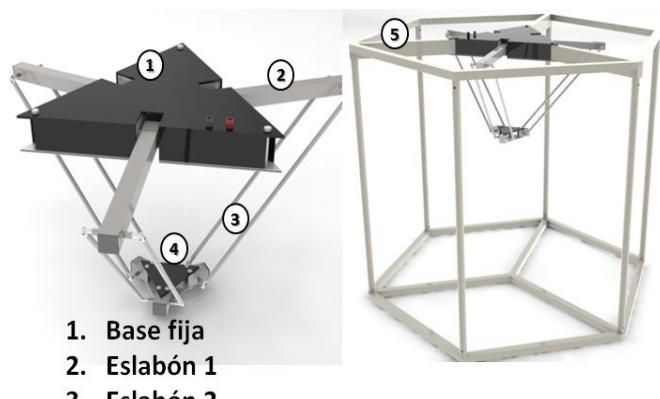
Altas velocidades y aceleraciones

Tecnologías no desarrolladas

El robot Delta como la mayoría de los robots industriales que son diseñados, construidos e implementados en la actualidad, tienen como principal objetivo el aumentar la productividad y calidad de los trabajos realizados por el ser humano en las diversas tareas complejas y/o riesgosas, en las cuales se pueden desempeñar de mejor manera, dichas tareas requieren de movimientos coordinados y precisos por parte del robot para poder llevarlas a cabo de forma eficiente. Por lo cual es de suma importancia el poder entender las propiedades cinemáticas y dinámicas de los robots, para así comprender y analizar su capacidad de movimiento proyectado hacia la planificación y generación de trayectorias en la elaboración de cualquier actividad.

Debido al continuo crecimiento de la robótica en la industria, cada vez se hace más necesario que los estudiantes afines a esta ciencia tengan un mayor conocimiento sobre los distintos robots y sus aplicaciones que rigen actualmente la industria mundial, en los cuales hay un gran porcentaje de manipuladores en paralelo.

Por lo tanto el principal objetivo de este paper se enfoca en la planeación y ejecución de trayectorias en el prototipo mecánico de un robot Delta diseñado e implementado en la Universidad Militar Nueva Granada (Figura 2), a través de una interfaz gráfica de usuario que pueda interactuar con el prototipo mecánico en tiempo real, permitiéndole simular y ejecutar trayectorias tanto en el espacio cartesiano como en el espacio de articulaciones, asemejando tareas reales implementadas en la industria actual.



1. Base fija
2. Eslabón 1
3. Eslabón 2
4. Base móvil
5. Estructura soporte

Figura 2. Prototipo Robot Delta, laboratorio robótica UMNG

## II. ROBOT DELTA

### • Espacio de Trabajo

El espacio de trabajo está definido por el conjunto de puntos que puede alcanzar el robot alrededor de sí mismo, con base en su configuración y tamaño de sus vínculos y articulaciones de la muñeca. La forma del espacio de trabajo para cada robot es exclusiva, ya que depende de su diseño [9].

El espacio de trabajo se puede subdividir en dos categorías así [11]:

- Espacio de trabajo alcanzable: Está compuesto por el conjunto de puntos alcanzables por el manipulador.
- Espacio de trabajo diestro: Está compuesto por el conjunto de puntos que el manipulador puede alcanzar con una orientación arbitraria de su efecto final.

Para una adecuada planeación de trayectorias tanto en el robot Delta como en cualquier otro robot industrial, es indispensable el poder conocer cuál es el espacio de trabajo en que dicho robot tendrá que ejecutar sus diversas tareas, que pueden ir desde un movimiento simple hasta movimientos coordinados con otros robots en algún proceso industrial.

El espacio de trabajo del robot Delta (Tabla 1) con el cual se van a diseñar las diferentes trayectorias tiene como principales propiedades [4]:

Coordenadas	Alcance mínimo	Alcance máximo
X	-342,8015 mm	342,4004 mm
Y	-296,9611 mm	296,9611 mm
Z	105,7695 mm	677,0132 mm

Tabla 1. Alcance máximo y mínimo de la plataforma móvil.

### • Cinemática Inversa

El principal objetivo de la cinemática inversa radica en encontrar los ángulos de las articulaciones, conociendo la configuración (posición y orientación) del efecto final del robot [5]. Este análisis es importante para la planeación y ejecución de trayectorias ya que se basa en la no linealidad de las ecuaciones encontrando múltiples soluciones, así, el procedimiento para encontrar la cinemática inversa depende de la configuración del robot y se puede resolver geométricamente, tanto para manipuladores en serie como en paralelo.

Aprovechando la simetría del robot Delta, el análisis cinemático inverso se desarrolla para cada una de las cadenas cinemáticas de forma independiente, dando como resultado un modelo matemático que describe a cada articulación como se puede encontrar en [4].

### • Cinemática Directa

El principal objetivo de la cinemática directa radica en encontrar la configuración (posición y orientación) en la que se encuentra el efecto final del manipulador, ya que en este análisis las variables de articulación se encuentran ya definidas [5].

El análisis cinemático directo del robot Delta se desarrolla partiendo de un sistema de ecuaciones no lineales que relaciona la posición de la plataforma móvil con los ángulos de sus eslabones, generando un modelo matemático que describe cada coordenada cartesiana como se puede encontrar en [4].

## III. PLANEACION TRAYECTORIAS

### • Espacio Cartesiano

La descripción de movimientos en el espacio cartesiano, considera métodos de generación de rutas en los cuales las formas de estas se especifican en términos de funciones que calculan la posición y la orientación cartesianas como funciones del tiempo (Figura 3), generalmente se prefiere en planificación de movimientos, donde no se incluyan obstáculos en el entorno y en los cuales la dinámica del robot no tenga mayor relevancia [6], [7].

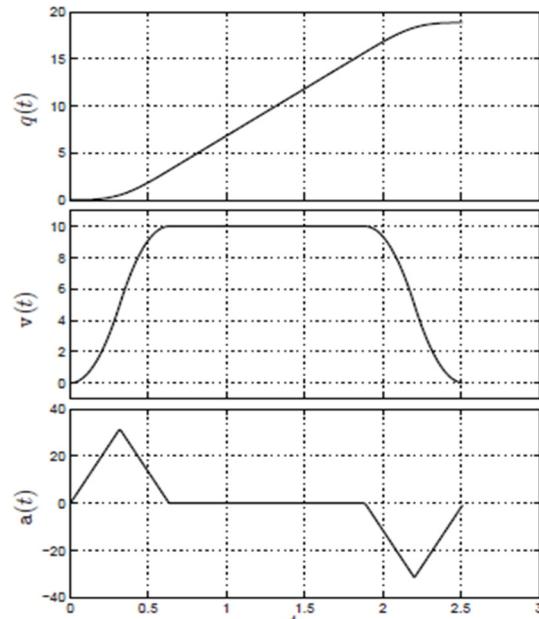


Figura 3. Posición, velocidad y aceleración en espacio cartesano [8].

La planeación de trayectorias en el espacio cartesiano se desarrolla principalmente a través del Jacobiano o matriz Jacobiana del manipulador en el cual se van a implementar dichas trayectorias. Esta matriz es una representación geométrica de los elementos que componen un mecanismo en función del tiempo, la cual permite la conversión de movimientos diferenciales o velocidades de las articulaciones individuales a movimientos diferenciales o velocidades en puntos de interés como puede ser el efecto final [9].

El Jacobiano o matriz Jacobiana constituye una de las más importantes herramientas para la caracterización de un manipulador, ya que es bastante útil para problemas como [10].

- ✓ Encontrar singularidades.
- ✓ Análisis de redundancias.
- ✓ Determinación del algoritmo de cinemática inversa.
- ✓ Descripción del mapeo entre las fuerzas aplicadas al efecto final los torques resultantes en las articulaciones.

La Matriz Jacobiana relaciona las velocidades de las articulaciones con la velocidad del sistema de referencia cartesiano

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

Así el Jacobiano o matriz Jacobiana del robot Delta es planteada de la siguiente forma [4]:

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

donde:

$J_x$  = Matriz asociada con la cinemática directa

$J_\theta$  = Matriz asociada con la cinemática inversa

Planteándose así el siguiente sistema de ecuaciones

$$\begin{bmatrix} j_{x11} & j_{x12} & j_{x13} \\ j_{x21} & j_{x22} & j_{x23} \\ j_{x31} & j_{x32} & j_{x33} \end{bmatrix} \begin{bmatrix} V_{px} \\ V_{py} \\ V_{pz} \end{bmatrix} = \begin{bmatrix} j_{\theta1} & 0 & 0 \\ 0 & j_{\theta2} & 0 \\ 0 & 0 & j_{\theta3} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} \quad (3)$$

Que generara de forma definitiva la estructura para el Jacobiano o matriz Jacobiana del robot Delta así

$$J = (J_\theta)^{-1} J_x = \left( \begin{bmatrix} j_{\theta1} & 0 & 0 \\ 0 & j_{\theta2} & 0 \\ 0 & 0 & j_{\theta3} \end{bmatrix} \right)^{-1} \begin{bmatrix} j_{x11} & j_{x12} & j_{x13} \\ j_{x21} & j_{x22} & j_{x23} \\ j_{x31} & j_{x32} & j_{x33} \end{bmatrix} \quad (4)$$

Dependiendo de qué matriz de las planteadas anteriormente sea singular, un mecanismo de lazo cerrado podría tener una configuración singular de cinemática directa, una configuración singular de cinemática inversa, o ambas.

La planeación de trayectorias se desarrollara por medio del algoritmo de cinemática inversa, el cual puede ser usado para obtener un modelo matemático que representara un movimiento diferencial como se ve en la ecuación 4.5.

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

Esta ecuación puede ser desarrollada fácilmente por un computador ya que es un proceso iterativo que va calculando en cada muestra la salida  $q(k)$ .

#### • Espacio de Articulaciones

La descripción de movimientos en el espacio de articulación, considera métodos de generación de rutas en los cuales las formas de estas (en espacio y tiempo) se especifican en términos de funciones de ángulos de articulación (Figura 4) [6].

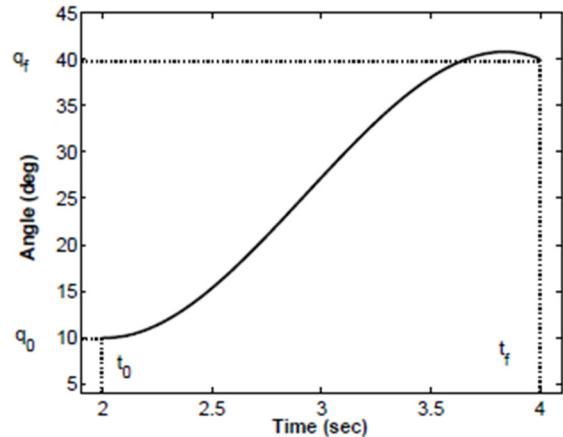


Figura 4. Representación típica en el espacio de articulación [11].

El tiempo requerido por cada segmento de la ruta debe ser el mismo para cada articulación, de manera que todas las articulaciones involucradas lleguen al punto vía al mismo tiempo [6].

La cinemática inversa permite el cálculo de un conjunto de ángulos de articulación que corresponderá a la configuración (posición y orientación) de destino, conociendo un punto inicial y un punto final se desea encontrar una función que para un  $t_o$  determine el valor inicial de la configuración y para un  $t_f$  determine un valor final de la configuración.

Para crear una función uniforme son evidentes al menos cuatro restricciones sobre  $\theta(t)$ , de las cuales dos serán restricciones para el valor inicial y final de la posición, mientras las otras dos restricciones corresponderán a condiciones en la velocidad inicial y final

$$\begin{aligned} \theta(0) &= \theta_0 \\ \theta(t_f) &= \theta_f \\ \dot{\theta}(0) &= 0 \\ \dot{\theta}(t_f) &= 0 \end{aligned}$$

Estas restricciones satisfacen así un polinomio mínimo de grado 3 (polinomio cubico) que tendrá como descripción de su posición, velocidad y aceleración respectivamente a

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

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

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

De manera que combinando las restricciones con las ecuaciones (6, 7 y 8) se puede encontrar fácilmente las constantes  $a_0, a_1, a_2, a_3$  donde

$$a_0 = \theta_0$$

$$a_1 = 0$$

$$a_2 = \frac{3}{t^2_f}(\theta_f - \theta_0)$$

$$a_3 = -\frac{2}{t^3_f}(\theta_f - \theta_0)$$

Una vez halladas todas las constantes del polinomio se puede conectar cualquier ángulo de articulación desde una posición inicial a una final deseadas, este caso es para una velocidad inicial y final cero pero si desea se puede aumentar el número de restricciones y aumentar el grado del polinomio con el fin de mayor control a la hora de generar la trayectoria, como por ejemplo dar restricciones a la velocidad y aceleración en cada punto que se analice.

Para mayor rendimiento computacional se prefiere llevar las ecuaciones a una forma matricial vectorial.

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

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

donde:

$$[C] = \text{Matriz de coeficientes } [a_n \dots a_0]^t$$

$$[M] = \text{Matriz de coeficientes potencias del tiempo } [b_n t^n \dots b_0 t^0]^t$$

$$[\theta] = \text{Matriz de restricción, velocidad y aceleración}$$

Obteniendo de como representación final

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

#### IV. EJECUCIÓN TRAYECTORIAS

Esta sección está dedicada al análisis y ejecución de las trayectorias que se implementaran en el robot Delta, con el objetivo de probar el comportamiento del robot frente al algoritmo de cinemática inversa en una trayectoria punto a punto.

#### Algoritmo de Ejecución

El algoritmo que se plantea y ejecuta para la planeación y ejecución de las trayectorias tiene como estructura de desarrollo:

1. Calculo de la cinemática inversa en los puntos vías.
2. Calculo de la velocidad entre puntos vía.
3. Calculo de la matriz Jacobiana.
4. Implementación de la ecuación 4.5.
5. Envío de los ángulos entregados en el paso 4 al circuito electrónico.
6. Cinemática directa de los ángulos entregados en el paso 4.

#### Trayectoria 1

La primera trayectoria implementada se diseñara para formar un triángulo equilátero, dicha trayectoria se planeara sobre el plano XY, manteniendo un valor constante en la coordenada Z, y se diseñara a través de cinco puntos en total de los cuales tres van a ser puntos vías y dos son los puntos inicial y final de la trayectoria. Las coordenadas de los puntos se observan en la tabla 2.

Puntos	Coordenada X	Coordenada Y	Coordenada Z
Inicial	0	0	30
Vía 1	15	0	30
Vía 2	0	15	30
Vía 3	-15	0	30
Final	0	0	30

Tabla 2. Puntos que conforman la trayectoria 1.

El comportamiento del robot Delta en el espacio de articulaciones que describe la trayectoria 1 se evidencia (Figuras 5-7) a continuación.

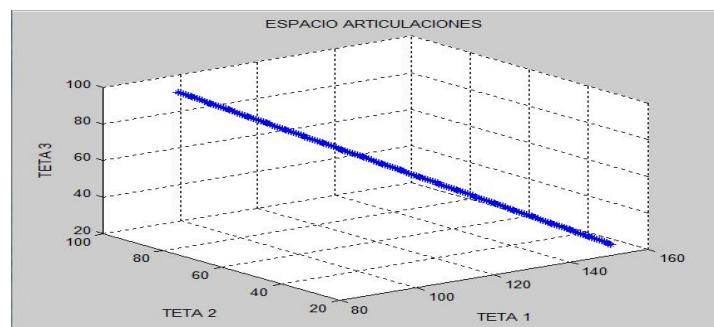
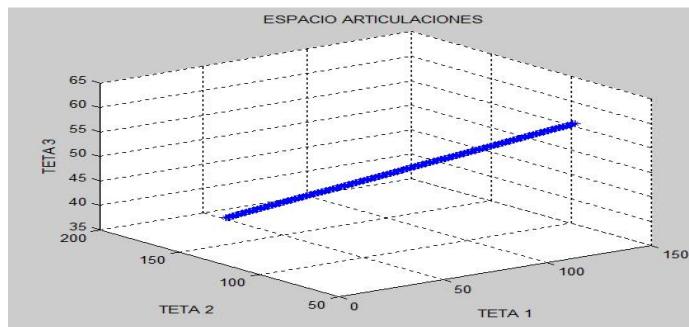
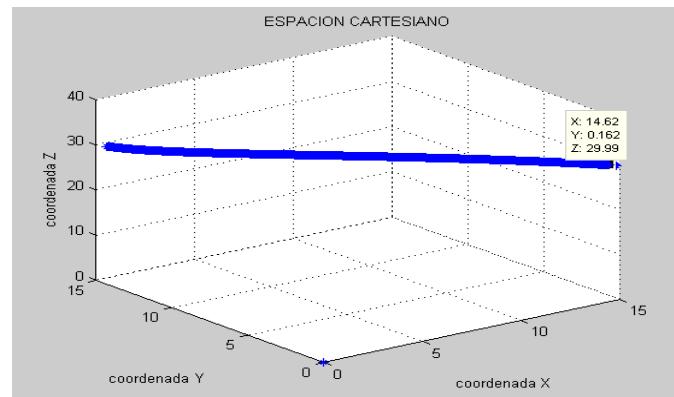


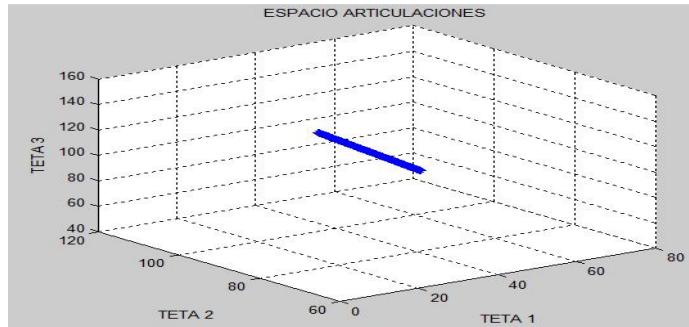
Figura 5. Comportamiento del punto vía 1 en espacio de articulaciones en MATLAB.



**Figura 6.** Comportamiento del punto vía 2 en espacio de articulaciones en MATLAB.

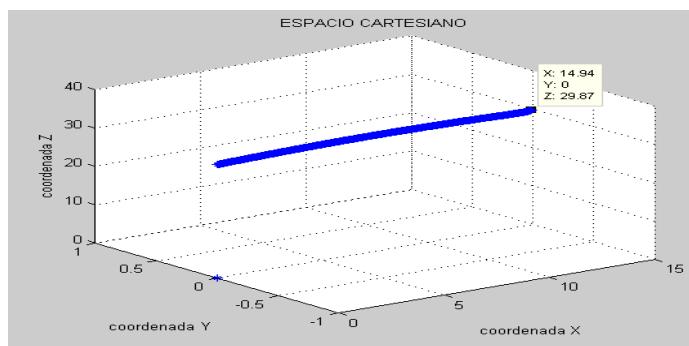


**Figura 9.** Comportamiento del punto vía 2 en espacio cartesiano en MATLAB.

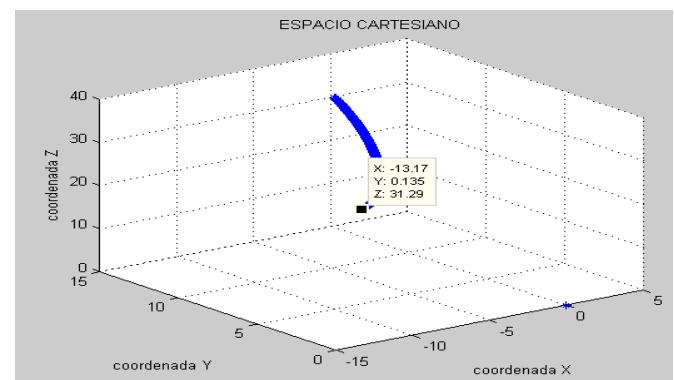


**Figura 7.** Comportamiento del punto vía 3 en espacio de articulaciones en MATLAB.

El comportamiento del robot Delta en el espacio cartesiano que describe la trayectoria 1 se evidencia (Figuras 8-10) a continuación.



**Figura 8.** Comportamiento del punto vía 1 en espacio cartesiano en MATLAB.



**Figura 10.** Comportamiento del punto vía 3 en espacio cartesiano en MATLAB.

En esta trayectoria, se eligió una velocidad constante entre los puntos vía la cual garantiza una línea recta entre dichos puntos como se pudo apreciar en las gráficas que describen la trayectoria en el espacio de articulaciones de el algoritmo de cinemática inversa.

Para un correcto seguimiento por parte del robot Delta de la trayectoria en el desarrollo de esta se implementaron 480 iteraciones que permiten apreciar mejor la línea recta que forman los 3 actuadores.

Por lo tanto se puede apreciar que los ángulos siempre están entre el intervalo (0 a 180) lo que garantiza una no singularidad en el robot Delta, si dado el caso los ángulos fueran mayores a 180 o menores a 0 se produciría una singularidad en el robot lo cual generaría un punto en el espacio de trabajo que el robot no podría alcanzar.

Se puede apreciar entonces el comportamiento total del robot Delta durante toda la ejecución de la trayectoria 1 tanto en el plano XY (Figura 11) como en el espacio cartesiano en cada eje coordenado (Figura 12).

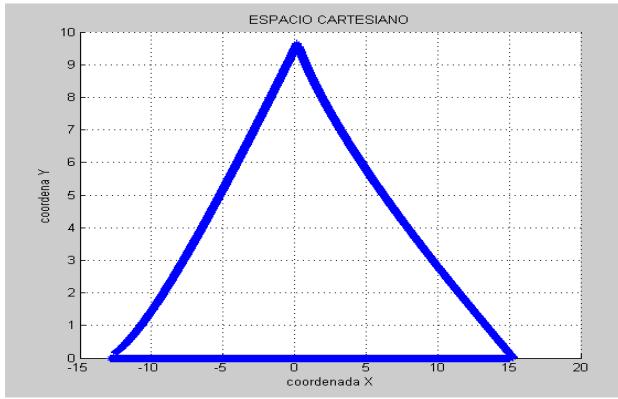


Figura 11. Comportamiento de trayectoria plano XY.

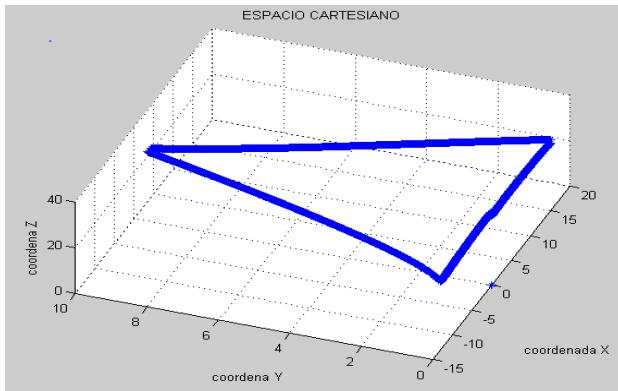


Figura 12. Comportamiento de trayectoria plano XYZ.

Se plantea también un análisis por medio del polinomio característico de orden tres con sus respectivas restricciones de posición y velocidad, para corroborar que se esté alcanzando la posición deseada en cada uno de los puntos vía por parte de los actuadores.

#### Punto vía 1

$$\begin{aligned} 39 + 0.2135t^2 - 0.0014t^3 \\ 39 + 0.2118t^2 - 0.0014t^3 \\ 39 + 0.2118t^2 - 0.0014t^3 \end{aligned}$$

#### Punto vía 2

$$\begin{aligned} 59 + 0.2118t^1 + 0.1729t^2 - 0.0034t^3 \\ 59 + 0.2118t^1 + 0.1729t^2 - 0.0034t^3 \\ 59 + 0.2118t^1 + 0.1729t^2 - 0.0034t^3 \end{aligned}$$

#### Punto vía 3

$$\begin{aligned} 75 + 2.6471t^1 + 0.0706t^2 - 0.0135t^3 \\ 75 + 2.6471t^1 + 0.0706t^2 - 0.0135t^3 \\ 75 + 2.6471t^1 + 0.0706t^2 + 0.0135t^3 \end{aligned}$$

A continuación se puede apreciar el comportamiento de cada uno de los actuadores (servomotores) siguiendo el posicionamiento generado por el polinomio anteriormente descrito durante los diferentes puntos vía que componen la trayectoria (Figuras 13-21).

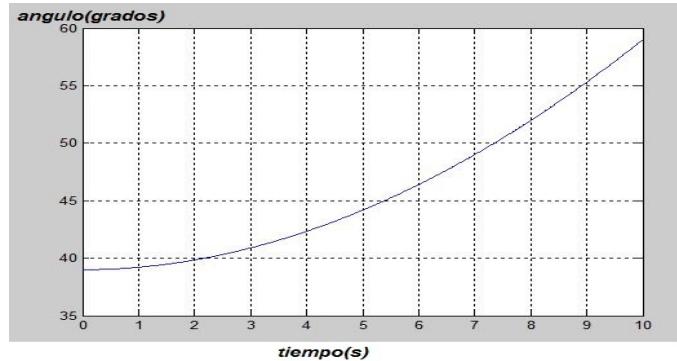


Figura 13. Comportamiento del actuador 1 en el punto vía 1 en MATLAB.

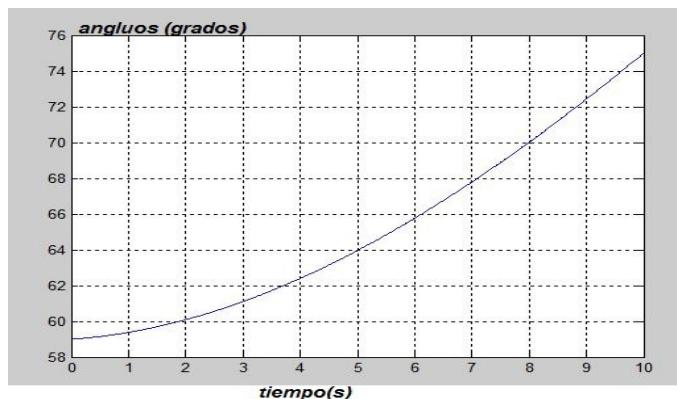


Figura 14. Comportamiento del actuador 1 en el punto vía 2 en MATLAB.

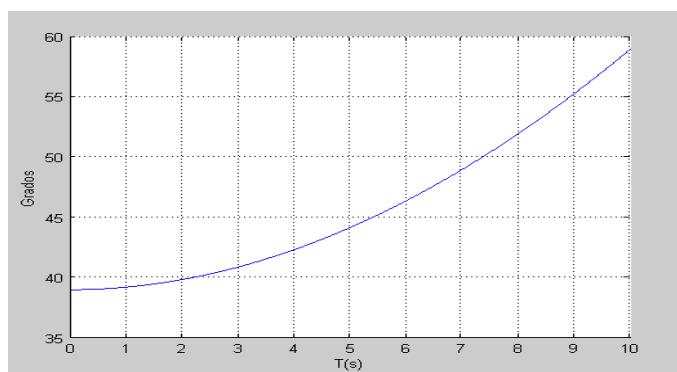


Figura 15. Comportamiento del actuador 1 en el punto vía 3 en MATLAB.

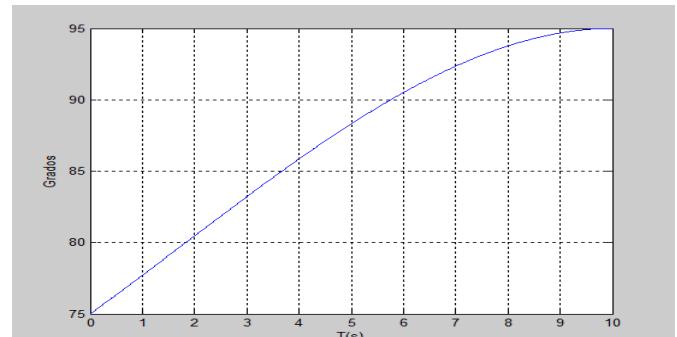


Figura 16. Comportamiento del actuador 2 en el punto vía 1 en MATLAB.

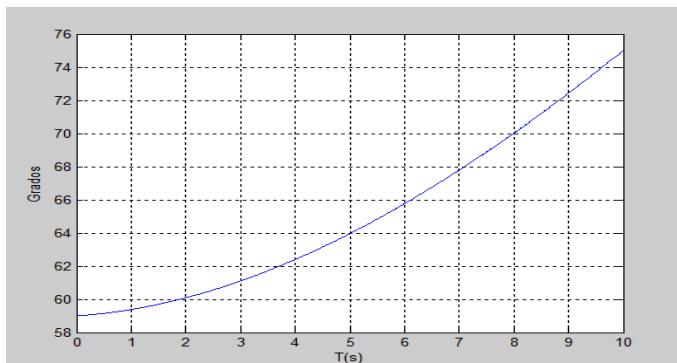


Figura 17. Comportamiento del actuador 2 en el punto vía 2 en MATLAB.

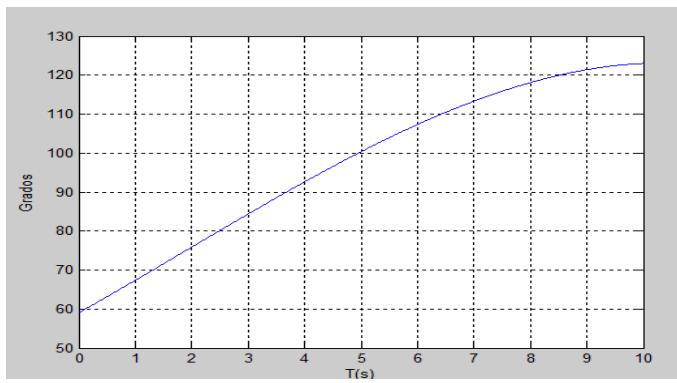


Figura 18. Comportamiento del actuador 2 en el punto vía 3 en MATLAB.

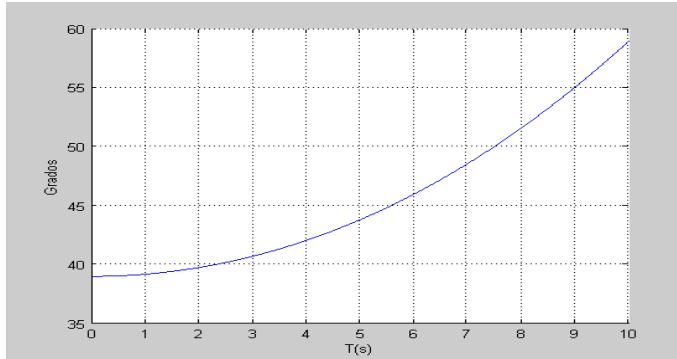


Figura 19. Comportamiento del actuador 3 en el punto vía 1 en MATLAB.

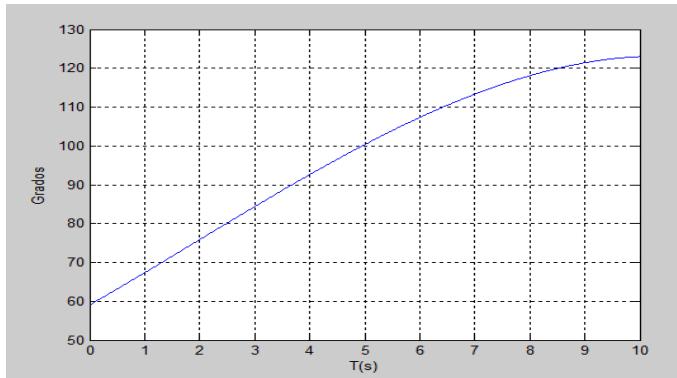


Figura 20. Comportamiento del actuador 3 en el punto vía 2 en MATLAB.

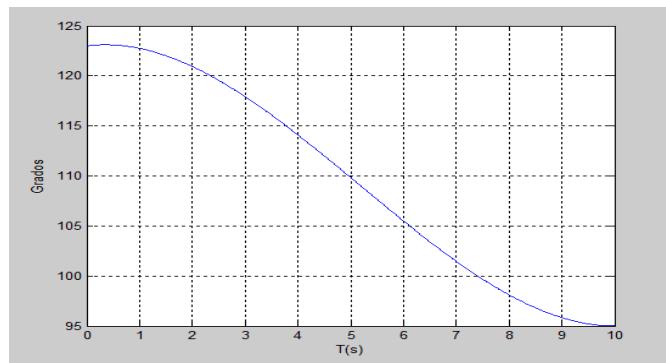


Figura 21. Comportamiento del actuador 3 en el punto vía 3 en MATLAB.

En las graficas se puede apreciar un movimiento continuo entre los puntos vía diferencia del algoritmo de cinemática inversa garantizando así que el efecto final pase exactamente cada punto vía, esto puede ser de una gran ventaja la hora de precisión.

Por último se genera una simulación en un entorno 3D para sistemas mecánicos conocido como SimMechanics™ de MATLAB, en el cual se podrá corroborar el comportamiento de los eslabones en la ejecución de la trayectoria previamente generada visualizando la ubicación del punto inicial y el primer punto vía de la trayectoria (Figuras 22), alcanzados por el robot Delta siguiendo una línea recta.

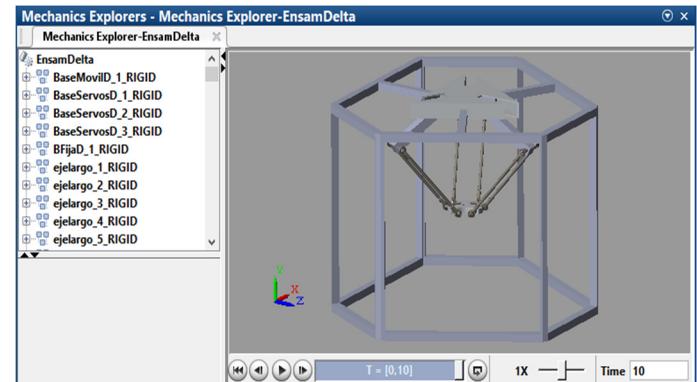


Figura 22. Simulación Robot Delta en MATLAB.

## Trayectoria 2

La segunda trayectoria a implementar se llevará a cabo variando la velocidad en el algoritmo de tal forma que en el espacio cartesiano se pueda formar una semicircunferencia con centro en el origen entre los puntos vías, donde las coordenadas de los puntos se pueden observar a continuación (Tabla 6).

Puntos	Coordinada X	Coordinada Y	Coordinada Z
Inicial	0	0	30
Vía 1	15	0	30
Vía 2	0	-15	30
Vía 3	0	0	30

Tabla 6. Puntos que conforman la trayectoria 2.

El comportamiento del robot Delta tanto en el espacio de articulación como en el espacio cartesiano (Figuras 23-25) que describe la trayectoria 2 se puede evidenciar a continuación.

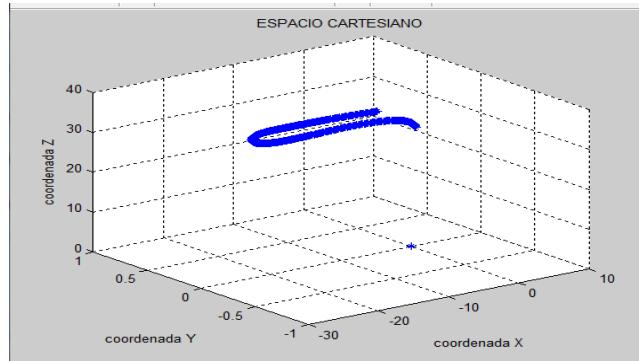


Figura 23. Comportamiento del punto vía 1 en espacio cartesiano.

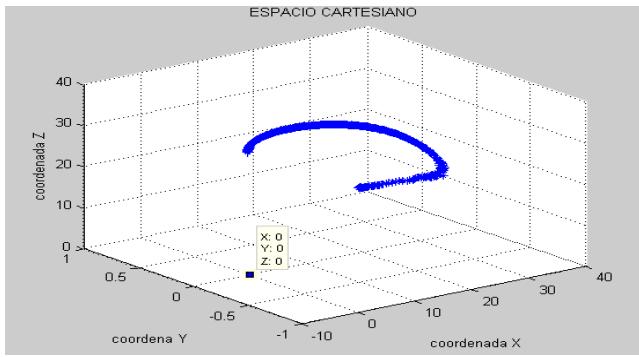


Figura 24. Comportamiento del punto vía 2 en espacio cartesiano.

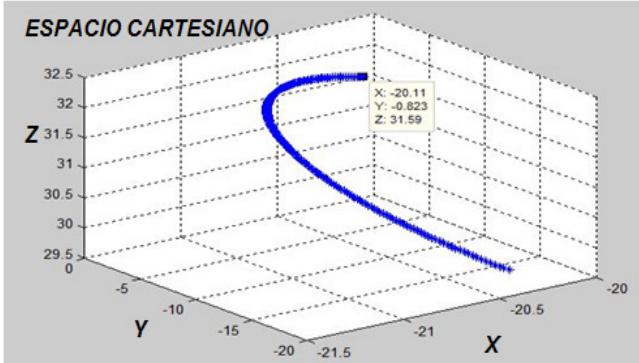


Figura 25. Comportamiento del punto vía 3 en espacio cartesiano.

Se genera una cuarta visualización de la trayectoria variando el número puntos (Figura 26).

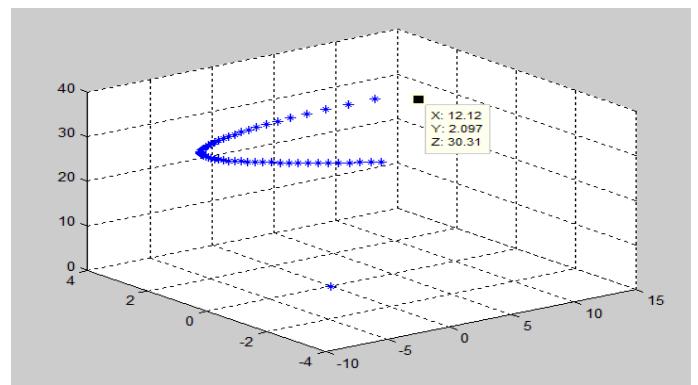


Figura 26. Comportamiento del punto vía 4 en espacio cartesiano.

Por último se genera una simulación en un entorno 3D para sistemas mecánicos conocido como SimMechanics™ de MATLAB, en el cual se podrá corroborar el comportamiento de los eslabones en la ejecución de la trayectoria previamente generada visualizando la ubicación del punto inicial y el punto final de la trayectoria (Figuras 27 y 28), alcanzados por el robot Delta siguiendo la semicircunferencia previamente planeada.

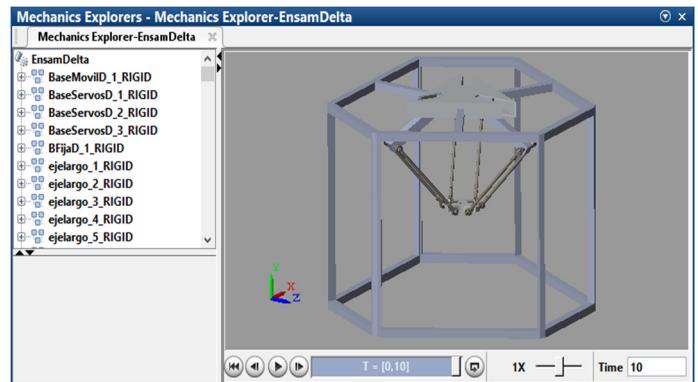


Figura 27. Simulación 1 Robot Delta trayectoria 2.

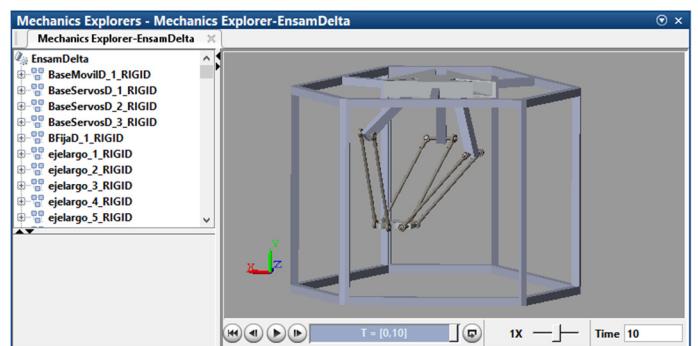


Figura 28. Simulación 2 Robot Delta trayectoria 2.

## V. CONCLUSIONES

Se corrobora el comportamiento del robot Delta con el algoritmo de trayectorias de la cinemática inversa, con el cual se requiere el desarrollo del Jacobiano o matriz Jacobiana, así

como también la simulación de un polinomio de tercer grado con el cual se pudo comparar las diferentes ventajas computacionales entre los dos algoritmos representados.

Se concluyó de forma práctica que el método de polinomios es de menor costo computacional que el algoritmo de cinemática inversa por iteraciones, pero cuya principal ventaja es la de poder predecir las singularidades del robot Delta por trabajar directamente con el Jacobiano.

Las trayectorias se trabajaron con un criterio de velocidad constante fundamental para la ejecución de movimientos en línea recta, así al variar el vector velocidad se generaban diferentes trayectorias y en ciertos casos las trayectorias pasaban de un proceso de interpolación a uno de aproximación generando un error de posición.

## REFERENCIAS

- [1] Barrientos, Antonio., Peñín, Luis Felipe., Balaguer, Carlos., Aracil, Rafael., 2007. Fundamentos de Robótica, 2<sup>nd</sup> ed. Mc Graw Hill.
- [2] Mellado Arteche, Martín., 2011. Robótica, 1<sup>st</sup> ed. Universidad Politécnica de Valencia.
- [3] Liu, Xin-Jun., Wang, Jinsong., 2014. Parallel Kinematics: Type, Kinematics, and Optimal Design. Springer.
- [4] Urrea, Lucas., Medina, Sergio. Diseño e Implementación de una Plataforma Robotica Tipo Delta. Trabajo de Grado Universidad Militar Nueva Granada. Facultad de Ingeniería. Ingeniería en Mecatrónica, 2012.
- [5] Saha, Subir Kumar., 2010. Introducción a la Robótica. 1<sup>st</sup> Ed. Mc Graw Hill.
- [6] Craig, John J., 2006. Robótica, 3<sup>rd</sup> ed. Pearson.
- [7] Nof, Shimon Y., 1999. Handbook of Industrial Robotics, 2<sup>nd</sup> ed. Wiley.
- [8] Biagiotti, Luigi., Melchiorri, Claudio., 2008. Trajectory Planning for Automatic Machines and Robots, Springer
- [9] Niku, Saeed B., 2010. Introduction to Robotics: Analysis, Control, Applications, 2<sup>nd</sup> ed. Wiley.
- [10] Siciliano, Bruno., Sciavicco, Lorenzo., Villani, Luigi., Oriolo, Giuseppe., 2009. Robotics: Modelling, Planning and Control. Springer.
- [11] Spong, Mark W., Hutchinson, Seth., Vidyasagar, M., 2005. Robot Modeling and Control, 1<sup>st</sup> ed. Wiley.