



Facultad de Ingeniería

ISSN: 0121-1129

revista.ingenieria@uptc.edu.co

Universidad Pedagógica y Tecnológica
de Colombia
Colombia

Flórez-Vergara, Diego Edissón; Castro-Riveros, Fabián Camilo; Castillo-Esteva, Ricardo
Andrés

Planeación y ejecución de trayectorias en el espacio de trabajo para un robot Delta

Facultad de Ingeniería, vol. 25, núm. 42, mayo-agosto, 2016, pp. 75-88

Universidad Pedagógica y Tecnológica de Colombia

Tunja, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=413945614007>

- 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 el espacio de trabajo para un robot Delta

Trajectory planning and execution in the workspace for a Delta robot

Planejamento e execução de trajetórias no espaço de trabalho para um robô Delta

Fecha de recepción: 11 de mayo de 2015
Fecha de aprobación: 20 de marzo de 2016

Diego Edissón Flórez-Vergara*
Fabián Camilo Castro-Riveros**
Ricardo Andrés Castillo-Esteba***

Resumen

En este artículo se expone una técnica de planeación y ejecución de trayectorias en el espacio cartesiano para ser aplicada en un robot delta. Con el fin de configurar y generar la trayectoria deseada, se diseña un software con interfaz gráfica (GUI), en el cual se implementa el algoritmo para calcular y visualizar los puntos intermedios de movimiento del robot en el espacio de trabajo. Para validar la trayectoria generada es utilizado un robot delta, diseñado y construido en el laboratorio de Robótica de la Universidad Militar Nueva Granada, el cual es controlado por medio de un circuito electrónico que permite la implementación adecuada de la técnica propuesta. De esta manera, es posible validar sobre el robot real las trayectorias calculadas por el algoritmo y visualizadas con el software de simulación.

Palabras clave: espacio de articulaciones, Jacobiano, robot Delta, singularidades.

Abstract

This paper describes a technique for trajectory planning and execution in Cartesian Space to be applied in a delta robot. In order to configure and generate the desired path, is designed a software with Graphical User Interface (GUI), in which, the proposed algorithm is implemented to calculate and display the robot movement through midpoints in the workspace.

* Universidad Militar Nueva Granada (Bogotá-Distrito Capital, Colombia). u1801272@unimilitar.edu.co.

** Universidad Militar Nueva Granada (Bogotá-Distrito Capital, Colombia).

*** Ph.D. Universidad Militar Nueva Granada (Bogotá-Distrito Capital, Colombia). ricardo.castillo@unimilitar.edu.co.

To validate the generated path is used a Delta robot designed and built in the Colombian Robotics Laboratory at Nueva Granada Military University. This robot is controlled by an electronic circuit that enables the proper implementation of the proposed technique. That makes it possible to validate the trajectories calculated by the algorithm and displayed in the simulation software, using a real robot.

Key Words: Delta robot, Jacobian, joint space, singularities.

Resumo

No artigo se expõe uma técnica de planejamento e execução de trajetórias no espaço cartesiano para ser aplicada em um robô Delta. Com a finalidade de configurar e gerar a trajetória desejada se desenha um software com interface gráfica (GUI), no qual se implementa o algoritmo para calcular e visualizar os pontos intermédios de movimento do robô no espaço de trabalho. Para validar a trajetória gerada é utilizado um robô Delta, desenhado e construído no laboratório de Robótica da Universidad Militar Nueva Granada, o qual é controlado por meio de um circuito eletrônico que permite a implementação adequada da técnica proposta. Desta maneira, é possível validar sobre o robô real as trajetórias calculadas pelo algoritmo e visualizadas com o software de simulação.

Palavras chave: espaço de articulações, Jacobiano, robô Delta, singularidades.

Cómo citar este artículo:

[1] D. E. Flórez-Vergara, F. C. Castro-Riveros and R. A. Castillo-Esteva, "Planeación y ejecución de trayectorias en el espacio de trabajo para un robot Delta", *Fac. Ing.*, vol. 25 (42), pp. 75–88, may.-ago. 2016.

I. INTRODUCCIÓN

Desde sus primeros modelos hasta los más influyentes en la actualidad, se han generado diversas definiciones sobre qué es un robot industrial; 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 el manipulador (sistema mecánico y accionadores) y el 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 con su fecha de fabricación
2. Por su grado de inteligencia
3. Por su funcionalidad
4. Por el tipo de control; la Asociación Francesa de Robótica Industrial (AFRI) clasifica los robots según el tipo de control utilizado en cuatro categorías:
 - 4.1. Tipo A o tele-manipulado, que tiene un control manual o telemando.
 - 4.2. Tipo B o secuencial: automático, es decir, que tiene ciclos preajustados, regulación mediante fines de carrera o topes, control por PLC y accionamientos, ya sean neumáticos, eléctricos o hidráulicos.
 - 4.3. 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.

- 4.4. 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.
 - 5.5. Robot industrial manipulador
 - 5.6. Robot industrial móvil

Un robot paralelo está compuesto por un efector final con n grados de libertad y una base fija, unidos entre sí por, al menos, dos cadenas cinemáticas independientes, el cual puede ser accionado por n actuadores simples [3].

El robot Delta (Fig. 1) es un manipulador paralelo que consta de tres cadenas cinemáticas cerradas de configuración PR(Ps), donde (Ps) representa el espacio paralelogramo de cuatro barras con cuatro articulaciones esféricas, y 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 de mayor tamaño a una base móvil de menor tamaño [4].

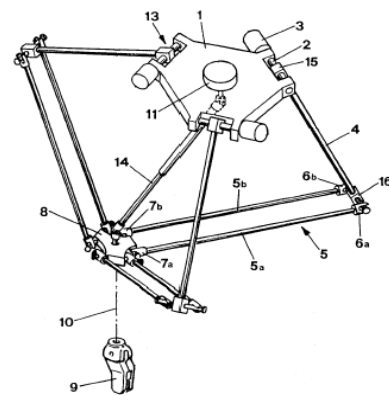


FIG. 1. Robot Delta de R. Clavel [5].

Algunas de las principales ventajas y desventajas del robot Delta, como también de muchas plataformas tipo paralelo se aprecian en la Tabla 1.

TABLA 1
VENTAJAS Y DESVENTAJAS DE ROBOTS
PARALELOS

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, tiene como principal objetivo aumentar la productividad y calidad de los trabajos realizados por el ser humano en las diversas tareas complejas o riesgosas, en las cuales se pueden desempeñar de mejor manera; dichas tareas requieren de movimientos coordinados y precisos del robot para poder llevarlas a cabo de forma eficiente; por lo cual es de suma importancia 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, se hace cada vez más necesario que los estudiantes afines a esta ciencia adquieran mayor conocimiento sobre los distintos robots y sus aplicaciones que rigen actualmente la industria mundial; entre los cuales hay un gran porcentaje de manipuladores tipo paralelo. Por lo tanto, el principal objetivo de este artículo 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 (Fig. 2), a través de una interfaz gráfica de usuario que pueda interactuar con el prototipo mecánico, permitiéndole simular y ejecutar trayectorias tanto en el espacio cartesiano como en el espacio de articulaciones, asemejando tareas reales implementadas en la industria actual.

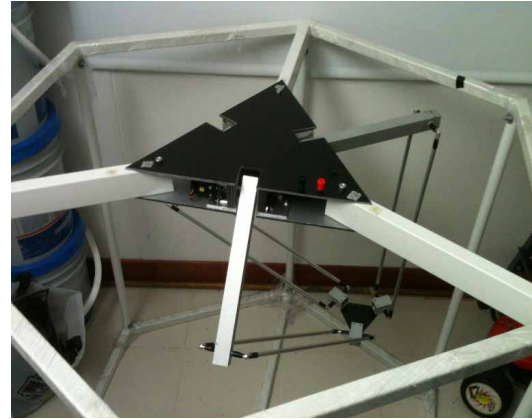


FIG. 2. Prototipo mecánico del Robot Delta, Laboratorio de robótica UMNG [6].

II. ROBOT DELTA

A. 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 el 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 [7].

El espacio de trabajo se puede subdividir en dos categorías así [8]:

- 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 efector final.

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

El espacio de trabajo del robot Delta con el cual se van a diseñar las diferentes trayectorias tiene como principales propiedades las expuestas en la Tabla 2 [6].

TABLA 2
ALCANCE MÁXIMO Y MÍNIMO DE LA
PLATAFORMA MÓVIL

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

B. Cinemática inversa

El principal objetivo de la cinemática inversa radica en encontrar los ángulos de las articulaciones, conociendo la configuración (posición y orientación) del efector final del robot [9].

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 cada articulación como se puede encontrar en [6].

C. Cinemática directa

El principal objetivo de la cinemática directa radica en encontrar la configuración (posición y orientación) en la que se encuentra el efector final del manipulador, ya que en este análisis las variables de articulación se encuentran ya definidas [9].

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 [6].

III. PLANEACIÓN TRAYECTORIAS

A. 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 (Fig. 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 [10, 11].

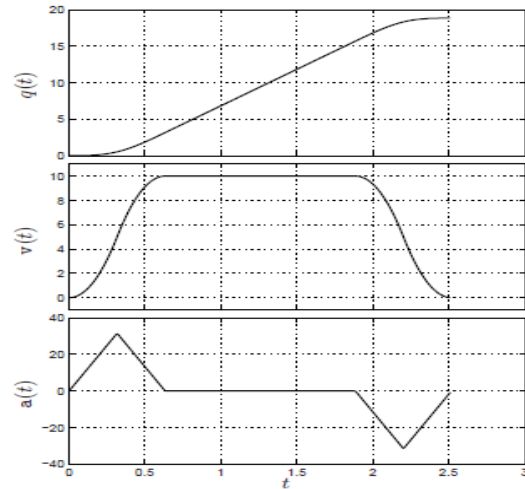


FIG. 3. Representación de formas típicas en el espacio cartesiano [12].

La planeación de trayectorias en el espacio cartesiano se desarrollará, 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, y permite la conversión de movimientos diferenciales o velocidades de las articulaciones individuales en movimientos diferenciales o velocidades en puntos de interés, como puede ser el efector final [7].

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 [13]:

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

La matriz jacobiana relaciona las velocidades de las articulaciones con la velocidad del sistema de referencia cartesiano, como se aprecia en (1):

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

Así, la matriz jacobiana del robot Delta es planteada de la forma (2) [6]:

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

J_x = Matriz asociada a la cinemática directa

J_θ = Matriz asociada a la cinemática inversa

Planteándose así el siguiente sistema de ecuaciones, que se puede ver representado en (3).

$$\begin{bmatrix} j_{x11} & j_{x12} & j_{x13} \\ j_{x21} & j_{x22} & j_{x23} \\ j_{x31} & j_{x32} & j_{x33} \end{bmatrix} \begin{bmatrix} V_{px} \\ V_{py} \\ V_{pz} \end{bmatrix} = \begin{bmatrix} j_{\theta 1} & 0 & 0 \\ 0 & j_{\theta 2} & 0 \\ 0 & 0 & j_{\theta 3} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{11} \\ \dot{\theta}_{12} \\ \dot{\theta}_{13} \end{bmatrix} \quad (3)$$

Que generará de forma definitiva la estructura para la matriz jacobiana del robot Delta, como se aprecia en (4) y (5).

$$J = (J_\theta)^{-1} J_x \quad (4)$$

$$J = \left(\begin{bmatrix} j_{\theta 1} & 0 & 0 \\ 0 & j_{\theta 2} & 0 \\ 0 & 0 & j_{\theta 3} \end{bmatrix} \right)^{-1} \begin{bmatrix} j_{x11} & j_{x12} & j_{x13} \\ j_{x21} & j_{x22} & j_{x23} \\ j_{x31} & j_{x32} & j_{x33} \end{bmatrix} \quad (5)$$

Dependiendo de qué matriz, de las planteadas anteriormente, sea singular, un mecanismo de lazo cerrado podría tener una configuración singular de cinemática directa, una de cinemática inversa o ambas. Así, la planeación de trayectorias se desarrollará por medio del algoritmo de cinemática inversa, el cual puede ser usado para obtener un modelo matemático que represente un movimiento diferencial, el cual se ve representado en (6).

$$q(k+1) = q(k) + (jq(k))^{-1} J_x * v(k) * \Delta t \quad (6)$$

$$\text{con } \Delta t = \frac{|t|}{\text{número Puntos}}$$

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

B. 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 sus ángulos de articulación (Fig. 4) [10], [14].

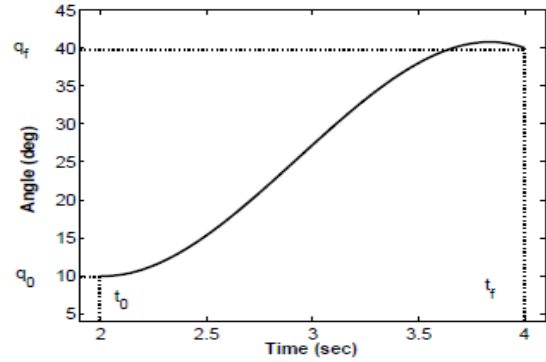


FIG. 4. Representación típica en el espacio de articulación [8].

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 [10].

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 , un valor final de la configuración.

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

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

$$\theta(t_f) = \theta_f$$

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

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

Estas restricciones satisfacen así un polinomio mínimo de grado 3 (polinomio cúbico), que será descrito para su posición, velocidad y aceleración en (7), (8) y (9), respectivamente.

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (7)$$

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (8)$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t \quad (9)$$

De manera que combinando las restricciones mencionadas con las representaciones matemáticas dadas en (7), (8) y (9) se pueden encontrar fácilmente las constantes a_0, a_1, a_2, a_3 , donde

$$a_0 = \theta_0$$

$$a_1 = 0$$

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

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

Una vez halladas todas las constantes del polinomio, se puede conectar cualquier ángulo de articulación desde una posición inicial a una posición final deseada; este caso es para una velocidad inicial y final cero, pero, si se desea, se puede aumentar el número de restricciones y aumentar el grado del polinomio, con el fin de obtener un 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; así, para mayor rendimiento computacional se prefiere llevar las ecuaciones a una forma matricial vectorial [15] observada en (10) y (11).

$$[\theta] = [M] * [C] \quad (10)$$

$$[C] = [M]^{-1}[\theta] \quad (11)$$

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

$$[M] = \text{Matriz coeficientes potencias del tiempo}$$

$$[b_n t^n \dots b_0 t^0]^t$$

$$[\theta] = \text{Matriz de restriccion, velocidad y aceleracion}$$

Obteniendo, finalmente, a (12)

$$\begin{bmatrix} \theta(to) \\ \theta(tf) \\ \dot{\theta}(to) \\ \dot{\theta}(tf) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & t & t^2 & t^3 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2t & 3t^2 \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (12)$$

IV. EJECUCIÓN TRAYECTORIAS

Esta sección está dedicada al análisis y ejecución de las trayectorias que se implementarán 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.

A. Algoritmo de ejecución

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

- A. Cálculo de la cinemática inversa en los puntos vías
- B. Cálculo de la velocidad entre puntos vía
- C. Cálculo de la matriz Jacobiana
- D. Implementación de (6)
- E. Envío de los ángulos entregados en el literal D al circuito electrónico
- F. Cinemática directa de los ángulos entregados en el literal D

1) Trayectoria 1: La primera trayectoria por implementar se diseñará para formar un triángulo equilátero; dicha trayectoria se planeará en el plano XY, manteniendo un valor constante en la coordenada Z, ejecutándose a través de cinco puntos en total, de los cuales, tres son puntos vías y dos son los puntos inicial y final de la trayectoria. Las coordenadas de los puntos se observan en la Tabla 3.

TABLA 3

PUNTOS QUE CONFORMAN LA TRAYECTORIA 1

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

El comportamiento simultáneo de las tres articulaciones del robot Delta en el seguimiento de los puntos que conforman la trayectoria, se puede apreciar en la Fig. 5, Fig. 6 y Fig. 7.

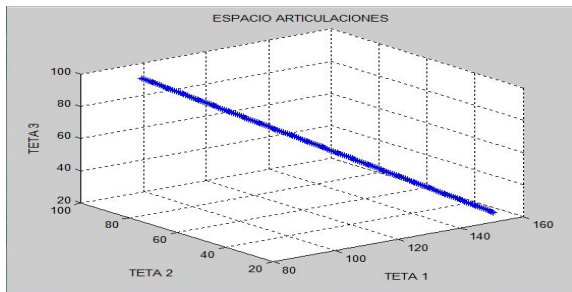


FIG. 5. Representación en el espacio de articulaciones del punto vía 1 con MATLAB.

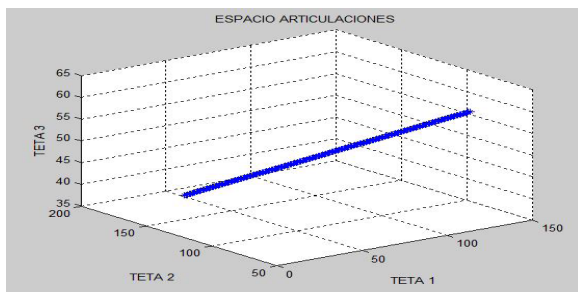


FIG. 6. Representación en el espacio de articulaciones del punto vía 2 con MATLAB.

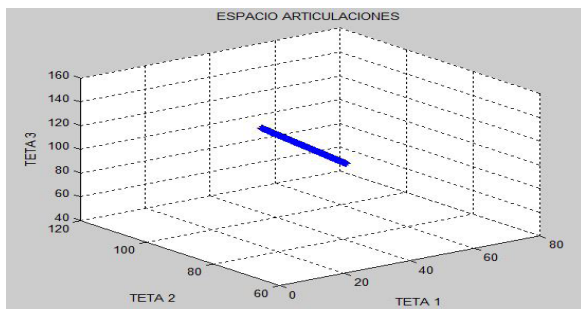


FIG. 7. Representación en el espacio de articulaciones del punto vía 3 con MATLAB.

De igual manera, el comportamiento del robot Delta en el seguimiento de los puntos que conforman la trayectoria en el espacio cartesiano se puede apreciar en la Fig. 8, Fig. 9 y Fig. 10.

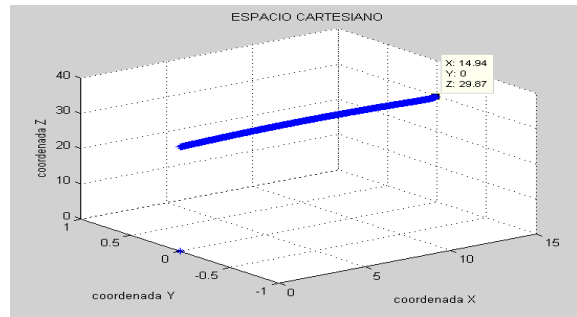


FIG. 8. Representación en el espacio cartesiano del punto vía 1 con MATLAB.

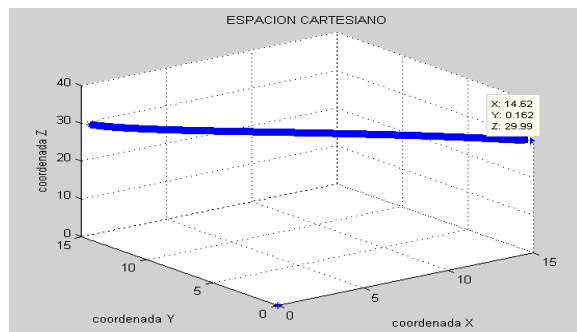


FIG. 9. Representación en el espacio cartesiano del punto vía 2 con MATLAB.

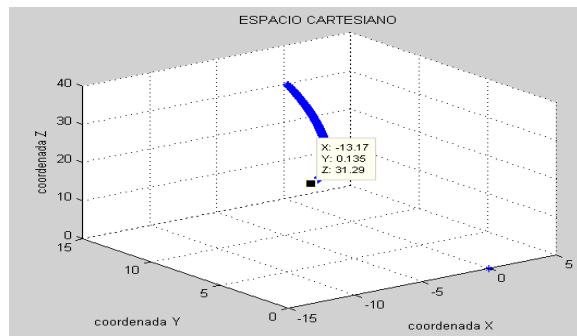


FIG. 10. Representación en el espacio cartesiano del punto vía 3 con MATLAB.

En la planeación de esta trayectoria se eligió una velocidad constante entre los puntos vía, la cual describe un comportamiento lineal entre estos, como se observó en las gráficas que describen la trayectoria en el espacio de articulaciones. Sin embargo, el seguimiento por el robot Delta de cualquier trayectoria que se planea y ejecute cuenta con un proceso iterativo

del algoritmo de cinemática inversa, el cual es de gran importancia, ya que, de acuerdo con el número de iteraciones, las trayectorias serán de:

- Interpolación: Con un número de iteraciones elevado y una velocidad de ejecución baja.
- Aproximación: Con un número de iteraciones bajo y una velocidad de ejecución alta.

El número de iteraciones o puntos vía permisibles para un adecuado costo computacional se determinó en el rango de 10-6000; por lo tanto, para un adecuado seguimiento por parte del robot Delta de la trayectoria 1 se implementaron 4800 iteraciones, que permiten apreciar la línea recta formada por los 3 actuadores a una velocidad y tiempo convenientes. De igual manera, se puede observar 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 aprecia, entonces, el comportamiento del robot Delta durante la ejecución de la trayectoria 1 tanto en el plano XY (Fig. 11) como en el espacio cartesiano en cada eje coordenado (Fig. 12).

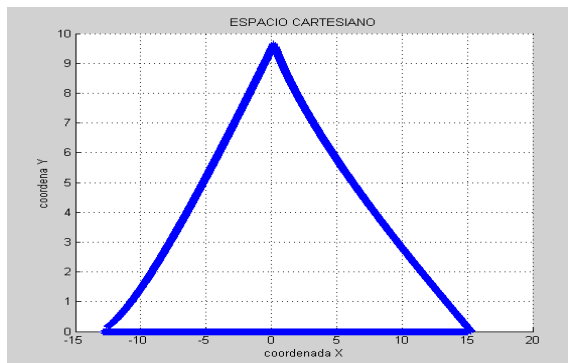


Fig. 11. Representación de trayectoria en plano XY.

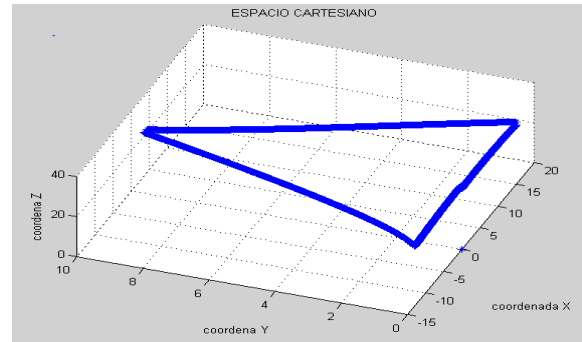


Fig. 12. Representación de trayectoria en el espacio XYZ.

Al realizar un análisis de la trayectoria variando los puntos vía (Tabla 4), la velocidad (Tabla 5) y el tiempo (Tabla 6) de ejecución, se puede observar el error que genera la variación de los distintos parámetros, así como la incidencia que tiene cada parámetro en el algoritmo de cinemática inversa.

TABLA 4

VARIACIÓN DE PUNTOS VÍA A VELOCIDAD Y
TIEMPO CONSTANTE

Número puntos	Coordenada inicial	Coordenada final	Error
6000	(15,0,30)	(0,0,30)	0
1000	(15,0,30)	(0,0,30)	0.4
100	(15,0,30)	(15,0,30)	0.4
10	(15,0,30)	(15,0,30)	1.45

TABLA 5

VARIACIÓN DE VELOCIDAD CON PUNTOS VÍA Y
TIEMPO CONSTANTE

Velocidad	Coordenada inicial	Coordenada final	Error
20	(15,0,30)	(0,0,30)	0
10	(15,0,30)	(0,0,30)	4.66
5	(15,0,30)	(15,0,30)	9.38
3	(15,0,30)	(15,0,30)	11.57

TABLA 6
VARIACIÓN DE TIEMPO CON PUNTOS VÍA Y
VELOCIDAD CONSTANTE

Tiempo	Coordenada inicial	Coordenada final	Error
100	(15,0,30)	(0,0,30)	0
50	(15,0,30)	(0,0,30)	3.4
10	(15,0,30)	(15,0,30)	13.2
3	(15,0,30)	(15,0,30)	14.7

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

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

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

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

Punto vía 2

$$59 + 0.2118t^1 + 0.1729t^2 - 0.0034t^3$$

$$59 + 0.2118t^1 + 0.1729t^2 - 0.0034t^3$$

$$59 + 0.2118t^1 + 0.1729t^2 - 0.0034t^3$$

Punto vía 3

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

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

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

Dichos polinomios describen el comportamiento de cada uno de los actuadores, servomotor1 (Fig.13-15), servomotor2 (Fig.16-18) y servomotor3 (Fig.19-21), donde se especifican las restricciones de posicionamiento que se deben cumplir en los diferentes puntos vía que componen la trayectoria, con el fin de obtener un movimiento continuo y preciso del efector final.

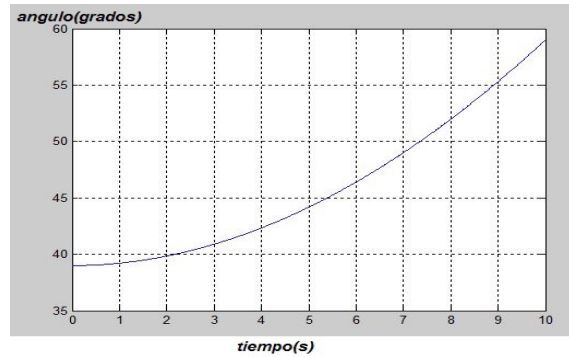


FIG. 13. Comportamiento en el punto vía 1 del actuador1 con MATLAB.

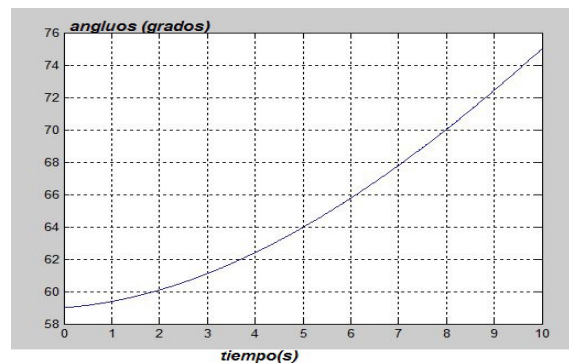


FIG. 14. Comportamiento en el punto vía 2 del actuador1 con MATLAB.

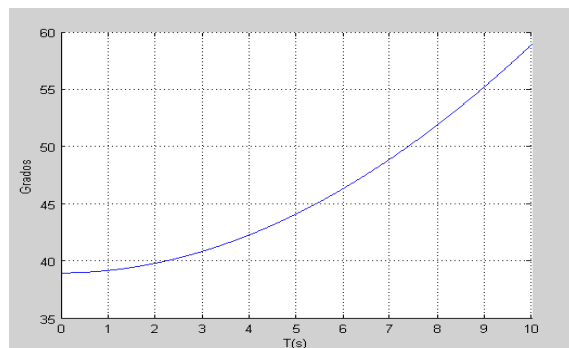


FIG. 15. Comportamiento en el punto vía 3 del actuador1 con MATLAB.

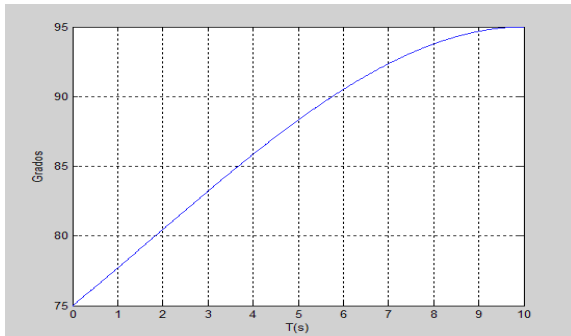


FIG. 16. Comportamiento en el punto vía 1 del actuador2 con MATLAB.

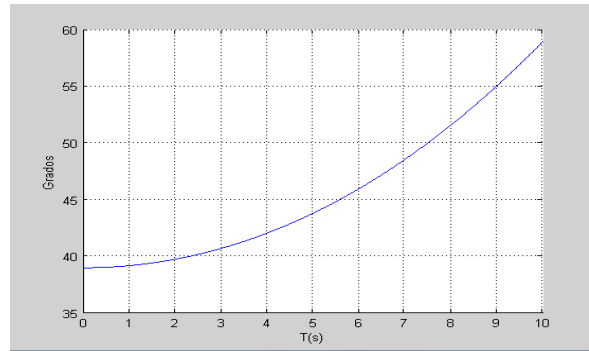


FIG. 19. Comportamiento en el punto vía 1 del actuador3 con MATLAB.

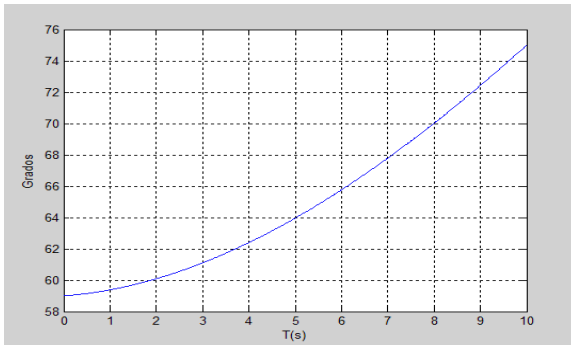


FIG. 17. Comportamiento en el punto vía 2 del actuador2 con MATLAB.

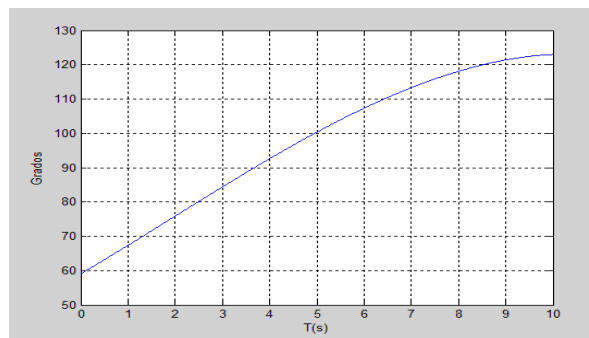


FIG. 20. Comportamiento en el punto vía 2 del actuador3 con MATLAB.

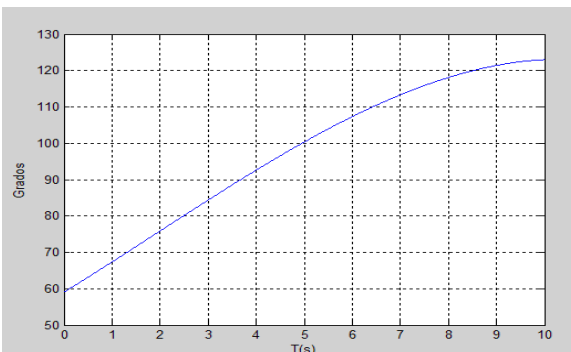


FIG. 18. Comportamiento en el punto vía 3 del actuador2 con MATLAB.

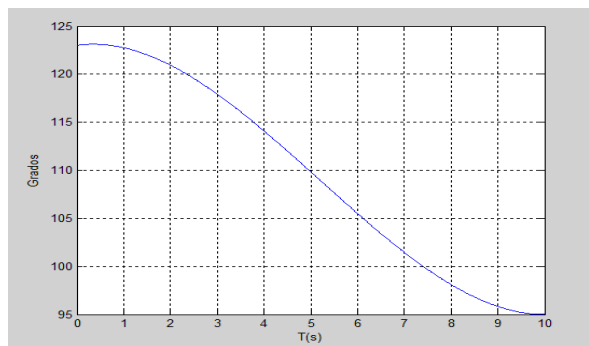


FIG. 21. Comportamiento en el punto vía 3 del actuador3 con MATLAB.

2. Trayectoria 2: La segunda trayectoria por implementar se diseñará para formar una trayectoria curva, más específicamente, una semicircunferencia en el espacio XYZ, manteniendo un valor constante en la coordenada Z, con el objetivo de unir dos puntos coordenados especificados por el usuario, que se pueden evidenciar en la Tabla 7.

TABLA 7

PUNTOS QUE CONFORMAN LA TRAYECTORIA 2

Puntos	Coordenada X	Coordenada Y	Coordenada Z
Inicial	20	0	30
Final	-10	0	30

En esta trayectoria el componente de velocidad del algoritmo de cinemática inversa se especifica en función del seno, el coseno y el radio de una circunferencia; de igual manera, debido al aumento de complejidad en la forma del algoritmo y, por ende, de la trayectoria, se prefiere llevar a cabo por un método más aproximado que el implementado en la trayectoria1, lo cual requiere de un número menor en el proceso iterativo del algoritmo de cinemática inversa y, en consecuencia, un aumento considerable en la velocidad de ejecución y en el error de posición de la trayectoria por parte del robot Delta.

El comportamiento del robot Delta en el espacio cartesiano, en su seguimiento de los puntos vía que describen la trayectoria2, se puede visualizar en las Fig. 22, con un radio de circunferencia de 2.5; Fig. 23, con un radio de circunferencia de 2, y Fig. 24, con un radio de circunferencia de 0.5.

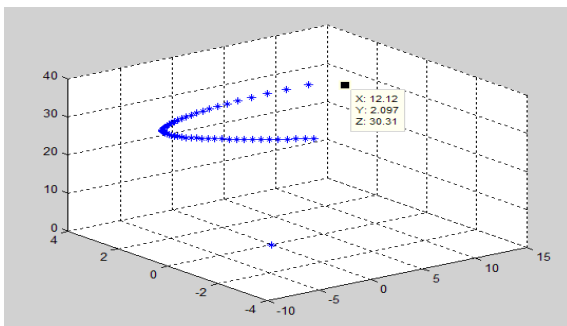


FIG. 22. Representación de la trayectoria2 con 100 iteraciones en MATLAB.

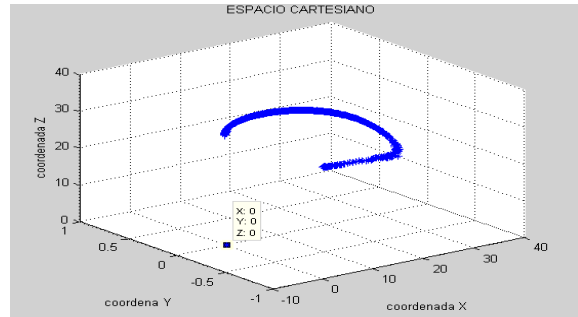


FIG. 23. Representación de la trayectoria2 con 1000 iteraciones en MATLAB.

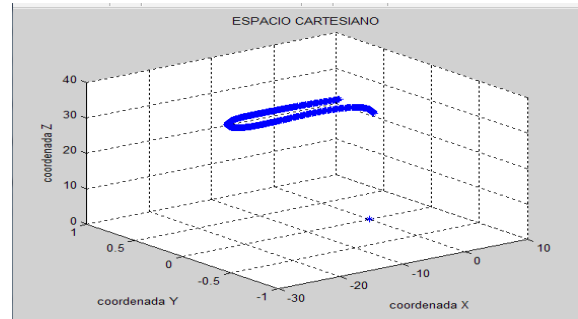


FIG. 24. Representación de la trayectoria2 con 1000 iteraciones en MATLAB.

B. Interfaz gráfica de usuario (GUI)

El entorno virtual (Fig. 27) con el cual el usuario interactúa para simular y ejecutar una trayectoria incorpora una simulación en un entorno 3D para sistemas mecánicos, conocido como SimMechanics™ [16], dentro del cual se corrobora el posicionamiento de los eslabones en la planeación de los puntos vía que conforman la trayectoria que se pretende ejecutar; esto con el fin de visualizar y comprender el comportamiento del robot Delta durante el seguimiento de una trayectoria, previniendo que este no entre en una configuración singular, que podría llegar a dañar seriamente el prototipo mecánico real una vez se le ordene empezar la ejecución de la trayectoria. En la Fig. 25 y Fig. 26 se observan, respectivamente, las simulaciones para el punto inicial y el punto vía 1 que componen la trayectoria2.

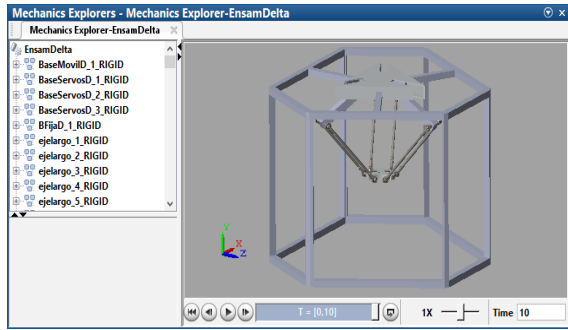


FIG. 25. Simulación del punto inicial de la trayectoria2 en SimMechanics™.

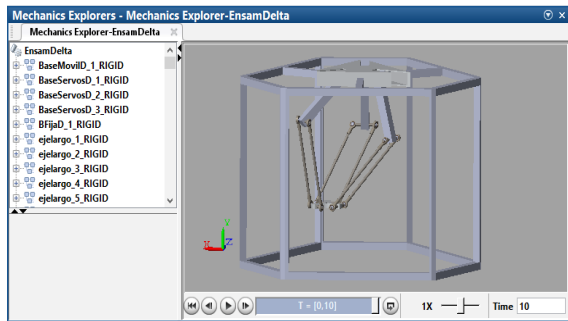


FIG. 26. Simulación del punto vía 1 de la trayectoria2 en SimMechanics™.



FIG. 27. Interfaz gráfica de usuario (GUI) en MATLAB.

C. Circuito electrónico

El diseño y construcción de un nuevo circuito electrónico pretende mejorar algunas propiedades del diseño electrónico original (Fig. 28), que permitan un comportamiento más real e idóneo, en aspectos como:

- Disminución del tamaño
- Portabilidad
- Facilidad de uso

- Avance en dispositivos electrónicos



FIG. 28. Circuito electrónico original, robot Delta [6].

La implementación de un ARDUINO® Leonardo [17] en lugar del PIC 18f4550 tiene como principal ventaja brindar la capacidad de programar el robot online, es decir, sin la necesidad de estar retirando el microcontrolador del circuito para reprogramarlo cada vez que se requiera efectuar un cambio en el código. Por otra parte, se utilizará la salida de alimentación de 3.3V del ARDUINO® Leonardo para alimentar directamente el dispositivo de comunicación inalámbrica Xbee, reduciendo el número de componentes electrónicos por manipular y, por consiguiente, el tamaño del PCB. Finalmente, se utilizará directamente el Jack de alimentación AC del ARDUINO® Leonardo, lo cual permitirá mayor portabilidad del robot Delta, al no depender exclusivamente de una fuente de alimentación del laboratorio para su adecuado funcionamiento; el circuito mejorado se ve en la Fig. 29.



FIG. 29. Circuito electrónico final, robot Delta.

4. CONCLUSIONES

Al considerarse el criterio de velocidad y tiempo constante, en la implementación del algoritmo de cinemática inversa, se logró la obtención de movimientos con mayor precisión y con un error de posición reducido, los cuales permitieron el seguimiento de las trayectorias por un proceso de interpolación y no de aproximación.

La implementación de una trayectoria por medio del algoritmo de cinemática inversa requiere de un alto costo computacional, al tratarse de un algoritmo iterativo, teniendo como factores importantes el consumo de tiempo y de espacio adicional en memoria.

Debido al diseño del prototipo mecánico del robot Delta en la Universidad Militar Nueva Granada, los eslabones tienen pequeños problemas de respuesta al variar su rango de velocidad, por lo cual se prefirió la ejecución de las trayectorias a velocidades bajas, reduciendo las vibraciones del mecanismo durante la ejecución de las trayectorias.

AGRADECIMIENTOS

A la Universidad Militar, por el apoyo y financiamiento por medio del proyecto de investigación ING-1764.

REFERENCIAS

- [1] A. Barrientos, L. F. Peñín, C. Balaguer and R. Aracil, *Fundamentos de robótica*, 2nd ed. Madrid, España: Mc Graw Hill, 2007.
- [2] M. Mellado-Arteche. *Robótica*. Valencia, España: Universidad Politécnica de Valencia, 2011.
- [3] J. P. Merlet, *Parallel Robots*, 2nd ed. Dordrecht, Netherlands: Springer, 2006.
- [4] X-J. Liu and J. Wang, *Parallel Kinematics: Type, Kinematics, and Optimal Design*. Heidelberg, Germany: Springer Berlin Heidelberg, 2014. DOI: <http://dx.doi.org/10.1007/978-3-642-36929-2>.
- [5] R. Clavel, "Device for the movement and positioning of an element in space," *U.S. Patent 4976582*, Dec. 1990.
- [6] L. Urrea and S. Medina, "Diseño e implementación de una plataforma robótica tipo Delta". Trabajo de Grado, Ingeniería en Mecatrónica, Facultad de Ingeniería, Universidad Militar Nueva Granada, Bogotá D.C., Colombia, Sep. 2012.
- [7] S. B. Niku, *Introduction to Robotics: Analysis, Control, Applications*, 2nd ed. Hoboken, USA: John Wiley & Sons Inc., 2010.
- [8] M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control*. USA: John Wiley & Sons, 2005.
- [9] S. K. Saha, *Introducción a la robótica*. Noida, India: McGraw-Hill, 2010.
- [10] J. J. Craig, *Robótica*, 3rd ed. Atlacomulco, México: Pearson Educación, 2006.
- [11] S. Y. Nof, *Handbook of Industrial Robotics*, 2nd ed. USA: John Wiley & Sons, 1999. DOI: <http://dx.doi.org/10.1002/9780470172506>.
- [12] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Heidelberg, Germany: Springer, 2008.
- [13] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, *Robotics: Modeling, Planning and Control*. London, United Kingdom: Springer, 2009. DOI: <http://dx.doi.org/10.1007/978-1-84628-642-1>.
- [14] R. N. Jazar, *Theory of Applied Robotics: Kinematics, Dynamics and Control*, 2nd ed. Boston, USA: Springer US, 2010. DOI: <http://dx.doi.org/10.1007/978-1-4419-1750-8>.
- [15] P. Corke, *Robotics, Vision and Control: Fundamental algorithms in MATLAB®*. Heidelberg, Germany: Springer, 2011. DOI: <http://dx.doi.org/10.1007/978-3-642-20144-8>.
- [16] MathWorks SimMechanics™. [Online]. Available: <http://www.mathworks.com/products/simmechanics/>.
- [17] ARDUINO® Leonardo. [Online] Available: <https://www.arduino.cc/en/Main/arduinoBoardLeonardo>.