

## PROTOTIPO DE ROBOT DELTA CON LABVIEW UTILIZANDO METODOS GEOMETRICOS

*Ignacio Javier Vázquez Cuevas<sup>1</sup>, José Alejandro López Corella<sup>2</sup>, Sergio Iván Hernández Ruiz<sup>3</sup>, Maribel Herrera Velarde<sup>4</sup>, Martín Ochoa Alegria<sup>5</sup>, Corrales Verduzco Haydee Guadalupe<sup>6</sup>*

Tecnológico Nacional de México / I.T. de Nogales, Metal Mecánica Ingeniería Mecatrónica, Eléctrica y Electrónica Ingeniería en Electrónica, Av. Tecnológico 911, Nogales, Sonora, México  
+52(631) 311 1881. Ext 1129.  
[jose.lc@nogales.tecnm.mx](mailto:jose.lc@nogales.tecnm.mx)

### RESUMEN.

En el presente trabajo se demuestra el desarrollo de un prototipo de Robot Delta utilizando LabVIEW, aplicando métodos geométricos para la solución de la Cinemática Inversa del Robot, primeramente se utilizó el modelo matemático para crear una simulación del robot en forma gráfica utilizando vectores en LabVIEW, en la simulación se crearon trayectorias de los movimientos con una interpolación lineal, posteriormente se desarrolló el prototipo en impresión 3D y la circuitería necesaria para que el prototipo siga las trayectorias de la simulación deseada de esta manera se pudo comparar el funcionamiento correcto del simulador en LabVIEW con el prototipo de robot delta desarrollado.

**Palabras Clave:** Robot Delta, LabVIEW, Impresión 3D.

### ABSTRACT.

In this work, the development of a prototype of the Delta Robot using LabVIEW is demonstrated, applying geometric methods for the solution of the Inverse Kinematics of the Robot. First, the mathematical model was used to create a simulation of the robot in graphic form using vectors in LabVIEW, In the simulation, the trajectories of the movements were created with a linear interpolation, later the prototype was developed in 3D printing and the circuitry necessary for the prototype to follow the trajectories of the desired simulation. In this way, the correct operation of the simulator could be compared in LabView. with the prototype delta robot developed

**Keywords:** Robot Delta, LabView, 3D printing.

## 1. INTRODUCCIÓN

Se han desarrollado algunos simuladores de Robots Delta con diferentes métodos matemáticos para la solución de la cinemática inversa, uno de los métodos para la solución de este modelado matemático es el uso de Redes Neuronales [1], otros de los métodos utilizados son números complejos en la cual se utiliza la rotación usual y variante de los complejos para modelar robots [2], también existe el Álgebra de Cuaterniones la cual es una herramienta matemática utilizada para la solución de la cinemática directa [4]. Los Cuaterniones se crean Vectores Dinámicos las cuales se definen con las mismas características de movimiento que los eslabones que conforman al robot; por lo que representan una herramienta muy útil para

obtener las ecuaciones cinemáticas de cadenas cerradas o abiertas [5].

Este trabajo se centra con el desarrollo de la Cinemática Inversa utilizando métodos geométricos como se demuestra en el artículo [6] que desarrollaron un robot paralelo de tres grados de libertad denominado manipulador delta, también en el artículo [7] demostraron el modelo matemático con métodos geométricos validados en Matlab utilizando SolidWorks para su implementación mecánica. El presente artículo toma como base el algoritmo geométrico descrito en [8], en la cual se demuestra y valida creando un simulador en Matlab, con el fin de migrarlo a LabVIEW para futuras aplicaciones en un prototipo.

Hoy en día los Robots Delta han tenido una gran aplicación, ya que pueden manipular objetos pesados, con una velocidad de operación alta, además tienen una mejor precisión y repetitividad, son muy utilizados en procesos en la etapa de empaquetamientos y en industrias alimenticias [8]. Los robots manipuladores se pueden clasificar de acuerdo a su estructura: serie, paralelo e híbridos.

Los Robots Delta también son conocidos como robots de cadena cerrada o paralela, estos están compuestos por dos plataformas paralelas, una base móvil y de menor dimensión que otra fija, tal como se observa en la figura 1. Además, están unidas por medio de varios brazos paralelos, cada brazo está controlado por un servomotor, estos generan un movimiento.

El robot Delta es considerado como uno de los robots paralelos que más éxito han tenido en cuanto a su diseño. Varios centenares de este tipo de robot se encuentran trabajando en la industria actualmente [7].

La idea básica detrás del diseño del robot paralelo Delta es el uso de paralelogramos. Un paralelogramo ofrece a un eslabón rendimiento para permanecer en una orientación fija con respecto a un eslabón de entrada. El uso de tales paralelogramos ubica la orientación de la plataforma móvil que sólo permanece con los tres grados de libertad de translación. Las juntas de revoluta de las palancas de rotación están actuando de dos maneras diferentes: con motores rotatorios o con actuadores lineales. Finalmente, un cuarto dispositivo se

usa para transmitir movimiento rotatorio de la base a un extremo final, montado en la plataforma móvil. [7]

## 2. DESARROLLO

### 2.1. Cinemática Inversa.

En la figura 1 muestra el esquema simplificado de un robot tipo Delta que consiste de dos plataformas; la fija, en donde se ubican los actuadores que para este tipo de robot son motores rotativos, colocados a la misma distancia del centro O, en los puntos 1,2 y 3 y la plataforma móvil que porta el efector del robot, ubicado en forma adecuada en el centro del triángulo. Las dos plataformas están unidas entre sí, por tres estructuras de barras idénticas conformadas por un brazo en cada uno de los motores y por antebrazo que está unido al brazo y a la plataforma móvil, por medio de juntas esféricas [9]

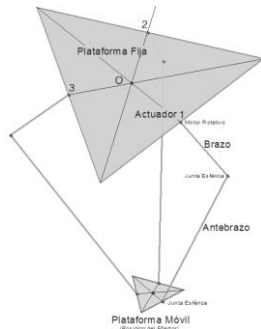


Figura 1.- Esquema Robot Delta.

Debido a la restricción de la junta A, el brazo describe una circunferencia de radio  $L1$ , mientras que con respecto del punto B, el punto antebrazo puede describir una esfera de Radio  $L2$ . La intersección de la circunferencia y la esfera se produce en dos puntos, se toma como solución el punto con menor valor en la coordenada y. Al determinar la posición del punto C, se puede obtener el ángulo  $\theta_1$  del actuador, como se muestra en la figura 2.

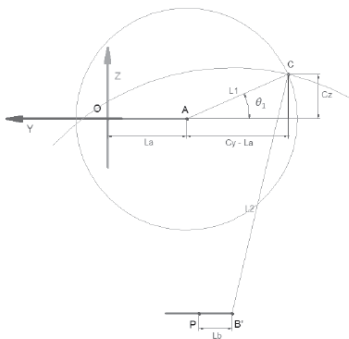


Figura 2.- Cadena cinemática de un brazo del Robot Delta.

Coordenadas del Punto P, B, A y B'

$$P(x_0, y_0, z_0); B(x_0, y_0 - L_b, z_0); A(0, -L_a, 0); B'(0, y_0 - L_b, z_0)$$

Con las coordenadas de los puntos descritos, se plantea un sistema de dos ecuaciones no lineales que permita encontrar la posición del punto C, con la cual se puede calcular el ángulo que forma el brazo con el plano horizontal, y se obtiene así, la solución esperada como se puede observar en las siguientes ecuaciones

$$(C_y - L_a)^2 + (C_z - A_z)^2 = L_1^2 \dots \dots \text{Ec1}$$

$$(C_y - B'_y)^2 + (C_z - B'_z)^2 = L_2^2 - x_0^2 \dots \dots \text{Ec2}$$

$$\theta_1 = \arctan\left(\frac{C_{z1}}{C_{y1} - L_a}\right) \dots \dots \text{Ec3}$$

Para los otros brazos, se usa la matriz de rotación con un ángulo de  $120^\circ$  para el brazo 2 y  $240^\circ$  para el 3. Esta matriz de rotación permite girar el sistema de coordenadas, de manera que se pueda usar la solución descrita para el cálculo de restantes ángulos, esta solución se validó con una aplicación en Matlab visto en las investigaciones anteriores [8]

### 2.2. Simulador en Labview.

Se desarrollaron las SubVI, para crear el simulador, en la figura 3 se puede observar la SubVI Rz la cual es la matriz de rotación en z, esta función recibe como parámetro el punto y el ángulo para rotarlo, como se explica anteriormente esta función es utilizada para rotar los puntos del segundo y tercer brazo del Robot Delta.

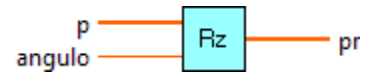


Figura 3.- matriz de rotación en Z.

En la figura 4 se muestra la matemática desarrollada en LabVIEW utilizando la estructura de Formula Node.

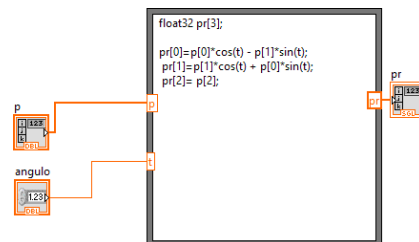


Figura 4.- Matemática de Rz.

También se creó la *SubVI Cadena Delta*, esta función recibe como parámetros los tamaños de los eslabones, así como el punto donde se desea que articule el robot resolviendo la cinemática inversa, se utilizó la solución del sistema de ecuaciones resuelto en Matlab [8]; Esta función da como resultado los puntos de articulación de la cadena cinemática, así como el ángulo del actuador, la función se puede observar en la figura 5.

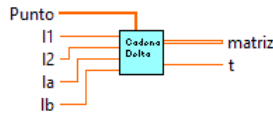


Figura 5.- SubVI Cadena delta.

En la figura 6 se observa la *SubVI Delta* utiliza internamente las funciones anteriormente mencionadas, esta función recibe como parámetros la posición que deseamos que articule el robot resolviendo la cinemática inversa, así como un arreglo donde se almacena los tamaños de los eslabones, esta función da como resultado las articulaciones de los tres actuadores y la gráfica del robot como se puede observar en la figura 7.

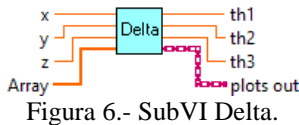


Figura 6.- SubVI Delta.

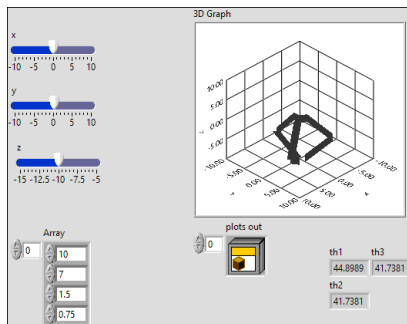


Figura 7.- Gráfica del Robot Delta.

En la figura 8 muestra el programa siguiendo el algoritmo geométrico, en la cual resuelve las tres cadenas cinemáticas con sus respectivas rotaciones para graficar el Robot.

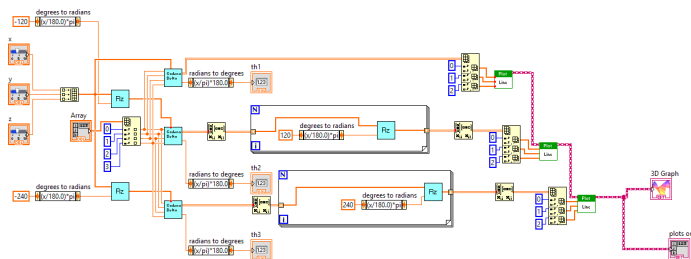


Figura 8.- Programa del Robot Delta.  
**2.3 Interpolación de la trayectoria**

Para realizar la animación del movimiento con las trayectorias se capturan las coordenadas en x,y,z para el movimiento entre los puntos, en la figura 9 se muestra el programa en donde se agregan estas coordenadas en una matriz de LabView

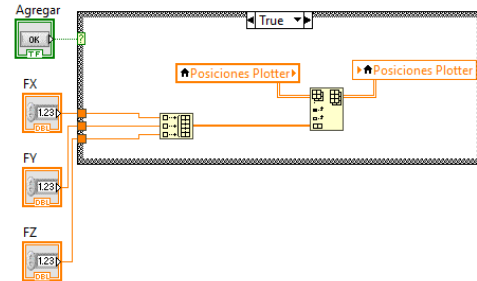


Figura 9.- Programa de captura de coordenadas

Para poder hacer la trayectoria grafica del movimiento, se utilizó un arreglo, se tomaron los controles numéricos que mandan las posiciones para la cinemática inversa en X, Y y Z y se fueron agregando con un botón al arreglo de posiciones "posiciones plotter" como se ve en la figura 10.

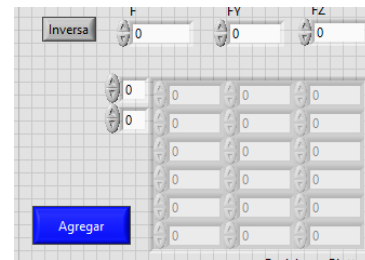


Figura 10.- Interfaz del usuario para la captura de puntos.

El arreglo generado puede contener n puntos, para la secuencia de la trayectoria y así mismo se creó una interpolación lineal para el movimiento entre los puntos, como se puede observar en la ecuación 4

$$\Delta \vec{I} = \frac{\vec{P}_f - \vec{P}_i}{n \text{ pasos}}$$

Donde:

$\Delta \vec{I}$  = Incremento para la interpolación

$P_i$  = Punto inicial de la trayectoria

$P_f$  = punto final de la trayectoria

$n \text{ pasos}$  = pasos de la interpolación lineal

Para el movimiento siguiendo la interpolación se realiza la suma vectorial del punto inicial con el incremento de la interpolación, y se crea el movimiento al nuevo punto.

$$\vec{P}_i = \vec{P}_i + \Delta\vec{I}$$

En la figura 11 se puede observar el programa que realiza la trayectoria lineal interpolando entre los puntos capturados para el movimiento, como se puede observar se tiene dos ciclos FOR anidados, el primero es para recorrer la matriz de los puntos del movimiento, el segundo FOR crea la animación con la interpolación lineal con divisiones de 5 pasos

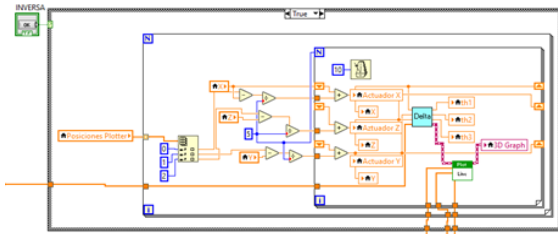


Figura 11.- Interfaz del usuario para la captura de puntos

## 2.4 Circuito eléctrico y Comunicación

Para controlar el robot se utilizó un Arduino NANO, por su tamaño pequeño a comparación del Arduino UNO fue más sencillo montarlo en el robot junto con sus demás componentes. Para la alimentación se utilizó una batería Li-po de 11.1 V para que sea controlado inalámbricamente, lo que lo hace muy portable y fácil de posicionar en cualquier lugar, además de que tiene un interruptor, como se puede observar en la figura 12 del diagrama esquemático del circuito eléctrico

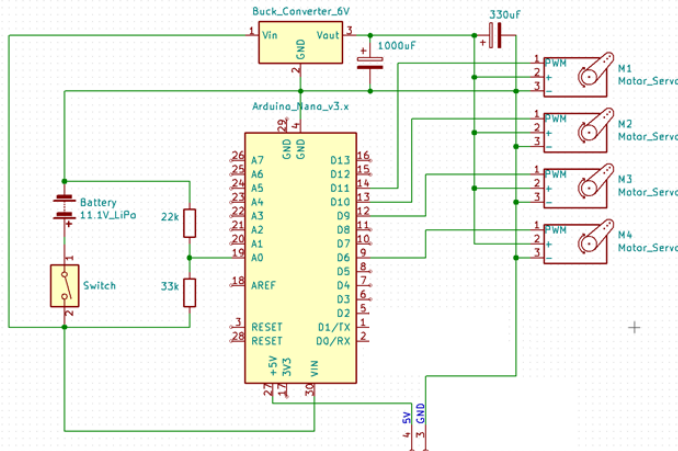


Figura 12.- Esquemático del circuito en KiCad

Se utilizaron 4 servo motores, 3 HJ S3315D para posicionar los brazos y un SG90 modificado para rotación continua para rotar al efector final.

Los servos HJ S3315D utilizan 6v y una cantidad significativa de corriente, así que utilizamos un regulador de voltaje para bajar el voltaje de la batería. El que utilizamos puede recibir 12.6v de entrada con una salida de 5A a 6v.

Los servos pueden crear bastante ruido eléctrico y fluctuaciones, así que se utilizaron capacitores para ayudar a reducirlo

Para comunicar LabVIEW con el IDE de Arduino, se utilizó NI-Visa, que se encarga de mandar la información de los actuadores de LabVIEW al monitor serial del Arduino, que controla al robot, como se puede observar en la figura 13.

- Para controlar al motor 1, o eslabón 1, se manda una C a Arduino desde th1, utilizando el actuador X para controlarlo.
- Para controlar al motor 2, o eslabón 2, se manda una A a Arduino desde th2, utilizando el actuador Y para controlarlo.
- Para controlar al motor 3, o eslabón 3, se manda una B a Arduino desde th3, utilizando el actuador Z para controlarlo.

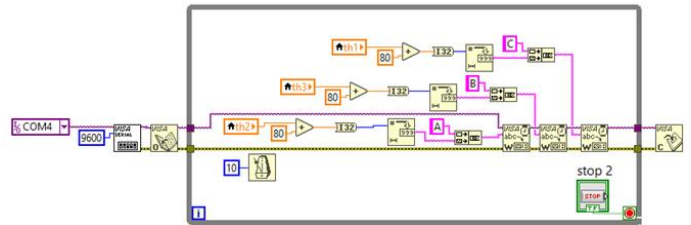


Figura 13.- Comunicación de LabVIEW con Arduino

Todo esto cambiando el valor de la coordenada en la cual se encuentra el efector final, lo que LabVIEW manda verdaderamente a Arduino son los valores de las articulaciones que tendrá cada motor en la posición que se desea tener el efector final.

Se programó el Arduino siguiendo las reglas establecidas para la comunicación desde LabVIEW usando el puerto serial, como se puede observar en la figura 14 del programa del Arduino.

```

ControlDelta
1 #include<Servo.h>
2 char lecl; //Lectura
3 int pos; //isusuib del servomotor
4 int serv04 = 6;
5
6 Servo serv01; //Declaramos los servos que se van a utilizar
7 Servo serv02;
8 Servo serv03;
9
10 void setup() {
11 // put your setup code here, to run once:
12 Serial.begin(9600);
13 serv01.attach(11); //Se declara en que PIN
14 serv02.attach(10);
15 serv03.attach(9);
16 pinMode(serv04, OUTPUT);
17 }
18
19 void loop() {
20 // put your main code here, to run repeatedly:
21 if (Serial.available() > 0)
22 {
23 lecl = Serial.read();
24 pos = Serial.parseInt(); //te convierte la informacion de ASCII a Numerico hasta que llegue a salto de carga
25 if (lecl == 'A')
26 serv01.write(pos);
27 if (lecl == 'B')
28 serv02.write(pos);
29 if (lecl == 'C')
    
```

```

30  servo3.write(pos);
31  if (lecl == 'D')
32    digitalWrite(servo4, HIGH);
33  else
34    digitalWrite(servo4, LOW);
35  }
36  }
    
```

Figura 14.- Programa Arduino

### 2.5 Impresión del robot en 3D

En la etapa de investigación y diseño del robot delta en Solid Works se encontró que ya existían diseños existentes por lo que se tomó la decisión de usar uno ya existente y acondicionarlo a nuestras necesidades [10].

Se tomaron los sólidos y se acondicionaron a el tamaño de los servo motores a utilizar y para comenzar a imprimir los archivos del robot, se hicieron los slices utilizando un infill de "honeycomb" de 20-25% en PLA para asegurar que sostuviera bien los motores y el efector final. A continuación, se muestran imágenes de los ensamblajes de las piezas que se imprimieron figura 15, 16, 17. Las cuales tomo un total de 27 horas en imprimirse.



Figura 15- Ensamble completo en SolidWorks

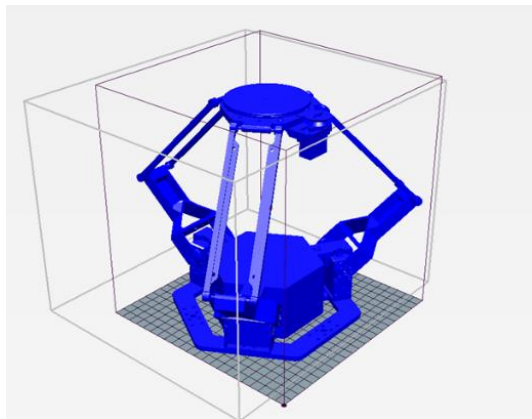


Figura 16.- Ensamble en Repetier.

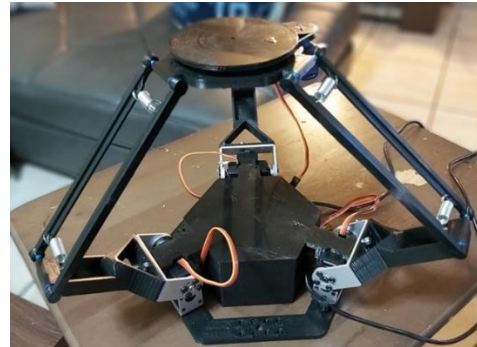


Figura 17.- Robot ensamblado.

### 3. RESULTADOS

Se realizó un análisis del prototipo del Robot Delta, dando distintos puntos para observar la trayectoria del Robot, en la cual se puede determinar la eficiencia del simulador y del prototipo. En la figura 18 y 19 muestra la interfaz del usuario del simulador utilizando los controladores e indicadores de LabVIEW, el indicador más importante del simulador es la gráfica en vectores, este simulador tiene como indicadores la posición actual del Robot y tiene como controladores las barras para mover el robot en xyz así como un botón para agregar las pntos de los movimiento que seguirá la trayectoria, así mismo se pudo observar que el movimiento del simulador corresponde al movimiento del prototipo

En la figura 17 se puede observar la foto del prototipo creado en Impresión 3D

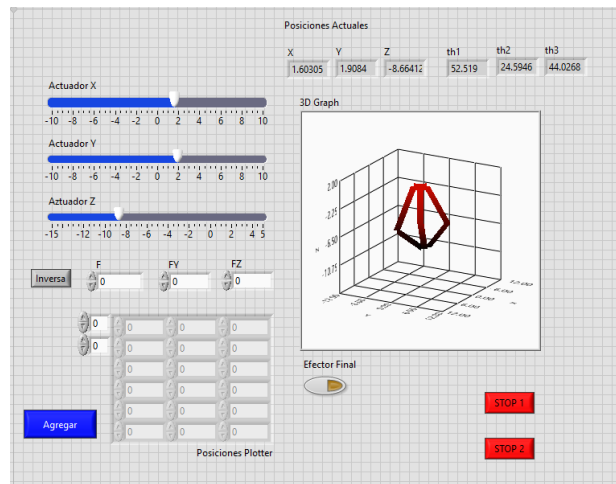


Figura 18.- Interfaz del simulador en LabVIEW.



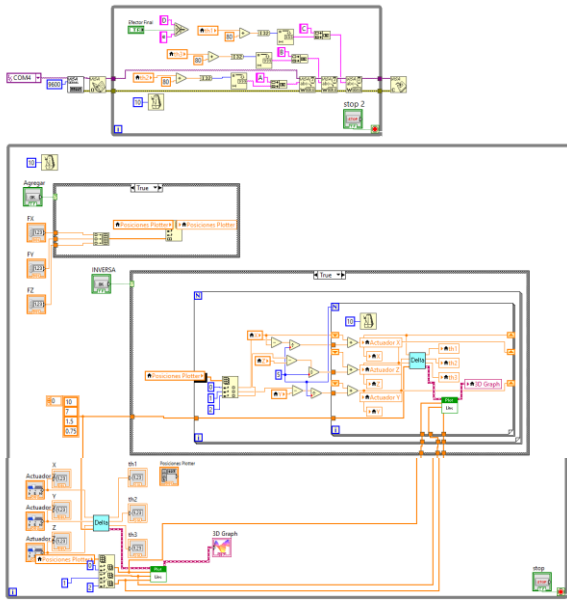


Figura 19.- Programa principal en LabVIEW.

#### 4. CONCLUSIONES

Para el desarrollo de simuladores y prototipos de robots se requiere de la aplicación de modelos matemáticos, así como de la aplicación de Softwares para el control del prototipo físico, en este caso se utilizó LabVIEW para el desarrollo de la simulación, este Software es muy versátil ya que cuenta con múltiples herramientas para la conexión y control a dispositivos, así como herramientas matemáticas para la integración del modelado matemático. Cabe destacar que al utilizarse métodos meramente geométricos y trigonométricos, como fue en este caso, la programación no fue tan robusta; sin embargo, es importante saber que existen también otros métodos, como lo son los métodos numéricos, inteligencia artificial así como algebra de cuaterniones, y que conllevan a los mismos resultados al buscar solución, por esta razón se propone para futuras investigaciones y aplicaciones del desarrollo del Robots, se aplicara otros métodos anteriormente mencionados.

Para el desarrollo del prototipo se utilizó la tecnología de impresión 3D, esta tecnología ha facilitado la rápida implementación y desarrollo de prototipos, se han reducido los tiempos de integración con simuladores como fue en este caso.

Como trabajo futuro se pretende enfocarse en la validación de las variables de desempeño del robot cuando este se somete a pruebas de repetibilidad (G R&R), evaluando el comportamiento de las posiciones angulares de los servomotores.

#### 5. REFERENCIAS

- [1] M. Dehghani, «Neural Network Solution for Forward Kinematics Problem of HEXA Parallel Robot,» American Control Conference, p. 6, 2008.
- [2] P. A. O. ESCOTO, MODELACIÓN Y SIMULACIÓN CINEMÁTICA DE UN ROBOT DELTA PLANAR TIPO RR, Ciudad Universitaria, México, D. F.: UNAM, 2006.
- [3] F. C. JIMÉNEZ, TESIS: ANÁLISIS CINEMÁTICO Y DINÁMICO DE UN ROBOT PARALELO ESPACIAL RUS, Departamento de Posgrado en Ingeniería Mecánica de la Facultad de Ingeniería: UNAM, 2008.
- [4] M. J. X. Atonatiúh, «Diseño y Control de un Robot Paralelo,» Asociación Mexicana de Mecatrónica A.C., vol. 6, pp. 219-225, 2007.
- [5] W. Pachacama, «Diseño y construcción de un prototipo de robot delta para aplicaciones pick & place,» Jornadas en Ingeniería Eléctrica y Electrónica, vol. 27, pp. 100-107, 2017.
- [6] J. GUDIÑO-LAU, «Diseño de un robot delta y su análisis cinemático,» Revista de Aplicación Científica y Técnica, vol. 3, n° 8, pp. 13-23, 2017.
- [7] C. A. P. Cortés, «OPTIMIZACIÓN DIMENSIONAL DE UN ROBOT PARALELO TIPO DELTA BASADO EN EL MENOR CONSUMO DE ENERGÍA,» Ciencia e Ingeniería Neogranadina, vol. 21, n° 1, pp. 77-88, 2011.
- [8] I. j. V. Cuevas, «SIMULADOR DEL ROBOT DELTA EN LABVIEW UTILIZANDO METODOS,» Mem. ELECTRO, vol. 40, n° 1405-2172, p. 219 – 224, 2018.
- [9] Ignacio Vázquez Cuevas, «Modelación y Diseño de un Simulador de un Robot Paralelo Manejado por un Controlador Manual Didáctico,» Congreso Nacional de Mecatrónica, vol. 8, pp. 329-324, 2009.
- [10] «<https://www.thingiverse.com/thing:3465651>,» [En línea].