

Diseño de un robot paralelo

Mauricio Cirilo Méndez Canseco
Dr. en Ingeniería,
Universidad Anáhuac México Sur
Coordinador de Ingeniería Mecatrónica
Universidad Anáhuac México Sur
mauricio.mendez@anahuac.mx; h1sim@yahoo.com

Resumen— En este artículo se presenta el diseño y construcción de un robot paralelo tipo Delta. Este tipo de robot es utilizado en la industria para propósitos de manipulación del producto a alta velocidad. Durante la fase de diseño se utilizó una metodología basada en el Álgebra de Cuaternios. En esta álgebra existe una transformación lineal con propiedades de una rotación que permite expresar vectores en función de la rotación dada por un ángulo alrededor de un eje definido en el espacio. De esta forma es posible representar un sistema mecánico espacial a través de vectores. La representación vectorial del robot delta sirve para diversos cálculos: obtención de las dimensiones del robot, simulación del robot para verificar su alcance espacial, implementar la cinemática directa para realizar una manipulación manual del robot y la cinemática inversa para programar una secuencia de movimientos punto a punto. También se muestra algunos componentes mecánicos utilizados en el robot, se especifica el control implementado en el robot y se explica la forma básica de programar movimientos de los servomotores. En Mathematica® se han desarrollado funciones que permiten realizar todos los cálculos cinemáticos de sistemas mecánicos espaciales, mecanismos o robots; y funciones que permiten la simulación de los mismos.

Palabras clave— Robot paralelo, robótica, diseño, control, Cuaternio, manipulación.

Abstract— This work presents the design and construction of a parallel robot type Delta. This type of robot is used in the industry for handling purposes product at high speed. One was used during the design phase Algebra Quaternion-based methodology. In this algebra has a linear transformation with rotation properties that allows expressing vectors depending on rotation given by an angle on a defined axis in the space. It is possible to represent a mechanical system space to through vector. Vector modeling of the delta robot is used to various calculations: obtaining the dimensions of the robot, the robot simulation to verify your spatial extent, implement the direct kinematics for perform a manual handling of the robot and the inverse kinematics to schedule a sequence of movements point to point. It also demonstrates some mechanical components used in the robot, the control implemented in the robot specifications and explains the basic form of schedule movements of the servo motors. Functions have been developed in Mathematica® allow to perform all calcula-

tions space mechanical systems, kinematics mechanisms or robots; and functions that allow the simulation.

Keywords— Parallel robot, robotics, design, control, Quaternion, handling.

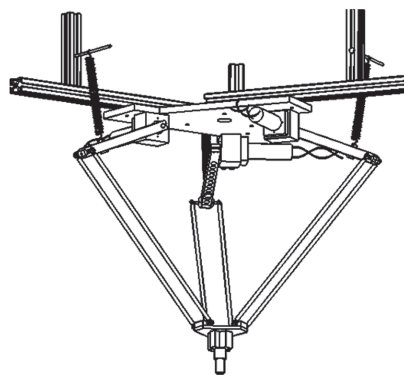
I. INTRODUCCIÓN

Un robot de cinemática paralela, también llamado robot de cadena cerrada o manipulador paralelo, consiste básicamente en una plataforma móvil unida a una base fija por medio de varios brazos. Típicamente cada brazo está controlado por un actuador. En general estos robots paralelos pueden manipular una carga mayor que los robots de cadena abierta, ya que comparten la carga entre varios brazos paralelos.

Los robots paralelos fueron introducidos hace ya algunas décadas por Gough (1975) y Stewart (1965). Clavel (1989) propuso el robot Delta, el cual es utilizado en aplicaciones de alta velocidad en la industria.

El robot paralelo tipo Delta, mostrado en la Fig. 1, es simétrico, espacial y compuesto por tres eslabonamientos idénticos los cuales conectan la base fija con la plataforma móvil, cuenta con 3 grados de libertad.

FIG. 1 MANIPULADOR PARALELO TIPO DELTA



Fuente: El autor

Las dimensiones de los eslabones se determinaron, al realizar un análisis con base en el modelado del robot mediante Cuaterniones.

Las piezas de las cadenas cinemáticas se fabricaron en aluminio y la plataforma móvil en Nylamid.

Se diseñó una interfaz computacional para poder manipular y controlar al robot desde una PC.

Este programa se realizó en el lenguaje C# e interactúa con la tarjeta de control de los tres servomotores utilizados.

II. VECTORES DINÁMICOS

Es posible obtener las ecuaciones cinemáticas de cadenas abiertas y cerradas a través de representarlas con vectores que contengan propiedades de movimiento similares al del sistema que se va a estudiar. A estos vectores los llamamos Vectores Dinámicos.

La obtención de Vectores Dinámicos con propiedades de rotación se facilita al utilizar el Álgebra de Cuaterniones. Lo que hace necesario conocer las propiedades de esta álgebra [1]. A continuación se presenta una transformación lineal con propiedades de rotación para definir dichos vectores.

A. Rotación con Álgebra de Cuaternios Q

Existe una transformación lineal $\rho(p, \cdot): Q \rightarrow Q$, donde $p \in Q$ está fijo, y preserva el producto interno, la norma y el ángulo:

$$\rho(p, q) = p * q * p^{-1} = \frac{1}{\|p\|^2} \cdot (p * q * \bar{p}) \forall q \in Q \quad (1)$$

Donde:

$*$: $Q \times Q \rightarrow Q$ es la operación multiplicativa de los Cuaterniones.

$\|\cdot\|: Q \rightarrow \mathbb{R}^+$ es la norma en espacio vectorial Q .

\cdot es la multiplicación escalar.

$\bar{p} \in Q$ es el conjugado de p .

La cual es una rotación en el espacio vectorial de los Cuaterniones según se demuestra en [1].

Los componentes de $p \in Q$ están relacionados con los parámetros físicos de la rotación, mediante las siguientes expresiones:

$$p_0 = \cos \frac{\theta}{2}, \quad p_v = \sin \frac{\theta}{2} w_v$$

θ es el ángulo de rotación.

$w = (w_1, w_2, w_3)$ es el eje de rotación.

Si $p \in Q$ es un Cuaternión unitario, entonces la rotación $\rho(p, \cdot): Q \rightarrow Q$ se simplifica a la siguiente forma:

$$\rho(p, q) = p * q * p^{-1} = p * q * \bar{p} \quad \forall q \in Q \quad (2)$$

B. Metodología para calcular vectores dinámicos

Antes de mostrar las ecuaciones cinemáticas para n cuerpos, ejemplificaremos el caso de dos cuerpos acoplados.

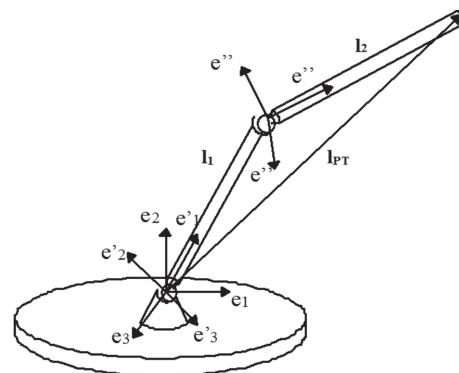
Todos los desarrollos de los resultados aquí mostrados se encuentran detallados en [2].

El modelado consiste en determinar a través de los eslabones el vector de posición L_{T2} del extremo final de los cuerpos acoplados a la base global considerando el movimiento de cada eslabón y los efectos que tiene cada uno de ellos en los cuerpos siguientes, como se ilustra en la Fig. 2:

$$L_{T2}(t) = T_v[P_1(t) * T_v^{-1} [I_1] * \bar{P}_1(t) + P_2(t) * P_1(t) * T_v^{-1} [I_2] * \bar{P}_2(t) * P_1(t)] \quad (3)$$

I_1 e I_2 son los vectores que representan a los dos cuerpos en la posición en la que se determinó el modelo.

FIG. 2 SISTEMAS DE REFERENCIA



Fuente: El autor

Donde el Cuaternión $P_1(t)$ es la rotación del primer cuerpo y está constituido por:

$$P_1(t) = (P_{10}, P_{11}, P_{12}, P_{13}) \text{ y}$$

$$P_{10} = \cos \frac{f_1(t)}{2}, \quad P_{v0}(t) = \sin \frac{f_1(t)}{2} \mathbf{ur1}$$

El eje de rotación $\mathbf{ur1}_v$ está definido respecto a la base global.

El Cuaternión $P_2(t)$ es la rotación del segundo cuerpo formado por:

$$P_2(t) = (P_{20}, P_{21}, P_{22}, P_{23}) \text{ y}$$

$$P_{20}(t) = \cos \frac{f_2(t)}{2}, \quad P_{v0}(t) = \sin \frac{f_2(t)}{2} \mathbf{ur2}_v$$

El vector $\mathbf{ur2}_v$ es el eje de rotación del segundo cuerpo definido en el sistema global y es afectado por la rotación $P_1(t)$.

$$\mathbf{ur2} = T_v [\rho(P_1(t), T_v^{-1}[\mathbf{u}_2])] \quad (4)$$

La velocidad la obtenemos al derivar la ecuación de posición y simplificar mediante las propiedades mencionadas, hasta obtener:

$$\dot{\mathbf{L}}_{T2}(t) = f_1'(t) \mathbf{u}_{r1} \times \mathbf{L}_1(t) + f_2'(t) \mathbf{u}_{r2} \times \mathbf{L}_2(t) + f_1'(t) \mathbf{u}_{r1} \times \mathbf{L}_2(t) \quad (5)$$

Donde: $\mathbf{u}_{r1} = \mathbf{u}_1$

Los términos \mathbf{u}_1 y \mathbf{u}_2 son los ejes de rotación definidos respecto a la base global antes de efectuar cualquier movimiento. Cuando el acoplamiento se mueve los ejes de rotación son afectados por el movimiento y esto se indica con los Cuaterniones que definen dicho movimiento; así, los ejes definidos en cada instante son \mathbf{u}_{r1} y \mathbf{u}_{r2} .

La velocidad puede escribirse de la siguiente forma:

$$\dot{\mathbf{L}}_{T2}(t) = \mathbf{w}_1(t) \times \mathbf{L}_1 + \mathbf{w}_2(t) \times \mathbf{L}_2(t) + \mathbf{w}_1(t) \times \mathbf{L}_2(t) \quad (6)$$

Donde: $\mathbf{w}_1(t) = f_1'(t) \mathbf{u}_{r1}$

$\mathbf{w}_2(t) = f_2'(t) \mathbf{u}_{r2}$

Y la aceleración está dada por:

$$\ddot{\mathbf{L}}_{T2}(t) = \alpha_1(t) \times \mathbf{L}_1(t) + \mathbf{w}_1(t) \times [\mathbf{w}_1(t) \times \mathbf{L}_1(t)] + \alpha_2(t) \times \mathbf{L}_2(t) + \mathbf{w}_2(t) \times [\mathbf{w}_2(t) \times \mathbf{L}_2(t)] + \alpha_1(t) \times \mathbf{L}_2(t) + \mathbf{w}_1(t) \times [\mathbf{w}_1(t) \times \mathbf{L}_2(t)] + 2 \mathbf{w}_1(t) \times [\mathbf{w}_2(t) \times \mathbf{L}_2(t)] \quad (7)$$

Donde:

$\mathbf{w}_1(t) = f_1'(t) \mathbf{u}_{r1}$ es la velocidad angular en $L_1(t)$.

$\alpha_1(t) = f_1''(t) \mathbf{u}_{r1}$ es la aceleración angular en $L_1(t)$.

$\mathbf{w}_2(t) = f_2'(t) \mathbf{u}_{r2}$ es la velocidad angular en $L_2(t)$.

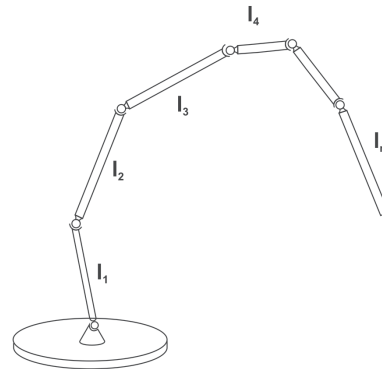
$\alpha_2(t) = f_2''(t) \mathbf{u}_{r2}$ es la aceleración angular en $L_2(t)$.

Todo expresado respecto a la base global.

C. Cinemática de n cuerpos rígidos

De acuerdo a la Fig. 3 se muestra el algoritmo desarrollado en [2] para calcular las ecuaciones cinemática de n cuerpos rígidos.

FIG. 3 SISTEMA DE N CUERPOS ACOPLADOS



Fuente: El autor

La ecuación de posición, velocidad y aceleración para n cuerpos acoplados son:

Posición L_{Tn} del extremo final del sistema de n cuerpos rígidos.

$$\mathbf{L}_{Tn} = \sum_{m=1}^n \mathbf{L}_m \quad (8)$$

Donde:

$$\mathbf{L}_m = \prod_{i=1}^m \mathbf{P}_i(t) * \mathbf{l}_m * \overline{\prod_{i=1}^m \mathbf{P}_i(t)} = T_v [\rho(\prod_{i=1}^m \mathbf{P}_i(t), \mathbf{l}_m)]$$

La velocidad:

$$\prod_{i=1}^m \mathbf{P}_i(t) = \mathbf{P}_m(t) * \mathbf{P}_{m-1}(t) * \dots * \mathbf{P}_1(t)$$

$$V_{Tn} = \sum_{m=1}^n \dot{L}_m = \sum_{m=1}^n \sum_{i=1}^m w_i \times L_m \tag{9}$$

Donde:

$$w_m = f'_m(t) T_v \left[\rho \left(\prod_{i=1}^{m-1} P_i(t), u_m \right) \right]$$

y u_m es el eje de rotación del elemento L_m .

La aceleración:

$$\begin{aligned} a_{Tn} &= \sum_{m=1}^n \ddot{L}_m \\ &= \sum_{m=1}^n \left[\sum_{i=1}^m (\alpha_i \times L_m + w_i \times w_i \times L_m) + \right. \\ &\quad \left. 2 \sum_{k=1}^m \sum_{i=1+k}^m (w_k \times w_i \times L_m) \right] \end{aligned} \tag{10}$$

Donde:

$$\alpha_m = f''_m(t) T_v \left[\rho \left(\prod_{i=1}^{m-1} P_i(t), u_m \right) \right]$$

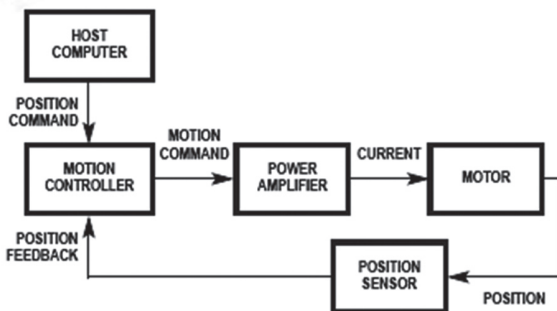
Este algoritmo sistematiza el cálculo de las ecuaciones de posición, velocidad y aceleración de n cuerpos acoplados de forma simplificada.

Esta operación es muy útil en la cinemática cuando tenemos que relacionar un sistema de referencia a partir de otro, mediante rotaciones la base que forma a cada uno de estos sistemas.

III. SISTEMA DE CONTROL

El sistema de control está basado en un controlador digital movimiento (DMC), amplificador y servomotor como el mostrado en la Fig. 4.

FIG. 4 SERVOSISTEMA



Fuente: El autor

El robot paralelo tipo Delta requiere tres servomotores para efectuar un movimiento y posicionarse en el espacio. El sistema utilizado se muestra en la Fig. 5.

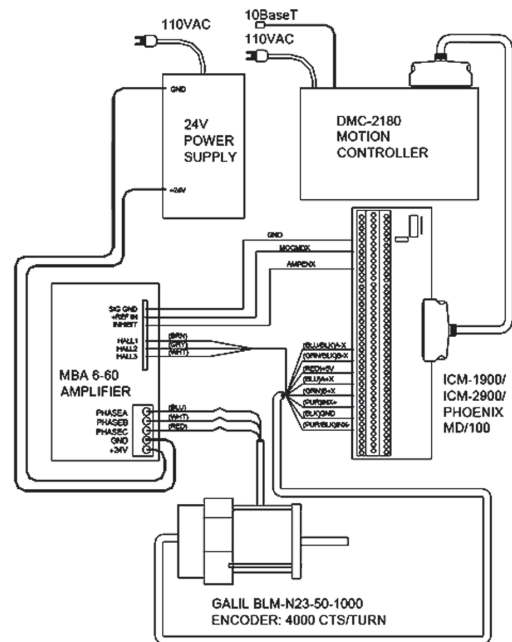
El DMC tiene un extenso conjunto de instrucciones para programar diversos tipos de movimien-

tos y aplicaciones. Las instrucciones están representadas por comandos en inglés formadas por dos letras ASCII. Por ejemplo para iniciar el movimiento del eje X e Y se escribe BG XY.

La programación básica en un DMC se puede apreciar en el ejemplo siguiente:

#A	etiqueta
PR 4000	distancia relativa para recorrer
SP 20000	velocidad
AC 200000	Aceleración
DC 200000	Desaceleración
BGX	iniciar movimiento
EN	fin de programa

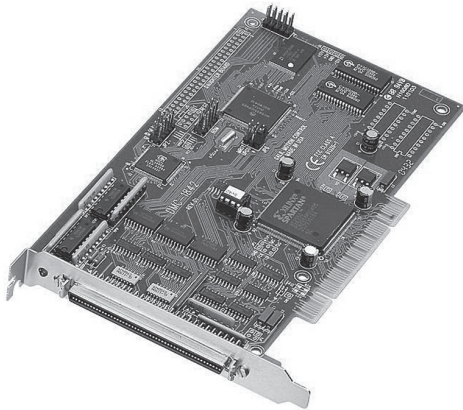
FIG. 5 SISTEMA DE CONTROL



Fuente: El autor

El DMC mostrado en la Fig. 6 utiliza una microcomputadora de 32 bits y diversos recursos para manejar aplicaciones complejas. Tiene sintonización avanzada PID, memoria no volátil multitarea para ejecutar las aplicaciones, entradas y salidas analógicas y digitales para implementar sensores externos. Maneja varios modos de movimiento: posicionamiento punto a punto, control de velocidad, interpolación lineal y circular, contorno y ECAM. La velocidad de comunicación con los encoders es de hasta 22 MHz.

FIG. 6 TARJETA DE CONTROL DE MOVIMIENTO



Fuente: El autor

Los servomotores utilizados que se muestran en la Fig. 7 tienen las siguientes características:
 Alta relación torque-inercia, ideal para movimiento punto a punto que requiere elevada aceleración.
 Torque continuo de 55 oz-in.
 Resolución del encoder de 1000 líneas.

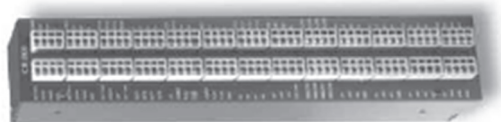
FIG. 7 SERVOMOTOR



Fuente: El autor

El módulo de interconexión Fig. 8 permite la distribución del cable SCSI que sale de la tarjeta de control en diferentes terminales de salida y entrada.

FIG. 8 MÓDULO DE INTERCONEXIÓN



Fuente: El autor

IV. DISEÑO MECÁNICO

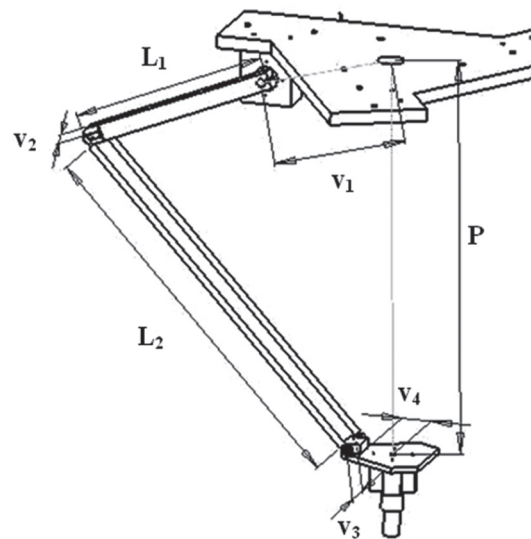
A partir de la configuración de un robot Delta hay que realizar un análisis del área de trabajo que se quiere cubrir con el robot para poder definir las dimensiones del mismo.

Este análisis se realiza a partir de las ecuaciones que definen al robot; las cuales se obtienen a partir de las cadenas cinemáticas cerradas que define cada uno de los brazos que lo conforman.

Las cadenas cinemáticas se definen como una suma de vectores, como se puede observar en la Fig. 9, la cual está dada por:

$$v_1 + L_1 + v_2 + L_2 + v_3 + v_4 = P$$

FIG. 9 CADENA CINEMÁTICA



Fuente: El autor

De acuerdo a la suma de vectores que se define en cada brazo con Cuaterniones en un ambiente de Mathematica® [2] se calcula de la siguiente manera:

```
(*Brazo 1*)
U11 = VeQu[teta1] * Degree, Ye[[1]]]
L1[1] = Rota[{U11}, {0, 0, -L01}] // TrigReduce // Chop;
eje[1, 1] = Rota[{U11}, Xe[[1]]];
eje[1, 2] = Rota[{U11}, Ye[[1]]];
U12 = VeQu[teta2] * Degree, eje[1, 1]] // Simplify;
U13 = VeQu[teta3] * Degree, eje[1, 2]] // Simplify;
a[1, 1] = Rota[{U11}, L01b Ye[[1]]];
a[1, 2] = Rota[{U13, U12, U11}, {0, 0, -L02b}];
a[1, 3] = Rota[{U11}, -L04b Ye[[1]]];
a[1, 4] = -L05b Xe[[1]];
```


El segundo brazo:

```
(*Brazo 2*)
U21 = VeOu[alfa1 * Degree, Ye[[2]]]
L1[2] = Rota[{U21}, {0, 0, -L01}] // TrigReduce // Chop;
eje[2, 1] = Rota[{U21}, Xe[[2]]];
eje[2, 2] = Rota[{U21}, Ye[[2]]];
U22 = VeOu[alfa2 * Degree, eje[2, 1]] // Simplify;
U23 = VeOu[alfa3 * Degree, eje[2, 2]] // Simplify;
a[2, 1] = Rota[{U21}, L01b Ye[[2]]];
a[2, 2] = Rota[{U23, U22, U21}, {0, 0, -L02b}];
a[2, 3] = Rota[{U21}, -L04b Ye[[2]]];
a[2, 4] = -L05b Xe[[2]];
```

El tercer brazo:

```
U31 = VeOu[gamma1 * Degree, Ye[[3]]]
L1[3] = Rota[{U31}, {0, 0, -L01}] // TrigReduce // Chop;
eje[3, 1] = Rota[{U31}, Xe[[3]]];
eje[3, 2] = Rota[{U31}, Ye[[3]]];
U32 = VeOu[gamma2 * Degree, eje[3, 1]] // Simplify;
U33 = VeOu[gamma3 * Degree, eje[3, 2]] // Simplify;
a[3, 1] = Rota[{U31}, L01b Ye[[3]]];
a[3, 2] = Rota[{U33, U32, U31}, {0, 0, -L02b}];
a[3, 3] = Rota[{U31}, -L04b Ye[[3]]];
a[3, 4] = -L05b Xe[[3]];
```

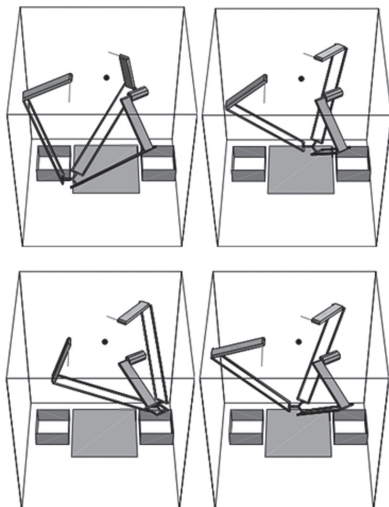
Y la suma de vectores que define a cada cadena cerrada es:

```
C1e = base + Dejemot[[1]] + L1[1] + A[1, 1] + A[1, 2] + A[1, 3] + A[1, 4];
C2e = base + Dejemot[[2]] + L1[2] + A[2, 1] + A[2, 2] + A[2, 3] + A[2, 4];
C3e = base + Dejemot[[3]] + L1[3] + A[3, 1] + A[3, 2] + A[3, 3] + A[3, 4];
```

Con esto es posible aplicar cinemática inversa para conocer la configuración total del robot de acuerdo a la posición a alcanzar.

Con el modelo obtenido y los comandos gráficos definidos en [1] es posible obtener diversas configuraciones del robot como se muestra en la Fig. 10.

FIG. 10 SIMULACIÓN DEL ROBOT

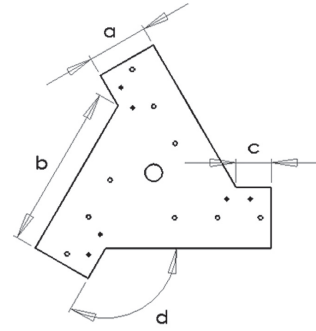


Fuente: El autor

Las ecuaciones permiten definir las longitudes de las piezas, con las cuales se obtuvieron los planos de las piezas que se van a maquinar para poder construir el robot tipo Delta.

En la plataforma fija, Fig. 11, se montan los tres servomotores del robot en forma de delta.

FIG. 11 PLATAFORMA FIJA

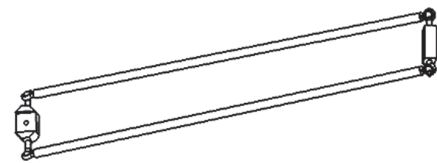


Fuente: El autor

El eslabón L_1 mostrado en la Fig. 9 es una barra de aluminio que acopla al servomotor con las barras cilíndricas paralelas.

Las barras paralelas son cilíndricas, Fig. 11 y en sus extremos se acoplan elementos esféricos que permiten el ajuste de la plataforma inferior debido por los tres brazos.

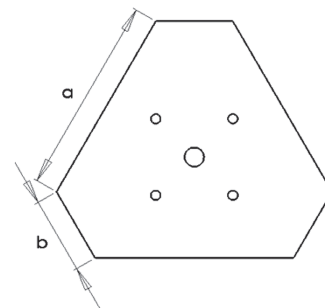
FIG. 12 BARRAS CILÍNDRICAS



Fuente: El autor

La plataforma móvil Fig. 12 une los tres pares de barras cilíndricas y en ella se monta el efector final.

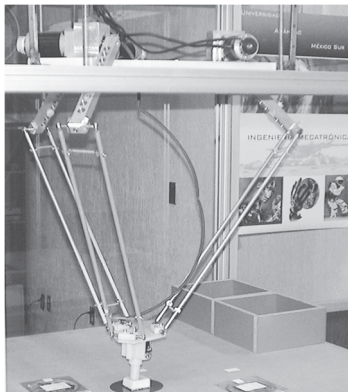
FIG. 13 BARRAS CILÍNDRICAS



Fuente: El autor

En la Fig. 13 se puede observar las piezas del robot ensambladas y montadas en una estructura de aluminio.

FIG. 14 ROBOT ENSAMBLADO



Fuente: El autor

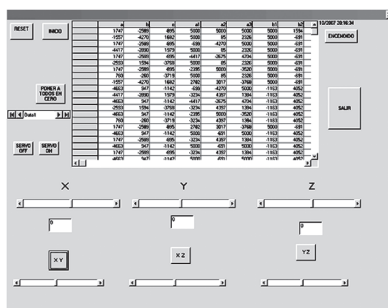
V. SOFTWARE DE CONTROL

La interfaz computacional para comunicarse con el robot y enviarle las instrucciones de la tarea que debe efectuarse tiene las siguientes funciones:

- Manejo manual del robot a través de scrolls para cada motor
- Manejo de servomotores de forma combinada
- Grabado de posiciones deseadas
- Ejecución de una secuencia de posiciones con movimiento punto a punto
- Mostrar las posiciones grabadas para la secuencia de movimientos

La interfaz mostrada en la Fig.14 se desarrolló en lenguaje C# de dot net y tiene implementada la cinemática inversa para obtener los ángulos necesarios que se requiere en cada servomotor para que el robot manipule los objetos requeridos.

FIG. 15 INTERFAZ COMPUTACIONAL



Fuente: El autor

En este programa es posible realizar una secuencia de posiciones de forma continua y suave que permite lograr altas velocidades durante su desarrollo.

VI. CONCLUSIONES

Se logró diseñar y construir un robot paralelo tipo “Delta” con tres Grados de Libertad con los recursos tecnológicos con los que cuenta la Facultad de Ingeniería de la Universidad Anáhuac México Sur; el cual, se muestra en un video en la página mendez.mecatronica.cc.

En el diseño mecánico se aplicó el concepto de Vectores Dinámicos definidos con el Álgebra de Cuaterniones para que con base en la cinemática inversa, se pudiera calcular las longitudes de los eslabones.

La interfaz computacional cumplió con los requerimientos de comunicación con el DMC para el control de los servomotores a través de una PC.

El desempeño general del robot “Hércules” cumplió con las expectativas planteadas al inicio del proyecto.

Los objetivos a mediano plazo son incorporarle un sistema de visión para que pueda manipular los objetos de manera automática e implementar otros tipos de control de movimiento.

REFERENCIAS

Published Papers from Conference Proceedings

- [1] Méndez M. “Dinámica de cuerpos rígidos con Quaterniones: una aplicación a los mecanismos”. Tesis de Doctorado en ingeniería, Universidad Anáhuac del Sur, 2000.
 - [2] Méndez M. “Simplificación del Modelado Cinemático de n cuerpos rígidos con Quaterniones”, Tercer Congreso Mexicano de Robótica, 2001
 - [3] Méndez M. “Dinámica de un sistema 2 GL con Quaterniones”, Congreso Nacional de Control Automático 2003
- Manuals
- [4] G Motion Control, “DMC-2x00 Manual”,2005