

Diseño y Control de un Robot Paralelo

Meneses Jiménez Xavier Atonatiúh , Méndez Canseco Mauricio Cirilo
y Cortés Bringas Eduardo

mauricio.mendez@anahuac.mx; h1sim@yahoo.com

Universidad Anáhuac México Sur.

Av. De las Torres 131, col. Olivar de los Padres, Del. Álvaro Obregón, México, D.F., CP 01780
Tel 01 (55) 56 28 88 00 ext. 225

Resumen

En este artículo se presenta el diseño y construcción de un robot paralelo tipo Delta. Además, se explica la manera en que se analiza la configuración mecánica y como se implementa la interfaz computacional para lograr programar una secuencia de movimientos punto a punto.

El Álgebra de Cuaterniones es utilizada para obtener Vectores Dinámicos [1] [2] [3] con el propósito de simular el comportamiento del robot paralelo y obtener sus ecuaciones cinemáticas. Los Vectores Dinámicos 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.

1. 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 (Figura 1) es simétrico, espacial y compuesto por tres eslabonamientos idénticos los cuales conectan la base

fija con el triángulo equilátero del efector final (Actuador) ubicado en uno de los bordes del mismo triángulo, contando con 3 grados de libertad, aumentando 1 grado de libertad si el actuador lo posee, y esto dependerá de la tarea a realizar.

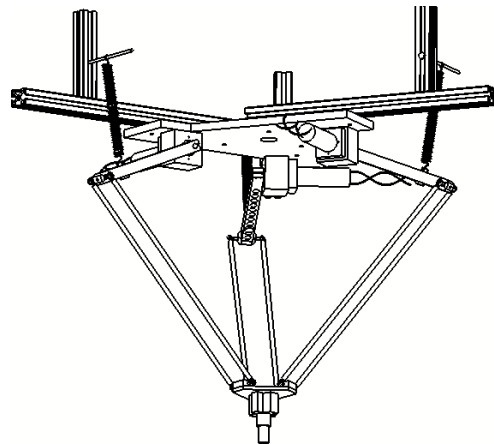


Figura 1. Manipulador paralelo tipo Delta "Hércules".

El robot construido y mostrado en la Figura 1 tiene por nombre "Hércules".

Las dimensiones de los eslabones se determinaron realizando un análisis con base en el modelado del robot utilizando Cuaterniones.

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

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.

2. 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 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.

2.1 Rotación con Álgebra de Cuaterniones Q.

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

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

Donde:

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

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

\bullet es la multiplicación escalar.

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

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

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

$$\mathbf{p}_0 = \cos \frac{\theta}{2}, \quad \mathbf{p}_v = \sin \frac{\theta}{2} \mathbf{w}_v$$

θ es el ángulo de rotación

$\mathbf{w} = (w_1, w_2, w_3)$ es el eje de rotación

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

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

2.2 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 a detalle en [2].

El modelado consiste en determinar a través de los eslabones el vector de posición \mathbf{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 (Figura. 2):

$$\mathbf{L}_{T2}(t) = T_v[\mathbf{P}_1(t) * T_v^{-1}[\mathbf{l}_1] * \bar{\mathbf{P}}_1(t) + \mathbf{P}_2(t) * \mathbf{P}_1(t) * T_v^{-1}[\mathbf{l}_2] * \bar{\mathbf{P}}_2(t) * \mathbf{P}_1(t)] \quad (3)$$

Siendo \mathbf{l}_1 y \mathbf{l}_2 son los vectores que representan a los dos cuerpos en la posición en la que se determinó el modelo.

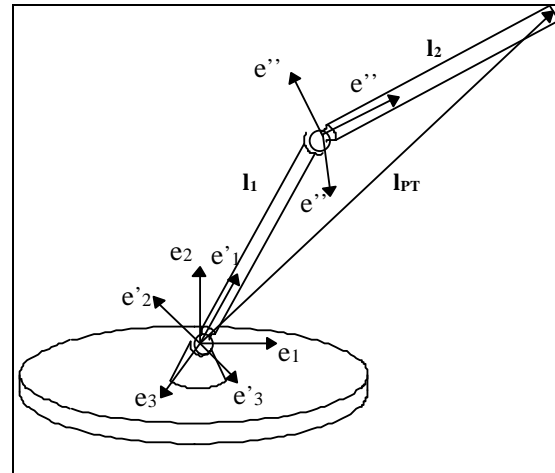


Figura 2. Sistemas de referencia.

Donde el Cuaternión $\mathbf{P}_1(t)$ es la rotación del primer cuerpo y está constituido por:

$$\mathbf{P}_1(t) = (P_{10}, P_{11}, P_{12}, P_{13}) \text{ y}$$

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

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

El Cuaternión $\mathbf{P}_2(t)$ es la rotación del segundo cuerpo formado por:

$$\mathbf{P}_2(t) = (P_{20}, P_{21}, P_{22}, P_{23}) \quad \text{y}$$

$$\mathbf{P}_{20}(t) = \cos \frac{f_2(t)}{2}, \mathbf{Q}_{v0}(t) = \sin \frac{f_2(t)}{2} \mathbf{ur}_{2v}$$

El vector \mathbf{ur}_2 es el eje de rotación del segundo cuerpo definido en el sistema global y es afectado por la rotación $\mathbf{P}_1(t)$.

$$\mathbf{ur}_2 = T_v [\rho(\mathbf{P}_1(t), T_v^{-1}[\mathbf{u}_2])] \quad (4)$$

La velocidad la obtenemos al derivar la ecuación de posición y simplificar utilizando 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 que cualquier movimiento se efectúe. 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 $\mathbf{L}_1(t)$.
 $\alpha_1(t) = f''_1(t) \mathbf{u}_{r1}$ es la aceleración angular en $\mathbf{L}_1(t)$.
 $\mathbf{w}_2(t) = f'_2(t) \mathbf{u}_{r2}$ es la velocidad angular en $\mathbf{L}_2(t)$.
 $\alpha_2(t) = f''_2(t) \mathbf{u}_{r2}$ es la aceleración angular en $\mathbf{L}_2(t)$.

Todo expresado respecto a la base global.

2.3 Cinemática de n cuerpos rígidos.

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

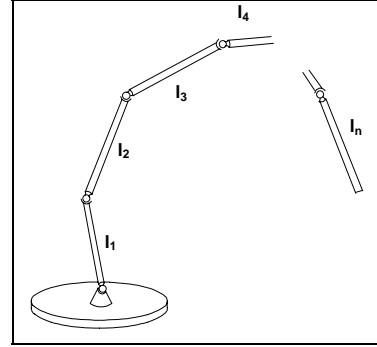


Figura 3. Sistema de n cuerpos acoplados.

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

Posición \mathbf{L}_{Tn} del extremo final del sistema de n cuerpos rígidos.

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

Donde:

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

y

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

Velocidad.

$$\mathbf{V}_{Tn} = \sum_{m=1}^n \dot{\mathbf{L}}_m = \sum_{m=1}^n \sum_{i=1}^m \mathbf{w}_i \times \mathbf{L}_m \quad (9)$$

Donde:

$$\mathbf{w}_m = f'_m(t) T_v [\rho(\prod_{i=1}^{m-1} \mathbf{P}_i(t), \mathbf{u}_m)]$$

y \mathbf{u}_m es el eje de rotación del elemento \mathbf{L}_m .

Aceleración.

$$\mathbf{a}_{Tn} = \sum_{m=1}^n \ddot{\mathbf{L}}_m = \sum_{m=1}^n [\sum_{i=1}^m (\alpha_i \times \mathbf{L}_m + \mathbf{w}_i \times \mathbf{w}_i \times \mathbf{L}_m) + 2 \sum_{k=1}^m \sum_{i=1+k}^m (\mathbf{w}_k \times \mathbf{w}_i \times \mathbf{L}_m)] \quad (10)$$

Donde:

$$\alpha_m = f''_m(t) T_v [\rho(\prod_{i=1}^{m-1} \mathbf{P}_i(t), \mathbf{u}_m)]$$

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, relacionando mediante rotaciones la base que forma a cada uno de estos sistemas.

3. Sistema de control

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

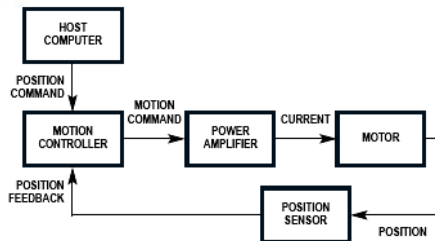


Figura 4. Servosistema.

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

El DMC tiene un extenso conjunto de instrucciones para programar diversos tipos de movimientos y aplicaciones. Las instrucciones están representadas por comandos en inglés formados 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	<i>etiqueta</i>
PR 4000	<i>distancia relativa a recorrer</i>
SP 20000	<i>velocidad</i>
AC 200000	<i>Aceleración</i>
DC 200000	<i>Desaceleración</i>
BGX	<i>iniciar movimiento</i>
EN	<i>fin de programa</i>

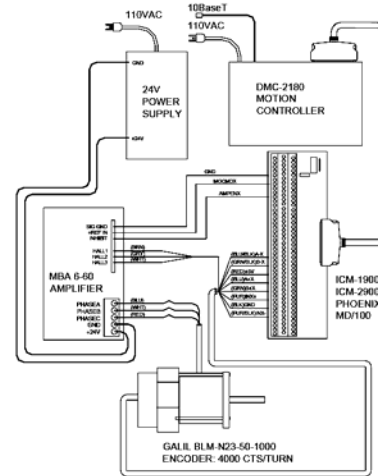


Figura 5. Sistema de control

El DMC (Figura 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.

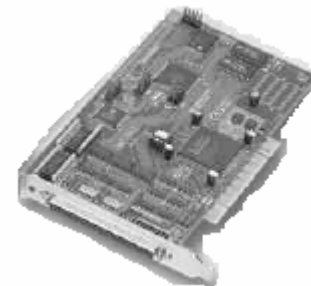


Figura 6. Tarjeta de control de movimiento.

Los servomotores (Figura 7) utilizados tienen las siguientes características:

- Alta relación torque-inercia, ideal para movimiento punto a punto que requieren elevada aceleración.
- Torque continuo de 55 oz-in.
- Resolución del encoder de 1000 líneas.



Figura 7. Servomotor

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

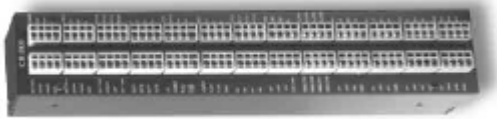


Figura 8. Módulo de interconexión.

4. 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 Figura 9, la cual está dada por:

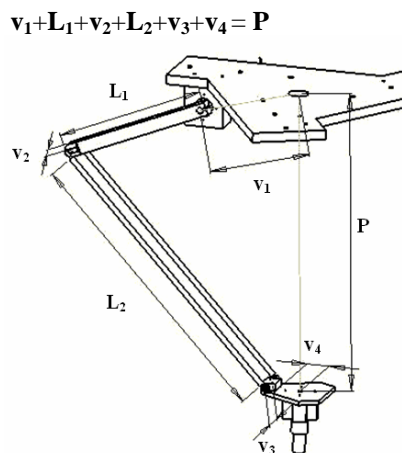


Figura 9. Cadena cinemática

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 = VeQu[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 = VeQu[alfa2 * Degree, eje[2, 1]] // Simplify;
U23 = VeQu[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 = VeQu[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 = VeQu[gamma2 * Degree, eje[3, 1]] // Simplify;
U33 = VeQu[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:

$$\begin{aligned} C1e &= \text{base} + \text{Dejemot}[[1]] + L1[1] + A[1, 1] + A[1, 2] + A[1, 3] + A[1, 4] \\ C2e &= \text{base} + \text{Dejemot}[[2]] + L1[2] + A[2, 1] + A[2, 2] + A[2, 3] + A[2, 4] \\ C3e &= \text{base} + \text{Dejemot}[[3]] + L1[3] + A[3, 1] + A[3, 2] + A[3, 3] + A[3, 4] \end{aligned}$$

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

El modelo obtenido y comandos gráficos definidos en [1] es posible obtener diversas configuraciones del robot como se muestra en la Figura 10.

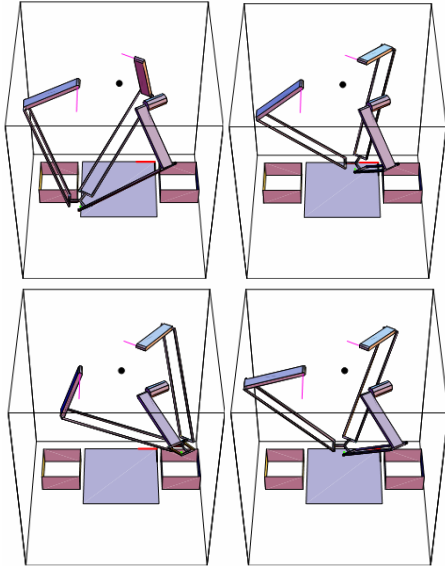


Figura 10. Simulación del robot.

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

En la plataforma fija (Figura 11) se montan los tres servomotores del robot como se aprecia en la Figura 9.

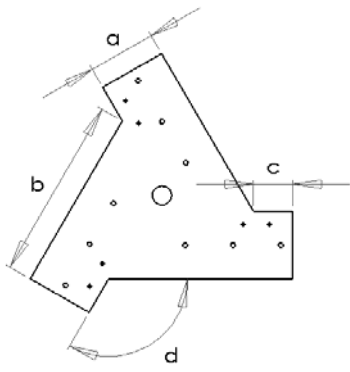


Figura 11. Plataforma fija.

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

Las barras paralelas son cilíndricas (Figura 11) y en sus extremos se acoplan elementos esféricos y permitir el ajuste de la plataforma inferior definido por los tres brazos.

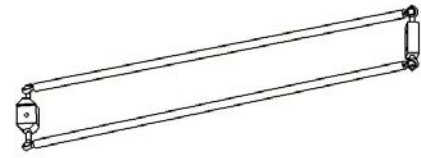


Figura 11. Barras cilíndricas.

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

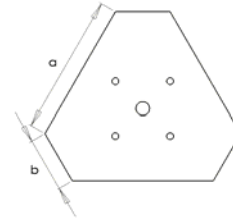


Figura 12. Barras cilíndricas.

En la siguiente foto (Figura 13) se puede observar las piezas del robot ensambladas y montadas en una estructura de aluminio.

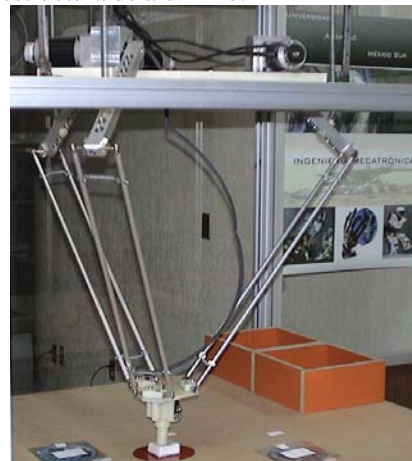


Figura 13. Robot ensamblado.

5. Software de Control

La interfaz computacional para comunicarse con el robot y enviarle las instrucciones de la tarea a efectuar 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 (Figura 14) se desarrollo en lenguaje C# de .net y se implementó la cinemática inversa para obtener los ángulos necesarios que se requiere en cada servomotor para que el robot manipule los objetos requeridos.

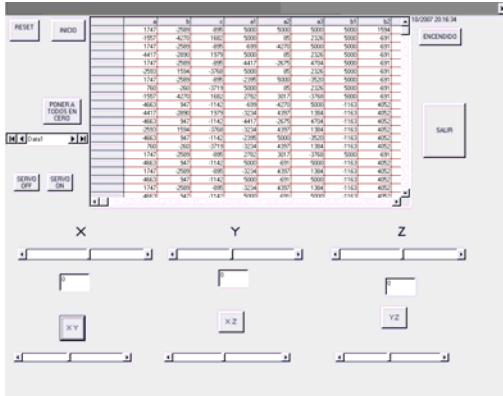


Figura 14. Interfaz computacional.

La implementación en la interfaz computacional se aplicó de forma diferente a la mencionada anteriormente para optimizar los cálculos al no implementar un método numérico. Para lograrlo de cada cadena cinemática planteada se calculo la norma del eslabón L_2 , la cual es conocida y de esta manera también se anula la información del ángulo de dicho eslabón, esto es:

$$|L_2| = |P - v_3 - v_4 - v_1 - L_1 - v_2|$$

Y se despeja el ángulo de cada eslabón L_1 para obtener $\theta(P)$.

6. 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.

En el diseño mecánico se aplicó el concepto de Vectores Dinámicos definidos con el Álgebra de Cuaterniones para que con base a 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 incorporar un sistema de visión para que pueda manipular los objetos de manera automática e implementar otros tipos de control de movimiento.

Referencias

- [1] Méndez M. “Dinámica de cuerpos rígidos con Cuaterniones: 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 Cuaterniones”, Tercer Congreso Mexicano de Robótica, 2001.
- [3] Méndez M. “Dinámica de un sistema 2 GL con Cuaterniones”, Congreso Nacional de Control Automático 2003.
- [4] G Motion Control, “DMC-2x00 Manual”, 2005