



Asignatura: Robótica

Código: 95-0482

Curso: R-6055

Año: 2010

Trabajo Práctico n° 1:
Implementación de una
matriz cinemática en DSP

Alumnos:

- BORNEO, Pablo N.
- DI VRUNO, Federico
- SERATTI, Federico

Profesor: Mas. Ing. GIANNETA, Hernan

JTP: Ing. GRANZELLA, Damian

ÍNDICE

Página

INTRODUCCIÓN TEÓRICA	2
CINEMÁTICA DEL ROBOT	2
RELACION ENTRE CINEMÁTICA DIRECTA E INVERSA	3
MODELO CINEMÁTICO.....	4
MÉTODO BASADO EN MATRICES DE TRANSFORMACIÓN HOMOGÉNEAS.....	4
ALGORITMO DE DENAVIT-HARTENBERG	7
AUMENTANDO LA DESTREZA DE ROBOTS REPETITIVOS	10
DESARROLLO DE LA PRÁCTICA	11
TABLA DE MOVIMIENTOS	13
CÁLCULO DE LAS MATRICES.....	13
DESARROLLO E IMPLEMENTACIÓN EN CODEWARRIOR DSP 56800-E	17
RESULTADOS DE LA SIMULACIÓN	21
GENERACIÓN DE PUNTOS.....	21
GRÁFICA DEL ESPACIO DE TRABAJO	25
CONCLUSIONES	28

INTRODUCCIÓN TEÓRICA

La Cinemática (del griego κινεω, kineo, movimiento) es la rama de la mecánica clásica que estudia las leyes del movimiento de los cuerpos sin tener en cuenta las causas que lo producen, limitándose, esencialmente, al estudio de la trayectoria en función del tiempo.

En la Cinemática se utiliza un sistema de coordenadas para describir las trayectorias, denominado sistema de referencia. La velocidad es el ritmo con que cambia la posición un cuerpo. La aceleración es el ritmo con que cambia su velocidad. La velocidad y la aceleración son las dos principales cantidades que describen cómo cambia su posición en función del tiempo.

CINEMÁTICA DEL ROBOT

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia. Así, la cinemática se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo, y en particular por las relaciones entre la posición y la orientación del extremo final del robot con los valores que toman sus coordenadas articulares. Existen dos problemas fundamentales para resolver la cinemática del robot, el primero de ellos se conoce como el **problema cinemático directo**, y consiste en determinar cual es la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas que se toma como referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot, el segundo denominado **problema cinemático inverso** resuelve la configuración que debe adoptar el robot para una posición y orientación del extremo conocidas.

Por otra parte, la cinemática del robot trata también de encontrar las relaciones entre las velocidades del movimiento de las articulaciones y las del extremo. Esta relación viene dada por el **modelo diferencial** expresado mediante la matriz Jacobiana.

El movimiento relativo en las articulaciones resulta en el movimiento de los elementos que posicionan la mano en una orientación deseada. En la mayoría de las aplicaciones de robótica, se está interesado en la descripción espacial del efector final del manipulador con respecto a un sistema de coordenadas de referencia fija.

La cinemática del brazo del robot trata con el estudio analítico de la geometría del movimiento de un robot con respecto a un sistema de coordenadas de referencia fijo como una función del tiempo sin considerar las fuerzas-momentos que originan dicho movimiento. Así pues, trata con la descripción analítica del desplazamiento espacial del robot como función del tiempo, en

particular las relaciones entre variables espaciales de tipo de articulación y la posición y orientación del efector final del robot.

Estudio de su movimiento con respecto a un sistema de referencia:

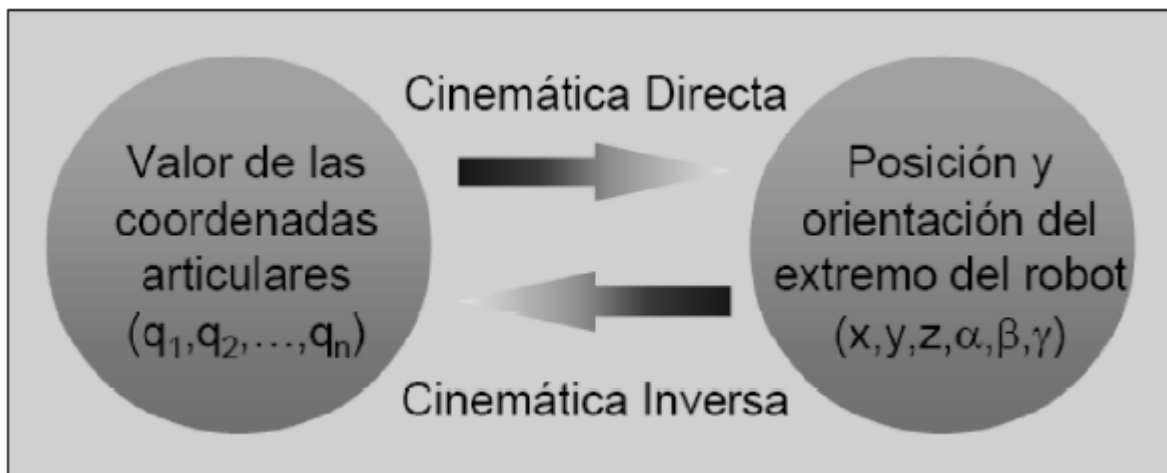
- Relación entre la localización del extremo del robot y los valores de sus articulaciones.
- Descripción analítica del movimiento espacial en función del tiempo.

Problema cinemático directo: Determinar la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas de referencia, conocidos los ángulos de las articulaciones y los parámetros geométricos de los elementos del robot.

Problema cinemático inverso: Determinar la configuración que debe adoptar el robot para alcanzar una posición y orientación del extremo conocidas.

Modelo diferencial (matriz Jacobiana): Relaciones entre las velocidades de las N articulaciones y las del extremo del robot.

RELACION ENTRE CINEMÁTICA DIRECTA E INVERSA



MODELO CINEMÁTICO

Método basado en relaciones geométricas (Trigonometría)

- No sistemática
- Es válido para robots de pocos grados de libertad

Método basado en matrices de transformación homogéneas

- Sistemático
- Es válido para robots con muchos grados de libertad.

MÉTODO BASADO EN MATRICES DE TRANSFORMACIÓN HOMOGÉNEAS

Se utiliza fundamentalmente el álgebra vectorial y matricial para representar y describir la localización de un objeto en el espacio tridimensional con respecto a un sistema de referencia fijo.

Dado que un robot se puede considerar como una cadena cinemática formada por objetos rígidos o eslabones unidos entre sí mediante articulaciones, se puede establecer un sistema de referencia fijo situado en la base del robot y describir la localización de cada uno de los eslabones con respecto a dicho sistema de referencia.

De esta forma, el problema cinemático directo se reduce a encontrar una matriz homogénea de transformación T que relacione la posición y orientación del extremo del robot respecto del sistema de referencia fijo situado en la base del mismo. Esta matriz T será función de las coordenadas articulares.

La resolución del problema cinemático directo consiste en encontrar las relaciones que permiten conocer la localización espacial del extremo del robot a partir de los valores de sus coordenadas articulares.

Así, si se han escogido coordenadas cartesianas y ángulos de Euler para representar la posición y orientación del extremo de un robot de seis grados de libertad, la solución al problema cinemático directo vendrá dada por las relaciones:

$$\begin{aligned}
x &= f_x(q_1, q_2, q_3, q_4, q_5, q_6) \\
y &= f_y(q_1, q_2, q_3, q_4, q_5, q_6) \\
z &= f_z(q_1, q_2, q_3, q_4, q_5, q_6) \\
\alpha &= f_\alpha(q_1, q_2, q_3, q_4, q_5, q_6) \\
\beta &= f_\beta(q_1, q_2, q_3, q_4, q_5, q_6) \\
\gamma &= f_\gamma(q_1, q_2, q_3, q_4, q_5, q_6)
\end{aligned}$$

En general, un robot de n grados de libertad está formado por n eslabones unidos por “ n ” articulaciones, de forma que cada par articulación-eslabón constituye un grado de libertad.

A cada eslabón se le puede asociar un sistema de referencia solidario a el y, utilizando las transformaciones homogéneas, es posible representar las rotaciones y traslaciones relativas entre los distintos eslabones que componen el robot.

Normalmente, la matriz de transformación homogénea que representa la posición y orientación relativa entre los sistemas asociados a dos eslabones consecutivos del robot se suele denominar matriz ${}^{i-1}A_i$.

Así pues, 0A_1 describe la posición y orientación del sistema de referencia solidario al primer eslabón con respecto al sistema de referencia solidario a la base, 1A_2 describe la posición y orientación del segundo eslabón respecto del primero, etc.

Del mismo modo, denominando 0A_k a las matrices resultantes del producto de las matrices ${}^{i-1}A_i$ con i desde 1 hasta k , se puede representar de forma total o parcial la cadena cinemática que forma el robot.

Cuando se consideran todos los grados de libertad, a la matriz 0A_n se le suele denominar **T**. Así, dado un robot de seis grados de libertad, se tiene que la posición y orientación del eslabón final vendrá dada por la matriz **T**:

$$\mathbf{T} = {}^0A_6 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6$$

Aunque para describir la relación que existe entre dos elementos contiguos se puede hacer uso de cualquier sistema de referencia ligado a cada elemento.

La forma habitual que se suele utilizar en robótica es la representación de Denavit- Hartenberg (D-H). Denavit y Hartenberg propusieron en 1955 un método

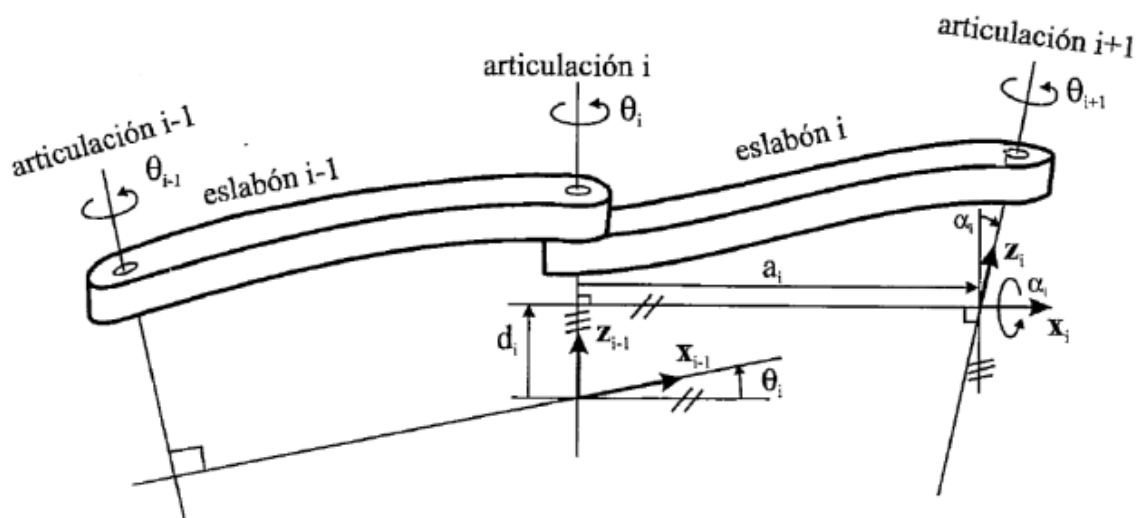
matricial que permite establecer de manera sistemática un sistema de coordenadas $\{ S_i \}$ ligado a cada eslabón i de una cadena articulada, pudiéndose determinar a continuación las ecuaciones cinemáticas de la cadena completa.

Según la representación de D-H, escogiendo adecuadamente los sistemas de coordenadas asociados a cada eslabón, será posible pasar de uno al siguiente mediante 4 transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón.

Estas transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia del elemento i con el sistema del elemento $i-1$.

Las transformaciones en cuestión son las siguientes (es importante recordar que el paso del sistema $\{ S_{i-1} \}$ al $\{ S_i \}$ mediante estas 4 transformaciones está garantizado solo si los sistemas $\{ S_{i-1} \}$ al $\{ S_i \}$ han sido definidos de acuerdo a unas normas determinadas que se expondrán posteriormente):

1. Rotación alrededor del eje z_{i-1} un Angulo θ_i .
2. Traslación a lo largo de z_{i-1} una distancia d_i ; vector $d_i (0,0,d_i)$.
3. Traslación a lo largo de x_i una distancia a_i ; vector $a_i (0,0,a_i)$.
4. Rotación alrededor del eje x_i un ángulo θ_i .



$${}^{i-1}A = T(z, \theta_i) T(0, 0, d_i) T(a_i, 0, 0) T(x, \alpha_i)$$

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C\theta_i & -C\alpha_i C\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ -S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde $a_i, d_i, \theta_i, \alpha_i$ son los parámetros D-H del eslabón i. De este modo, basta con identificar los parámetros $a_i, d_i, \theta_i, \alpha_i$ para obtener las matrices A y relacionar así todos y cada uno los eslabones del robot.

Como se ha indicado, para que la matriz ${}^{i-1}A_i$, definida anteriormente, relacione los sistemas $\{ S_{i-1} \}$ y $\{ S_i \}$, es necesario que los sistemas se hayan escogido de acuerdo a unas determinadas normas. Estas, junto con la definición de los 4 parámetros de Denavit Hartenberg, conforman el siguiente algoritmo para la resolución del problema cinemática directo: se define el Algoritmo de Denavit-Hartenberg para la obtención del modelo cinemática directo.

ALGORITMO DE DENAVIT-HARTENBERG

Denavit y Hartenberg propusieron un método sistemático para descubrir y representar la geometría espacial de los elementos de una cadena cinemática, y en particular de un robot, con respecto a un sistema de referencia fijo. Este método utiliza una matriz de transformación homogénea para descubrir la relación espacial entre dos elementos rígidos adyacentes, reduciéndose el problema cinemático directo a encontrar una matriz de transformación homogénea 4 X 4 que relacione la localización espacial del robot con respecto al sistema de coordenadas de su base.

D-H 1. Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerara como eslabón 0 a la base fija del robot.

D-H 2. Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n .

D-H 3. Localizar el eje de cada articulación. Si esta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.

D-H 4. Para i de 0 a $n-1$ situar el eje z_i sobre el eje de la articulación $i + 1$.

D-H 5. Situar el origen del sistema de la base $\{ S_0 \}$ en cualquier punto del eje z_0 .

Los ejes x_0 e y_0 se situaran de modo que formen un sistema dextrógiro con z_0 .

D-H 6. Para i de 1 a $n-1$, situar el sistema $\{ S_i \}$ (solidario al eslabón i) en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i . Si ambos ejes se cortasen se situaría $\{ S_i \}$ en el punto de corte. Si fuesen paralelos $\{ S_i \}$ se situaría en la articulación $i + 1$.

D-H 7. Situar x_i en la línea normal común a z_{i-1} y z_i .

D-H 8. Situar y_i de modo que forme un sistema dextrógiro con x_i y z_i .

D-H 9. Situar el sistema $\{ S_n \}$ en el extremo del robot de modo que z_n , coincida con la dirección de z_{n-1} y x_n sea normal a z_{n-1} y z_n .

D-H 10. Obtener θ_i como el ángulo que hay que girar en torno a z_{i-1} para que x_{i-1} y x_i , queden paralelos.

D-H 11. Obtener d_i , como la distancia, medida a lo largo de z_{i-1} , que habría que

desplazar $\{ S_{i-1} \}$ para que x_i y x_{i-1} quedasen alineados.

DH 12. Obtener a_i como la distancia medida a lo largo de x_i (que ahora coincidiría con x_{i-1}) que habría que desplazar el nuevo $\{ S_{i-1} \}$ para que su origen coincidiese con $\{ S_i \}$.

DH 13. Obtener α_i como el ángulo que habría que girar entorno a x_i , (que ahora coincidiría con x_{i-1}), para que el nuevo $\{ S_{i-1} \}$ coincidiese totalmente con $\{ S_i \}$.

DH 14. Obtener las matrices de transformación ${}^{i-1}A_i$ definidas anteriormente.

DH 15. Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot $T = {}^0A_1, {}^1A_2, \dots, {}^{n-1}A_n$

DH 16. La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base en función de las n coordenadas articulares.

Los cuatro parámetros de D-H $(a_i, d_i, \theta_i, \alpha_i)$ dependen únicamente de las características geométricas de cada eslabón y de las articulaciones que le unen con el anterior y siguiente. En concreto estos representan:

- θ_i Es el ángulo que forman los ejes x_{i-1} y x_i , medido en un plano perpendicular al eje z_{i-1} , utilizando la regla de la mano derecha. Se trata de un parámetro variable en articulaciones giratorias.
- d_i Es la distancia a lo largo del eje z_{i-1} desde el origen del sistema de coordenadas $(i-1)$ -ésimo hasta la intersección del eje z_{i-1} con el eje x_i . Se trata de un parámetro variable en articulaciones prismáticas.
- a_i Es la distancia a lo largo del eje x_i que va desde la intersección del eje z_{i-1} con el eje x_i hasta el origen del sistema i -ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes z_{i-1} y z_i .
- α_i Es el ángulo de separación del eje z_{i-1} y el eje z_i , medido en un plano perpendicular al eje x_i , utilizando la regla de la mano derecha.

Para obtener el modelo cinemática directo de un robot procedamos de la siguiente manera:

- Establecer para cada elemento del robot un sistema de coordenadas cartesianas ortonormal (x_i, y_i, z_i) donde $i=1,2,\dots,n$ (n = numero de gdl). Cada sistema de coordenadas corresponderá a la articulación $i+1$ y estará fijo en el elemento i (algoritmo D-H).
- Encontrar los parámetros D-H de cada una de las articulaciones.
- Calcular las matrices ${}^{i-1}A_i$
- Calcular la matriz $T_n = {}^0A_1 {}^1A_2 \dots {}^{n-1}A_n$

AUMENTANDO LA DESTREZA DE ROBOTS REPETITIVOS

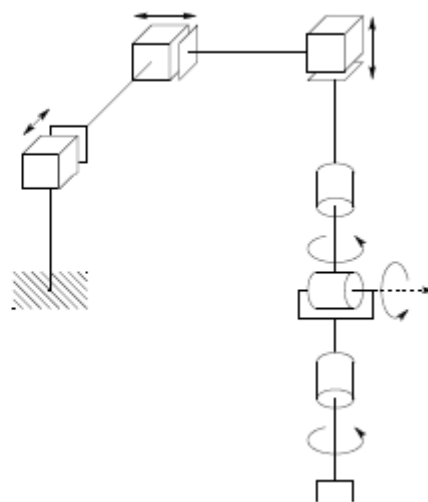
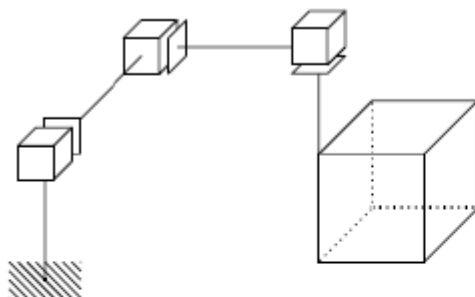
Se usan las coordenadas redundantes para definir tareas adicionales.

El mando de configuración está surgiendo como una manera eficaz de controlar los movimientos de un robot que tiene más grados de libertad y en el cual es necesario definir la trayectoria del efector del extremo y / o el objeto para ser manipulado. Pueden usarse los grados extras o redundantes de libertad para dar destreza de robot y versatilidad. En mando de configuración, la configuración del robot se representa matemáticamente por un juego de variables de configuración que son un vector de coordenadas generalizado y que es más pertinente a la tarea global que es el vector de coordenadas de la junta que aparecen en los acercamientos convencionales a controlar. El vector de la coordenada generalizado consiste en las coordenadas del efector del extremo en el espacio de la tarea, más varias funciones de cinemática que involucran grados redundantes de libertad. La tarea básica del sistema de mando es hacer las coordenadas del efector del extremo seguir la trayectoria deseada. Las funciones de la cinemática pueden seleccionarse para definir una tarea adicional por ejemplo, la anulación de obstáculos u optimización de la cinemática para reforzar la manipulabilidad. En efecto, la tarea adicional define la trayectoria en los grados redundantes de libertad. Las variables de configuración pueden usarse en un esquema de mando adaptable que no exige manipular el conocimiento del modelo matemático complicado de la dinámica del robot o los parámetros del objeto.

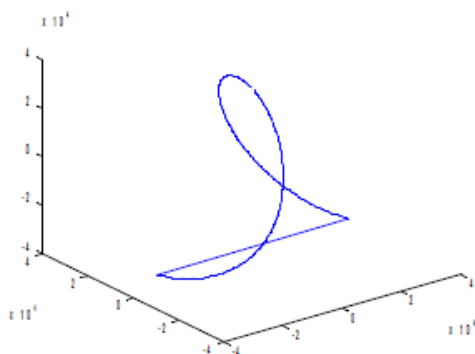
DESARROLLO DE LA PRÁCTICA

Implemente el código C en CW para el DSP56800/E de la cadena cinemática directa para el robot sin gripper, usando como setpoint, una trayectoria lineal continua a cada eje.

Defina los límites y área de trabajo del manipulador.



Imprima el resultado del vector (x,y,z) usando la función plot3(x,y,z) de Matlab.



Link	a_i	α_i	d_i	θ_i
1	0	0	d_1	θ_1^*
2	0	-90	d_2^*	0
3	0	0	d_3^*	0

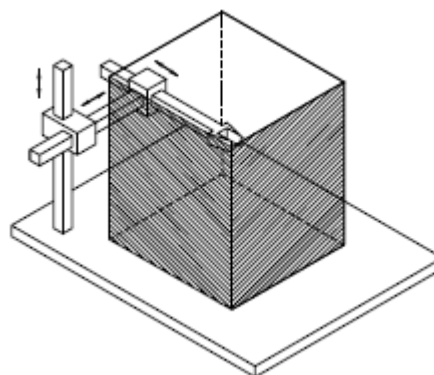
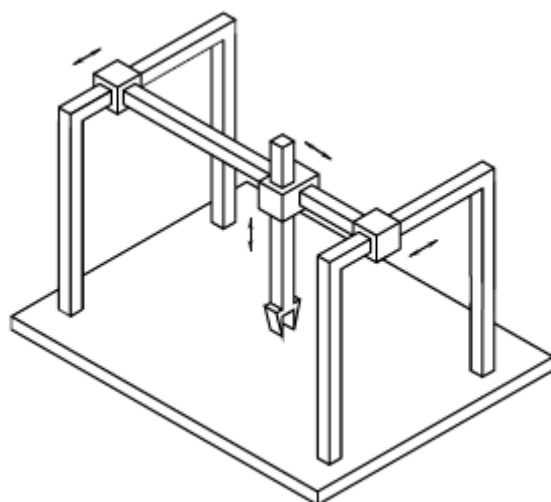
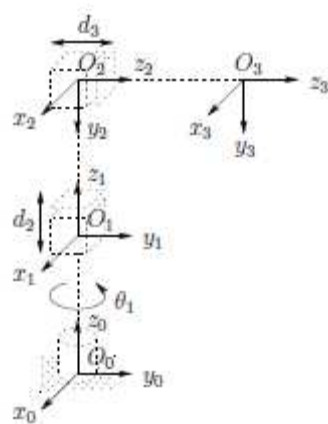


TABLA DE MOVIMIENTOS

	Rotación en Z	Traslación en Z	Traslación en X	Rotación en X
1	90	l1	0	-90
2	90	d1	0	90
3	90	d2	0	90
4	0	d3	0	0

CÁLCULO DE LAS MATRICES

Articulación #1

$$r_z = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(90^\circ) & -\sin(90^\circ) & 0 & 0 \\ \sin(90^\circ) & \cos(90^\circ) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$r_x = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\theta & -S\theta & 0 \\ 0 & S\theta & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-90^\circ) & -\sin(-90^\circ) & 0 \\ 0 & \sin(-90^\circ) & \cos(-90^\circ) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$t_z = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$${}^0A_1 = r_z \cdot r_x \cdot t_z = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Articulación #4

$$t_z = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$${}^3A_4 = t_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Obtención de la Matriz Homogénea

$$T = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} -1 & 0 & 0 & -d_1 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & -1 & -d_3 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix}$$

Finalmente:

$$x = -d_1$$

$$y = d_2$$

$$z = -d_3 + l_1$$

DESARROLLO E IMPLEMENTACIÓN

EN CODEWARRIOR DSP 56800-E

```
/** #####
**      Filename   : TP_Cinematica.C
**      Project    : TP_Cinematica
**      Processor   : 56F8367
**      Version     : Driver 01.14
**      Compiler    : Metrowerks DSP C Compiler
**      Date/Time   : 25/04/2010, 10:57 p.m.
**      Abstract    :
**          Main module.
**          This module contains user's application code.
**      Settings    :
**      Contents    :
**          No public methods
**
** #####*/
/* MODULE TP_Cinematica */

/* Including needed modules to compile this module/procedure */
#include "Cpu.h"
#include "Events.h"
#include "DFR1.h"
#include "MEM1.h"
#include "TFR1.h"
#include "MFR1.h"
#include "CCNT1.h"
#include "Inhr1.h"
/* Including shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

#include "stdio.h"

#define MXLONG 10
#define D1INICIAL 0
#define D2INICIAL 0
#define D3INICIAL 0
#define LARGOL1 1
#define PASONORMALIZADO 32767/(2*MXLONG)
Frac16 c,i;
Frac16 pasonormalizado=PASONORMALIZADO;
//Word16 c16[MXLONG];
//Word32 c32[MXLONG];

void main(void)
{
    /* Write your local variable definition here */

    Word16 d1_16,d2_16,d3_16,d1,d2,d3,x_16,y_16,z_16,l_16;
    Word32 d1_32,d2_32,d3_32,l_32;
```

```

l_32=L_mult(pasonormalizado,LARGOL1);
l_16=extract_l(l_32);

//UWord32 CycleCount;          /* Number of cycles          */

/** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! */
PE_low_level_init();
/** End of Processor Expert internal initialization.                      */

/* Write your code here */

/* Calibrate the cycle counter                                          */
//CCNT1_cycleCountCalibrate();

for(;;)
{

    d1=D1INICIAL;
    d2=D2INICIAL;
    d3=D3INICIAL;

    d1_32=(L_mult(pasonormalizado,d1));
    d1_16=extract_l(d1_32);
    x_16=-d1_16;

    d2_32=(L_mult(pasonormalizado,d2));
    d2_16=extract_l(d2_32);
    y_16=d2_16;

// Recorro el área límite sobre una de las superficies laterales

    for(d3=D3INICIAL;d3<MXLONG;d3++)
    {
        d3_32=(L_mult(pasonormalizado,d3));
        d3_16=extract_l(d3_32);
        z_16=add(-d3_16,l_16);
        printf ("%d \t %d \t %d \n",x_16,y_16,z_16);
    }
    d3--;

    for(d2=D2INICIAL;d2<MXLONG;d2++)
    {
        d2_32=(L_mult(pasonormalizado,d2));
        d2_16=extract_l(d2_32);
        y_16=d2_16;
        printf ("%d \t %d \t %d \n",x_16,y_16,z_16);
    }
    d2--;

    for(d3;d3>=D3INICIAL;d3--)
    {
        d3_32=(L_mult(pasonormalizado,d3));
        d3_16=extract_l(d3_32);
        z_16=add(-d3_16,l_16);
        printf ("%d \t %d \t %d \n",x_16,y_16,z_16);
    }

    for(d2;d2>=D2INICIAL;--d2)
    {

```

```

        d2_32=(L_mult(pasonormalizado,d2));
        d2_16=extract_1(d2_32);
        y_16=d2_16;
        printf ("%d \t %d \t %d \n",x_16,y_16,z_16);
    }

// Muevo el brazo hasta el límite sobre el eje x

    for(d1=D1INICIAL;d1<=MXLONG;++d1)
    {
        d1_32=(L_mult(pasonormalizado,d1));
        d1_16=extract_1(d1_32);
        x_16=-d1_16;
        printf ("%d \t %d \t %d \n",x_16,y_16,z_16);
    }

// Vuelvo a recorrer el área límite de esta superficie

    for(d3=D3INICIAL;d3<MXLONG;d3++)
    {
        d3_32=(L_mult(pasonormalizado,d3));
        d3_16=extract_1(d3_32);
        z_16=add(-d3_16,l_16);
        printf ("%d \t %d \t %d \n",x_16,y_16,z_16);
    }
    d3--;

    for(d2=D2INICIAL;d2<MXLONG;d2++)
    {
        d2_32=(L_mult(pasonormalizado,d2));
        d2_16=extract_1(d2_32);
        y_16=d2_16;
        printf ("%d \t %d \t %d \n",x_16,y_16,z_16);
    }
    d2--;

    for(d3;d3>=D3INICIAL;d3--)
    {
        d3_32=(L_mult(pasonormalizado,d3));
        d3_16=extract_1(d3_32);
        z_16=add(-d3_16,l_16);
        printf ("%d \t %d \t %d \n",x_16,y_16,z_16);
    }

    for(d2;d2>=D2INICIAL;--d2)
    {
        d2_32=(L_mult(pasonormalizado,d2));
        d2_16=extract_1(d2_32);
        y_16=d2_16;
        printf ("%d \t %d \t %d \n",x_16,y_16,z_16);
    }

/* Start the cycle-counter                                     */
//cycleCountStart();
//CCNT1_cycleCountStart();

//local_c32=(L_mult(pulse2rad,i));
//local_c16=extract_1(local_c32);

```

```

//testResult[i]=TFR1_tfri6SinPIx(local_c16);
/* Obtain and store the number of cycles */
//CycleCount = cycleCountStop();
//CycleCount=CCNT1_cycleCountStop();

/* Check the obtained value for validity
if (CycleCount == CYCLECOUNT_CALIBRATION_FAILURE) {
    printf ("Error calibracion");
}*/

//c32[i]= local_c32;
//c16[i]= local_c16;
//printf ("i=%d \t c16=%d \t Sen=%d \t \n",i,c16[i],testResult[i]);

//CCNT1_cycleCountReport();

}
}

/* END TP_Cinematica */
/*
** #####
**      This file was created by Processor Expert 3.00 [04.35]
**      for the Freescale 56800 series of microcontrollers.
** #####
**/

```

RESULTADOS DE LA SIMULACIÓN

GENERACIÓN DE PUNTOS

Eje X	Eje Y	Eje Z	Movimiento
0	0	3276	Incremento z
0	0	0	
0	0	-3276	
0	0	-6552	
0	0	-9828	
0	0	-13104	
0	0	-16380	
0	0	-19656	
0	0	-22932	
0	0	-26208	
0	0	-26208	Incremento y
0	3276	-26208	
0	6552	-26208	
0	9828	-26208	
0	13104	-26208	
0	16380	-26208	
0	19656	-26208	
0	22932	-26208	
0	26208	-26208	
0	29484	-26208	
0	29484	-26208	Decremento z
0	29484	-22932	
0	29484	-19656	
0	29484	-16380	
0	29484	-13104	
0	29484	-9828	
0	29484	-6552	
0	29484	-3276	
0	29484	0	
0	29484	3276	
0	29484	3276	Decremento y
0	26208	3276	
0	22932	3276	
0	19656	3276	
0	16380	3276	
0	13104	3276	
0	9828	3276	
0	6552	3276	

0	3276	3276	
0	0	3276	
0	0	3276	Incremento x
-3276	0	3276	
-6552	0	3276	
-9828	0	3276	
-13104	0	3276	
-16380	0	3276	
-19656	0	3276	
-22932	0	3276	
-26208	0	3276	
-29484	0	3276	
-32760	0	3276	
-32760	0	3276	Incremento z
-32760	0	0	
-32760	0	-3276	
-32760	0	-6552	
-32760	0	-9828	
-32760	0	-13104	
-32760	0	-16380	
-32760	0	-19656	
-32760	0	-22932	
-32760	0	-26208	
-32760	0	-26208	Incremento y
-32760	3276	-26208	
-32760	6552	-26208	
-32760	9828	-26208	
-32760	13104	-26208	
-32760	16380	-26208	
-32760	19656	-26208	
-32760	22932	-26208	
-32760	26208	-26208	
-32760	29484	-26208	
-32760	29484	-26208	Decremento z
-32760	29484	-22932	
-32760	29484	-19656	
-32760	29484	-16380	
-32760	29484	-13104	
-32760	29484	-9828	
-32760	29484	-6552	
-32760	29484	-3276	
-32760	29484	0	
-32760	29484	3276	
-32760	29484	3276	Decremento y
-32760	26208	3276	
-32760	22932	3276	
-32760	19656	3276	
-32760	16380	3276	
-32760	13104	3276	

-32760	9828	3276	
-32760	6552	3276	
-32760	3276	3276	
-32760	0	3276	
-32760	0	3276	Incremento z
-32760	0	0	
-32760	0	-3276	
-32760	0	-6552	
-32760	0	-9828	
-32760	0	-13104	
-32760	0	-16380	
-32760	0	-19656	
-32760	0	-22932	
-32760	0	-26208	
-32760	0	-26208	Decremento x
-29484	0	-26208	
-26208	0	-26208	
-22932	0	-26208	
-19656	0	-26208	
-16380	0	-26208	
-13104	0	-26208	
-9828	0	-26208	
-6552	0	-26208	
-3276	0	-26208	
0	0	-26208	Incremento y
0	0	-26208	
0	3276	-26208	
0	6552	-26208	
0	9828	-26208	
0	13104	-26208	
0	16380	-26208	
0	19656	-26208	
0	22932	-26208	
0	26208	-26208	
0	29484	-26208	Incremento x
0	29484	-26208	
-3276	29484	-26208	
-6552	29484	-26208	
-9828	29484	-26208	
-13104	29484	-26208	
-16380	29484	-26208	
-19656	29484	-26208	
-22932	29484	-26208	
-26208	29484	-26208	
-29484	29484	-26208	
-32760	29484	-26208	
-32760	29484	-26208	Decremento z
-32760	29484	-26208	
-32760	29484	-22932	

-32760	29484	-19656	
-32760	29484	-16380	
-32760	29484	-13104	
-32760	29484	-9828	
-32760	29484	-6552	
-32760	29484	-3276	
-32760	29484	0	
-32760	29484	3276	
-32760	29484	3276	Decremento x
-29484	29484	3276	
-26208	29484	3276	
-22932	29484	3276	
-19656	29484	3276	
-16380	29484	3276	
-13104	29484	3276	
-9828	29484	3276	
-6552	29484	3276	
-3276	29484	3276	
0	29484	3276	
0	29484	3276	Decremento y
0	26208	3276	
0	22932	3276	
0	19656	3276	
0	16380	3276	
0	13104	3276	
0	9828	3276	
0	6552	3276	
0	3276	3276	
0	0	3276	

Nota: Los valores generados por el CodeWarrior responden al formato de notación fraccional.

GRÁFICA DEL ESPACIO DE TRABAJO

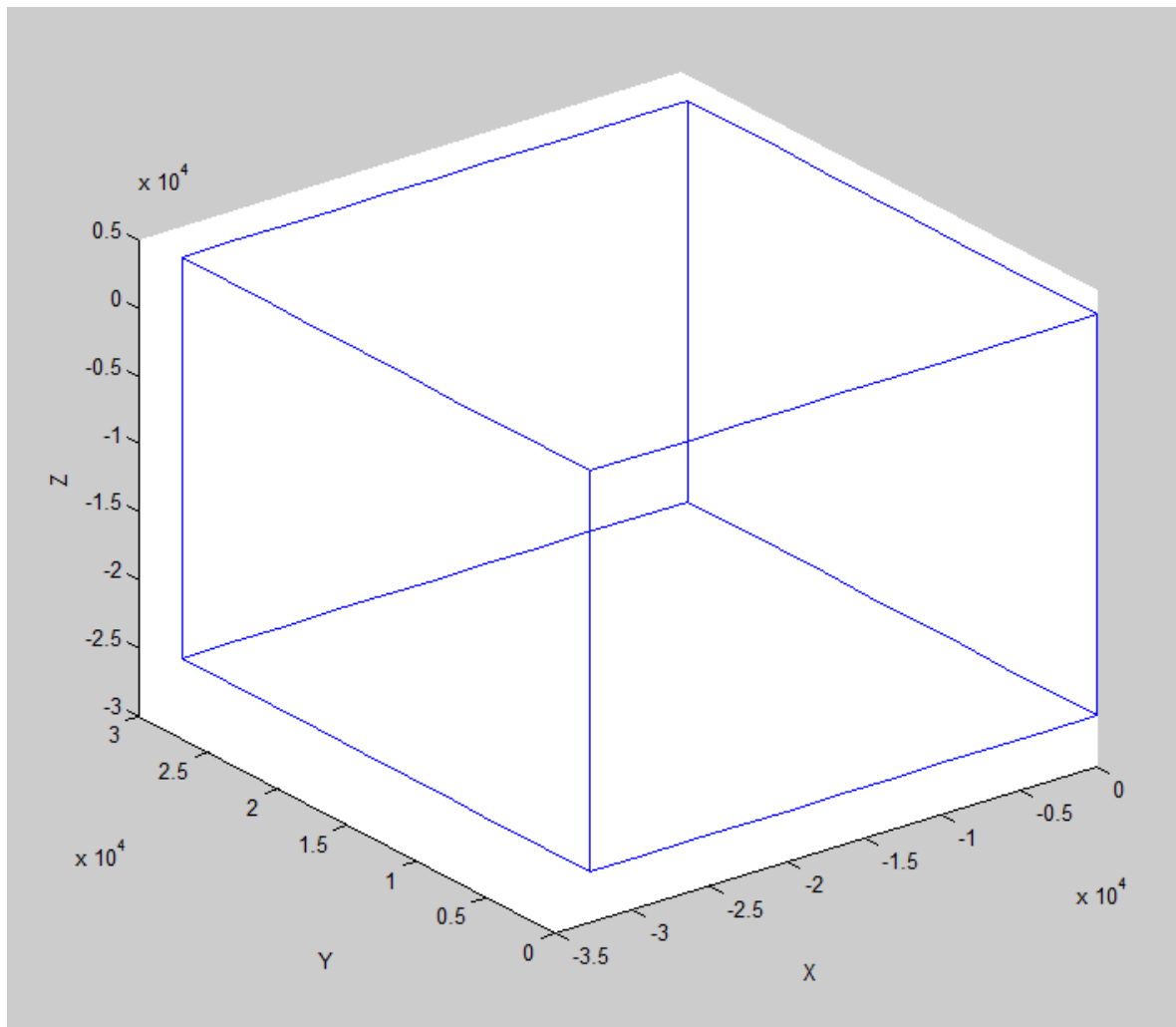


Figura 1

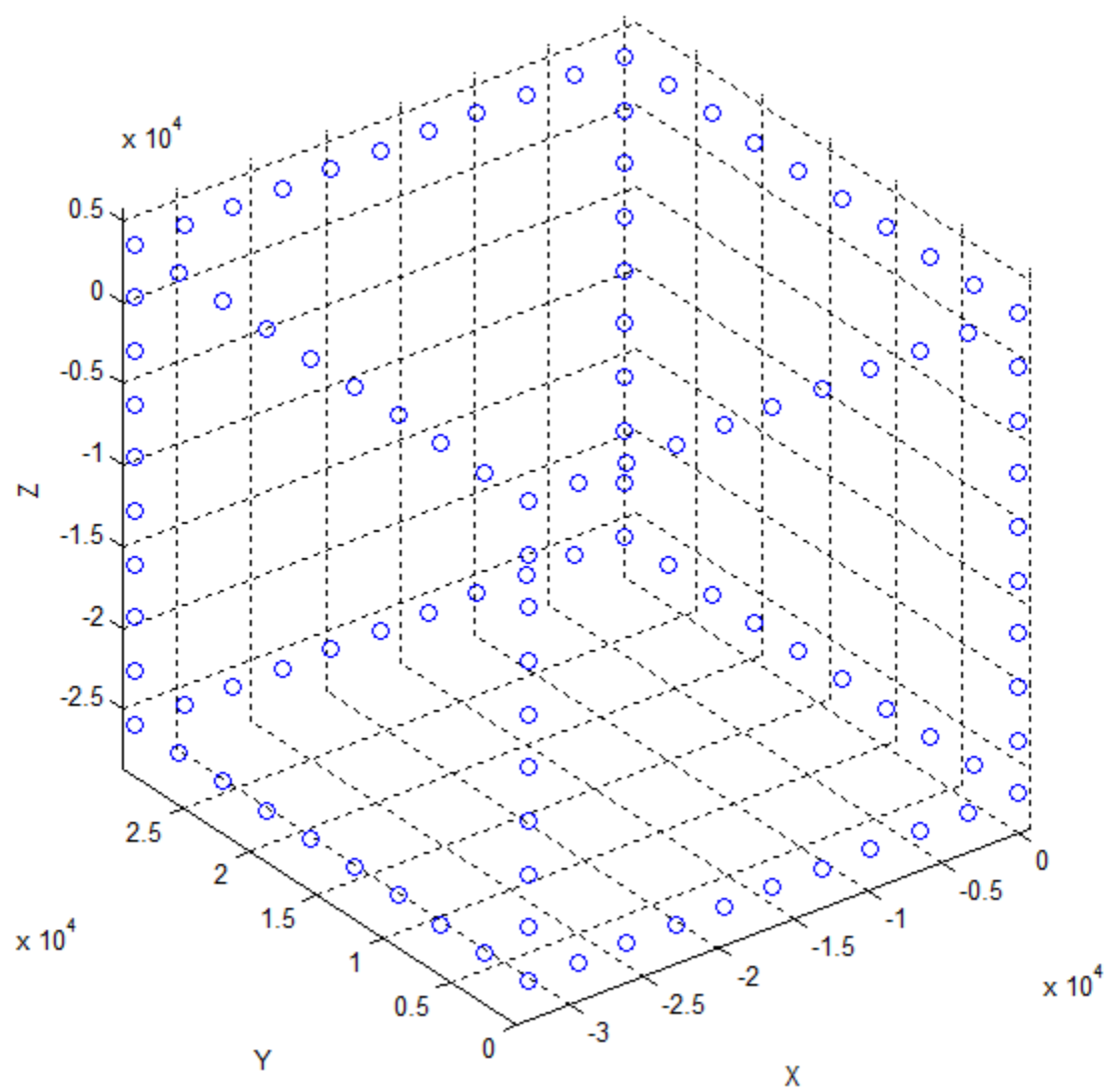


Figura 2

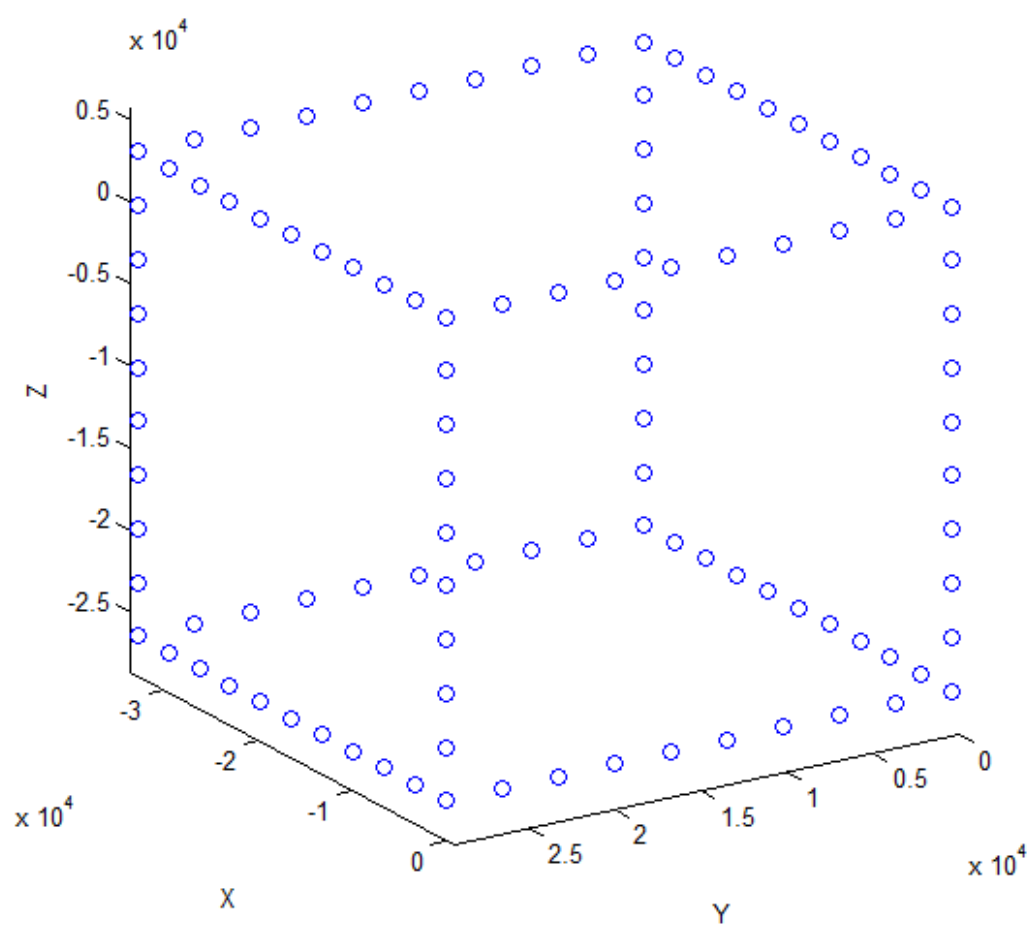


Figura 3

CONCLUSIONES

La realización de este trabajo práctico nos permitió verificar la simulación de la cinemática de un brazo robótico.

Teniendo en cuenta que partimos de un brazo que constaba de tres articulaciones lineales, resulta lógica la gráfica obtenida como espacio de trabajo del mismo. En la gráfica podemos observar que llevando el brazo hasta sus movimientos límites se dibuja un volumen con forma prismática. Las dimensiones de dicho volumen se podrían modificar poniendo otros valores como máximos para cada eje, congruente siempre con la física del brazo robótico. Como caso particular, vemos que el área de trabajo es un cubo.

Para el desarrollo de esta práctica utilizamos herramientas tales como CodeWarrior para generar el código fuente y Matlab para realizar los gráficos. Para dichos softwares utilizamos herramientas que fueron de gran utilidad, y en algunos casos eran desarrolladas por primera vez, según nuestra experiencia.

Cuando compilamos el código por primera vez y realizamos la gráfica correspondiente, verificamos que habíamos calculado en forma errónea los valores límite, ya que el dibujo mostraba trayectorias no deseadas y que no correspondían con nuestras directivas. Es decir, físicamente estábamos destruyendo el robot, ya que lo haríamos ir a puntos del espacio a los cuales le es imposible acceder.

Una vez solucionado esto, delimitamos en forma correcta los movimientos de cada eje y obtenemos como resultado el gráfico de la figura 1. El mismo muestra las trayectorias lineales límite de cada dirección, formando un volumen cúbico.

Luego, volvimos a repetir dicha gráfica, pero esta vez en vez de utilizar la función plot3 del Matlab (que extrapola los puntos del gráfico), utilizamos otra que dibuja solo los puntos generados por la simulación. Esto se ve en las figuras 2 y 3. De esta forma podemos observar los puntos del espacio a los cuales es capaz de acceder nuestro brazo robótico. Dos puntos contiguos determinan la resolución del mismo. Esta es una característica de la tecnología y la forma constructiva del robot.