UTN Facultad Regional Buenos Aires

Robótica

Trabajo Práctico Nº 1:

Implementación de una matriz cinemática en DSP

Profesor: Mas. Ing Hernán Giannetta

Alumnos:

Sebastián Lopreto
Pablo Moras
Jonathan Rubstein

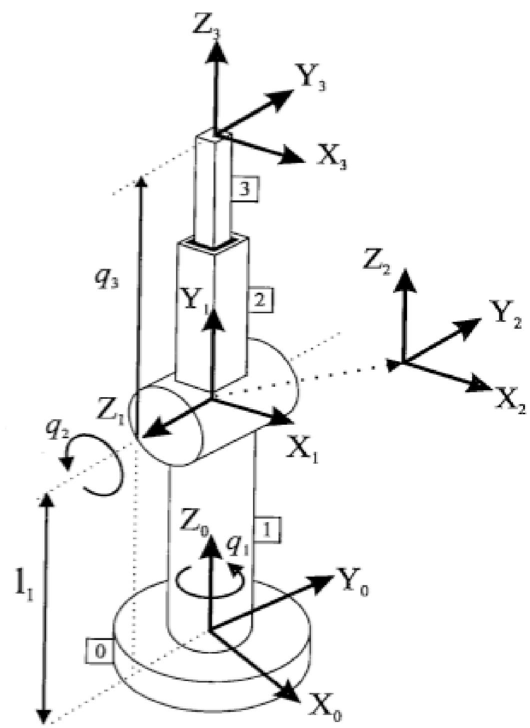
Curso: R6055

Fecha de entrega: 8 de junio de 2009

Enunciado

Implemente el código C en CW para el DSP56800/E de la cadena cinemática directa de la figura, usando la matriz homogénea, y 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) utilizando la función plot3(x,y,z) de Matlab.



Articulación	θ	d	a	α
1	q1	L1	0	90
2	q2	0	0	-90
3	0	q3	0	0

Introducción

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia sin considerar las fuerzas que intervienen. 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 a resolver en la cinemática del robot; el primero de ellos se conoce como el problema cinemático directo, y consiste en determinar cuál 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.

En este trabajo práctico, vamos a utilizar la cinemática directa para encontrar la posición en cada instante del extremo manipulador del robot. A continuación desarrollamos los distintos métodos de resolución.

Problema cinemático directo

Como hemos dicho en el párrafo anterior, la resolución del problema cinemático directo permite conocer cuál es la posición y orientación que adopta el extremo del robot cuando cada una de las variables que fijan la posición u orientación de sus articulaciones toma valores determinados.

Dado que son las variables articulares las que pueden ser leídas directamente de los correspondientes sensores por la unidad de control de robot, el modelo cinemático directo será utilizado por éste, entre otros fines, para presentar al usuario información relativa a la localización del extremo del robot.

Así, 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:

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

La obtención del modelo cinemático directo puede ser abordado mediante dos enfoques diferentes denominados métodos geométricos y métodos basados en cambios de sistemas de referencia.

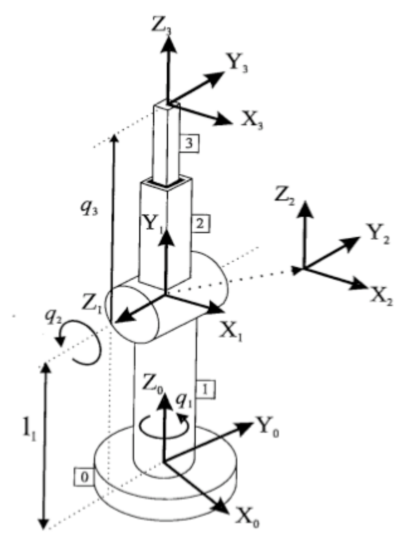
Los primeros son adecuados para casos simples, pero al no ser sistemáticos, su aplicación queda limitada a robots con pocos grados de libertad. Los métodos basados en cambios de sistemas de referencia, permiten de una manera sistemática abordar la obtención del modelo cinemático directo del robot para robots de n grados de libertad, siendo éstos, por tanto, los más frecuentemente utilizados, en particular los que usan las matrices de transformación homogénea.

<u>Resolución del problema cinemático directo mediante matrices de transformación homogénea</u>

En este método 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 de transformación homogénea 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.

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

El enunciado establece que el robot que debemos utilizar es el siguiente:



Vamos a calcular las matrices de transformación homogénea para todas las articulaciones.

$$ {}^{0}A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 & 0 \\ \sin q_1 & \cos q_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 & 0 \\ \sin q_1 & 0 & -\cos q_1 & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}A_2 = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & 0 \\ \sin q_2 & \cos q_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos q_2 & 0 & -\sin q_2 & 0 \\ \sin q_2 & 0 & \cos q_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Finalmente, debemos obtener:

$$ {}^{0}A_3 = {}^{0}A_1 \cdot {}^{1}A_2 \cdot {}^{2}A_3 $$

$$ {}^{0}A_3 = \begin{bmatrix} \cos q_1 \cdot \cos q_2 & -\sin q_1 & -\cos q_1 \cdot \sin q_2 & -q_3 \cdot \cos q_1 \cdot \sin q_2 \\ \sin q_1 \cdot \cos q_2 & \cos q_1 & -\sin q_1 \cdot \sin q_2 & q_3 \cdot \sin q_1 \cdot \sin q_2 \\ \sin q_2 & 0 & \cos q_2 & q_3 \cdot \cos q_2 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Para transformar un determinado punto expresado en el sistema de referencia de la última articulación, al mismo punto expresado en el sistema de referencia de la base, debemos hacer lo siguiente:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos q_1 \cdot \cos q_2 & -\sin q_1 & -\cos q_1 \cdot \sin q_2 & -q_3 \cdot \cos q_1 \cdot \sin q_2 \\ \sin q_1 \cdot \cos q_2 & \cos q_1 & -\sin q_1 \cdot \sin q_2 & q_3 \cdot \sin q_1 \cdot \sin q_2 \\ \sin q_2 & 0 & \cos q_2 & q_3 \cdot \cos q_2 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Por lo tanto, si queremos conocer las coordenadas del extremo del robot respecto al sistema de referencia de la base:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos q_1 \cdot \cos q_2 & -\sin q_1 & -\cos q_1 \cdot \sin q_2 & -q_3 \cdot \cos q_1 \cdot \sin q_2 \\ \sin q_1 \cdot \cos q_2 & \cos q_1 & -\sin q_1 \cdot \sin q_2 & q_3 \cdot \sin q_1 \cdot \sin q_2 \\ \sin q_2 & 0 & \cos q_2 & q_3 \cdot \cos q_2 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -q_3 \cdot \cos q_1 \cdot \sin q_2 \\ q_3 \cdot \sin q_1 \cdot \sin q_2 \\ q_3 \cdot \cos q_2 + L_1 \\ 1 \end{bmatrix}$$

Finalmente, sabemos que las coordenadas del extremo del robot respecto al sistema de referencia de la base son:

$$\begin{aligned} x &= -q_3 \cdot \cos q_1 \cdot \sin q_2 \\ y &= q_3 \cdot \sin q_1 \cdot \sin q_2 \\ z &= q_3 \cdot \cos q_2 + L_1 \end{aligned}$$

Para este trabajo práctico, vamos a variar los parámetros de las articulaciones para generar un movimiento en el extremo del robot y así poder obtener algunas conclusiones.

Programación

Para poder realizar la prueba, generamos un programa en C utilizando el CodeWarrior, el cual nos ofrece una plataforma de trabajo sobre el DSP56800/E. En dicho programa, fuimos variando los parámetros de las articulaciones para generar todos los casos posibles y analizar el comportamiento del robot.

Para simplificar el análisis, decidimos usar para los valores de q_1, q_2 y q_3 el tipo de dato frac16, la cual es la notación utilizada para los puntos flotantes en este tipo de microprocesadores.

Por simplicidad, a continuación transcribimos el código genérico utilizado. Se debe tener en cuenta las restricciones de variabilidad de los parámetros para cada caso.

```c
/**###################################################################
**     Filename  : TP1.C
**     Project   : TP1
**     Processor : 56F8367
**     Version   : Driver 01.13
**     Compiler  : Metrowerks DSP C Compiler
**     Date/Time : 24/05/2009, 10:38
**     Abstract  :
**         Main module.
**         This module contains user's application code.
**     Settings  :
**     Contents  :
**         No public methods
**
**     (c) Copyright UNIS, a.s. 1997-2008
**     UNIS, a.s.
**     Jundrovska 33
**     624 00 Brno
**     Czech Republic
**     http      : www.processorexpert.com
**     mail      : info@processorexpert.com
**
###################################################################*/
/* MODULE TP1 */

/* Including needed modules to compile this module/procedure */
#include "Cpu.h"
#include "Events.h"
#include "TFR1.h"
#include "MFR1.h"
#include "MEM1.h"
#include "XFR1.h"
#include "AFR1.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"

/* Librerías extra */
#include "stdio.h"

//*********************************************************************
//        DEFINES
//*********************************************************************

#define MXRAD 100
#define PULSE2RAD 32767/MXRAD
```

```
#define LIMIT 50   /* Valor que define el rango de trabajo de las variables*/
#define L1 0

//*****************************************************************
//        GLOBAL VARIABLES
//*****************************************************************

Frac16 i,k;
Frac16 pulse2rad = PULSE2RAD, sin, cos;
Frac16 resultado [LIMIT] [3];
Word16 c16_art, c16_aux;
Word32 c32_art;

//*****************************************************************
//        PROGRAM CODE
//*****************************************************************

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

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

 /* Write your code here */

 for(i=0,k=0;i<LIMIT;i++) {

         /* Calculo los valores del seno y el coseno correspondientes */

                         c32_art = (L_mult(pulse2rad,i));
                         c16_art = extract_l(c32_art);
                         sin = TFR1_tfr16SinPIx(c16_art);
                         cos = TFR1_tfr16CosPIx(c16_art);
                         k = c16_art;

                         /* Calculo de la coordenada x */

                         c16_aux = mult(cos,sin);
                         c16_aux = mult(c16_aux,k);
                         c16_aux = negate(c16_aux);
                         resultado [i][0] = c16_aux;

                         /* Calculo de la coordenada y */

                         c16_aux = mult(sin,sin);
                         c16_aux = mult(c16_aux,k);
                         resultado [i][1] = c16_aux;
```

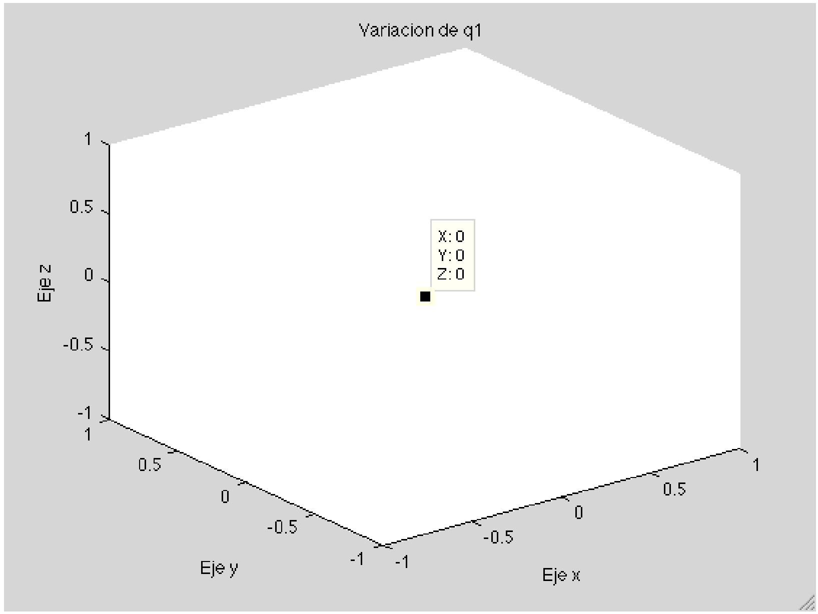
```
                /* Calculo de la coordenada z */

                c16_aux = mult(cos,k);
                c16_aux = add(c16_aux,L1);
                resultado [i][2] = c16_aux;

                /* Muestro el resultado por pantalla */

    printf ("%d,%d,%d;",resultado[i][0],resultado[i][1],resultado[i][2]);

  }
}

/* END TP1 */
/*
**
#######################################################
**
**     This file was created by UNIS Processor Expert 2.99 [04.17]
**     for the Freescale 56800 series of microcontrollers.
**
**
#######################################################
*/
```

Código para Matlab

```
/* Se carga la matriz con los puntos calculados con el Codewarrior */

a=[x0,y0,z0;x1,y1,z1;x2,y2,z2;....];

/* Se gráfica la matriz */

figure();plot3(a(:,1),a(:,2),a(:,3));
```

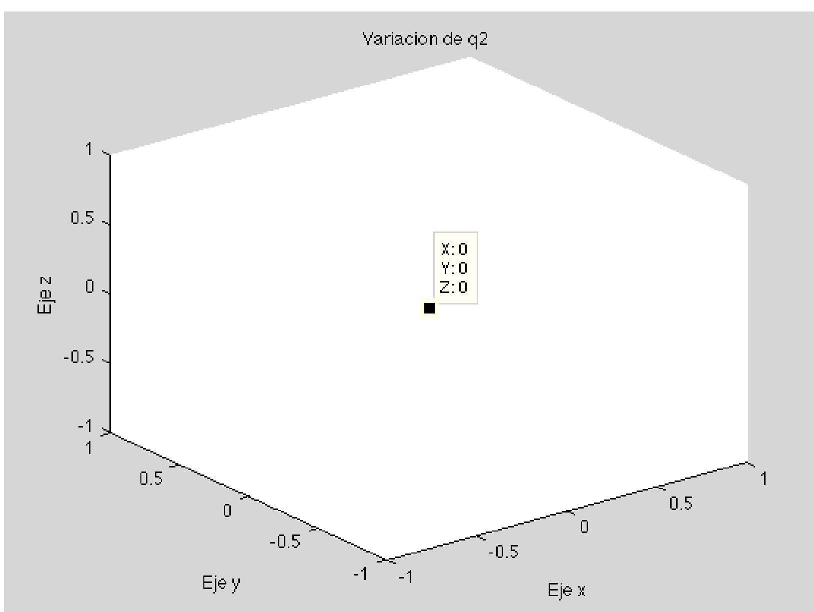
Gráficos

- Variación de q_1 manteniendo q_2=0 y q_3=0



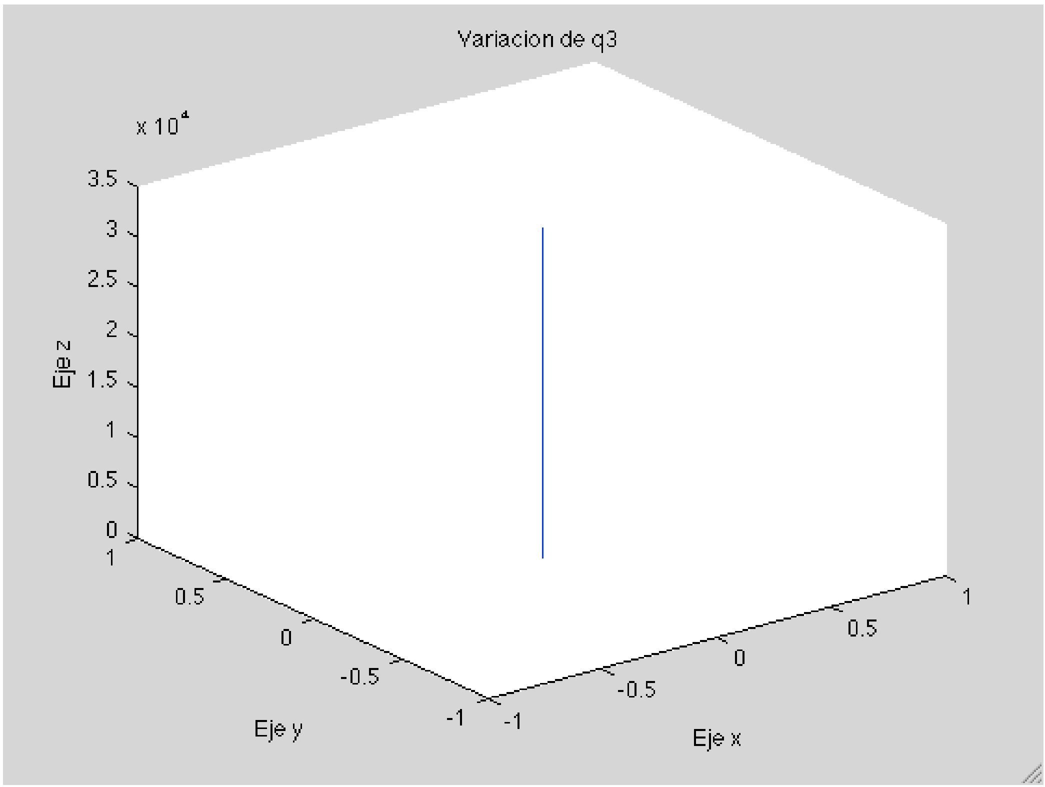
En este caso, q_1 puede variar en todo el rango (0 a 360º), para observar el movimiento total. También, por simplicidad, elegimos L_1 igual a cero.
Como resultado, vemos que el extremo del robot permanece siempre en su posición. Esto se debe a que la articulación 1 gira sobre si misma, y al hecho de que no hay inclinación ni longitud de brazo.

- Variación de q_2 manteniendo q_1=0 y q_3=0



Como en el caso anterior, q_2 también puede variar en todo el rango (0 a 360º). Se observa que como la rotación y la longitud del brazo son nulos, entonces el extremo del brazo queda fijo en un punto.

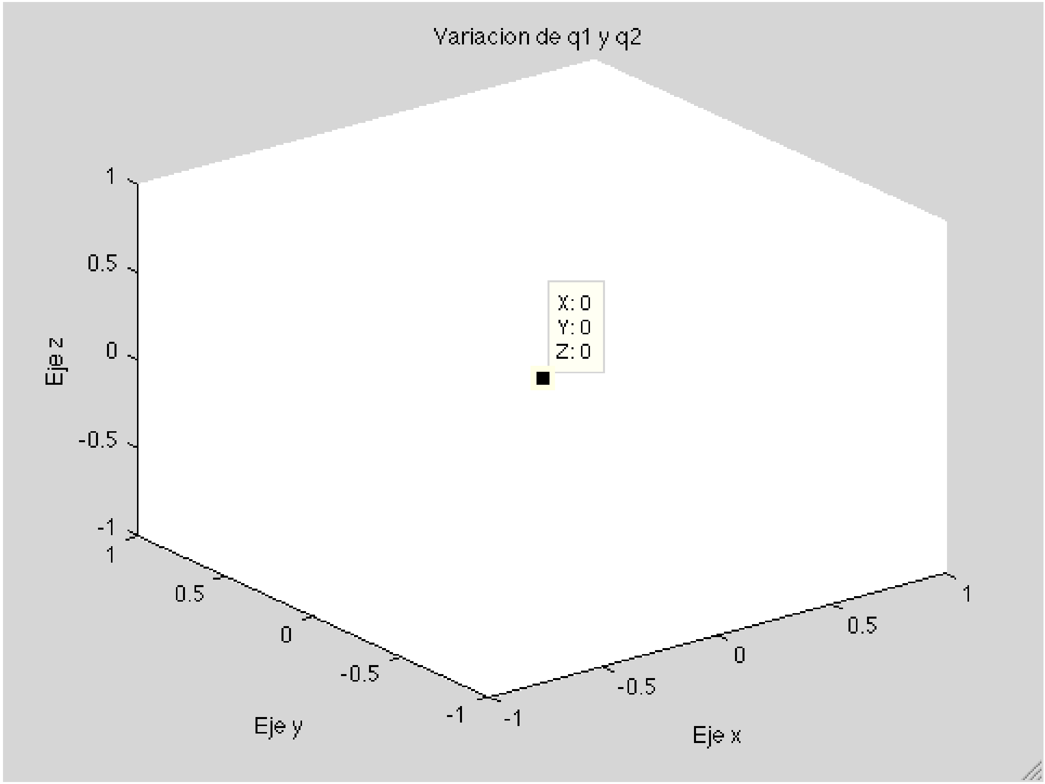
- Variación de q_3 manteniendo q_1=0 y q_2=0



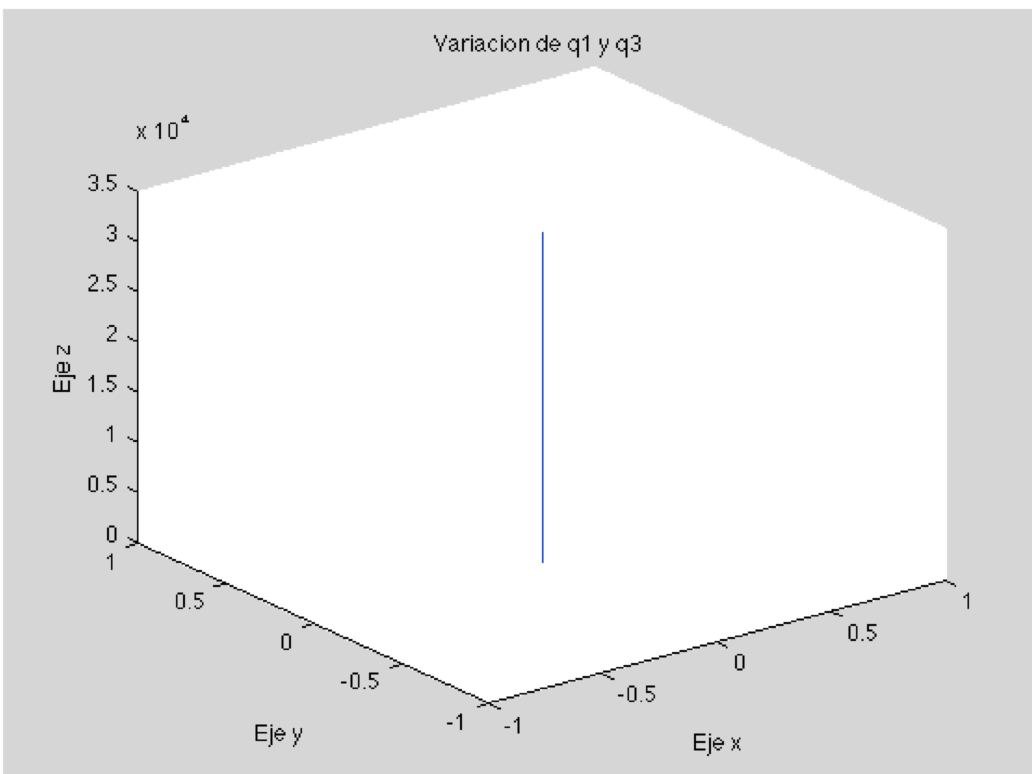
En este caso, q_3 está limitado solamente a los valores positivos del rango (0 a 32767). Como vemos, podemos obtener una trayectoria vertical del extremo del brazo aún, cuando no hay ni rotación ni inclinación. Hay que tener cuidado con los valores negativos del rango, ya que significaría un desplazamiento hacia abajo más allá de lo permitido.

- Variación de q_1 y q_2 manteniendo q_3=0

Como hemos visto anteriormente, al no haber un desplazamiento en la longitud del brazo, no se produce ninguna modificación en la localización del extremo del robot. Por lo tanto, el gráfico que mostramos a continuación es igual a los dos primeros casos:

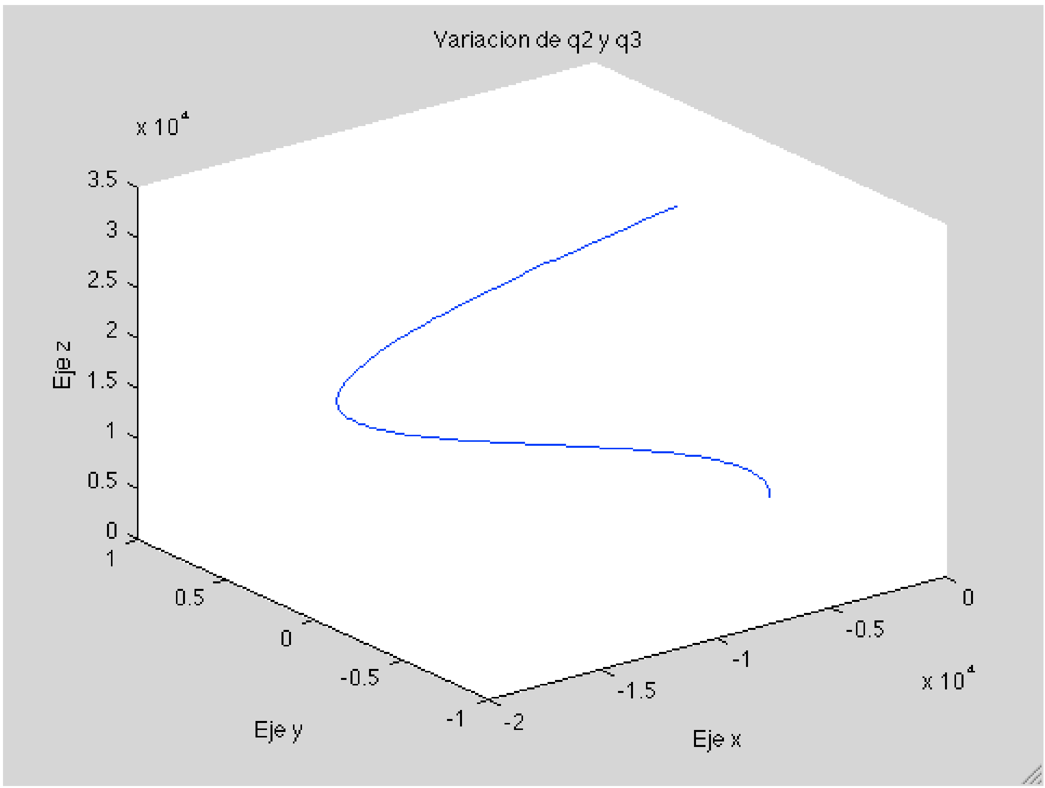


- Variación de q_1 y q_3 manteniendo q_2=0



En este caso, al haber un incremento en la longitud del brazo y en la rotación, se genera un desplazamiento del extremo del robot. Se tuvo solamente en cuenta el incremento positivo de la longitud del brazo.

- Variación de q_2 y q_3 manteniendo $q_1 = 0$



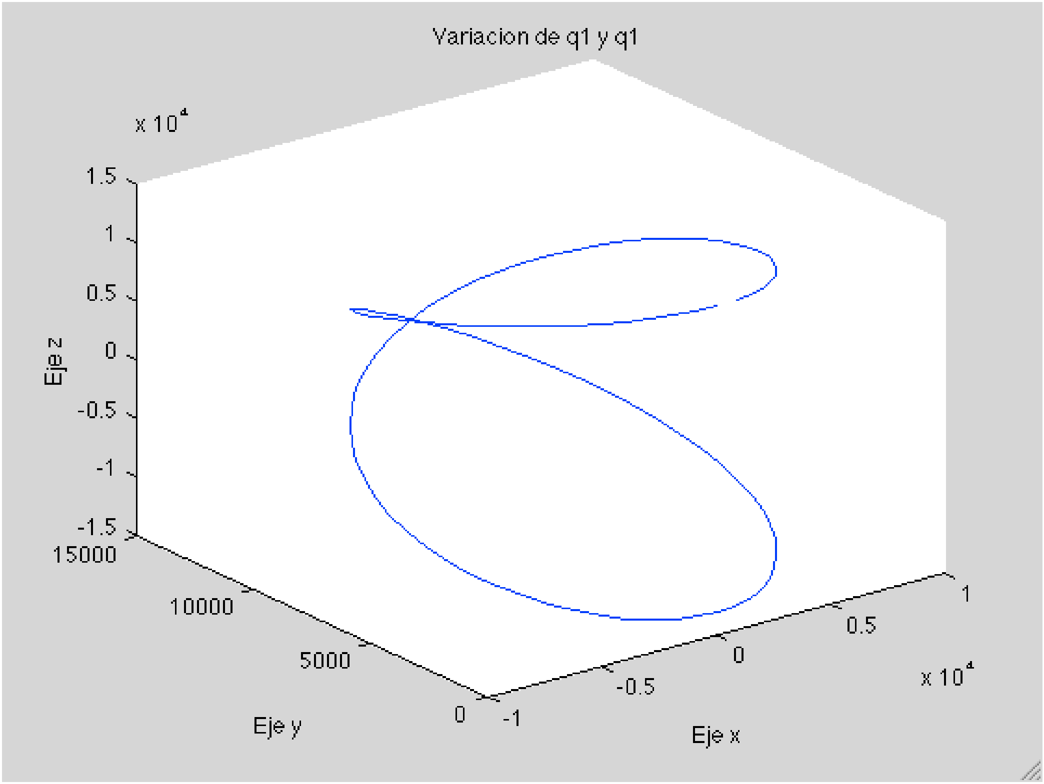
Para este caso, observamos que el extremo tiene una trayectoria elíptica. Se limitó la variación el movimiento solamente para los valores positivos de z.

- Variación de q_1 y q_2 manteniendo q_3=cte

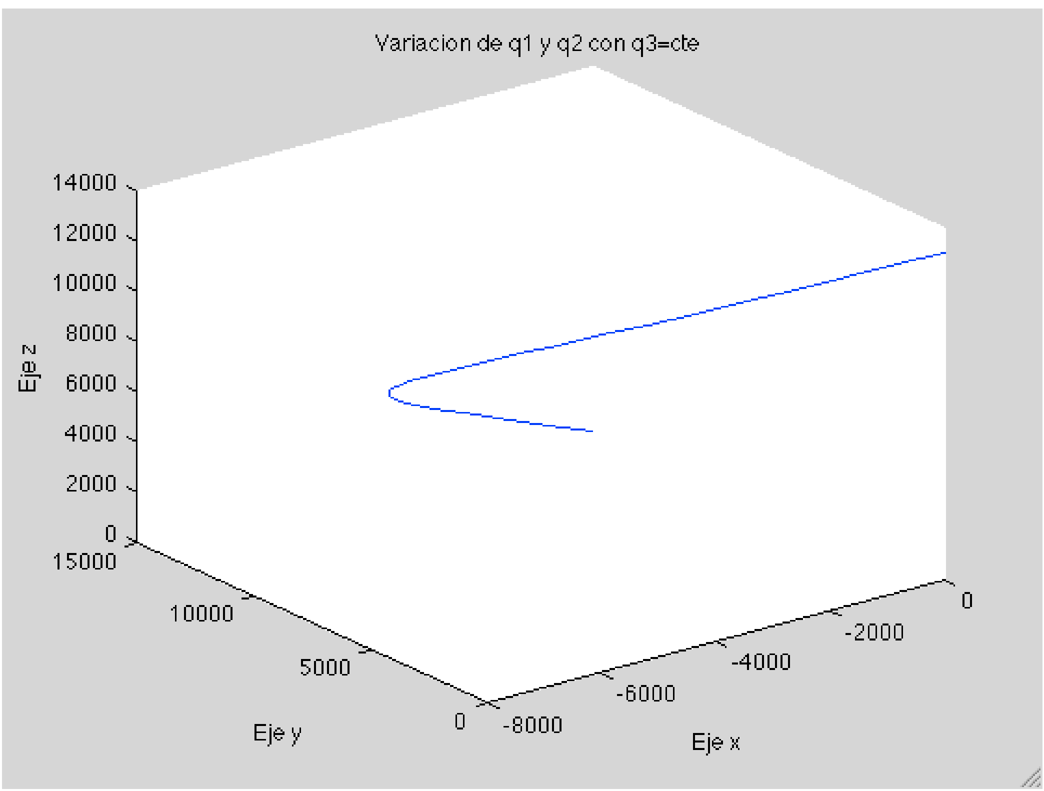
Para este caso, vemos como es un movimiento completo de las tres articulaciones y podemos empezar a observar el área de movimiento del robot.

En la primera figura, observamos la variación en todo el rango posible, mientras que en la segunda se limitó a los valores positivos de z, o sea, el primer cuadrante.

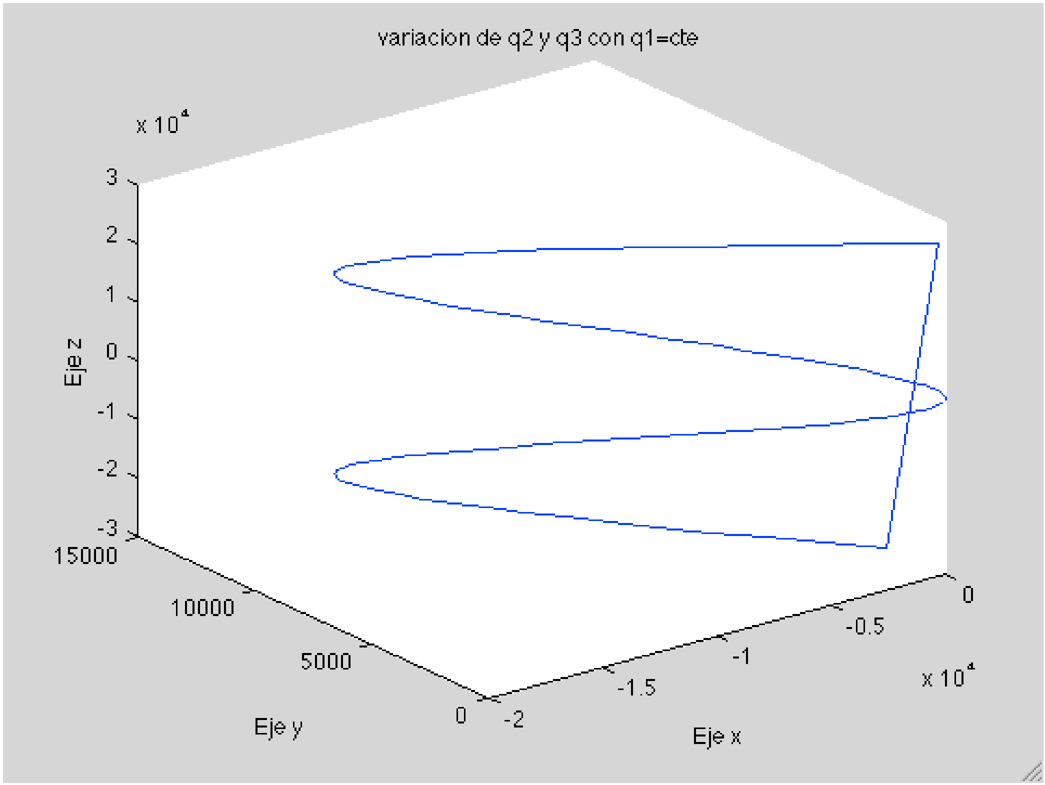
Rango completo:



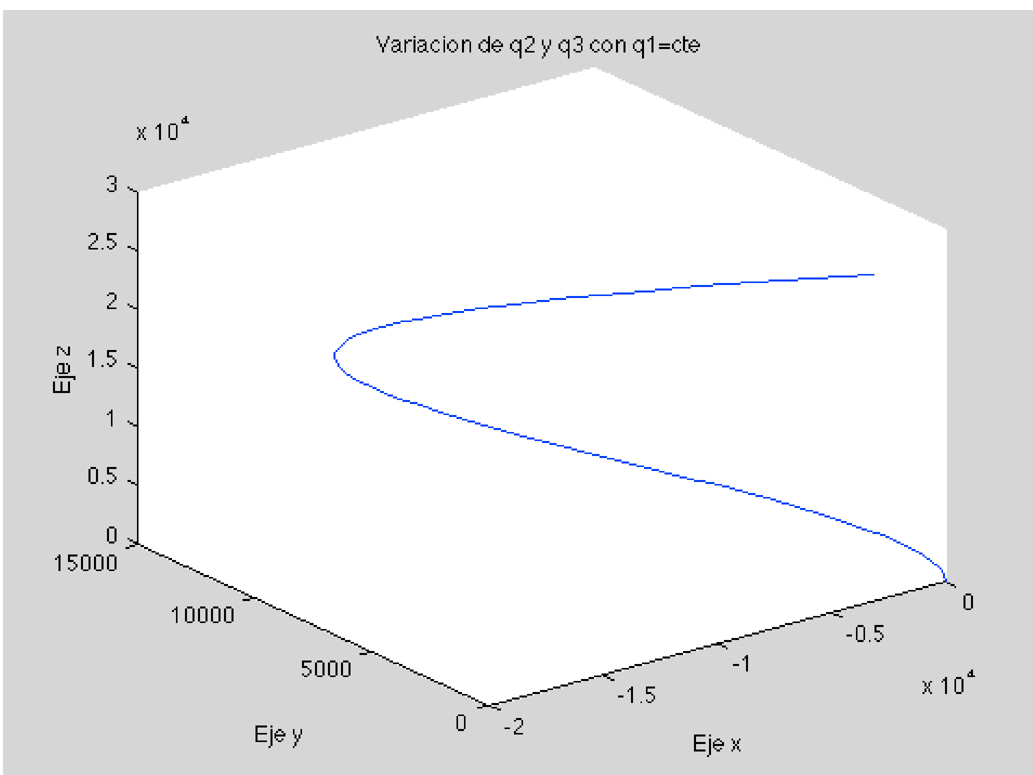
Limitado a los valores positivos de z:



- Variación de q_2 y q_3 manteniendo q_1=cte

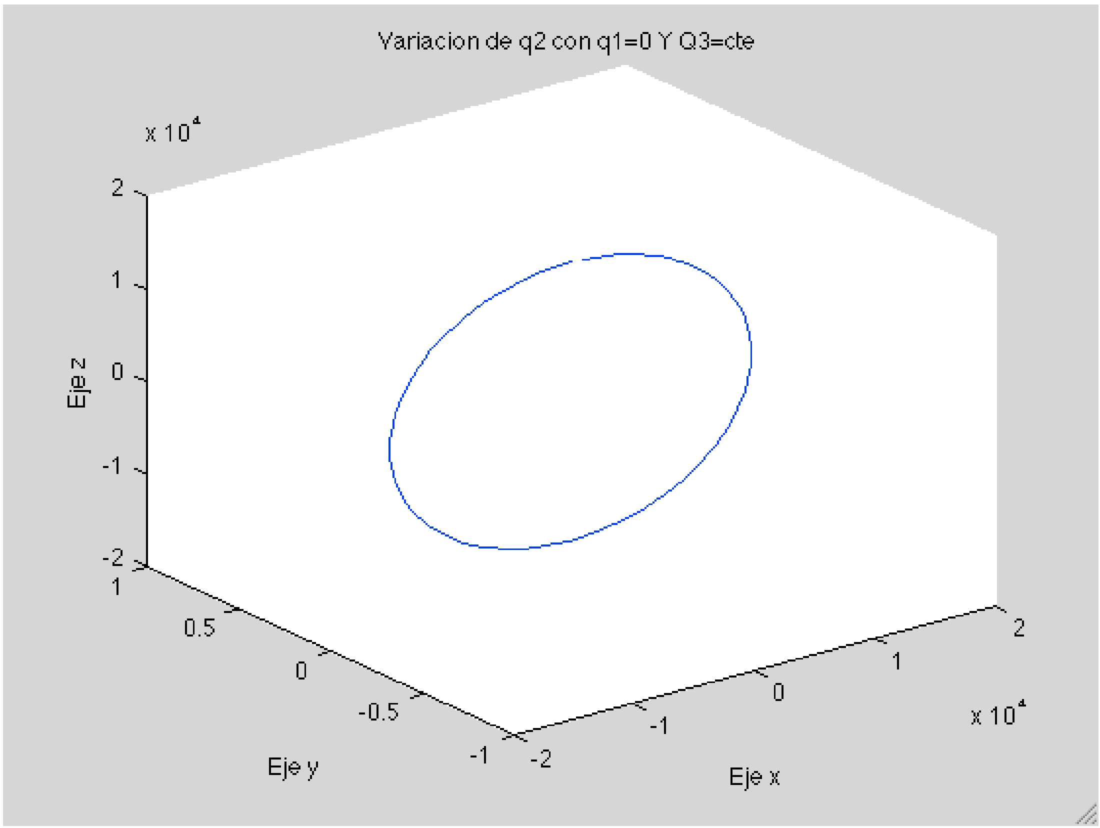


Limitado a valores positivos de z:



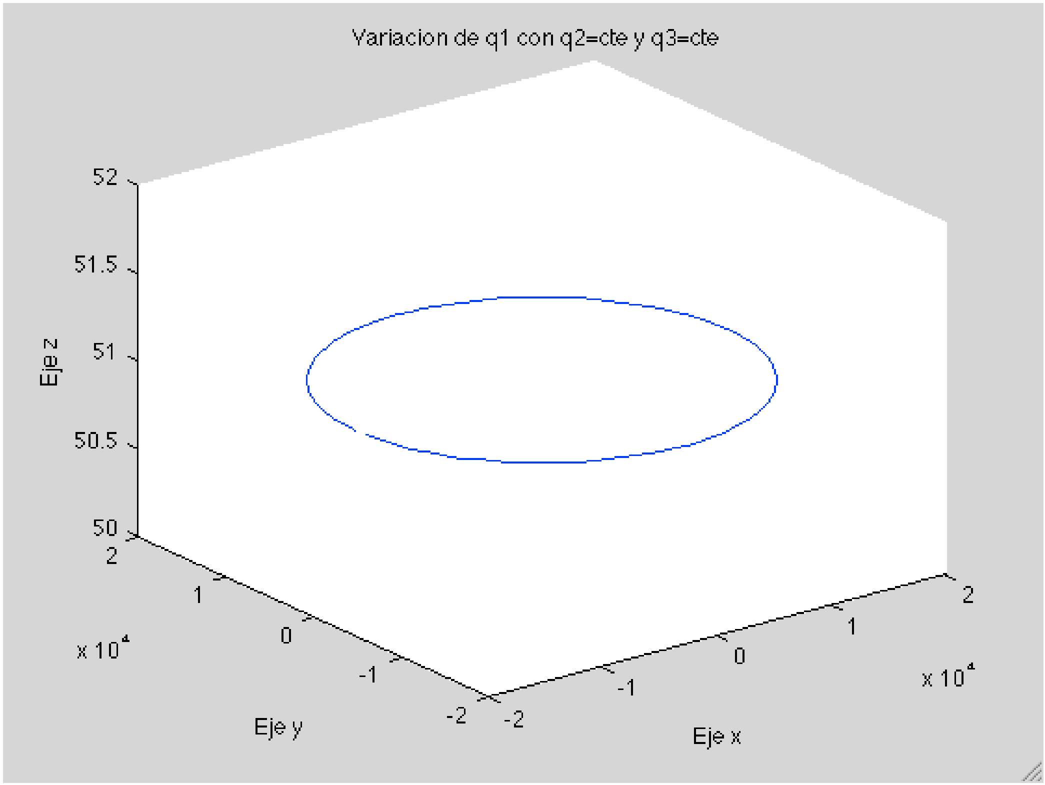
En los próximos dos gráficos, queremos estudiar cuál sería el área máxima de trabajo del robot, para lo cual hacemos:

1. Variamos q_2, con q_1=0 y q_3=cte



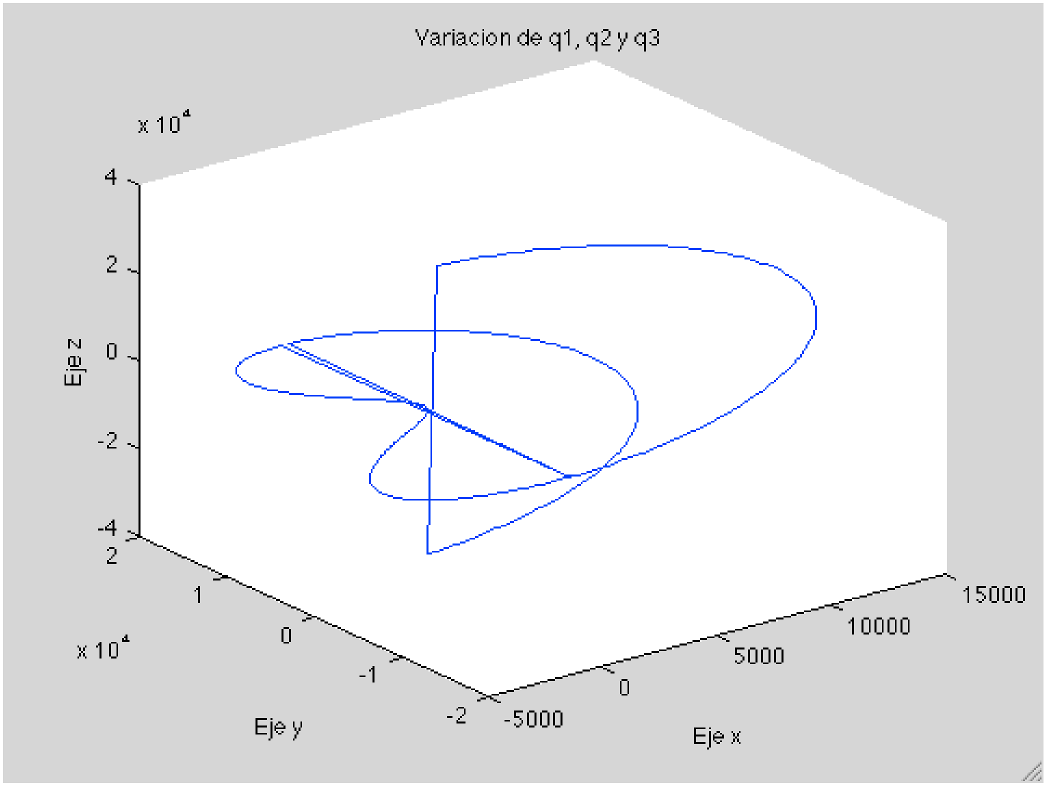
Podemos ver que la periferia del círculo es la distancia máxima a la que puede llegar el brazo del robot. Hay que considerar, por supuesto, los valores posibles de z (en los casos anteriores, dijimos que solo tomábamos los valores de z positivos, obteniendo solo medio círculo).

2. Variamos q_1 con q_2=cte y q_3=cte

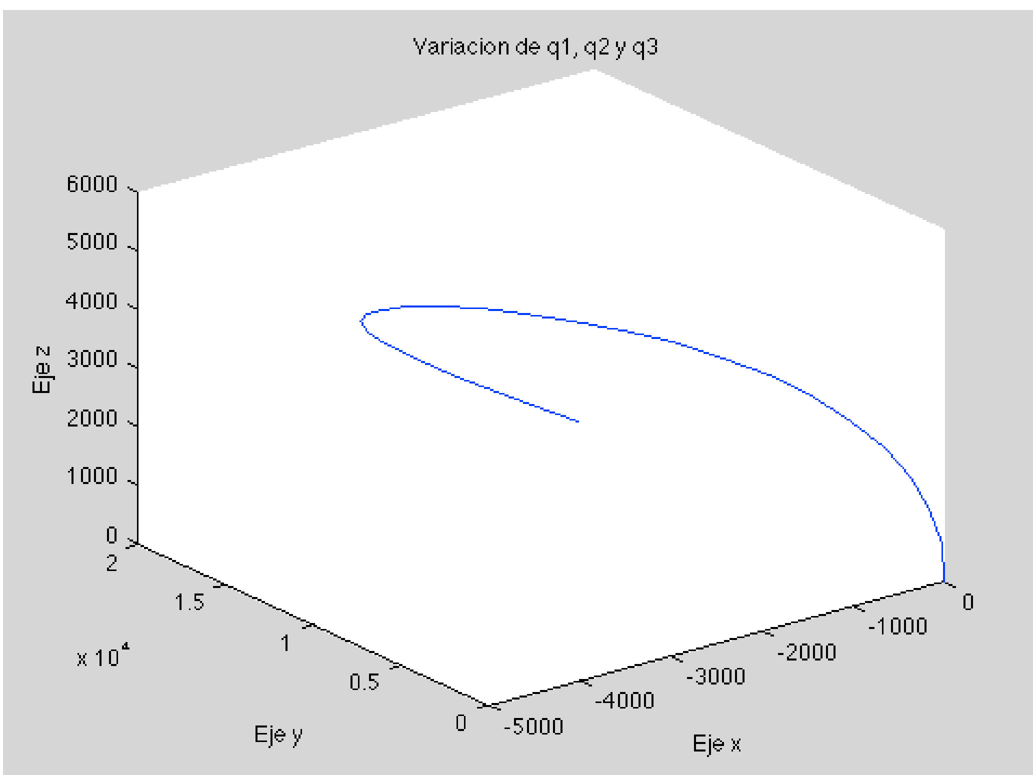


Nuevamente, vemos el área de máxima de trabajo del brazo del robot, en este caso horizontal. En este caso, se debe restringir a la rotación máxima permitida por el robot, por ejemplo 90º, con lo que se obtendría solamente un cuarto de círculo en el primer cuadrante.

- Variación de q_1, q_2 y q_3 – Incremento lineal para los tres movimientos



Limitado a valores de z positivo



Conclusiones

Este trabajo práctico nos resultó de gran ayuda para la comprensión e incorporación de la teoría cinemática para la robótica. La utilización de las matrices de transformación homogéneas son de gran ayuda para el cálculo de la posición y orientación de las articulaciones de los robots, y son un método simple y sistemático para utilización en robots con una cantidad de grados de libertad considerables.

La utilización del CodeWarrior fue una herramienta excelente, ya que nos permite desarrollar y simular virtualmente nuestros programas sobre un DSP. Una de las ventajas también importantes es que viene con un set de funciones ya armadas para la utilización en nuestros programas, lo cual nos permite enfocarnos directamente sobre el problema.

También utilizamos el Matlab para graficar los resultados, lo cual nos permitió de una forma fácil, observar cómo se iba desplazando el brazo del robot, de acuerdo a los parámetros que iban variando.

Es importante destacar que cuando resolvemos los ejercicios matemáticamente, no tenemos en cuenta los límites físicos que tiene el robot. Gracias a que pudimos graficar el desplazamiento, pudimos ver y tomar precauciones en el rango de variación de los parámetros.

Por último, de acuerdo a todos los gráficos anteriores, podemos observar que no es posible generar un movimiento en línea recta variando los 3 parámetros al mismo tiempo. Para lograr esto, se debería utilizar la teoría de la cinemática inversa, y calcular para una trayectoria dada, los parámetros necesarios de las articulaciones.