UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Materia: Robótica
División: R6055
Tema: Implementación de una matriz cinemática en DSP
Profesor: Ing. Hernán Gianetta
Alumnos:

- Bole, Nicolás
- Ríos, Leandro
- Zbucki, Hernán

Objetivos

- Aprender a utilizar el software de simulación CodeWarrior para micros DSP como así también los comandos DSP.
- Poner en práctica los conceptos vistos en la teoría de las matrices homogéneas de cada grado de libertad del robot.
- Implementar matrices y argumentos en C.
- Visualizar la cinemática directa resultante del robot al variar los argumentos ϑ_1, ϑ_2 y l_3 de distintas formas.

Enunciado

- Implemente el código C de CodeWarrior para DSP56800/E de la cadena cinemática directa de la figura 1, usando el método de la matriz homogénea. Se usarán como setpoint trayectoria lineal continua a cada eje. Defina los límites y área del trabajo del manipulador. Las articulaciones del manipulador obedecen a los siguientes movimientos.

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

- Imprima el resultado del vector (X,Y,Z) usando la función plot3(X,Y,Z) en MATLAB

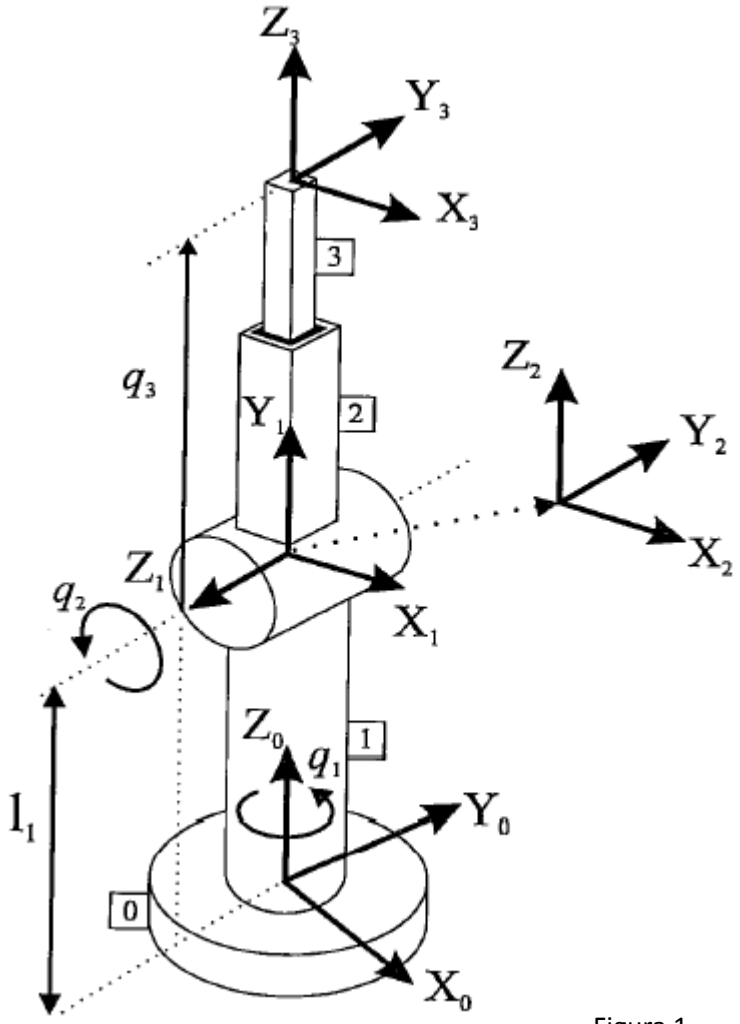

Figura 1

Introducción

La resolución de la cinemática directa del robot es de vital importancia para entender la capacidad de movimiento y de flexibilidad que nos puede brindar el robot que estamos diseñando. La cinemática directa se basa en el conocimiento de los movimientos particulares de cada articulación, obteniéndose como resultado la trayectoria del extremo del robot. Cabe acotar que este método es limitado a la hora de una análisis completo del robot porque no comprende los rozamientos, las cargas ni las velocidades y aceleraciones. Es decir, sin interiorizarnos en la dinámica del robot, vamos a poder analizar los posibles movimientos de un robot.

Los métodos para desarrollar la cinemática directa pueden ser distintos según el autor; el más simple es por uso de la trigonometría. Como se ve en el ejemplo de la figura 2, estos dos links pueden producir las siguientes coordenadas cartesianas en el extremo:

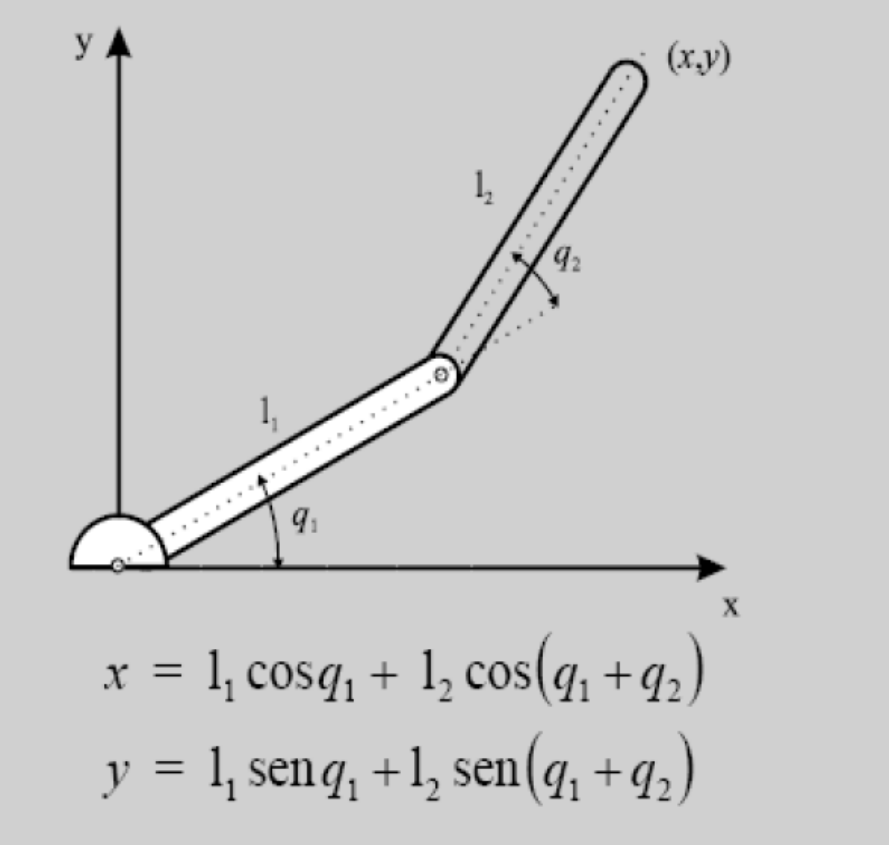

Figura 2

Como se puede ver, las coordenadas resultan de una simple descomposición trigonométrica, lo cual en este caso es simple. Pero en sistemas más complicados y con más grados de libertad la tarea puede resultar ardua.

Para ello se calcula la matriz homogénea: Es una matriz tipificada en la cual se encuentra una matriz de rotación 3x3, una de traslación 1X3, y una última de 1x1 para el escalado. Como ejemplo de uso de este método, resolveremos el robot de la Figura 1 de la siguiente forma.

Para la primera articulación:

$$ {}^{0}A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & l_1.C_1 \\ S_1 & C_1 & 0 & l_1.S_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Para la segunda articulación:

$$ {}^{1}A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & l_2.C_2 \\ S_2 & C_2 & 0 & l_2.S_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

La multiplicación de ambas matrices resulta:

$$ {}^{0}A_2 = \begin{bmatrix} C_{1+2} & -S_{1+2} & 0 & l_1C_1+l_2C_{1+2} \\ S_{1+2} & C_{1+2} & 0 & l_2S_2+l_2S_{1+2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Multiplicando a esta última matriz homogénea por vector de coordenadas P'=[0,0,0,1] (Origen de coordenadas del sistema visto desde el extremo de la cadena):

$$ X = l_1.\cos(\vartheta_1)+l_2.\cos(\vartheta_1+\vartheta_2) $$
$$ Y = l_2.\mathrm{sen}(\vartheta_2)+l_2.\mathrm{sen}(\vartheta_1+\vartheta_2) $$

<u>Resolución de la matriz homogénea del manipulador del enunciado</u>

- Matriz homogénea de la primera articulación:

$$ {}^{0}A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- Matriz homogénea de la segunda articulación:

$$ {}^{1}A_2 = \begin{bmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- Matriz homogénea de la tercera articulación:

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

- Multiplicación de las matrices homogéneas ${}^{0}A_1$ y ${}^{1}A_2$

$$ {}^{0}A_2 = \begin{bmatrix} C_1C_2 & -S_1 & -C_1S_2 & 0 \\ S_1C_2 & C_1 & -S_1S_2 & 0 \\ S_2 & 0 & C_2 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

- Matriz homogénea del manipulador

$$ {}^{0}A_3 = \begin{bmatrix} C_1C_2 & -S_1 & -C_1S_2 & -q_3C_1S_2 \\ S_1C_2 & C_1 & -S_1S_2 & -q_3S_1S_2 \\ S_2 & 0 & C_2 & q_3C_2+l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Código de implementación

Se implemento el siguiente código en el software CodeWarrior 56800E para simular la generación de las coordenadas de movimiento del extremo del robot. Se fueron modificando la generación de argumentos, dejando constantes algunos mientras otros variaban, alternadamente. Se implementó la matriz homogénea como un Array bidimensional de 16 posiciones.

```
/** ###################################################################
**     Filename  : TPN1.C
**     Project   : TPN1
**     Processor : 56F8367
**     Version   : Driver 01.13
**     Compiler  : Metrowerks DSP C Compiler
**     Date/Time : 11/05/2009, 19:38
**     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 TPN1 */


/* Including needed modules to compile this module/procedure */
#include "Cpu.h"
#include "Events.h"
#include "TFR1.h"
#include "MFR1.h"
#include "MEM1.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 MXRAD 90
#define PULSE2RAD 32767/MXRAD

Frac16 c,i,L1;
```

```
Frac16 pulse2rad=PULSE2RAD ,testResult[MXRAD],testResult2[MXRAD];

/* Argumento 1 – Ángulo q1 */
Frac16 Alfa1[MXRAD];
/* Argumento 2 – Ángulo q2*/
Frac16 Alfa2[MXRAD];
/* Argumento 3 – Distancia l3*/
Frac16 Desp3[MXRAD];
/*Matriz Homogenea*/
Frac16 Total[4][4],x,y,z;

Word16 c16[MXRAD];
Word32 c32[MXRAD];

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

        /*
        La init 56800 esta implementada en la función de abajo.
        */

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

 /* Write your code here */

 for(L1=0;;)
        {
                c=0;
                for(i=0;i<MXRAD;i++)
                        {
                                /*Valorizacion de los argumentos*/
                                c32[i]=(L_mult(pulse2rad,i));
                                Alfa1[i]=extract_l(c32[i]);
                                Alfa2[i]=extract_l(c32[i]);
                                Desp3[i]=extract_l(c32[i]);


/*Elemento Fila 1 Columna 1*/
Total[0][0]=mult(TFR1_tfr16CosPIx(Alfa1[i]),TFR1_tfr16SinPIx(Alfa2[i]));

/*Elemento Fila 1 Columna 2*/
Total[0][1]=negate(TFR1_tfr16SinPIx(Alfa1[i]));

/*Elemento Fila 1 Columna 3*/
Total[0][2]=negate(mult(TFR1_tfr16CosPIx(Alfa1[i]),TFR1_tfr16SinPIx(Alfa2[i])));
```

```
/*Elemento Fila 1 Columna 4*/
Total[0][3]=mult(negate(mult(TFR1_tfr16CosPIx(Alfa1[i]),TFR1_tfr16SinPIx(Alfa2[i]))),Desp3[i]);

/*Elemento Fila 2 Columna 1*/
Total[1][0]=mult(TFR1_tfr16SinPIx(Alfa1[i]),TFR1_tfr16CosPIx(Alfa2[i]));

/*Elemento Fila 2 Columna 2*/
Total[1][1]=TFR1_tfr16CosPIx(Alfa1[i]);

/*Elemento Fila 2 Columna 3*/
Total[1][2]=negate(mult(TFR1_tfr16SinPIx(Alfa1[i]),TFR1_tfr16CosPIx(Alfa2[i])));

/*Elemento Fila 2 Columna 4*/
Total[1][3]=mult(negate(mult(TFR1_tfr16SinPIx(Alfa1[i]),TFR1_tfr16SinPIx(Alfa2[i]))),Desp3[i]);

/*Elemento Fila 3 Columna 1*/
Total[2][0]=TFR1_tfr16SinPIx(Alfa2[i]);

/*Elemento Fila 3 Columna 2*/
Total[2][1]=0;

/*Elemento Fila 3 Columna 3*/
Total[2][2]=TFR1_tfr16CosPIx(Alfa2[i]);

/*Elemento Fila 3 Columna 4*/
Total[2][3]=add(mult(TFR1_tfr16CosPIx(Alfa2[i]),Desp3[i]),L1);

/*Elemento Fila 4 Columna 1*/
Total[3][0]=0;

/*Elemento Fila 4 Columna 2*/
Total[3][1]=0;

/*Elemento Fila 4 Columna 3*/
Total[3][2]=0;

/*Elemento Fila 4 Columna 4*/
Total[3][3]=1;

/* Calculamos las coordenadas cartesianas del punto extremo del robot ( X Y Z)*/

x=Total[0][3];
y=Total[1][3];
z=Total[2][3];

/*Imprimo X,Y,Z  y los argumentos q1,q2,l3*/
printf ("%d,%d,%d,%d,%d,%d",x,y,z,Alfa1[i],Alfa2[i],Desp3[i]);
printf("\n");
```

```
                        }
                }
}

/* END TPN1 */
/*
** ###################################################################
**
**     This file was created by UNIS Processor Expert 2.99 [04.17]
**     for the Freescale 56800 series of microcontrollers.
**
** ###################################################################
*/
```

Plots 3D del movimiento del extremo de la cadena cinemática del manipulador

Se grafican a continuación los movimientos posibles del extremo de la cadena en el espacio cartesiano X, Y, Z. Para ellos se fueron variando los valores de q1, q2, y q3.

Cabe acotar que para los siguientes análisis, el valor de L_1 fue considerado nulo (L_1=0). Las variaciones de q_1, q_2 y q_3 fueron hechas en términos de la notación frac16 desde los intervalos continuos 0 a 32767 y de -32767 a 0. Esas trayectorias lineales quedan plasmadas en movimientos en octantes positivos y negativos que luego se irá restringiendo en función del área de trabajo requerida.

a. Variando al ángulo q_1 en los intervalos descritos anteriormente y, dejando constantes en 0 grados al ángulo q_2 y al desplazamiento q_3: (Figura 3)

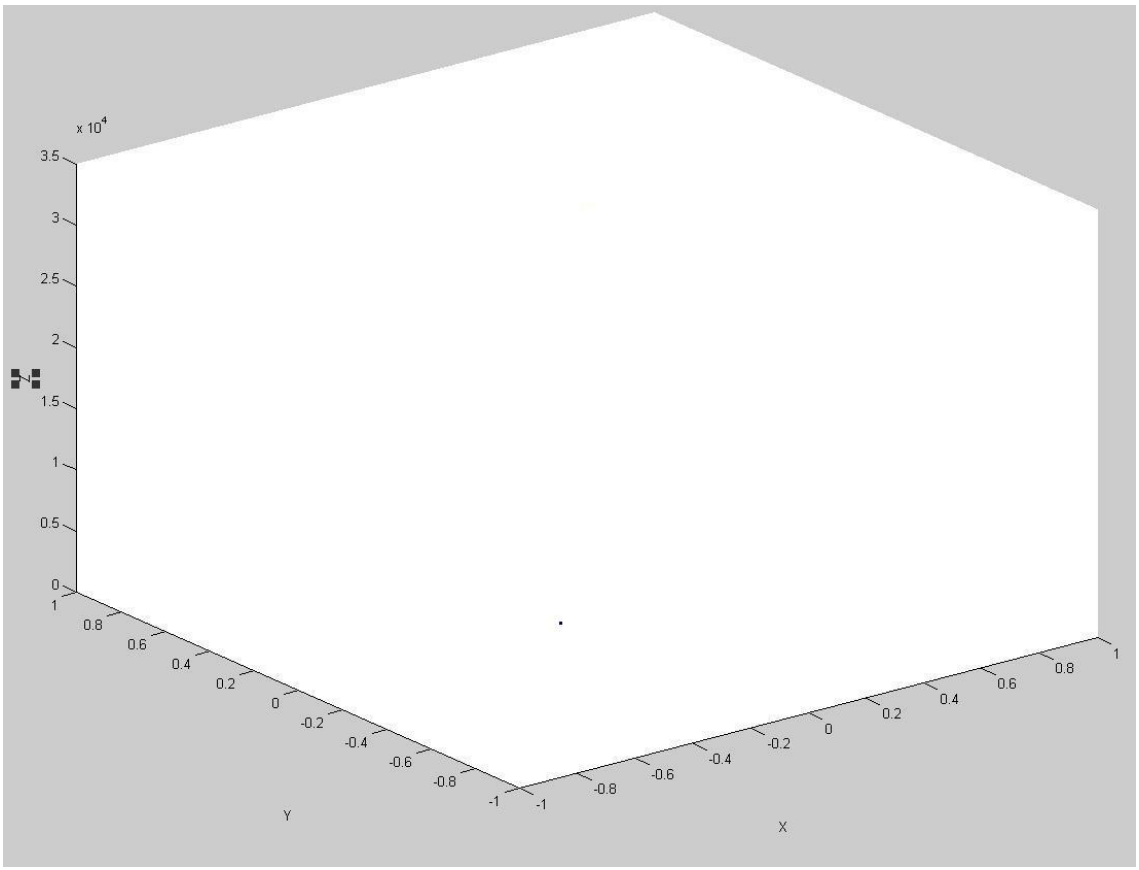

Figura 3

Se puede observar que al mover el ángulo q1 y al no tener ni longitud de brazo ni inclinación, el punto extremo queda fijo en el sistema coordenado.

b. Variando al ángulo q_2 en los intervalos descritos anteriormente y, dejando constantes en 0 grados al ángulo q_1 y al desplazamiento q_3: (Figura 3)

En este caso sucede exactamente lo mismo; modificando q_2 tenemos inclinación pero al no tener q_3 (longitud de brazo), el extremo queda fijo en el espacio cartesiano tridimensional.

c. Variando la longitud q_3 en los intervalos descritos anteriormente y, dejando constantes a 0 grados los ángulos q_1 y q_2

Aquí, sin la necesidad de modificar la inclinación o la rotación, al darle distintos valores a la distancia del brazo del robot se puede observar una trayectoria variable según la longitud que se desee, tanto para arriba (valores positivos del eje Z), como para abajo (valores negativos del eje Z). Figura 4

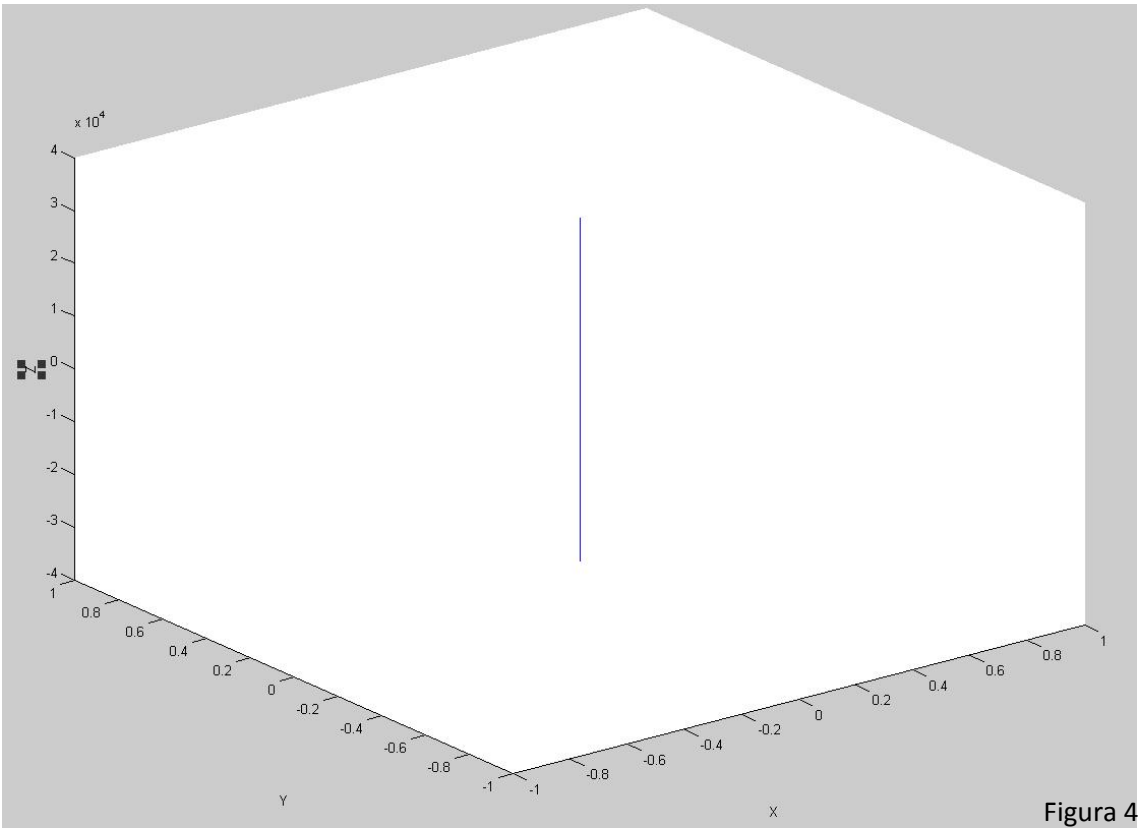

Figura 4

Como no es viable traspasar ciertos límites en el movimiento, restringimos el mismo a los valores positivos del eje Z; los resultados se muestran en la Figura 5 con un patrón similar al de la figura anterior.

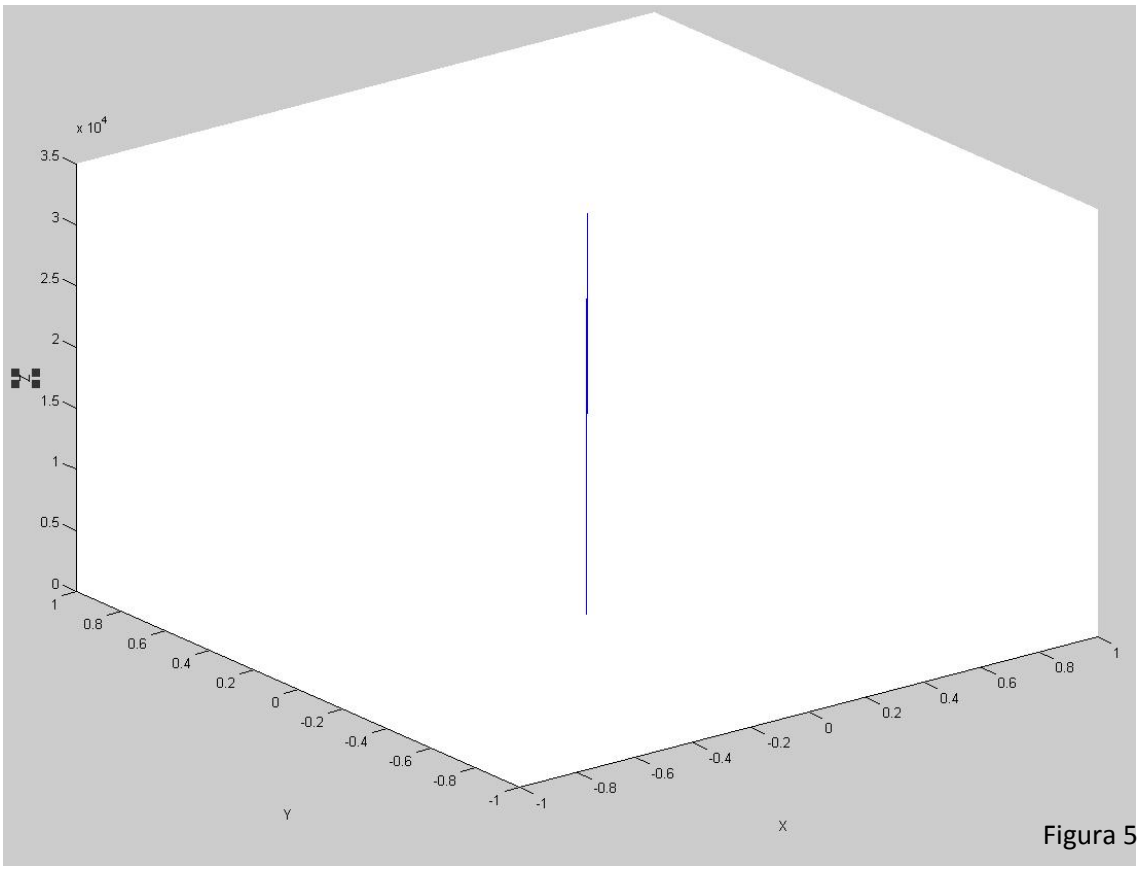

Figura 5

d. Variando los ángulos q_1 y q_2 en la misma forma y dejando constante a cero el valor de q_3:

   En este otro caso se puede distinguir el mismo efecto que veíamos anteriormente en la Figura 3. Al no tener longitud el brazo del robot, los movimientos de las articulaciones 1 y 2 no marcan ningún dibujo en el espacio, y así el punto queda fijo en el centro de coordenadas.

e. Variamos el ángulo q_1 y el desplazamiento q_3 en la misma forma y dejamos constante con ángulo nulo al q_2:

   También aquí se aprecia cómo se modifica la longitud del brazo del robot sin importar el ángulo de rotación que le demos. (Figura 6) Se debe restringir el espacio, como se hizo antes, para que la longitud del brazo no crezca en coordenadas negativas ya que si el robot no estuviera elevado o en una plataforma especial, el extremo del mismo podría estrellarse contra el suelo. (Figura 7)

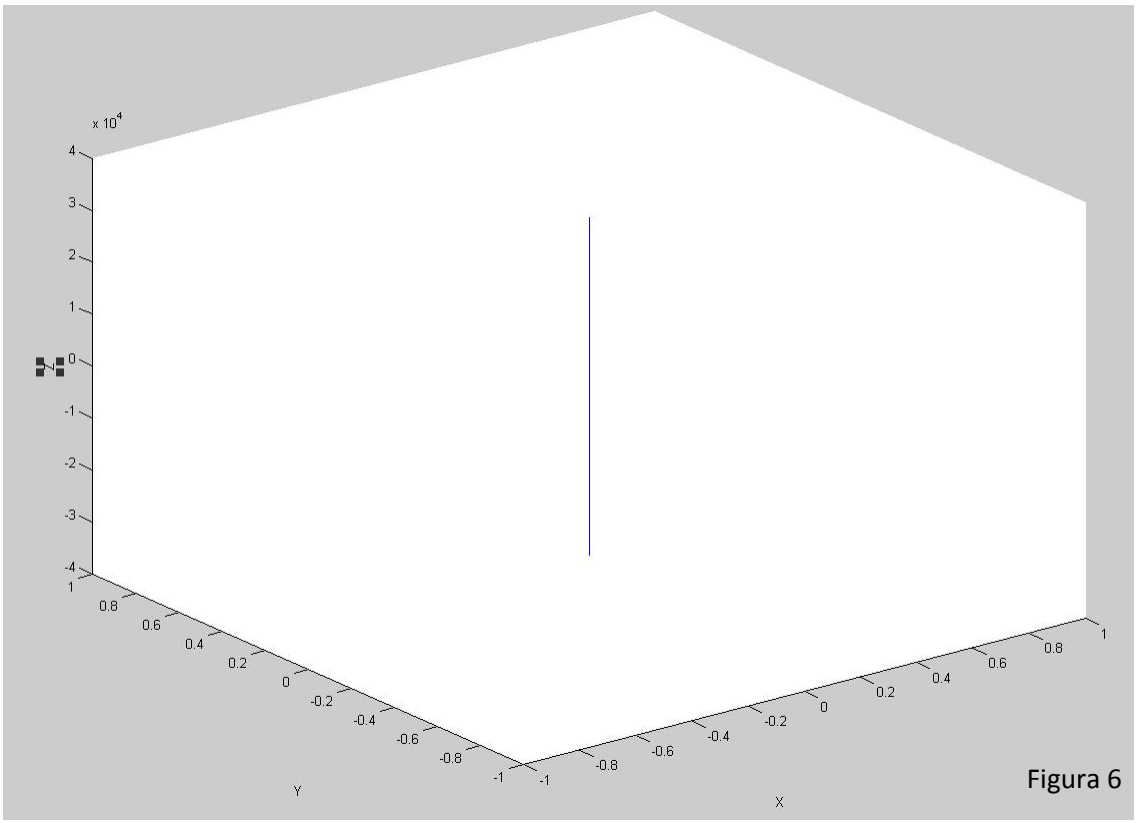
x 10^4
4
3
2
1
0
-1
-2
-3
-4
Z
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Y
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
X

Figura 6

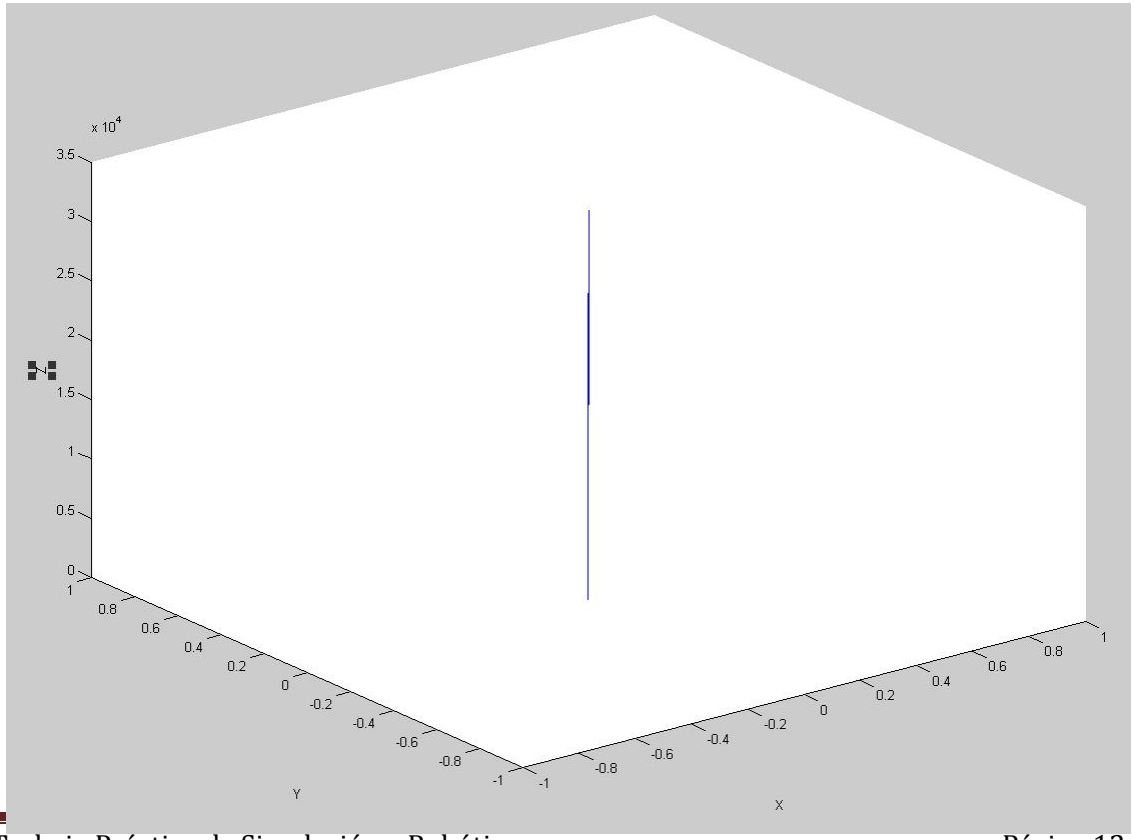
x 10^4
3.5
3
2.5
2
1.5
1
0.5
0
Z
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Y
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
X

Figura 7

f. Variamos el ángulo q_2 (ángulo de inclinación) y el desplazamiento q_3 en la misma forma y dejamos constante con ángulo nulo al q_1 (ángulo de rotación):

El diagrama de movimiento resultante se da sobre el plano Y=0, debido a que no estamos rotando la figura. Se ve claramente que el movimiento es elíptico sobre uno de los lados del plano X=0, precisamente el negativo. (Figura 8) Claramente se ve que es necesario restringir el binomio ángulo de inclinación y longitud del brazo debido a que muchos de los valores cruzan a los negativos de coordenadas del eje Z, situación generalmente indeseable. (Figura 9)

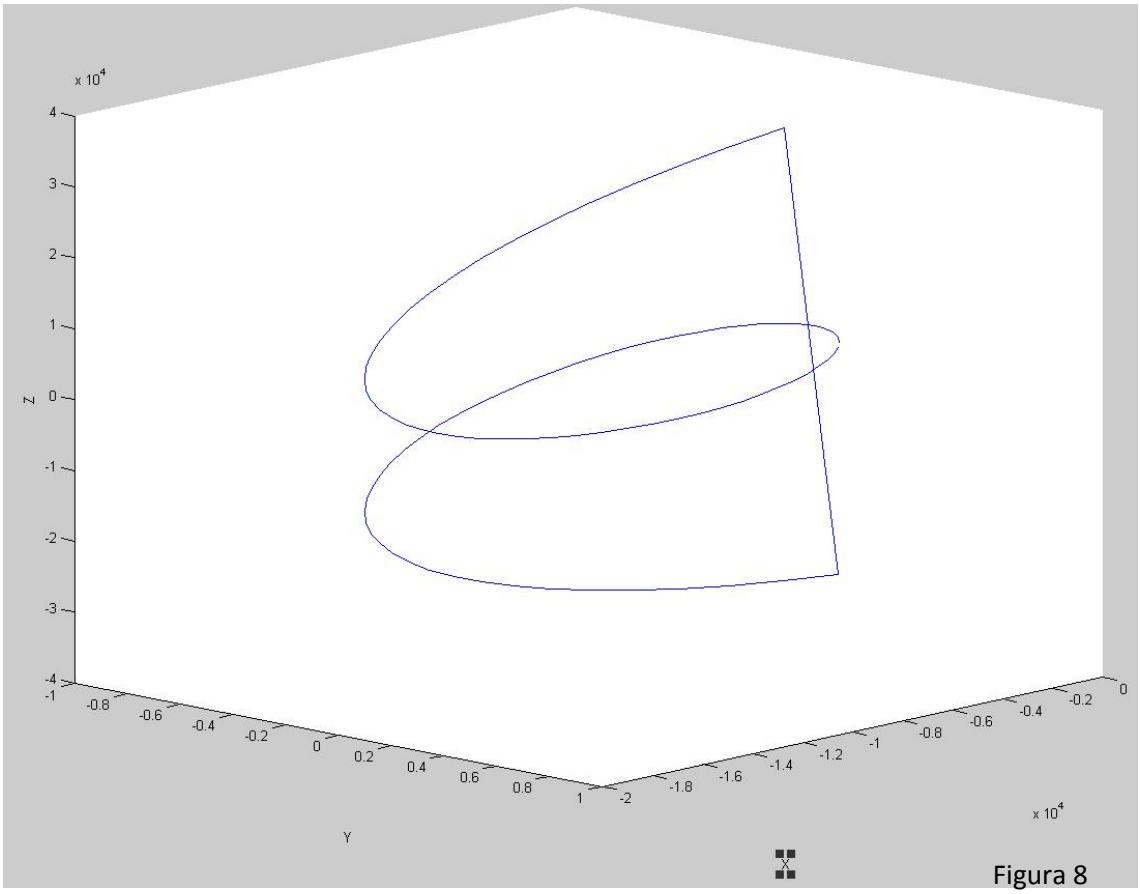

Figura 8

El movimiento que realiza el brazo, mientras se achica en tamaño y rota, resulta helicoidal. Por lo que se ve, también tenemos dos puntos críticos en los cuáles las coordenadas para el eje Z son nulas. Amén de esto, variando de distinta forma a q2 y q3 podemos cubrir casi todo un plano, en este caso paralelo al eje Y, siendo de gran utilidad para robots con grippers o con punta de soldadura. En los próximos diagramas se verá más claramente el movimiento en las tres dimensiones.

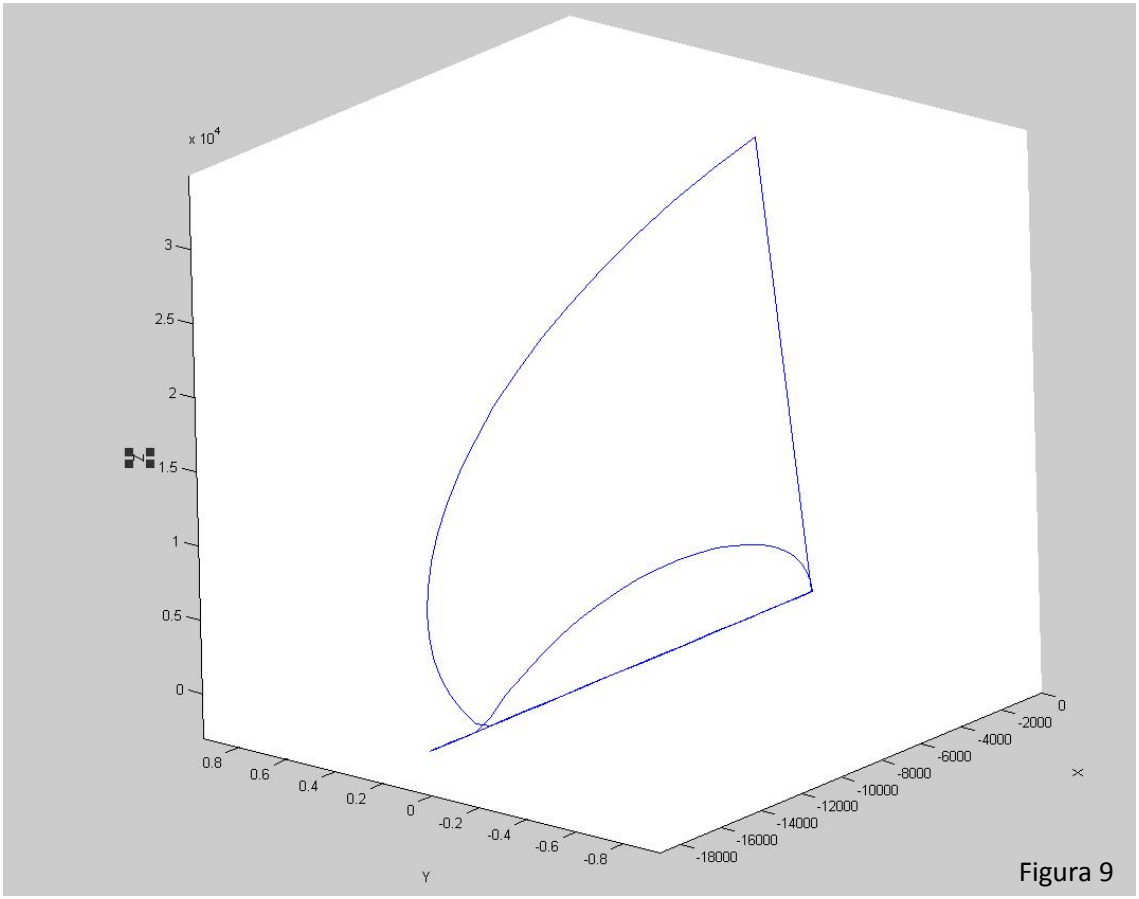

Figura 9

g. Variamos el ángulo q_2 (ángulo de inclinación) y el desplazamiento q_3 en la misma forma y el ángulo q_1 (ángulo de rotación):

Como se puede observar, el brazo del robot hace dos curvas que empiezan en el origen de coordenadas y que finalizan en lo más alto (cuándo el brazo está extendido hacia arriba) y en lo más bajo, cuando el intervalo de variación de los parámetros cruza los valores negativos.
El movimiento del brazo cubre 2 octantes, y los barre de en lo sentidos descritos. El dibujo nos termina dando una idea de buena autonomía, restringida sólo al octante de valores positivos. (Figura 10).
Se puede completar la idea que inicio el diagrama anterior; ahora el movimiento puede llegar a coordenadas X, Y, Z. Resulta entonces que el campo máximo de movimiento del robot sin restricciones es una esfera que lo rodea cuyo radio máximo depende de la longitud máxima que puede tomar q_3 Pero sin embargo, en la realidad este movimiento no es viable, por lo tanto lo debemos restringir a valores positivos de las tres coordenadas, suponiendo de alguna forma que el robot está situado en la arista (en la esquina y en el suelo) de una habitación. En función del recinto de trabajo real, se puede restringir el tamaño máximo de q_3 como así también los ángulo de inclinación y rotación. (Figura 11)

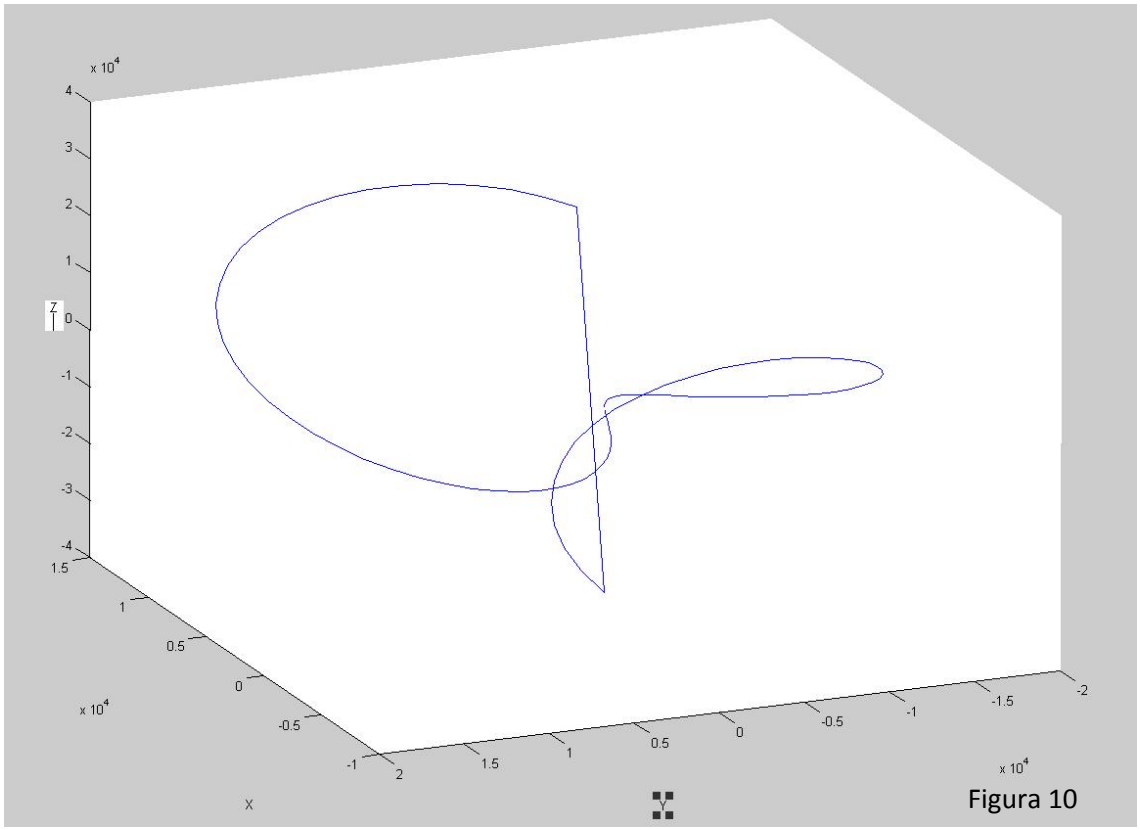

Figura 10

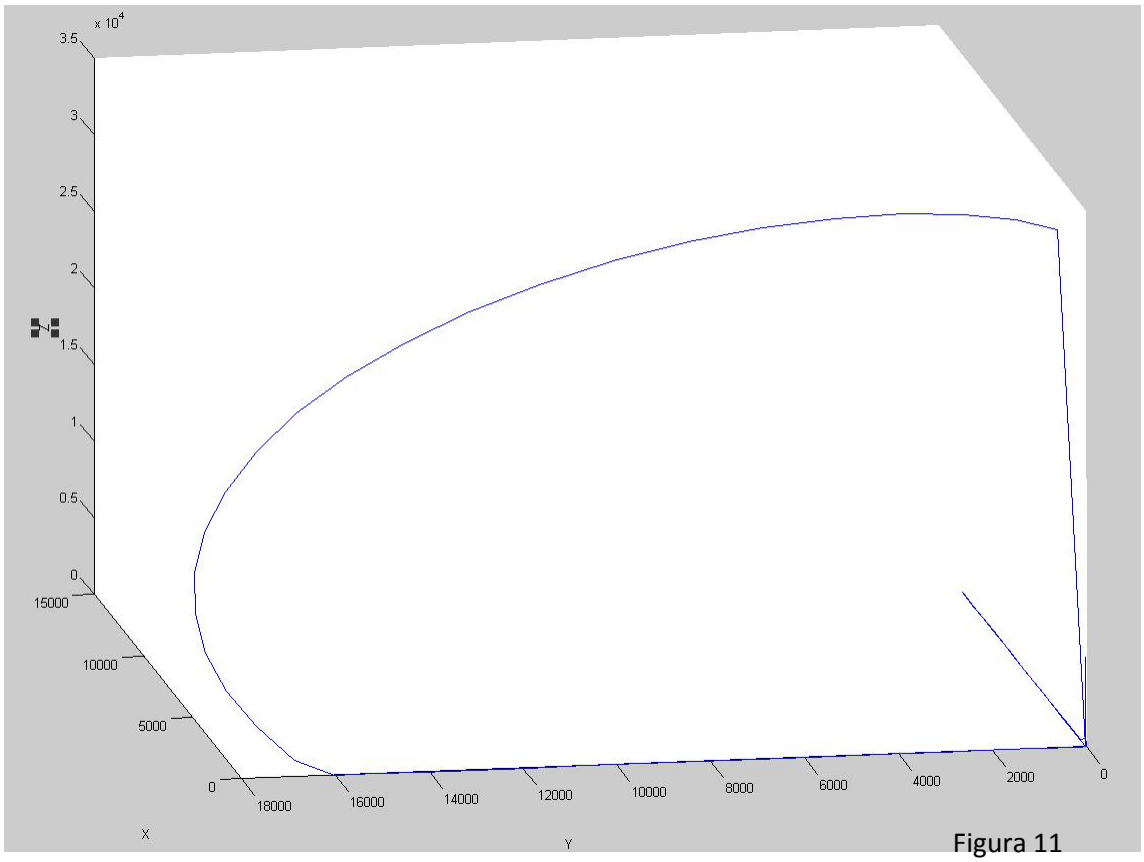

Figura 11

h. Mantenemos a q_3 con un valor constante distinto de cero (10000), dejando a q_2 nulo y modificando el ángulo q_1 de rotación.

Este caso resulta repetido a los primeros analizados; variamos el ángulo de rotación dejando constante el de inclinación, y sin importar la longitud que tome el extremo del brazo se describirá un punto en el sistema de coordenadas. La única diferencia con los casos anteriores es que el valor de la coordenada Z del punto es el valor de la longitud q_3. Figura 12

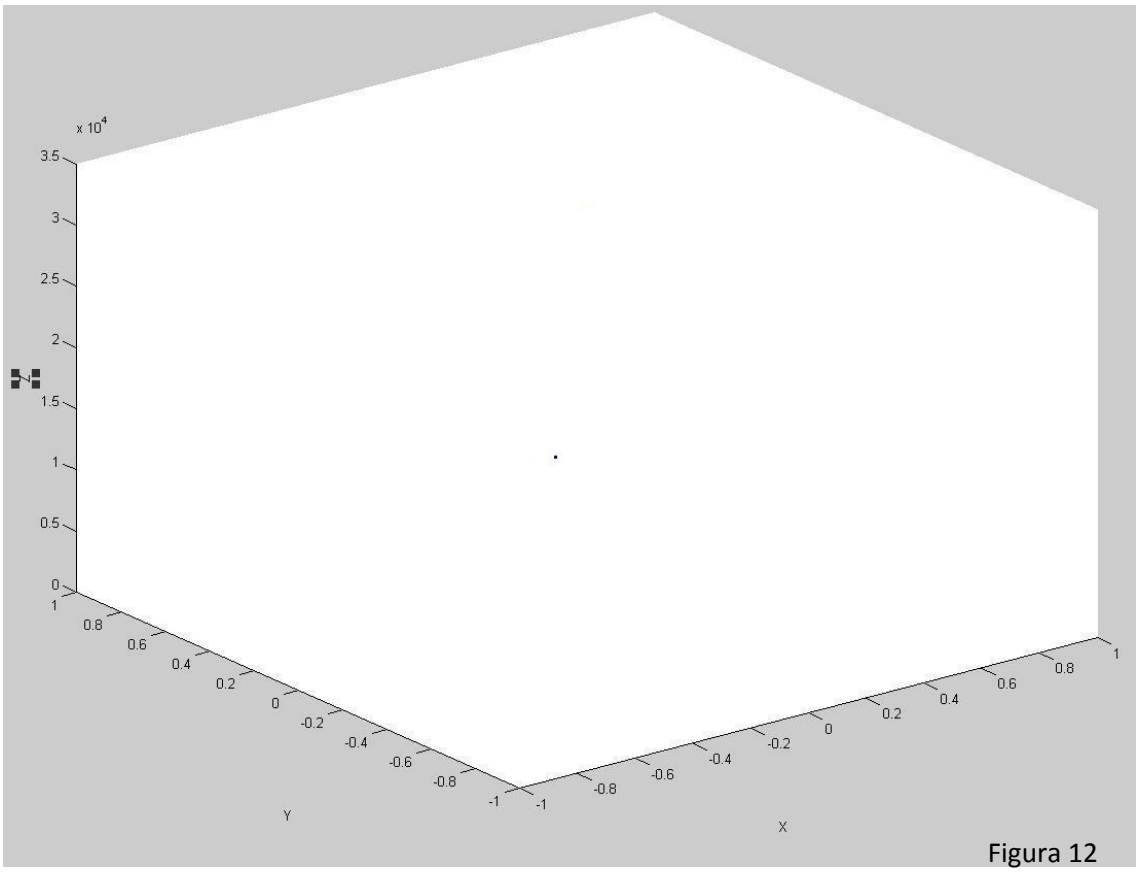

Figura 12

i. Mantenemos a q_3 con un valor constante distinto de cero (10000), modificando el ángulo q_2 de inclinación y el de rotación q_1 por igual.

En este diagrama se comienza a ver el abanico de movimientos que se genera con modificar los ángulos de rotación y de inclinación. Al no tocar la longitud q_3 se aprecia con claridad el espacio de movimiento que se puede obtener como así lo valores máximos de las trayectorias para el q_3 dado. (Figura 13) Tenemos dos espacios en donde surge los movimientos para q1=q2 variables. Debemos restringir los movimientos en donde las coordenadas Z son negativas por el mismo motivo que explicamos anteriormente (Figura 14). El movimiento queda enmarcado en el 1er octante del espacio describiendo una curva: dicha

curva da una idea de cuánto se puede extender un brazo fijo (por ejemplo) en el espacio, modificando sólo dos articulaciones (2 GDL).

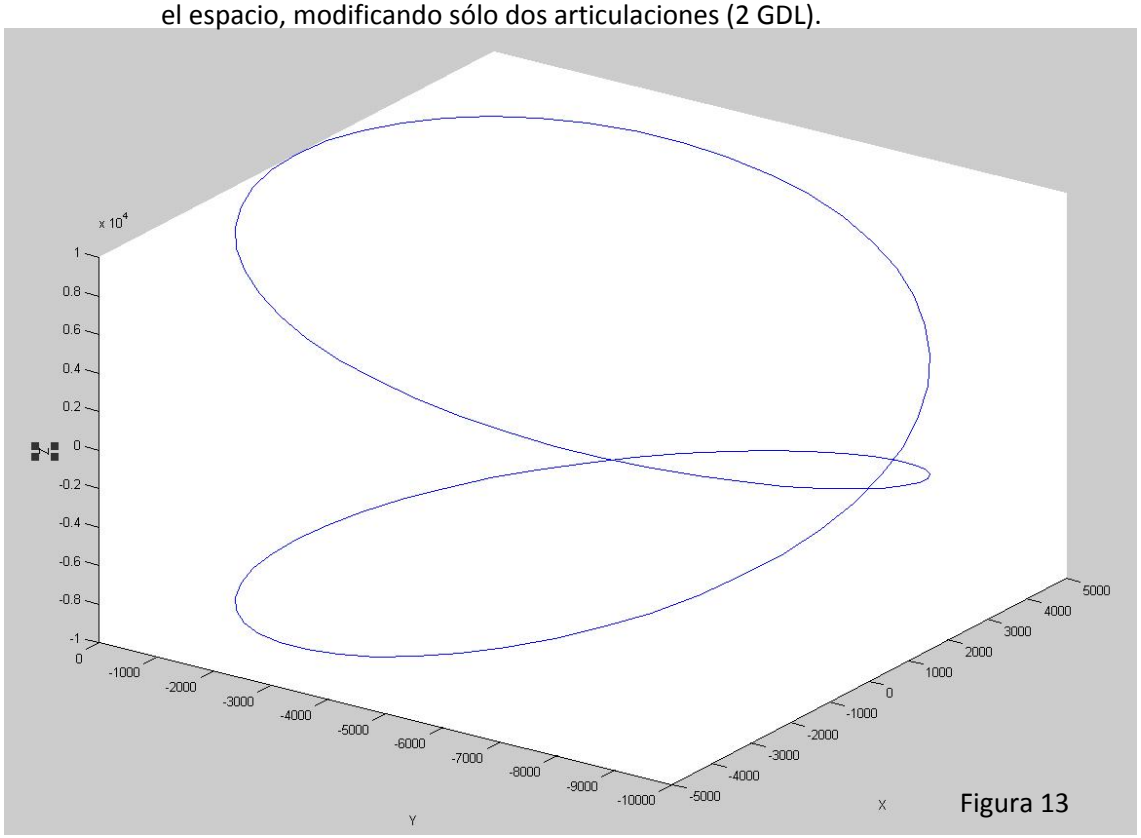

Figura 13

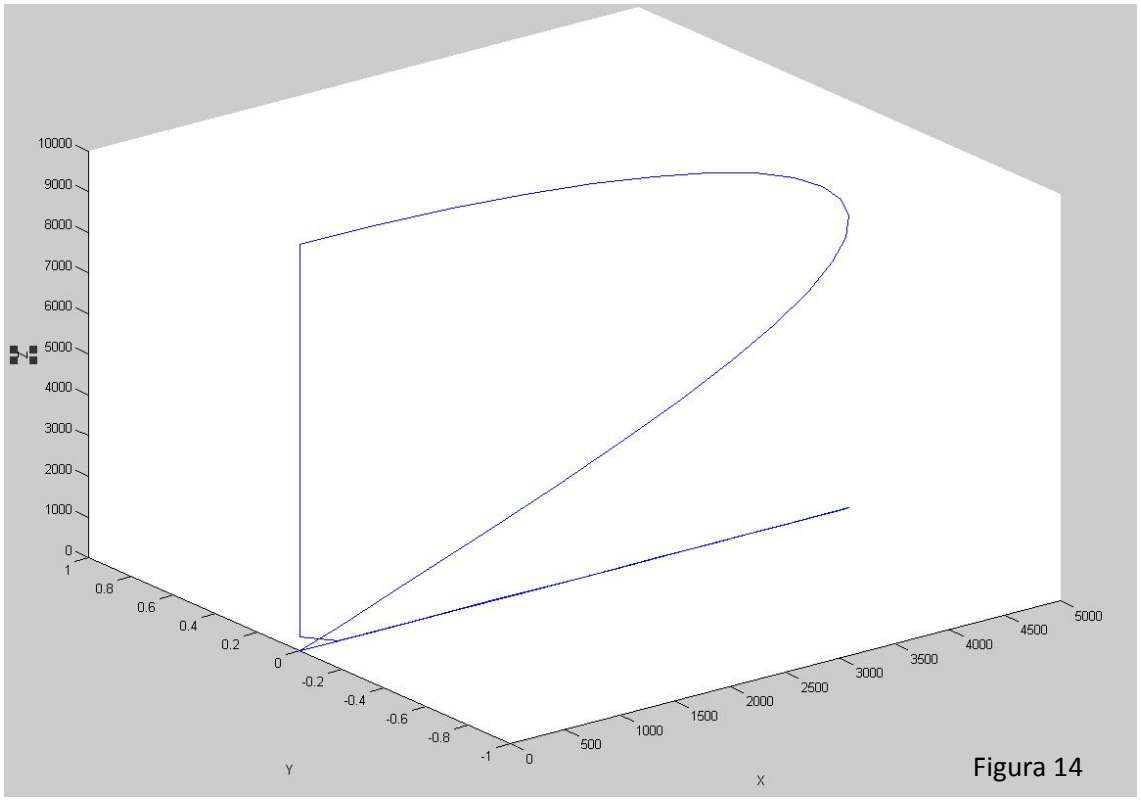

Figura 14

j. Variando el ángulo de rotación q_2, dejando constante en cero a q_1 y q_3=10000 Acá se describe una circunferencia en el espacio, lo que denota los máximos perimetrales del extremo del robot para un determinado q_3. (Figura 15). Se restringe el diagrama para valores positivos de coordenadas X,Y, Z.

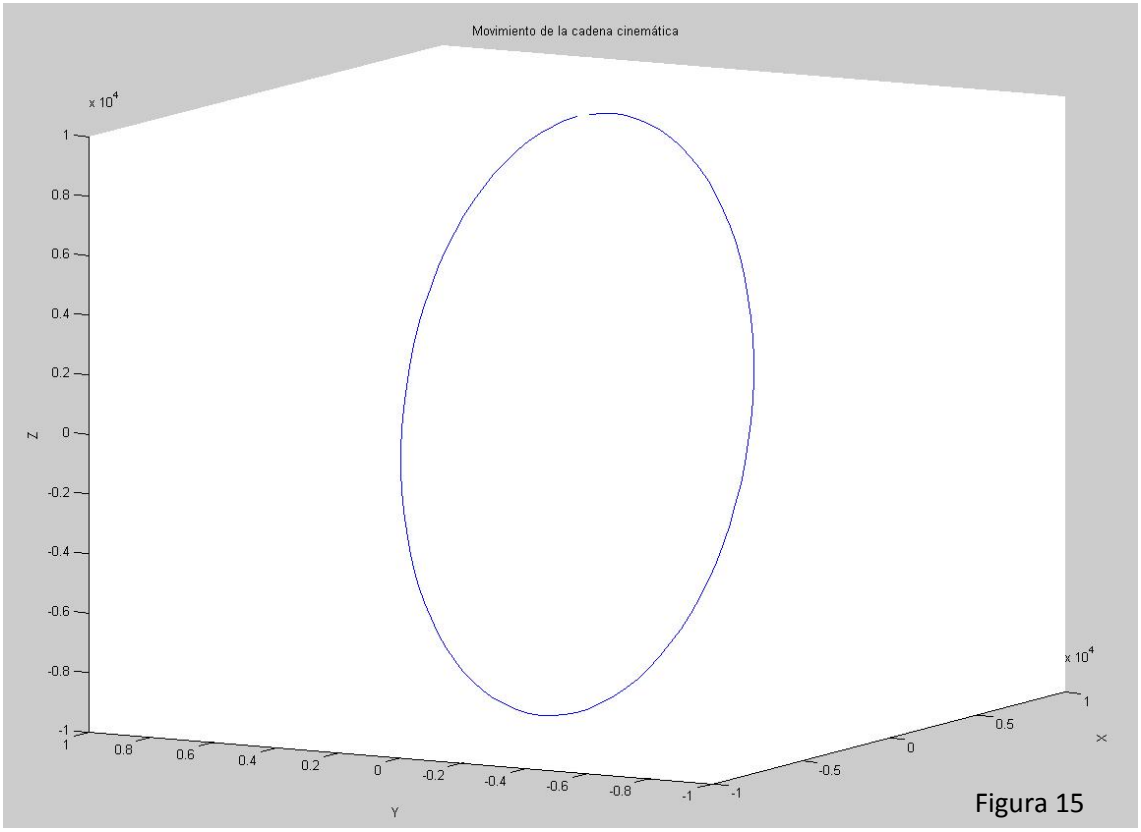

Figura 15

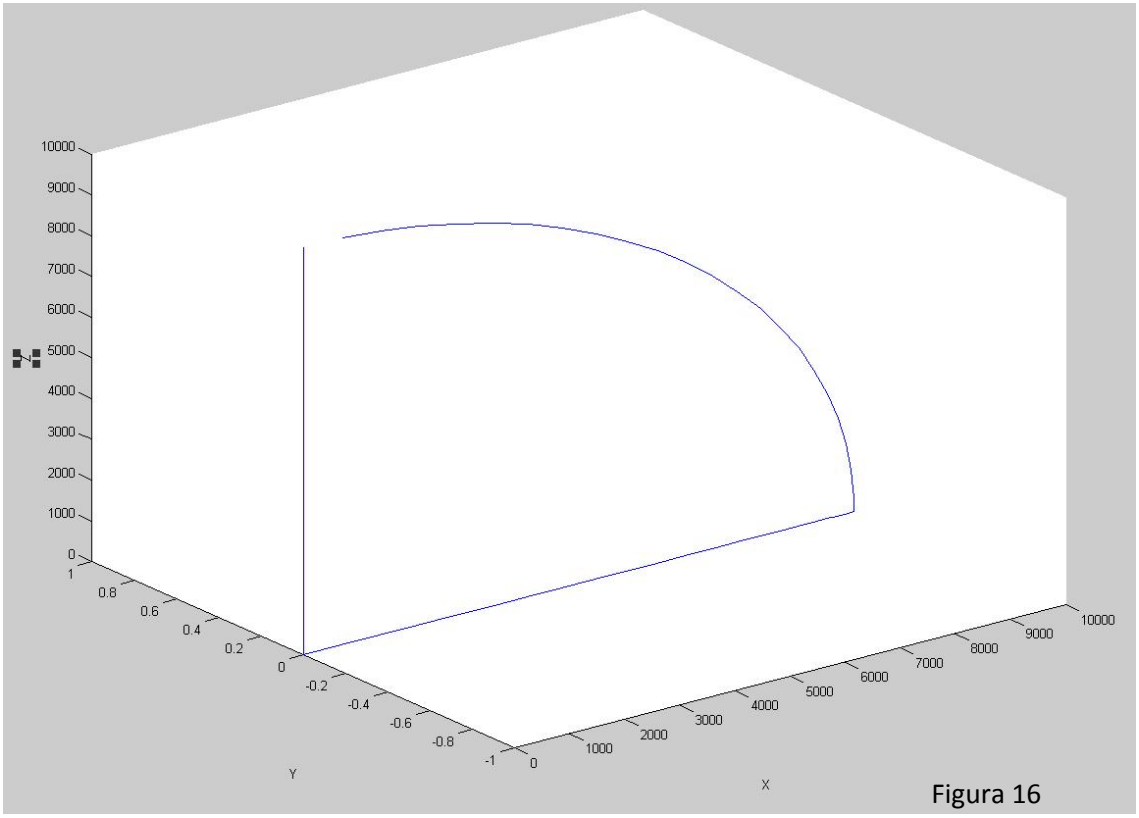

Figura 16

k. Modificamos q_1 y dejamos constante en 10000 a q_2 y q_3

Como pasó en el diagrama anterior, se enmarca una circunferencia contenida en un plano del eje Z. Se denota la rotación del robot para un determinado q_3 y un ángulo q_2, ambos fijos. (Figura 17) Se restringe el movimiento circular sólo para el 1er octante, como se hizo hasta ahora. (Figura 18)

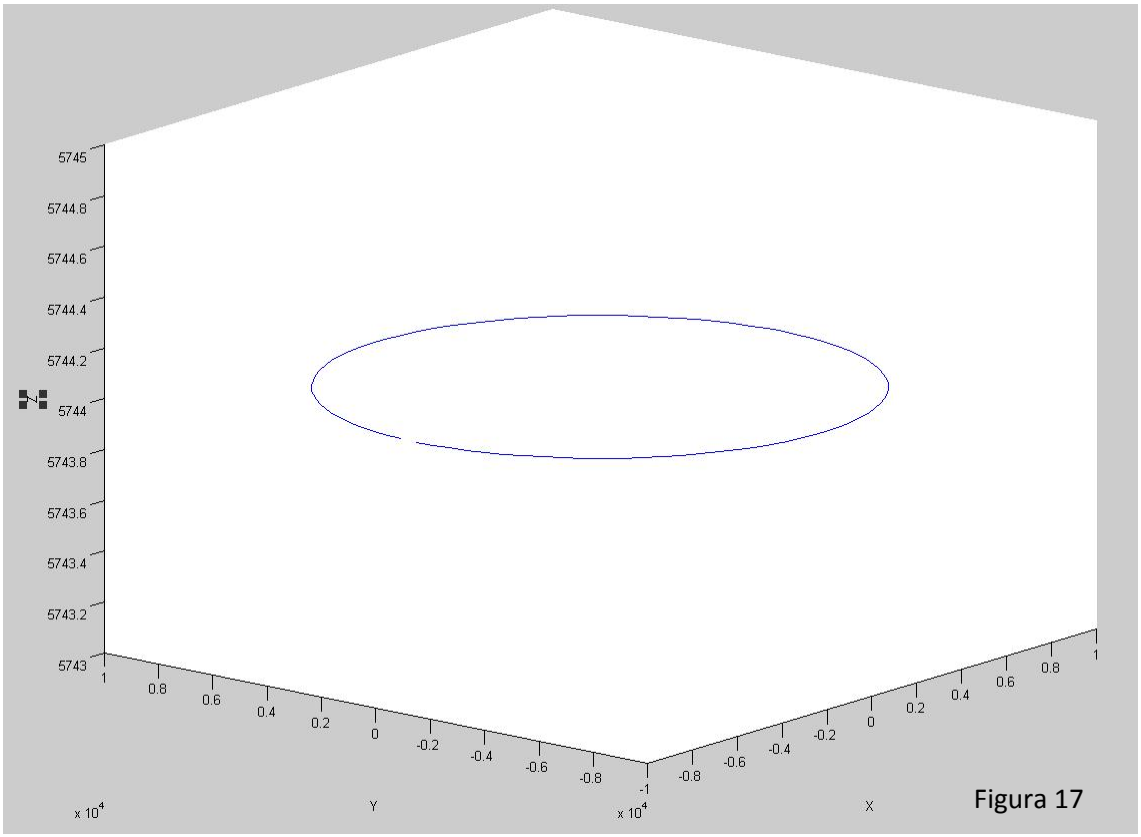

Figura 17

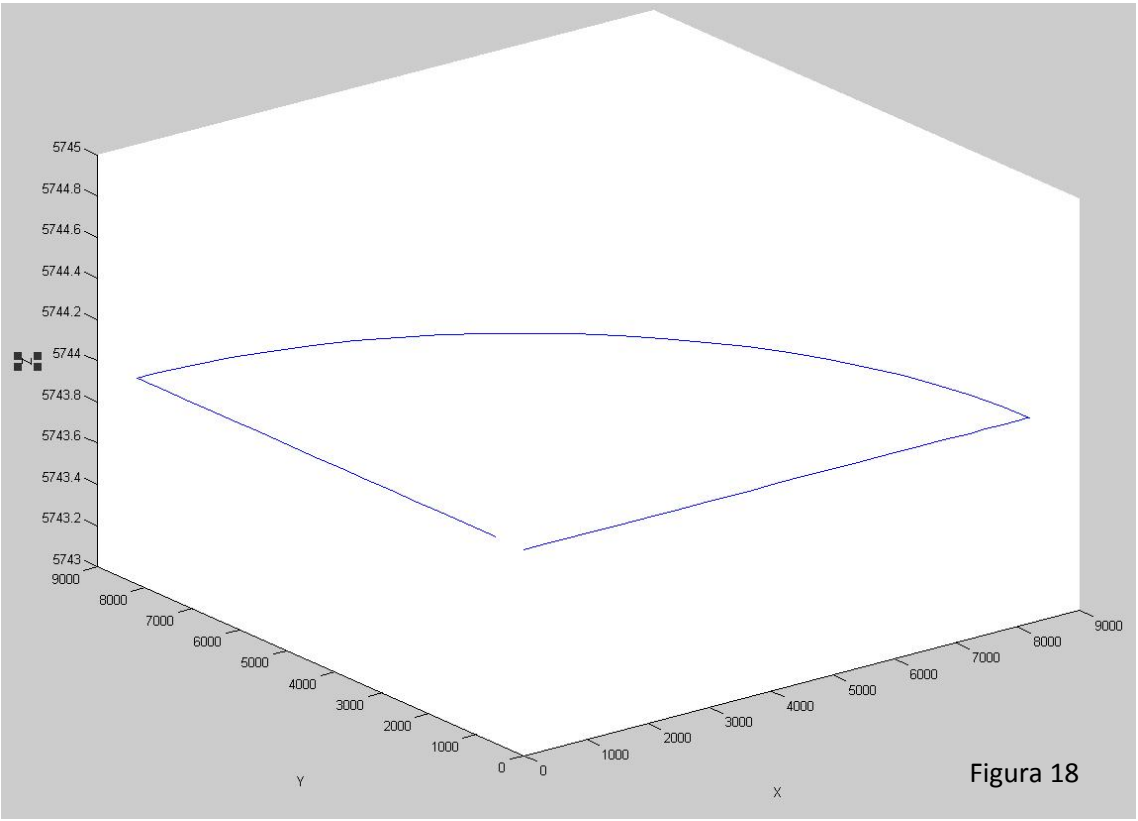

Figura 18

Conclusiones Finales

- Aprendimos la importancia de las matrices homogéneas: Si bien en la teoría se dificulta ver el significado práctico de éstas con un robot real, la implementación y simulación nos mostró lo simple de conceptualizar los movimientos y capacidades de este último con métodos algebraicos.
- Es muy importante las restricciones espaciales en los movimientos. La mayoría de las simulaciones mostraron coordenadas negativas las cuales hay que tener en cuenta para que el robot no choque contra el suelo o las paredes. Amén de eso, es necesario restringir aún más los movimientos para que no se choquen con otros robots o máquinas en una planta.
- Se verificó que el método de las matrices homogéneas es análogo al de hacer los procedimientos trigonométricos, pero facilitando el trabajo si el robot a analizar tiene más de 2 GDL.
- Chequeamos la implementación en C de la matriz homogénea con lo que el sentido común nos indicó. En muchas de las simulaciones, eran de esperarse movimientos como el de la figura 17 o 18, mientras que en otros movimientos, ni nos esperábamos obtener tales resultados.
- Aprendimos lo valioso que es manejar el recurso DSP y los microcontroladores que lo soportan. También así nos resulto de gran ayuda el uso de los *beans* en CodeWarrior: Las funciones más complejas vienen pre-implementadas en el soft y pueden ser usadas en proyectos propios como el nuestro. Acercarnos a esta tecnología nos brinda sabiduría para animarnos a encarar proyectos desafiantes y que requieran de lo último.