

**UNIVERSIDAD TECNOLÓGICA NACIONAL - FRBA**



## **ROBOTICA**

### **Cinemática de un Manipulador SCARA de 3 Grados de Libertad**

**ALUMNOS: Perilli, Hernán  
Granzella, Damián**

**CODIGO: 95-0482**

**CURSO: R6055**

**DOCENTE: Ing. Hernán Giannetta**

**AÑO: 2009**

# MANIPULADOR SCARA DE 3 GRADOS DE LIBERTAD

## Introducción a la 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 a resolver en la cinemática del robot. El primero de ellos se lo conoce como el **problema cinemática 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 es el denominado problema cinemático inverso, el cual resuelve la configuración que debe adoptar el robot para una posición y orientación para una posición conocida.

Denavit y Hartenberg propusieron un método sistemático para describir y presentar la geometría espacial de los elementos de una escala cinemática, y en particular de un robot, con respecto a un sistema de referencia fijo. Este método utiliza una matriz de transformación homogénea para describir la relación espacial entre dos elementos rígidos adyacentes, reduciéndose el problema cinemático directo a encontrar una matriz de transformación homogénea de 4x4 que relacione la localización espacial del extremo del robot con respecto al sistema de coordenadas de su base.

Por otro lado, 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.

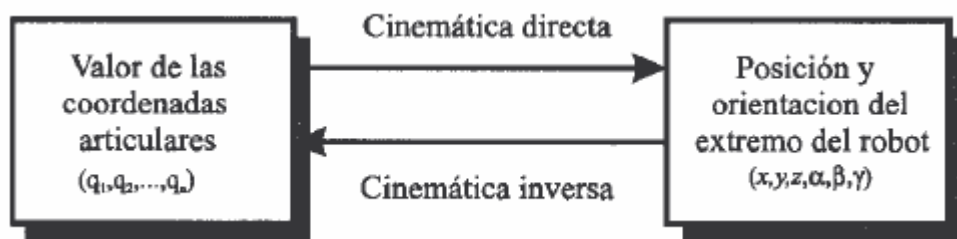


Diagrama de relación entre cinemática directa e indirecta.

## El Problema Cinemático Directo

Como ya se ha descrito anteriormente, 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 ó 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.

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

## Resolución del problema cinemática directo mediante matrices de transformación homogénea

La resolución del problema cinemática 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 dado por las relaciones:

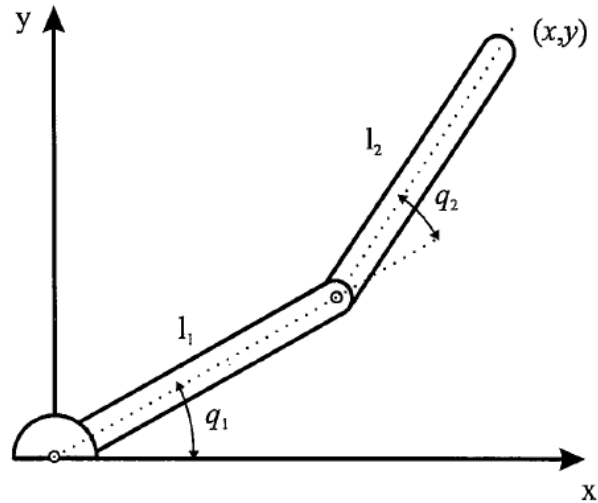
$$\begin{aligned}x &= f_x(q_1, q_2, q_3, q_4, q_5, q_6) \\y &= f_y(q_1, q_2, q_3, q_4, q_5, q_6) \\z &= f_z(q_1, q_2, q_3, q_4, q_5, q_6) \\\alpha &= f_\alpha(q_1, q_2, q_3, q_4, q_5, q_6) \\\beta &= f_\beta(q_1, q_2, q_3, q_4, q_5, q_6) \\\gamma &= f_\gamma(q_1, q_2, q_3, q_4, q_5, q_6)\end{aligned}$$

La obtención de estas relaciones no es en general complicada, sino incluso en ciertos casos fácil de encontrar mediante simples consideraciones geométricas. Por ejemplo para el caso de un robot con 2 grados de libertad es fácil comprobar que:

$$\begin{aligned}x &= l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\y &= l_1 \sin q_1 + l_2 \sin(q_1 + q_2)\end{aligned}$$

Para robots de más grados de libertad podría plantearse un método sistemático de basado en la utilización de las matrices de transformación homogénea.

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 homogénea que representa la posición y orientación relativa entre los sistemas asociados a dos eslabones consecutivos, 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 segundo eslabón respecto al 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. Así, por ejemplo, la posición y orientación del sistema solidario con el segundo eslabón del robot con respecto al sistema de coordenadas de la base se puede expresar mediante la matriz  ${}^0A_2$ :



Robot Planar de 2 Grados de Libertad

$${}^0A_2 = {}^0A_1 {}^1A_2$$

De manera análoga, la matriz  ${}^0A_3$  representa la localización del sistema del tercer eslabón:

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

Cuando se consideran todos los grados de libertad, a la matriz  ${}^0A_n$  se la suele denominar T.

Aunque para escribir 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 al sistema de referencia del elemento  $i$  con el del elemento  $i-1$ . Las transformaciones son las siguientes:

1. Rotación alrededor del eje  $z_{i-1}$  un ángulo  $\theta_i$ .
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  $\alpha_i$ .

Dado que el producto de matrices no es conmutativo, las transformaciones se han de realizar en el orden indicado. De este modo se tiene que:

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

Y realizando el producto entre las matrices queda:

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

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

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

### Algoritmo de Denavit-Hartenberg para la obtención del modelo cinemático directo

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

**D-H 2.** Numerara 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 ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo de 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 articulación i+1.

**D-H 5.** Situar el origen del sistema de base  $\{S_0\}$  en cualquier punto del eje  $z_0$ . Los ejes  $x_0$  e  $y_0$  se situarán 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  $z_{n-1}$  y  $z_n$ .

**D-H 10.** Obtener  $\theta_i$  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  $x_{i-1}$  y  $x_i$  quedasen alineados.

**D-H 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 al nuevo  $\{S_{i-1}\}$  para que su origen coincidiese con  $\{S_i\}$ .

**D-H 13.** Obtener  $\alpha_i$  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  $\{S_i\}$ .

**D-H 14.** Obtener las matrices de transformación  ${}^{i-1}A_i$ .

**D-H 15.** Obtener la matriz T de transformación que relaciona el sistema de la base con el del extremo del robot.

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

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

$\theta_i$ : 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 de  $z_{i-1}$  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  $z_i$ .

$\alpha_i$ : Es el ángulo de separación del eje  $z_{i-1}$  y el eje  $z_i$ , medido en un plano perpendicular a un eje  $x_i$ , utilizando la regla de la mano derecha.

## Resolución del problema cinemático directo de un manipulador SCARA de 3 grados de libertad

### Manipulador SCARA

En la figura a continuación, se muestra un robot con la configuración SCARA (Selective Compliance Assembly Robot Arm). Este robot fue creado por un grupo de industrias electrónicas japonesas, en colaboración con dos universidades, para insertar los componentes de forma vertical. Como se puede observar esta configuración está formada por dos articulaciones de rotación con respecto a dos ejes paralelos entre sí y perpendiculares al plano de trabajo, y una de desplazamiento en una dirección paralela a la de los ejes de rotación.



Figura 1. Configuración SCARA

El volumen de trabajo de este robot, suponiendo unas longitudes de los brazos de  $L_1$  y  $L_2$ , un rango de giro de 360 grados y un rango de desplazamiento de  $L_3$ , es de  $\pi (L_1 + L_2)L_3$ . En un robot real los giros estarán limitados a un ángulo máximo obteniéndose un volumen de trabajo complejo que vendrá dado por el fabricante. En la Figura 2 se muestra el volumen de trabajo del robot de la Figura 1, siendo este volumen el típico de la configuración SCARA.

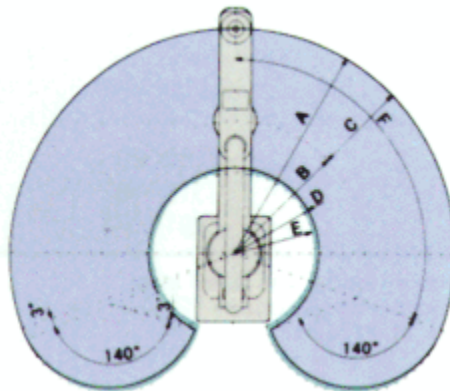
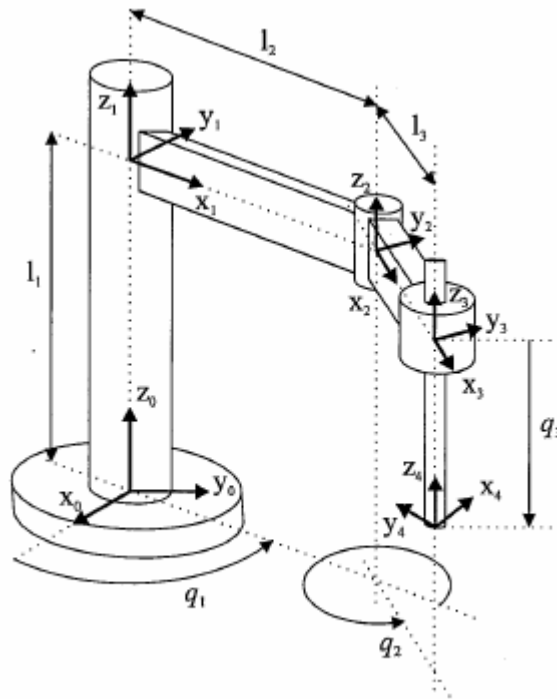


Figura 2. Volumen de Trabajo del Robot SCARA

En nuestro caso de análisis, el siguiente esquema es el que utilizaremos para tal fin, indicando cada una de sus partes constitutivas, así como sus articulaciones, y subsistemas.



Asignación de sistemas de referencia de un robot SCARA.

El sistema de referencia  $\{0\}$ , es el sistema de coordenadas base, y asumimos que está fijado al entorno del Robot. El sistema de referencia  $\{1\}$  coincide en el origen con  $\{0\}$  pero puede ser rotado un ángulo  $q_1$  alrededor del eje  $z_1$ . Al final del enlace 1, de longitud  $l_1$  se sitúa el sistema de referencia  $\{2\}$ , que puede rotar un ángulo  $q_2$  alrededor del eje  $z_2$ . El sistema de coordenadas  $\{3\}$  situado al final del enlace 2 se puede desplazar a lo largo del eje  $z$ , siendo  $d_3$  el desplazamiento de dicho enlace.

A fin de usar las transformaciones homogéneas para pasar de sistema de referencia a otro, se definen una serie de parámetros asociados a la estructura del manipulador, que posteriormente se utilizan en el cálculo del modelo directo.

Además para cada sistema de coordenadas  $\{i\}$ , se define el parámetro  $a_{i-1}$  como la distancia del eje de la articulación  $i-1$  y el eje de la articulación  $i$ , medida sobre la línea perpendicular común y  $\alpha_{i-1}$  como el ángulo entre estos dos ejes.

Obteniendo todos los parámetros anteriores para cada articulación del manipulador SCARA se obtiene los denominados parámetros de Denavit-Hartenberg del robot SCARA

Articulación	$\theta$	$d$	$a$	$\alpha$
1	$q_1$	$l_1$	0	0
2	0	0	$l_2$	0
3	$q_2$	0	$l_3$	0
4	0	$-q_3$	0	0

Parámetros de Denavit-Hartenberg



La matriz de transformación homogénea que realiza el paso del sistema {i} al {i-1} en función de los anteriores parámetros es:

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

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

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

A partir de la matriz de cada articulación se puede calcular la posición y orientación en el espacio cartesiano del extremo del manipulador como:

$$T = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4$$

Con los datos obtenidos y recordando que

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

obtendremos las matrices correspondientes a cada una de las articulaciones.

$${}^0A_1 = \begin{bmatrix} cq_1 & -sq_1 & 0 & 0 \\ sq_1 & cq_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2A_3 = \begin{bmatrix} cq_2 & -sq_2 & 0 & l_3cq_2 \\ sq_2 & cq_2 & 0 & l_3sq_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Finalmente obtendremos la matriz de transformación homogénea como  $T = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4$

$$T = \begin{bmatrix} cq_1cq_2 - sq_1sq_2 & -cq_1sq_1 - sq_2cq_2 & 0 & cq_1cq_2l_3 - sq_1sq_2l_3 + cq_1l_2 \\ sq_1cq_2 + cq_1sq_2 & -sq_1sq_2 + cq_1cq_2 & 0 & sq_1cq_2l_3 + cq_1sq_2l_3 + sq_1l_2 \\ 0 & 0 & 1 & -q_3 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## Entorno de Trabajo del Manipulador SCARA. Simulación con MatLab

Ya obtenida la matriz de transformación homogénea, podremos utilizarla para obtener, junto con la ayuda de la herramienta Matlab, el entorno y volumen de trabajo en la cual el manipulador SCARA de 3 GDL podrá moverse y trabajar sobre la misma. La idea para obtener el volumen físico de trabajo de manipulador, es generar una gran cantidad de trayectoria, realizando movimientos del extremo final muy cercanas unas a las otras, y modulando todas sus articulaciones desde su punto mínimo hasta el máximo, tratando de obtener todas las combinaciones posibles entre estas. Sabemos que las combinaciones son infinitas, ya que los puntos intermedios de cada articulación son infinitos, pero servirá para darnos una muy certera noción del espacio.

Se tomará como ejemplo de análisis, y respetando la configuración del gráfico del SCARA es cuestión los siguientes valores:

### Ejemplo 1

$$\begin{aligned} l_1 &= 50; & -\frac{3}{4}\pi &\leq q_1 \leq \frac{3}{4}\pi \\ l_2 &= 20; & -\frac{\pi}{2} &\leq q_2 \leq \frac{\pi}{2} \\ l_3 &= 10; & 0 &\leq q_3 \leq 15 \end{aligned}$$

Con estos valores, y con las pretensiones detalladas, se ha desarrollado un script para correr en MatLab (archivo con extensión .m) con el fin de simular y generar el espacio buscado. A continuación se expone el script en cuestión.

```
%*****
%Inicio del Script para generar volumen de trabajo del SCARA
%*****

L1 = 50;
L2 = 20;
L3 = 10;
XYZ = [];
UVW = [0;0;0;1];

a = 0;
b = 0;

indx = 1;
for q3 = 0:1/2:15
    if a == 0, a = 1; else a = 0; end;
    if a == 0,
        for q1 = -pi*3/4:pi/50:pi*3/4
            if b == 0, b = 1; else b = 0; end;
            if b == 0,
                for q2 = -pi*1/2:pi/10:pi*1/2
                    A1 = [cos(q1) -sin(q1) 0 0; sin(q1) cos(q1) 0 0; 0 0 1 L1; 0 0 0 1];
                    A2 = [1 0 0 L2; 0 1 0 0; 0 0 1 0; 0 0 0 1];
                    A3 = [cos(q2) -sin(q2) 0 L3*cos(q2); sin(q2) cos(q2) 0 L3*sin(q2); 0 0 1 0; 0 0 0 1];
                    A4 = [1 0 0 0; 0 1 0 0; 0 0 1 -q3; 0 0 0 1];
                    T = A1 * A2 * A3 * A4;
                    XYZ(:,indx) = T * UVW;
                    indx = indx + 1;
                end
            end
        end
    else
        for q2 = pi*1/2:-pi/10:-pi*1/2
            A1 = [cos(q1) -sin(q1) 0 0; sin(q1) cos(q1) 0 0; 0 0 1 L1; 0 0 0 1];
            A2 = [1 0 0 L2; 0 1 0 0; 0 0 1 0; 0 0 0 1];
            A3 = [cos(q2) -sin(q2) 0 L3*cos(q2); sin(q2) cos(q2) 0 L3*sin(q2); 0 0 1 0; 0 0 0 1];
            A4 = [1 0 0 0; 0 1 0 0; 0 0 1 -q3; 0 0 0 1];
            T = A1 * A2 * A3 * A4;
            XYZ(:,indx) = T * UVW;
            indx = indx + 1;
        end
    end
end
else
    for q1 = pi*3/4:-pi/50:-pi*3/4
```

```

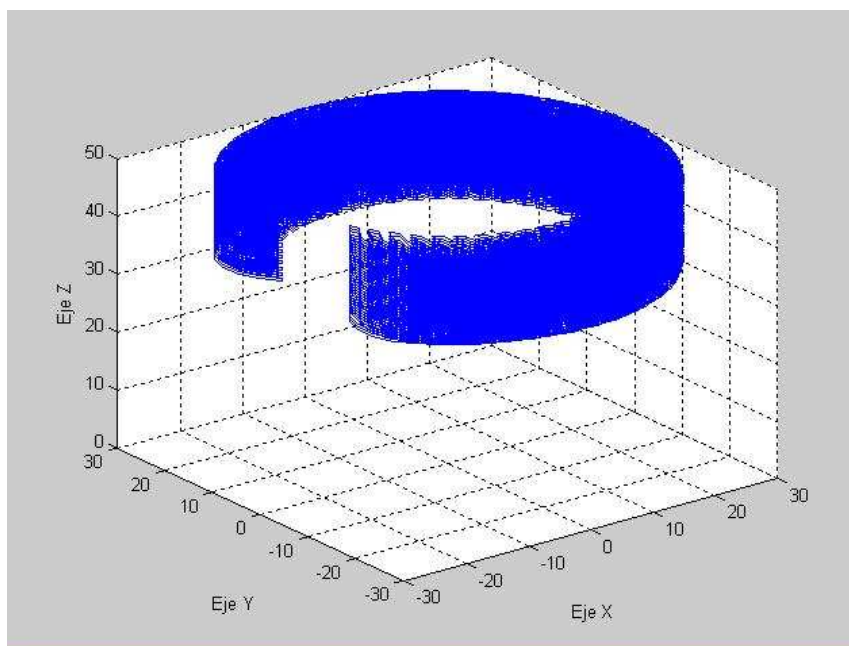
        if b == 0, b = 1; else b = 0; end;
        if b == 0,
            for q2 = -pi*1/2:pi/10:pi*1/2
                A1 = [cos(q1) -sin(q1) 0 0; sin(q1) cos(q1) 0 0; 0 0 1 L1; 0 0 0 1];
                A2 = [1 0 0 L2; 0 1 0 0; 0 0 1 0; 0 0 0 1];
                A3 = [cos(q2) -sin(q2) 0 L3*cos(q2); sin(q2) cos(q2) 0 L3*sin(q2); 0 0 1 0; 0 0 0 1];
                A4 = [1 0 0 0; 0 1 0 0; 0 0 1 -q3; 0 0 0 1];
                T = A1 * A2 * A3 * A4;
                XYZ(:,indx) = T * UVW;
                indx = indx + 1;
            end
        else
            for q2 = pi*1/2:-pi/10:-pi*1/2
                A1 = [cos(q1) -sin(q1) 0 0; sin(q1) cos(q1) 0 0; 0 0 1 L1; 0 0 0 1];
                A2 = [1 0 0 L2; 0 1 0 0; 0 0 1 0; 0 0 0 1];
                A3 = [cos(q2) -sin(q2) 0 L3*cos(q2); sin(q2) cos(q2) 0 L3*sin(q2); 0 0 1 0; 0 0 0 1];
                A4 = [1 0 0 0; 0 1 0 0; 0 0 1 -q3; 0 0 0 1];
                T = A1 * A2 * A3 * A4;
                XYZ(:,indx) = T * UVW;
                indx = indx + 1;
            end
        end
    end
end;

end;
end

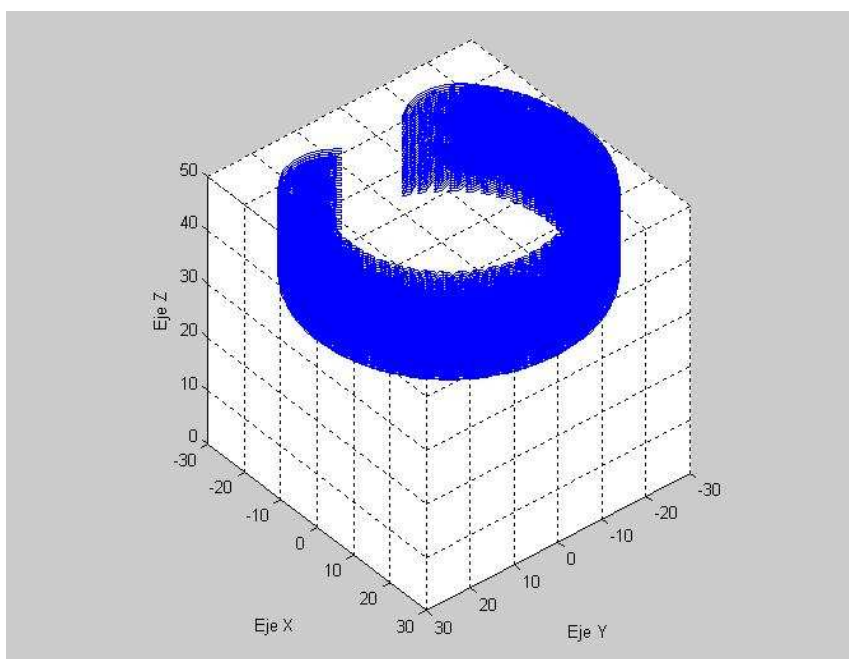
X = XYZ(1,:);
Y = XYZ(2,:);
Z = XYZ(3,:);
plot3(X,Y,Z);
colormap;
grid on;
xlabel('Eje X');
ylabel('Eje Y');
zlabel('Eje Z');
axis ([-30 30 -30 30 0 50]);
%*****
% Fin
%*****

```

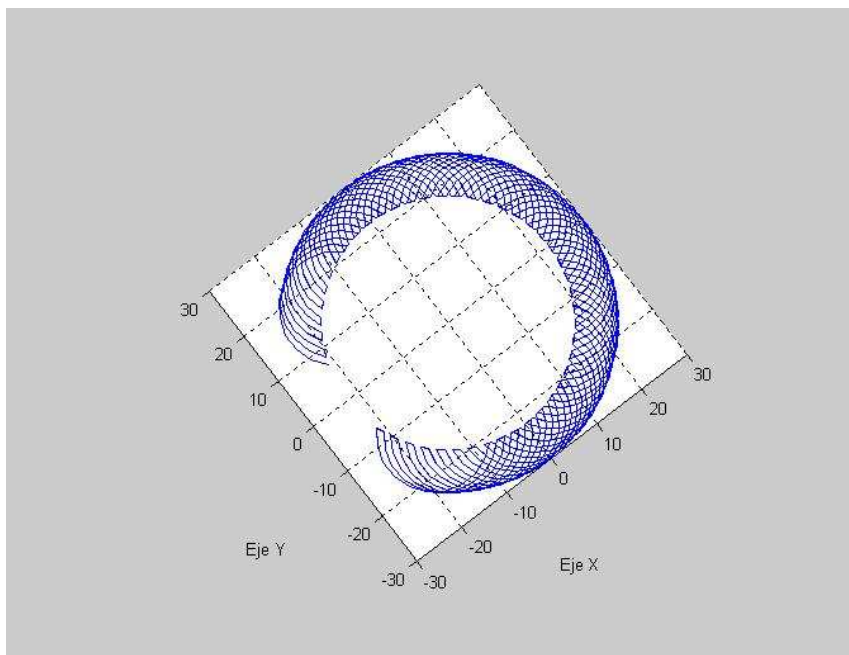
Ejecutando descrito recientemente, se obtiene las siguientes imágenes de nuestro manipulador.



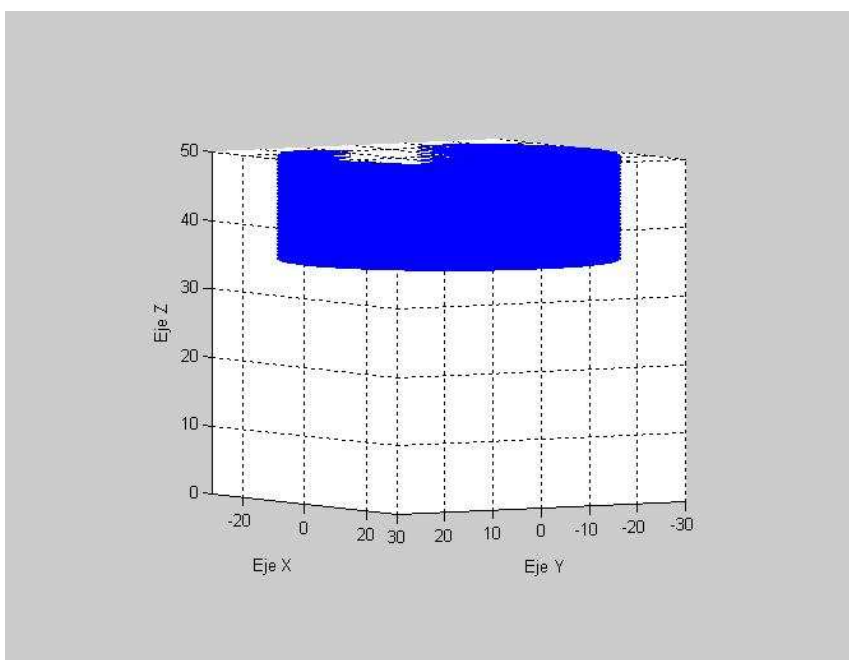
Volumen y Zona de Trabajo del Manipulador SCARA de 3 GDL



Volumen y Zona de trabajo desde Otro Ángulo



Vista Superior de la Zona de Trabajo del SCARA



Vista Semilateral de la Zona de Trabajo del SCARA

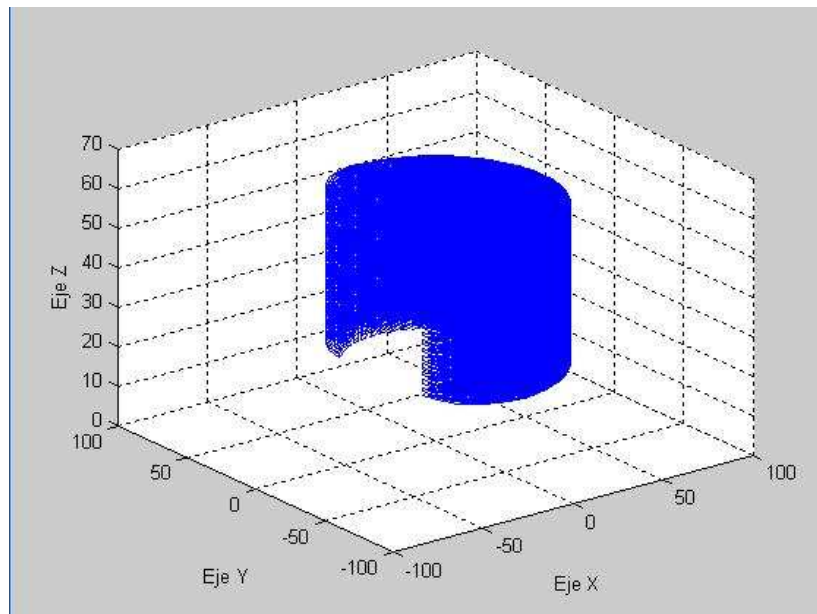
De similar manera se podrían obtener las distintas zonas de trabajo dependiendo las dimensiones y configuraciones del manipulador. Se sobreentiende que existe también una cantidad infinita de configuraciones, pero para terminar de cerrar un poco más el concepto, se mostrarán las siguientes configuraciones, adaptando oportuna y adecuadamente el script del MatLab.

### Ejemplo 2

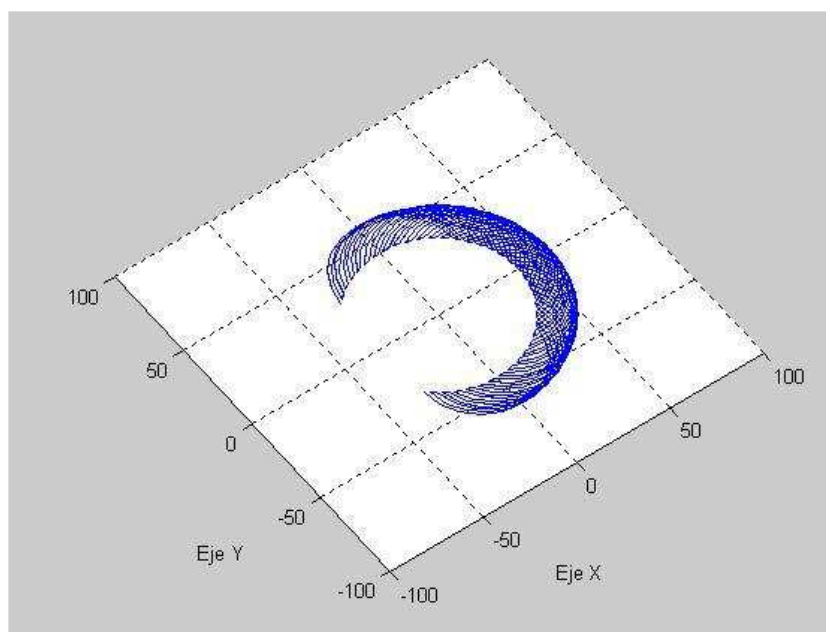
$$l_1 = 60; \quad -\frac{\pi}{2} \leq q_1 \leq \frac{\pi}{2}$$

$$\begin{aligned} 12 &= 30; & -\pi/2 \leq q_2 \leq \pi/2 \\ 13 &= 30; & 0 \leq q_3 \leq 40 \end{aligned}$$

A partir de esto se obtienen las siguientes imágenes:



Vista de Volumen Ejemplo 2

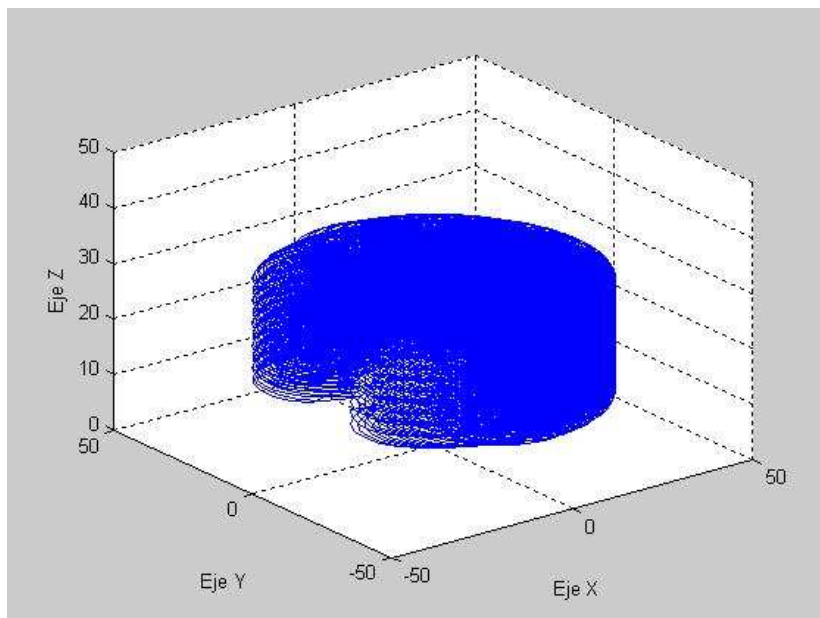


Vista Superior Ejemplo 2

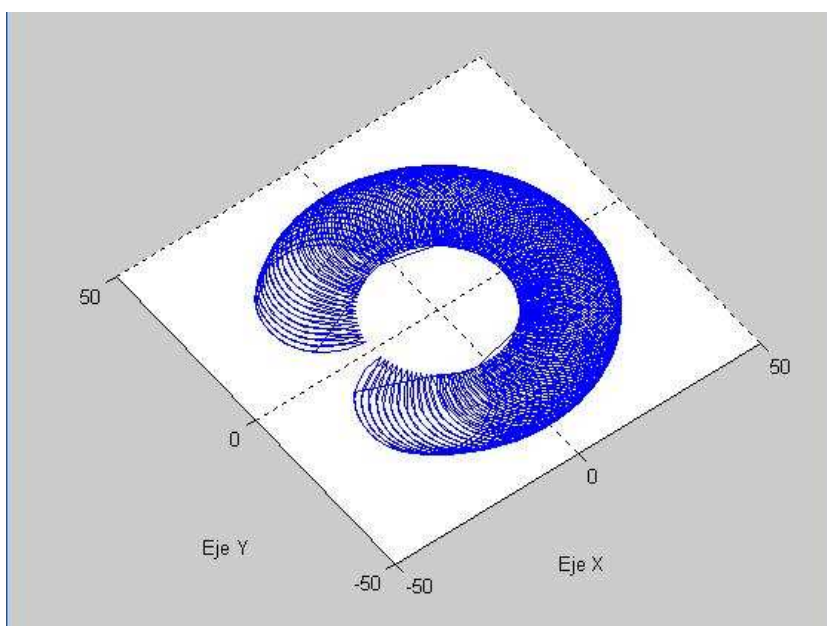
### Ejemplo 3

$$\begin{aligned} 11 &= 30; & -\frac{3}{3}\pi &\leq q_1 \leq \frac{3}{4}\pi \\ 12 &= 25; & -\frac{\pi}{2} &\leq q_2 \leq \frac{\pi}{2} \\ 13 &= 15; & 0 &\leq q_3 \leq 20 \end{aligned}$$

A partir de esto se obtienen las siguientes imágenes:



Vista Volumen Ejemplo 3



Vista Superior Ejemplo 3



Cabe hacer la aclaración que las variables del tipo prismáticas en estos ejemplos son adimensionales, ya que la simulación es a modo de ejemplo. En la realidad, estos valores corresponden a medidas de longitud.

Se observa pues entonces que con esta herramienta se puede obtener una idea muy certera del campo de trabajo que obtendrá el manipulador, pudiendo en etapa de diseño del mismo, confirmar las prestaciones o bien llevar a corrección.

## CINEMÁTICA INVERSA

El objetivo del problema cinemática inverso consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot  $q = [q_1, q_2, \dots, q_n]^T$  para que su extremo se posicione y oriente según una determinada localización espacial.

Así como es posible abordar el problema cinemática directo de una manera sistemática a partir de la utilización de matrices de transformación homogéneas, e independientemente de la configuración del robot, no ocurre lo mismo con el problema cinemática inverso, siendo el procedimiento de obtención de las ecuaciones fuertemente independiente de la configuración del robot.

Se han desarrollado algunos procedimientos genéricos susceptibles de ser programados, de modo que una computadora pueda, a partir del conocimiento de la cinemática del robot, obtener la n-uplas de valores que posicionan y orientan su extremo. El inconveniente de estos procedimientos es que se trata de procesos numéricos iterativos cuya velocidad de convergencia e incluso su convergencia en sí no está siempre garantizada.

A la hora de resolver el problema cinemático inverso es mucho más adecuado encontrar una solución cerrada. Esto es, encontrar una relación matemática explícita de la forma:

$$q_k = f_k(x, y, z, \alpha, \beta, \gamma)$$

$$k = 1 \dots n \quad \text{Grados de Libertad}$$

Este tipo de solución presenta, entre otras, las siguientes ventajas:

1. En muchas ocasiones, el problema cinemática inverso ha de resolverse en tiempo real, por ejemplo, en el seguimiento de una determinada trayectoria. Una solución del tipo iterativo no garantiza tener la solución en el momento adecuado.
2. Al contrario de lo que ocurría en el problema cinemática directo, con cierta frecuencia la solución del problema cinemática inverso no es única. Existiendo diferentes n-uplas que posicionan y orientan al extremo del robot del mismo modo. En estos casos una solución cerrada permite incluir determinadas reglas o restricciones que aseguren que la solución obtenida sea la más adecuada de entre las posibles.

No obstante, a pesar de la dificultades comentadas, la mayor parte de los robots poseen cinemáticas relativamente simples, que facilitan en cierta medida la resolución de su problema cinemático inverso. Por ejemplo, si se consideran sólo los tres primeros grados de libertad de muchos robots, éstos tienen una estructura planar, esto es, los tres primeros elementos quedan contenidos en un plano. Asimismo, en muchos robots, se da la circunstancia de que los tres grados de libertad últimos, dedicados fundamentalmente a orientar el extremo del robot, corresponden a

giros sobre ejes que se cortan en un punto. De nuevo esta situación facilita el cálculo de la n-upla correspondiente a la posición y orientación deseadas. Por lo tanto, para los casos citados y otros, es posible establecer ciertas pautas generales que permitan plantear y resolver el problema cinemática inverso de una manera sistemática.

Los métodos geométricos permiten generalmente obtener los valores de las primeras variables articulares, que son las que consiguen posicionar el robot. Para ello se utilizan relaciones trigonométricas y geométricas sobre los elementos del robot. Se suele recurrir a la resolución de triángulos formados por los elementos y articulaciones del robot.

Como alternativa para resolver el mismo problema se puede recurrir a manipular directamente las ecuaciones correspondientes al problema cinemático directo. Es decir, puesto que se establece la relación

$$\begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = [t_{ij}]$$

donde los elementos  $t_{ij}$  son función de las coordenadas articulares, es posible pensar que mediante ciertas combinaciones de las 12 ecuaciones planteadas, se puedan despejar las  $n$  variables articulares  $q_i$  en función de las componentes de los vectores  $n$ ,  $a$ ,  $o$  y  $p$ .

### **Resolución del Problema Cinemática Inverso por Métodos Geométricos Aplicado al Manipulador SCARA de 3 GDL**

Como se ha indicado, este procedimiento es adecuado para robots de pocos grados de libertad o para el caso que se consideren sólo los primeros grados de libertad, dedicados a posicionar el extremo.

El procedimiento en sí se basa en encontrar suficiente número de relaciones geométricas en las que intervendrán las coordenadas del robot, sus coordenadas articulares y las dimensiones físicas de sus elementos.

A continuación se desarrollará este método para nuestro manipulador SCARA de 3 grados de libertad. Para dicho análisis consideraremos el siguiente gráfico.

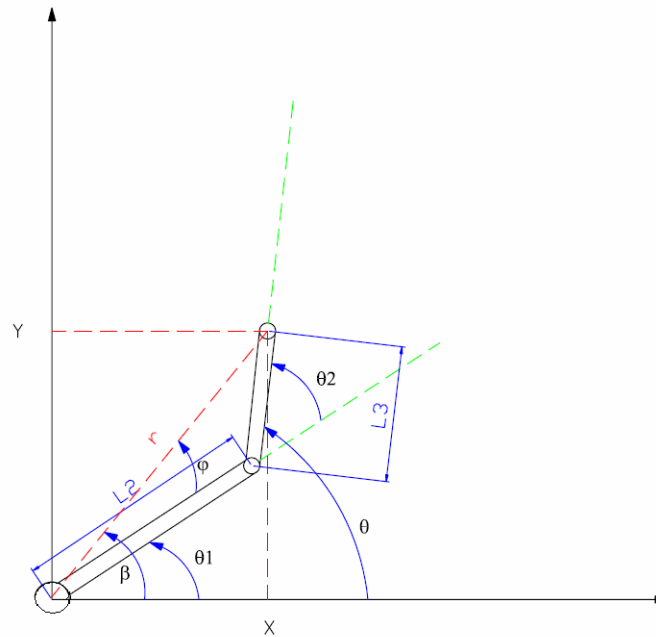


Diagrama Esquemático de la Vista Superior del SCARA

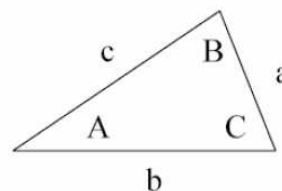
Según el gráfico indicado sabemos que

$$r^2 = x^2 + y^2$$

En este caso debemos aplicar la ley de cosenos para un triángulo general, que se detalla a continuación.

$$\frac{\text{sen}A}{a} = \frac{\text{sen}B}{b} = \frac{\text{sen}C}{c}$$

$$a^2 = b^2 + c^2 - 2bc \cos A$$



Aplicando esta ley obtenemos:

$$r^2 = l_2^2 + l_3^2 - 2l_2l_3 \cos(180^\circ + q_2)$$

También sabemos que

$$\cos(180^\circ + q_2) = -\cos q_2$$

Reemplazando esto en la ecuación anterior, y despejando  $q_2$  obtenemos

$$q_2 = \arccos \frac{x^2 + y^2 - l_2^2 - l_3^2}{2l_2l_3}$$

Por otro lado vemos que  $q_1 = \beta - \varphi$ . Debemos hallar entonces los valores de  $\beta$  y  $\varphi$ .

$$\operatorname{sen} \beta = \frac{y}{\sqrt{x^2 + y^2}} \quad ; \quad \beta = \arcsen \frac{y}{\sqrt{x^2 + y^2}}$$

Aplicando ley de cosenos

$$\cos \varphi = \frac{x^2 + y^2 + l_2^2 - l_3^2}{2l_2\sqrt{x^2 + y^2}} \quad ; \quad \varphi = \arccos \frac{x^2 + y^2 + l_2^2 - l_3^2}{2l_2\sqrt{x^2 + y^2}}$$

Reemplazando obtenemos finalmente

$$q_1 = \arcsen \frac{y}{\sqrt{x^2 + y^2}} - \arccos \frac{x^2 + y^2 + l_2^2 - l_3^2}{2l_2\sqrt{x^2 + y^2}}$$

Para obtener el valor de  $q_3$  se puede ver y deducir fácilmente que

$$q_3 = l_1 - z$$

### Resolución del Problema Cinemática Inverso a Partir de la Matriz de transformación Homogénea Aplicado al Manipulador SCARA de 3 GDL

En principio es posible tratar de obtener el modelo cinemática inverso de un robot a partir del conocimiento de su modelo directo. Es decir, suponiendo conocidas las relaciones que expresan el valor de la posición y orientación del extremo del robot en función de sus coordenadas articulares, obtener por manipulación de aquellas las relaciones inversas.

Sin embargo, en la práctica esta tarea no es trivial siendo en muchas ocasiones tan complejas que obliga a desecharlas. Además, puesto que el problema cinemática directa, contiene en el caso de un robot de 6 GDL 12 ecuaciones, y se buscan sólo 6 relaciones, una por cada grado de libertad, existirán necesariamente ciertas dependencias entre las 12 expresiones de partida, con lo cual la elección de qué ecuaciones escoger deber hacerse con cuidado.

El primer paso para resolver problema cinemática inverso es obtener la expresión

$$\begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = [t_{ij}]$$

correspondiente al manipulador SCARA en cuestión. Es decir, obtener la matriz T que relaciona el sistema de referencia {S0} asociado a la base con el sistema {S4} asociado a su extremo. Recordando la expresión obtenida con anterioridad

$$T = {}^0A_4 = \begin{bmatrix} cq_1cq_2 - sq_1sq_2 & -cq_1sq_2 - sq_2cq_2 & 0 & cq_1cq_2l_3 - sq_1sq_2l_3 + cq_1l_2 \\ sq_1cq_2 + cq_1sq_2 & -sq_1sq_2 + cq_1cq_2 & 0 & sq_1cq_2l_3 + cq_1sq_2l_3 + sq_1l_2 \\ 0 & 0 & 1 & -l_1q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Obtenida la expresión T en función de las coordenadas articulares  $q_1$ ,  $q_2$  y  $q_3$  y supuesta una localización de destino para el extremo del robot definida por los vectores n, o, a y p, se podría intentar manipular las 12 ecuaciones resultantes de T a fin de despejar  $q_1$ ,  $q_2$  y  $q_3$  en función de n, a, o y p.

Sin embargo este procedimiento directo es complejo. En lugar de ello aplicaremos el siguiente procedimiento:

Puesto que  $T = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4$  se tendrá que

$$({}^0A_1)^{-1} T = {}^1A_2 {}^2A_3 {}^3A_4$$

$$({}^1A_2)^{-1} ({}^0A_1)^{-1} T = {}^2A_3 {}^3A_4$$

$$({}^2A_3)^{-1} ({}^1A_2)^{-1} ({}^0A_1)^{-1} T = {}^3A_4$$

Sabiendo que  $T = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$  es conocida, los miembros a la izquierda de las expresiones son función de las variables articulares  $(q_1, \dots, q_k)$ , mientras que los miembros de la derecha son de las variables articulares  $(q_{k1}, \dots, q_n)$ .

De este modo, de la primera de las expresiones de se tendrá  $q_1$  aislado del resto de las variables articulares y tal vez será posible obtener su valor sin la complejidad que se tendría abordando directamente la manipulación. A su vez, una vez obtenida  $q_1$ , la segunda expresión permitirá tener el valor de  $q_2$  aislado respecto de  $q_3$ . Por último, conocidos  $q_1$  y  $q_2$  se podrá obtener  $q_3$  de la expresión sin excesiva dificultad.

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} n_x & n_y & n_z & -n^T p_x \\ o_x & o_y & o_z & -n^T p_y \\ a_x & a_y & a_z & -n^T p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0A_1^{-1} = \begin{bmatrix} cq_1 & -sq_1 & 0 & 0 \\ sq_1 & cq_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} cq_1 & sq_1 & 0 & 0 \\ -sq_1 & cq_1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$${}^2A_3^{-1} = \begin{bmatrix} cq_2 & -sq_2 & 0 & l_3cq_2 \\ sq_2 & cq_2 & 0 & l_3sq_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} cq_2 & sq_2 & 0 & -l_3c^2q_2 - l_3s^2q_2 \\ -sq_2 & cq_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3A_4^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left({}^0A_1\right)^{-1} {}^0T_4 = {}^1A_2 {}^2A_3 {}^3A_4$$

$$\begin{bmatrix} cq_1 & sq_1 & 0 & 0 \\ -sq_1 & cq_1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} cq_2 & -sq_2 & 0 & l_3cq_2 + l_2 \\ sq_2 & cq_2 & 0 & l_3sq_2 \\ 0 & 0 & 1 & -q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Es este punto, habría que buscar alguna ecuación que vincule a  $q_1$  con valores constantes y con  $p_x$ ,  $p_y$  y/o  $p_z$ , para así obtener el valor del mismo. En este caso no será posible, por lo cual podemos obtener en principio la siguiente relación:

$$cq_1p_x - sq_1p_y = l_3cq_2 + l_2$$

Luego:

$$\left({}^1A_2\right)^{-1} \left({}^0A_1\right)^{-1} {}^0T_4 = {}^2A_3 {}^3A_4$$

$$\begin{bmatrix} cq_1 & sq_1 & 0 & -l_2 \\ -sq_1 & cq_1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} cq_2 & -sq_2 & 0 & l_3cq_2 \\ sq_2 & cq_2 & 0 & l_3sq_2 \\ 0 & 0 & 1 & -q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$({}^2A_3)^{-1}({}^1A_2)^{-1}({}^0A_1)^{-1}{}^0T_4 = {}^3A_4$$

$$\begin{bmatrix} cq_1cq_2 - sq_1sq_2 & sq_1cq_2 + cq_1sq_2 & 0 & -cq_2l_2 - l_3c^2q_2 - l_3s^2q_2 \\ -cq_1sq_2 - sq_1cq_2 & -sq_1sq_2 + cq_1cq_2 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Aquí podemos rápidamente deducir que

$$q_3 = l_1 - p_z$$

Por otro lado podemos obtener

$$(-sq_2cq_1 - cq_2sq_1)p_x + (-sq_2sq_1 - cq_2cq_1)p_y = 0$$

Recordando el primer paso, tenemos un sistema de 2 ecuaciones con 2 incógnitas

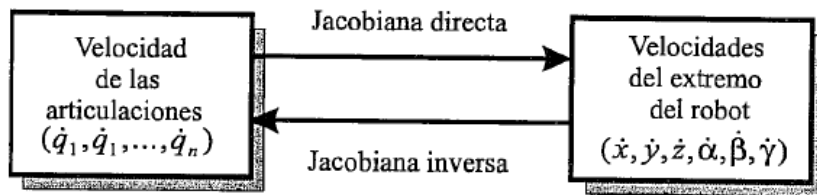
$$\begin{cases} cq_1p_x - sq_1p_y = l_3cq_2 + l_2 \\ (-sq_2cq_1 - cq_2sq_1)p_x + (-sq_2sq_1 - cq_2cq_1)p_y = 0 \end{cases}$$

Por medio de resolución matemática y trigonométrica, deberíamos llegar al mismo resultado que el obtenido de manera geométrica, o al menos, los resultados finales debería ser los mismos.

## Matriz Jacobiana

El modelado cinemática de un robot busca las relaciones entre las variables articulares y la posición y orientación del extremo del robot. En esta relación no se tiene en cuenta las fuerzas o pares que actúan sobre el robot (actuadores, cargas, fricciones, etc.) y que pueden generar el movimiento del mismo. Sin embargo, sí que debe permitir conocer la relación entre sus respectivas derivadas. Así, el sistema de control del robot, debe establecer qué velocidades debe imprimir a cada articulación, para conseguir que el extremo desarrolle una trayectoria temporal concreta, por ejemplo una línea recta a velocidad constante.

Para este y otros fines es de gran utilidad disponer de la relación entre las velocidades de las coordenadas articulares y de la posición y orientación del extremo del robot. La relación entre ambos vectores de velocidad se obtiene a través de la denominada Matriz Jacobiana.



## Relaciones diferenciales

El método más directo para obtener las relaciones entre las velocidades articulares y del extremo del robot consiste en diferenciar las ecuaciones correspondientes al modelo cinemática directo.

Así, conocidas las ecuaciones que resuelven el problema cinemática directo de un robot de  $n$  grados de libertad:

$$x = f_x(q_1, \dots, q_n) \quad y = f_y(q_1, \dots, q_n) \quad z = f_z(q_1, \dots, q_n) \alpha \beta \gamma$$

$$\alpha = f_\alpha(q_1, \dots, q_n) \quad \beta = f_\beta(q_1, \dots, q_n) \quad \gamma = f_\gamma(q_1, \dots, q_n)$$

Si se derivan ambos miembros del conjunto de ecuaciones anteriores se obtendrá:

$$J = \begin{bmatrix} x' \\ y' \\ z' \\ \alpha' \\ \beta' \\ \gamma' \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \dots & \frac{\partial f_x}{\partial q_n} \\ \vdots & \dots & \vdots \\ \frac{\partial f_\gamma}{\partial q_1} & \dots & \frac{\partial f_\gamma}{\partial q_n} \end{bmatrix}$$

La matriz  $J$  se denomina matriz Jacobiana. Puesto que cada uno de los elementos de la Jacobiana dependerá de los valores instantáneos de las coordenadas articulares, el valor de la Jacobiana será diferente en cada uno de los puntos del espacio articular.

Desarrollaremos el análisis del manipulador SCARA de 3 grados de libertad que es nuestro elemento en cuestión. Según lo visto recientemente, recordaremos la matriz de transformación homogénea:



$$T = \begin{bmatrix} cq_1cq_2 - sq_1sq_2 & -cq_1sq_1 - sq_2cq_2 & 0 & cq_1cq_2l_3 - sq_1sq_2l_3 + cq_1l_2 \\ sq_1cq_2 + cq_1sq_2 & -sq_1sq_2 + cq_1cq_2 & 0 & sq_1cq_2l_3 + cq_1sq_2l_3 + sq_1l_2 \\ 0 & 0 & 1 & -q_3 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

En este caso, la matriz Jacobiana estará dada por:

$$J = \begin{bmatrix} \frac{\partial p_x}{\partial q_1} & \frac{\partial p_x}{\partial q_2} & \frac{\partial p_x}{\partial q_3} \\ \frac{\partial p_y}{\partial q_1} & \frac{\partial p_y}{\partial q_2} & \frac{\partial p_y}{\partial q_3} \\ \frac{\partial p_z}{\partial q_1} & \frac{\partial p_z}{\partial q_2} & \frac{\partial p_z}{\partial q_3} \end{bmatrix}$$

De la matriz de transformación homogénea obtenemos directamente los valores de  $p_x$ ,  $p_y$  y  $p_z$ .

$$p_x = cq_1cq_2l_3 - sq_1sq_2l_3 + cq_1l_2$$

$$p_y = sq_1cq_2l_3 + cq_1sq_2l_3 + sq_1l_2$$

$$p_z = -q_3 + l_1$$

Sólo resta obtener las derivadas parciales en cada caso para finalmente obtener la matriz Jacobiana. De esta manera quedaría:

$$J = \begin{bmatrix} sq_1cq_2l_3 + cq_1sq_2l_3 - sq_1l_2 & -cq_1sq_2l_3 - sq_1cq_2l_3 & 0 \\ cq_1cq_2l_3 - sq_1sq_2l_3 + sq_1l_2 & cq_1cq_2l_3 - sq_1sq_2l_3 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

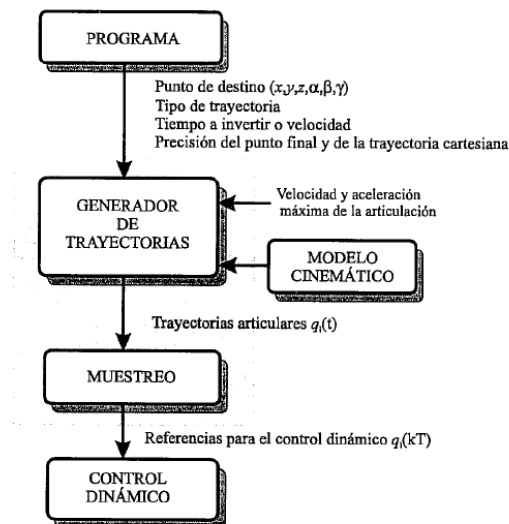
## Control Cinemático

El control cinemático establece cuáles son las trayectorias que debe seguir cada articulación del robot a lo largo del tiempo para lograr los objetivos fijados por el usuario, como punto destino trayectoria cartesiana del efector final del robot, tiempo invertido por el usuario, etc. Estas trayectorias se seleccionarán atendiendo a las restricciones físicas propias de los accionamientos y a ciertos criterios de calidad de trayectoria, como suavidad y precisión de la misma.

### Funciones del Control Cinemático

De manera general, el control cinemático deberá realizar las siguientes funciones:

1. Convertir la especificación de movimiento dada en el programa en una trayectoria analítica en espacio artesiano.
2. Muestrear la trayectoria cartesiana obteniendo un número finito de puntos de dicha trayectoria.
3. Utilizando la transformación homogénea inversa, convertir cada uno de estos puntos en sus correspondientes coordenadas articulares. Debe tenerse en cuenta aquí la posibilidad de soluciones múltiples de la transformación homogénea inversa, así como la posibilidad de ausencia de solución y puntos singulares, de modo que se asegure la continuidad de la trayectoria.
4. Interpolación de los puntos articulares obtenidos, generando para cada variable particular una expresión  $q_i(t)$  que pase o se aproxime por ellos de modo que, siendo una trayectoria realizable por los actuadores, se transforme en una trayectoria cartesiana lo más próxima a la especificada por el usuario.
5. Muestreo de la trayectoria articular para generar referencias al control dinámico.



Como se describió anteriormente, la matriz Jacobiana establece las relaciones diferenciales entre variables articulares y cartesianas, que se puede expresar como:

$$\dot{j}(t) = J(q)\dot{q}(t)$$

Esta relación es lineal y para una localización  $q$  determinada permite conocer las velocidades cartesianas a partir de las correspondientes articulares. Además, suponiendo  $J(q)$  invertible, lo que ocurrirá siempre que las dimensiones del espacio articular y de la tarea sean iguales y  $q$  no sea un punto singular, se podrá expresar:

$$\dot{q}(t) = J(q)^{-1} \dot{j}(t)$$

La implementación de este procedimiento en la unidad de control de un robot, exigirá la discretización de la trayectoria  $j(t)$  y su derivada en un número finito de puntos que posteriormente, y tras ser convertidos en velocidades articulares deberán ser interpolados.

Por lo tanto, para estudiar el control cinemático de un robot se deberá fundamentalmente conocer qué tipo de interpoladores pueden ser eficaces para unir los puntos articulares por los que se quiere pasar. Asimismo, será conveniente establecer criterios para seleccionar cuántos y qué puntos serán muestreados en la trayectoria del espacio cartesiano

### **Tipos de Trayectorias**

Para realizar una tarea determinada el robot debe moverse desde un punto inicial a un punto final. Este movimiento puede ser realizado según infinitas trayectorias espaciales. De todas ellas hay algunas que, bien por su sencillez de implementación por panel del control cinemático o bien por su utilidad y aplicación a diversas tareas, son las que en la práctica incorporan los robots comerciales. De este modo, puede encontrarse que los robots dispongan de trayectorias punto a punto, coordinadas y continuas.

#### **Trayectorias Punto a Punto**

En este tipo de trayectorias cada articulación evoluciona desde su posición inicial a la final sin realizar consideración alguna sobre el estado o evolución de las demás articulaciones. Normalmente, cada actuador trata de llevar a su articulación al punto de destino en el menor tiempo posible, pudiéndose distinguir dos casos: movimiento eje a eje y movimiento simultáneo de ejes.

#### **Movimiento de Ejes**

Sólo se mueve un eje cada vez. Comenzará a moverse la primera articulación, y una vez que esta haya alcanzado su punto final lo hará la segunda, y así sucesivamente. Este tipo de movimiento da obviamente como resultado un mayor tiempo de ciclo, teniendo como única ventaja un menor consumo de potencia instantánea por parte de los actuadores.

#### **Movimiento simultáneo de ejes**

En este caso todos los actuadores comienzan simultáneamente a mover las articulaciones del robot a una velocidad específica para cada una de ellas. Dado que la distancia a recorrer y las velocidades serán en general diferentes, cada una acabará su movimiento en un instante diferente.

El movimiento del robot no acabará hasta que se alcance definitivamente el punto final, lo que se producirá cuando el eje que más tarde concluya su movimiento. De esta manera, el tiempo total invertido en el movimiento coincidirá con el del eje que más tiempo emplee en realizar su movimiento particular, pudiéndose dar la circunstancia de que el resto de los actuadores hayan forzado su movimiento a una velocidad y aceleración elevada, viéndose obligados finalmente a esperar a la articulación más lenta.

Por los motivos expuestos, las trayectorias punto a punto no están implementadas salvo en robots muy simples o con unidades de control muy limitadas

#### **Trayectorias coordinadas o isócronas**

Para evitar que algunos actuadores trabajen forzando sus velocidades y aceleraciones. Teniendo que esperar después la conclusión del movimiento de la articulación más lenta, puede hacer un cálculo previo, averiguando cuál es esta articulación y qué tiempo invertirá. Se ralentizará

entonces el movimiento del resto de los ejes para que inviertan el mismo tiempo en su movimiento, acabando todos ellos simultáneamente. Se tiene así que todas las articulaciones se coordinan comenzando y acabando su movimiento a la vez, adaptándose todas a la más lenta.

El tiempo total invertido en el movimiento es el menor posible y no se piden aceleraciones y velocidades elevadas a los actuadores de manera inútil. Desde el punto de vista del usuario la trayectoria que describe el extremo del robot no es significativa, siendo ésta impredecible. Aunque como es obvio, un conocimiento del modelo y control cinemático del robot permitiría su cálculo.

### **Trayectoria Continua**

Cuando se pretende que la trayectoria que sigue el extremo del robot sea conocida por el usuario (trayectoria en el espacio cartesiano o de la tarea), es preciso calcular de manera continua las trayectorias articulares.

Típicamente, las trayectorias que el usuario pretende que el robot describa son trayectorias en línea recta o en arco de círculo. Para conseguirlo habrá que seguir los pasos indicados en el epígrafe 6.1 El resultado será que cada articulación sigue un movimiento aparentemente caótico con posibles cambios de dirección y velocidad y sin coordinación con el resto de las articulaciones. Sin embargo el resultado será que el extremo del robot describirá la trayectoria deseada.

### **Generación de Trayectorias**

Normalmente el usuario del robot indica el movimiento que éste debe realizar especificando las localizaciones espaciales por las que debe pasar el extremo, junto con otros datos, como instantes de paso, velocidades o tipos de trayectorias. Así, por ejemplo, es frecuente especificar que el robot debe ir de un punto inicial hasta otro final, siguiendo en cartesianas una línea recta a velocidad constante.

Puesto que estos puntos están excesivamente separados es preciso seleccionar puntos intermedios suficientemente cercanos como para que el control del robot consiga ajustar no sólo el punto final al especificado, sino también la trayectoria seguida a la indicada en el programa.

Para ello es preciso establecer un interpolador entre las localizaciones expresadas en el espacio de la tarea que dará como resultado una expresión analítica de la evolución de cada coordenada. La interpolación más frecuente es la lineal, en la que cada coordenada evoluciona a velocidad constante desde su valor inicial  $j^i$  hasta el final  $j^f$

$$j(t) = (j^f - j^i) \frac{t - t_i}{t_f - t_i} + j^i$$

donde  $t^i$  y  $t^f$  son los instantes de tiempo en los que se pretende alcanzar la localización inicial y final, respectivamente.

### **Interpolación de Trayectorias**

Como se ha indicado, una de las funciones del control cinemático es la de unir una sucesión de puntos en el espacio articular por los que se quiere que pasen las articulaciones del robot en un instante determinado. Además, junto con las condiciones de posición-tiempo, es conveniente añadir restricciones en la velocidad y aceleración de paso por los puntos, de manera que se asegure la suavidad de la trayectoria y se limiten las velocidades y aceleraciones máximas. Estas restricciones garantizarán que los actuadores están capacitados para implementar la trayectoria final.

Para ello deberá seleccionarse algún Upo de función (frecuentemente polinómica) cuyos parámetros o coeficientes se ajustarán al imponer las condiciones de contorno: posiciones, velocidades y aceleraciones. En la selección de esta función debe considerarse que tanto el cálculo de sus parámetros, como su posterior utilización para generar puntos de consigna al control dinámico, debe hacerse en tiempo real, por lo que la simplicidad de la función será un factor a valorar.

Se van a presentar a continuación las funciones interpoladoras utilizadas con mayor frecuencia. Cada una de ellas ha sido desarrollada para un solo grado de libertad, debiendo quedar claro que el mismo cálculo deberá repetirse para cada uno de los grados de libertad del robot. Cabe indicar que si bien las técnicas de interpolación que a continuación se describen están planteadas para el espacio articular son igualmente aplicables para el espacio de la tarea.

### Interpoladores Lineales

Supóngase que se pretende que una de las articulaciones  $q$  del robot, pase sucesivamente por los valores  $q_i$  en los instantes  $t_i$ . Una primera solución a este problema consistiría en mantener constante la velocidad de movimiento entre cada 2 valores sucesivos  $(q^{i-1}; q^i)$  de la articulación.

La trayectoria entre dos puntos  $q^{i-1}; q^i$  sería entonces:

$$q(t) = (q^i - q^{i-1}) \frac{t - t^{i-1}}{T} + q^{i-1}$$

Como es obvio, esta trayectoria asegura la continuidad de la posición, pero origina saltos bruscos en la velocidad  $\dot{q}$  de la articulación, y consecuentemente precisa de aceleraciones  $\ddot{q}$  de valor infinito lo que en la práctica no es posible.

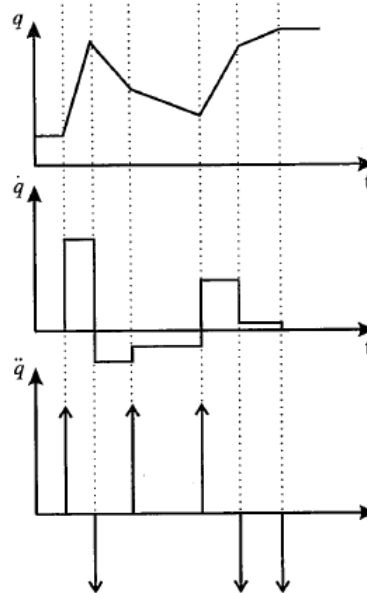
La selección de los instantes de paso  $i$  por los puntos  $q$  podrá haberse hecho según los diferentes criterios expuestos son:

1. Intentando que cada articulación  $q$  alcance el punto de destino en el menor tiempo posible sin considerar las demás articulaciones, lo que resultará en velocidades  $\dot{q}$  constantes e iguales a la máxima.
2. Ajustando los instantes de paso a los de la articulación que más tiempo precise, resultando movimientos coordinados.
3. Seleccionando los tiempos a partir de las especificaciones dadas en el espacio de la tarea de modo que el extremo del robot describa una trayectoria predeterminada.

### Interpoladores Cúbicos

Para asegurar que la trayectoria que une los puntos por los que tiene que pasar la articulación considerada presente continuidad en velocidad, puede recurrirse a utilizar un polinomio de grado 3 que una cada pareja de puntos adyacentes. De este modo, al tener cuatro parámetros disponibles se podrán imponer cuatro condiciones de contorno, dos de posición y dos de velocidad. Los valores de las velocidades de paso por cada punto deben por tanto ser conocidas a priori. Esta selección no

es obvia y se tratará posteriormente. Se consigue así una trayectoria compuesta por una serie de polinomios cúbicos, cada uno válido entre dos puntos consecutivos. Este conjunto de polinomios concatenados, escogidos de modo que exista continuidad en la posición y velocidad, se denominan splines (cúbicos, por ser de tercer grado).

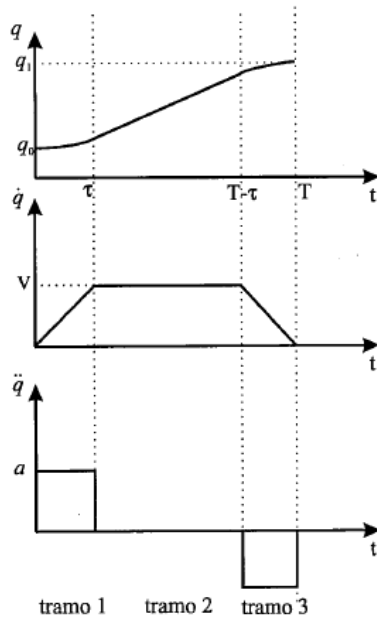


Posición, velocidad y aceleración para un interpolador lineal.

### Interpoladores a Tramos

En los interpoladores vistos hasta el momento, se utiliza un polinomio de un grado determinado (1 o 3) para unir dos puntos consecutivos de la trayectoria. El uso de polinomios de tercer grado permite asegurar que el polinomio pasa por los dos puntos y al mismo tiempo permite imponer los valores de velocidad de paso por los mismos. Sin embargo, al contrario de lo que ocurre con el interpolador de primer grado (lineal), la velocidad de la articulación durante el recorrido está variando continuamente, lo que exige un control continuo de la misma.

Una alternativa que proporciona una solución intermedia consiste en descomponer en tres tramos consecutivos la trayectoria que une dos puntos  $q_0$ ,  $q_1$ . En el tramo central se utiliza un interpolador lineal, y por lo tanto la velocidad se mantiene constante, no siendo preciso imprimir aceleración alguna al actuador. En los tramos inicial y final se utiliza un polinomio de segundo grado, de modo que en el tramo 1 la velocidad varíe linealmente desde la velocidad de la trayectoria anterior a la de la presente y en el tramo 3 varíe desde la velocidad de la trayectoria presente hasta la de la siguiente. Se tiene entonces que en los tramos inicial y final la aceleración toma valores constantes distintos de cero, mientras que en el tramo intermedio la aceleración es nula.



Interpolador con tres tramos

## Simulación de la Trayectoria del Manipulador por Cinemática Directa

Con el manipulador SCARA en estudio, y con los datos que hasta ahora venimos desarrollando, vamos a simular la trayectoria del brazo desde un punto inicial hasta uno final, obteniendo y pudiendo visualizar y tener una idea de cómo varían los estados de las articulaciones y como estaría trabajando el manipulador.

Para esto vamos a utilizar al Toolbox Hemero para MatLab, el cual nos permitirá generar lo recientemente mencionado.

Partimos de que las dimensiones del brazo son:

$$l_1 = 50; \quad l_2 = 20; \quad l_3 = 30$$

Como primer ejemplo vamos a partir del estado inicial

$$q_1 = 0; \quad q_2 = 0; \quad q_3 = 0$$

y pretendemos llegar de manera eficiente a

$$q_1 = \pi/4; \quad q_2 = \pi/4; \quad q_3 = 20$$

En este caso, como en los anteriores se deberá realizar un script para poder ejecutar la aplicación, con la ayuda del Hemero, para poder obtener los resultados. El script utilizado para esta ocasión es el que se detalla a continuación.

```
% *****
```

```
% Analisis cinematico del manipulador SCARA
% *****
% Define las dimensiones de los eslabones del manipulador.
L1 = 50;
L2 = 20;
L3 = 10;

% Variables y buffers.
XYZ = [];
% Punto de inicial en coordenadas articulares
q3 = 0;
q2 = 0;
q1 = 0;

A1 = [cos(q1) -sin(q1) 0 0; sin(q1) cos(q1) 0 0; 0 0 1 L1; 0 0 0 1];
A2 = [1 0 0 L2; 0 1 0 0; 0 0 1 0; 0 0 0 1];
A3 = [cos(q2) -sin(q2) 0 L3*cos(q2); sin(q2) cos(q2) 0 L3*sin(q2); 0 0 1 0; 0 0 0 1];
A4 = [1 0 0 0; 0 1 0 0; 0 0 1 -q3; 0 0 0 1];

T = A1 * A2 * A3 * A4;
TG(1, :, 1) = T(1, :);
TG(2, :, 1) = T(2, :);
TG(3, :, 1) = T(3, :);

% Punto destino en coordenadas articulares
q3 = 20;
q2 = pi/4;
q1 = pi/4;

A1 = [cos(q1) -sin(q1) 0 0; sin(q1) cos(q1) 0 0; 0 0 1 L1; 0 0 0 1];
A2 = [1 0 0 L2; 0 1 0 0; 0 0 1 0; 0 0 0 1];
A3 = [cos(q2) -sin(q2) 0 L3*cos(q2); sin(q2) cos(q2) 0 L3*sin(q2); 0 0 1 0; 0 0 0 1];
A4 = [1 0 0 0; 0 1 0 0; 0 0 1 -q3; 0 0 0 1];

% Apartir de las anteriores calcula la matriz de transformacion homogenea.
T = A1 * A2 * A3 * A4;
TG(1, :, 2) = T(1, :);
TG(2, :, 2) = T(2, :);
TG(3, :, 2) = T(3, :);
```



```
SCARA = [0 0 q1 L1 0; 0 L2 0 0 0; 0 L3 q2 0 0; 0 0 0 -q3 1];
```

```
%Q = [q1 q2 0 -q3];
```

```
stol = 0.1; ilimit = 1000;
```

```
% TG contiene dos matrices 3x3 de traslacion, una con las coordenadas del punto de inicio
```

```
% y la siguiente matriz de traslacion contiene las coordenadas del punto destino de brazo robot.
```

```
Q = IKINE(SCARA, stol, ilimit, TG)
```

```
%subplot(2,1,2);
```

```
%PLOTBOT(SCARA, Q, 'l');
```

```
%axis ([-30 30 -30 30 0 50]);
```

```
q = transpose (Q);
```

```
qd = [0 0; 0 0; 0 0; 0 0];
```

```
qdd = [0 0; 0 0; 0 0; 0 0];
```

```
to = 0;
```

```
tij = 1;
```

```
[Qcoef, time, q_t, qd_t, qdd_t] = pol3(q, qd, to, tij, 0.1);
```

```
%plot(time, q_t, 'K');
```

```
puntos = transpose(q_t);
```

```
figure(1);
```

```
%subplot(3,1,1);
```

```
PLOTBOT(SCARA, puntos, 'l');
```

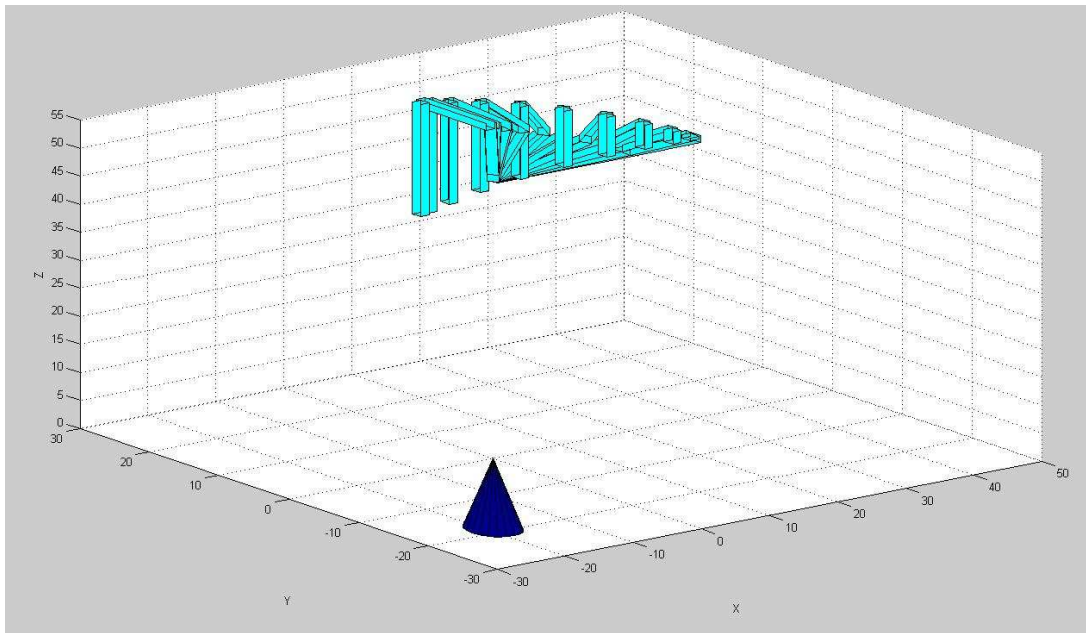
```
axis ([-30 50 -30 30 0 55]);
```

```
% *****
```

```
% Fin
```

```
% *****
```

Ejecutando este script desde Matlab podemos observar los resultados.



Trayectoria del SCARA

Se puede observar claramente cómo el brazo y las articulaciones van cambiando su valor y recorrido partiendo desde el punto inicial, hasta que su extremo logra alcanzar el punto final.

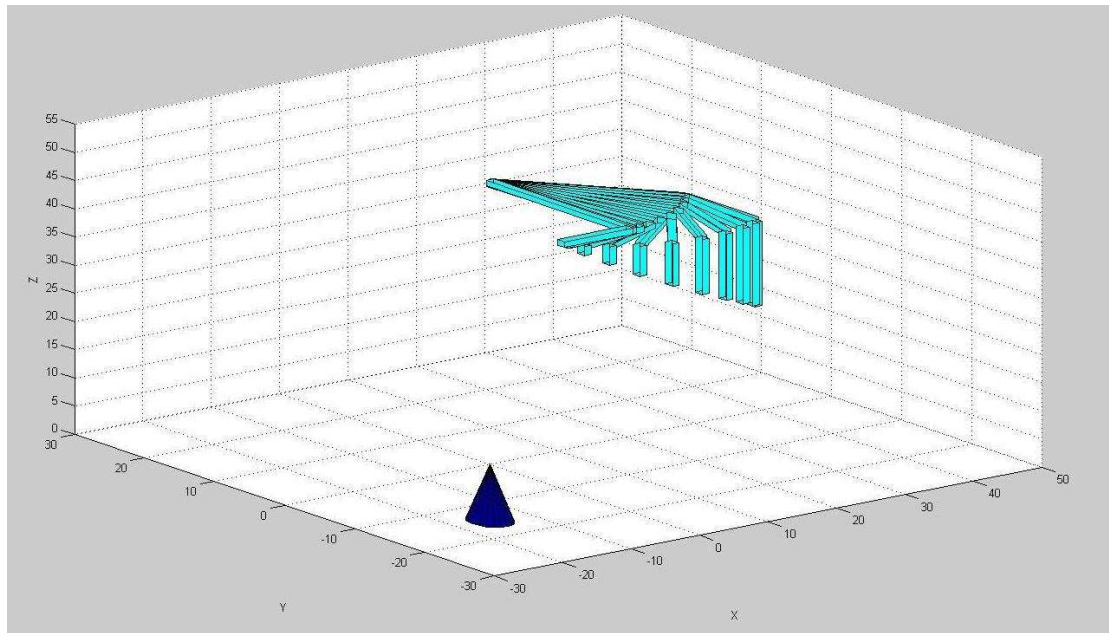
De similar manera lo podríamos observar si el punto de partida sería

$$q_1 = -\pi/4; \quad q_2 = -\pi/4; \quad q_3 = 15$$

y si deseamos finalizar en

$$q_1 = -\pi/2; \quad q_2 = -\pi/2; \quad q_3 = 0$$

Nuevamente modificando el script correspondiente con los datos correspondientes, y corriendo el mismo en MatLab, obtenemos:



Trayectoria del SCARA

Así se podrían obtener los distintos recorridos y ver cómo evolucionan las articulaciones del SCARA y qué trayectoria mantiene el extremo del mismo.

Otro análisis importante en cuestión de la cinemática, es ver cómo evolucionan las articulaciones, no sólo en su posición sino también en cuanto a velocidad y aceleración, de qué manera y como son sus curvas. Para esto, continuaremos utilizando la herramienta Hemero y el MatLab, buscando obtener las variables mencionadas, y graficarlas para ver su evolución. Para hacerlo más interesante y descriptivo, buscaremos ir desde el punto inicial al final en tres casos distintos, pero con distintas trayectorias cinemáticas, las cuales ya fueron mencionadas y descriptas con anterioridad. Para este caso las realizaremos con Polinómica de grado 3, Polinómica de grado 5 y en funciones lineales con enlaces parabólicos.

A continuación se muestra el script utilizado:

```
% *****
% Analisis cinematico directo del manipulador SCARA
% *****

% Define las dimensiones de los eslabones del manipulador.
L1 = 50;
L2 = 20;
L3 = 10;

% Variables y buffers.
XYZ = [];

% Estas son las coordenadas (origen) del extremo del efector final respecto al sistema S4.
UVW = [0;0;0;1];
```

% Punto de inicial en coordenadas articulares

q3 = 0;

q2 = -pi/4;

q1 = -pi/4;

A1 = [cos(q1) -sin(q1) 0 0; sin(q1) cos(q1) 0 0; 0 0 1 L1; 0 0 0 1];

A2 = [1 0 0 L2; 0 1 0 0; 0 0 1 0; 0 0 0 1];

A3 = [cos(q2) -sin(q2) 0 L3\*cos(q2); sin(q2) cos(q2) 0 L3\*sin(q2); 0 0 1 0; 0 0 0 1];

A4 = [1 0 0 0; 0 1 0 0; 0 0 1 -q3; 0 0 0 1];

T = A1 \* A2 \* A3 \* A4;

TG(1, :, 1) = T(1, :);

TG(2, :, 1) = T(2, :);

TG(3, :, 1) = T(3, :);

% Punto destino en coordenadas articulares

q3 = 15;

q2 = -pi/2;

q1 = -pi/2;

A1 = [cos(q1) -sin(q1) 0 0; sin(q1) cos(q1) 0 0; 0 0 1 L1; 0 0 0 1];

A2 = [1 0 0 L2; 0 1 0 0; 0 0 1 0; 0 0 0 1];

A3 = [cos(q2) -sin(q2) 0 L3\*cos(q2); sin(q2) cos(q2) 0 L3\*sin(q2); 0 0 1 0; 0 0 0 1];

A4 = [1 0 0 0; 0 1 0 0; 0 0 1 -q3; 0 0 0 1];

% Apartir de las anteriores calcula la matriz de transformacion homogenea.

T = A1 \* A2 \* A3 \* A4;

TG(1, :, 2) = T(1, :);

TG(2, :, 2) = T(2, :);

TG(3, :, 2) = T(3, :);

SCARA = [0 0 q1 L1 0; 0 L2 0 0 0; 0 L3 q2 0 0; 0 0 0 -q3 1];

%Q = [q1 q2 0 -q3];

stol = 0.1; ilimit = 1000;

% TG contiene dos matrices 3x3 de traslacion, una con las coordenadas del punto de inicio

% y la siguiente matriz de traslacion contiene las coordenadas del punto destino de brazo robot.

Q = IKINE(SCARA, stol, ilimit, TG)

```
%subplot(2,1,2);
%PLOTBOT(SCARA, Q, 'l');
%axis ([-30 30 -30 30 0 50]);

q = transpose (Q);
qd = [0 0; 0 0; 0 0; 0 0];
qdd = [0 0; 0 0; 0 0; 0 0];
to = 0;
tij = 1;

[Qcoef, time, q_t, qd_t, qdd_t] = pol3(q, qd, to, tij, 0.1);
puntos = transpose(q_t);

figure(1);
PLOTBOT(SCARA, puntos, 'l');
axis ([-30 50 -30 30 0 55]);

figure(2);
subplot(3,1,1);
plot(time,q_t(1,:), 'r');
hold on;
plot(time,q_t(2,:), 'g');
plot(time,q_t(3,:), 'b');
plot(time,q_t(4,:), 'k');
xlabel('tiempo(seg)'), ylabel('q(rad)'), grid;
hold off;

subplot(3,1,2);
plot(time,qd_t(1,:), 'r');
hold on;
plot(time,qd_t(2,:), 'g');
plot(time,qd_t(3,:), 'b');
plot(time,qd_t(4,:), 'k');
xlabel('tiempo(seg)'), ylabel('q''(rad)'), grid;
hold off;

subplot(3,1,3);
plot(time,qdd_t(1,:), 'r');
hold on;
```

```
plot(time,qdd_t(2,:), 'g');  
plot(time,qdd_t(3,:), 'b');  
plot(time,qdd_t(4,:), 'k'),  
xlabel('tiempo(seg)'), ylabel('q""(rad)'), grid;  
hold off;
```

```
[Qcoef, time, q_t, qd_t, qdd_t] = pol5(q, qd, qdd, to, tij, 0.1);  
puntos2 = transpose(q_t);
```

```
figure(3);  
PLOTBOT(SCARA, puntos2, 'l');  
axis ([-30 50 -30 30 0 55]);
```

```
figure(4);  
subplot(3,1,1);  
plot(time,q_t(1,:), 'r');  
hold on;  
plot(time,q_t(2,:), 'g');  
plot(time,q_t(3,:), 'b');  
plot(time,q_t(4,:), 'k');  
xlabel('tiempo(seg)'), ylabel('q(rad)'), grid;  
hold off;
```

```
subplot(3,1,2);  
plot(time,qd_t(1,:), 'r');  
hold on;  
plot(time,qd_t(2,:), 'g');  
plot(time,qd_t(3,:), 'b');  
plot(time,qd_t(4,:), 'k');  
xlabel('tiempo(seg)'), ylabel('q'(rad)'), grid;  
hold off;
```

```
subplot(3,1,3);  
plot(time,qdd_t(1,:), 'r');  
hold on;  
plot(time,qdd_t(2,:), 'g');  
plot(time,qdd_t(3,:), 'b');  
plot(time,qdd_t(4,:), 'k'),  
xlabel('tiempo(seg)'), ylabel('q""(rad)'), grid;  
hold off;
```

```

qdd = [10 10; 10 10; 10 10; 10 10];
[Qcoef, time, q_t, qd_t, qdd_t, intervals] = parab(q, qdd, to, tij, 0.1);
puntos3 = transpose(q_t);

figure(5);
PLOTBOT(SCARA, puntos2, 'l');
axis ([-30 50 -30 30 0 55]);

figure(6);
subplot(3,1,1);
plot(time,q_t(1,:), 'r');
hold on;
plot(time,q_t(2,:), 'g');
plot(time,q_t(3,:), 'b');
plot(time,q_t(4,:), 'k');
xlabel('tiempo(seg)'), ylabel('q(rad)'), grid;
hold off;

subplot(3,1,2);
plot(time,qd_t(1,:), 'r');
hold on;
plot(time,qd_t(2,:), 'g');
plot(time,qd_t(3,:), 'b');
plot(time,qd_t(4,:), 'k');
xlabel('tiempo(seg)'), ylabel('q'(rad)'), grid;
hold off;

subplot(3,1,3);
plot(time,qdd_t(1,:), 'r');
hold on;
plot(time,qdd_t(2,:), 'g');
plot(time,qdd_t(3,:), 'b');
plot(time,qdd_t(4,:), 'k');
xlabel('tiempo(seg)'), ylabel('q'''(rad)'), grid;
hold off;

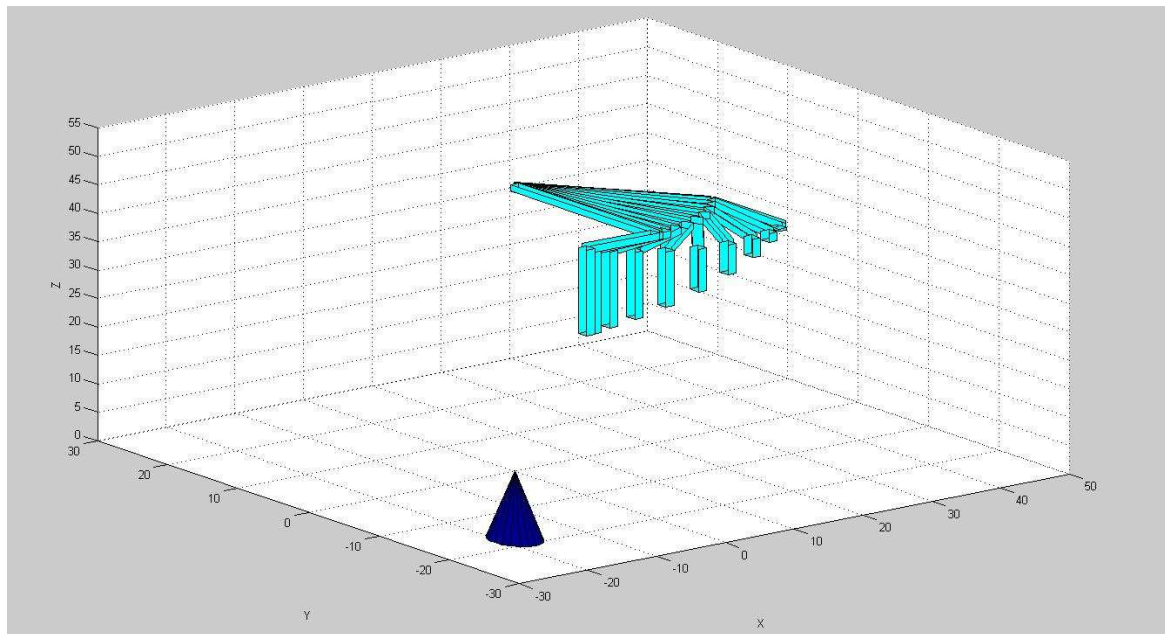
% *****
% Fin

% *****

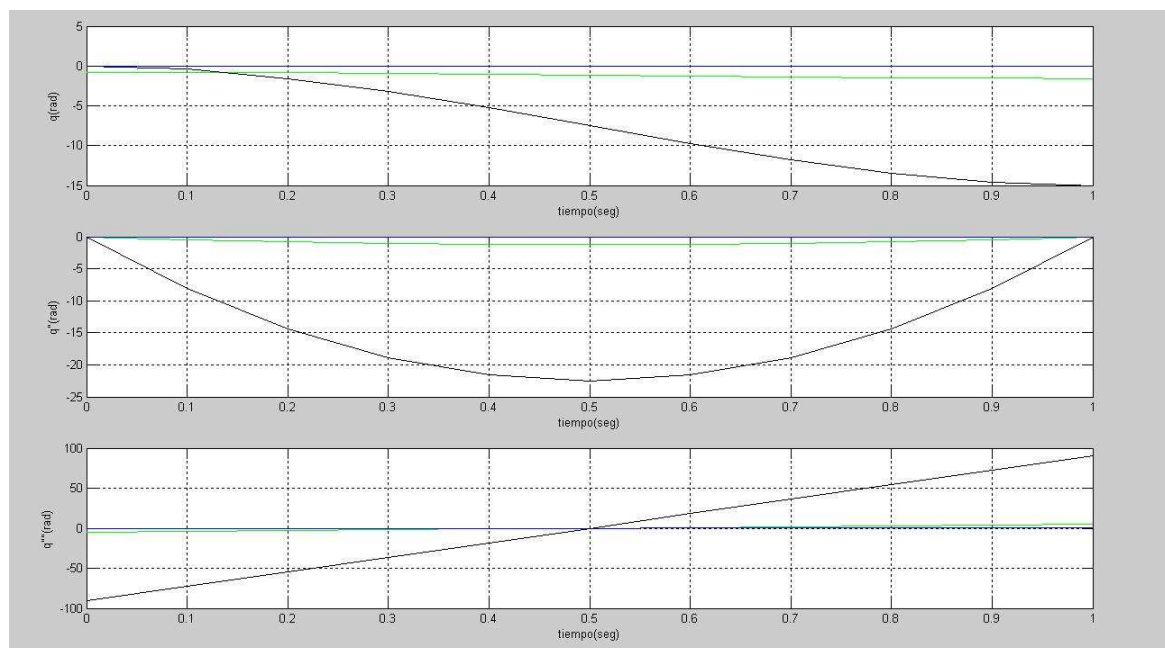
```

Los resultados son los siguientes:

## Polinómica de grado 3



Trayectoria Polinómica de grado 3

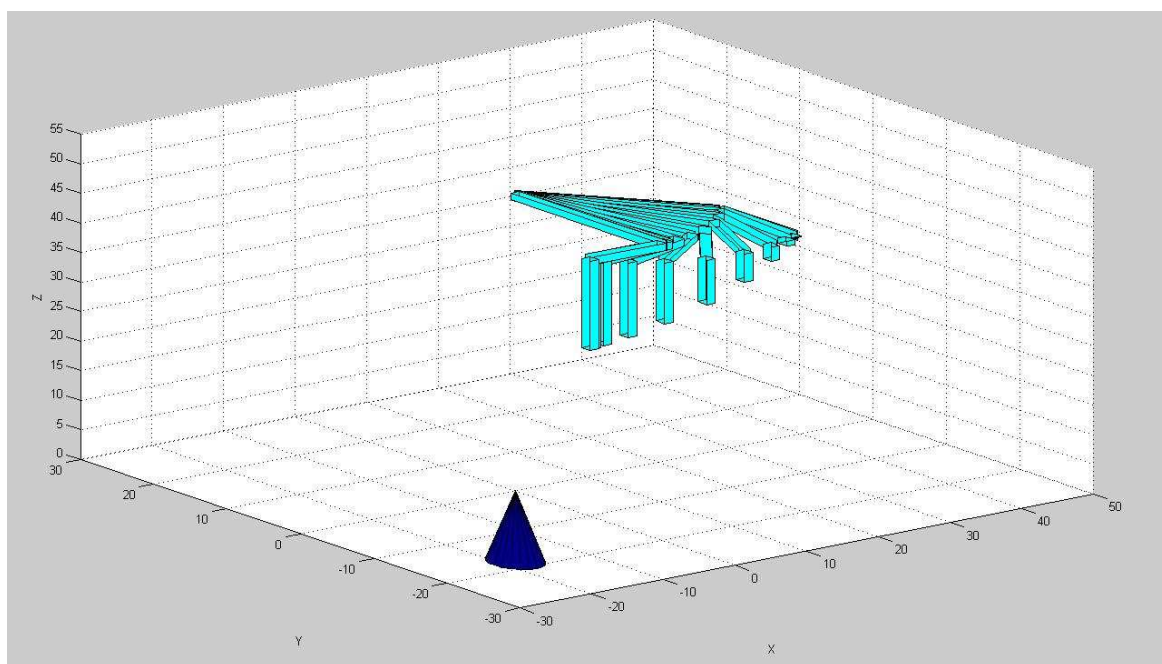


Curvas de Poligámica de grado 3

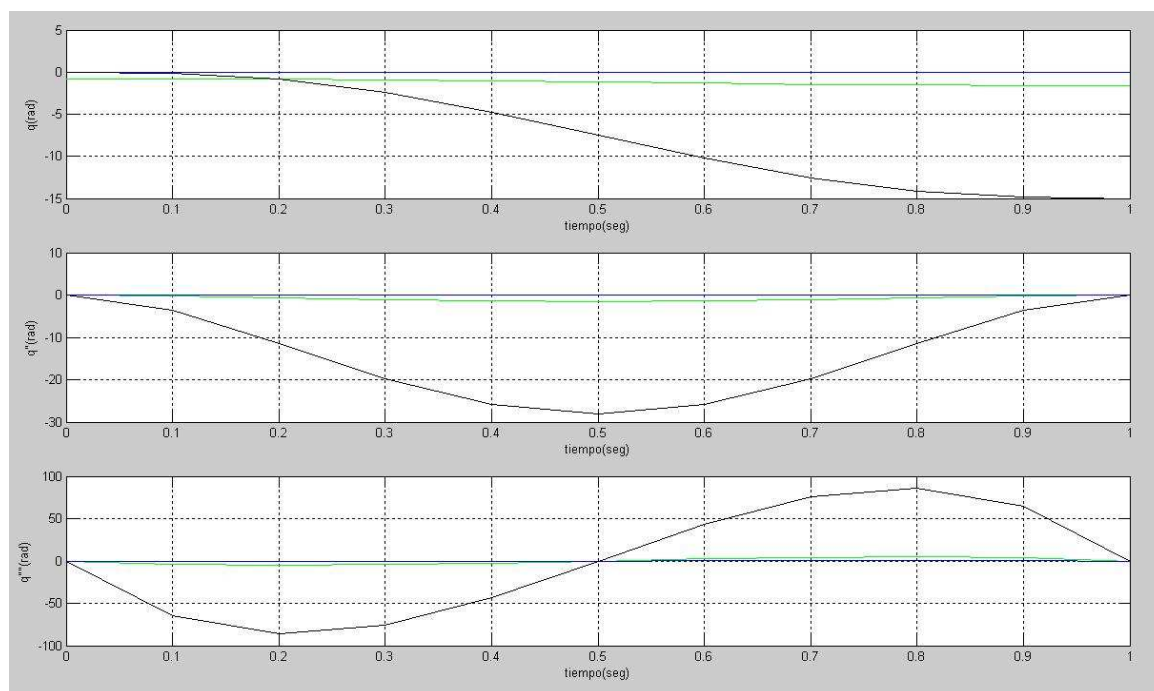
La primera de las curva corresponde a la posición de las articulaciones, la segunda a la velocidad y la tercera a la aceleración, siempre en función del tiempo.



## Polinómica de grado 5

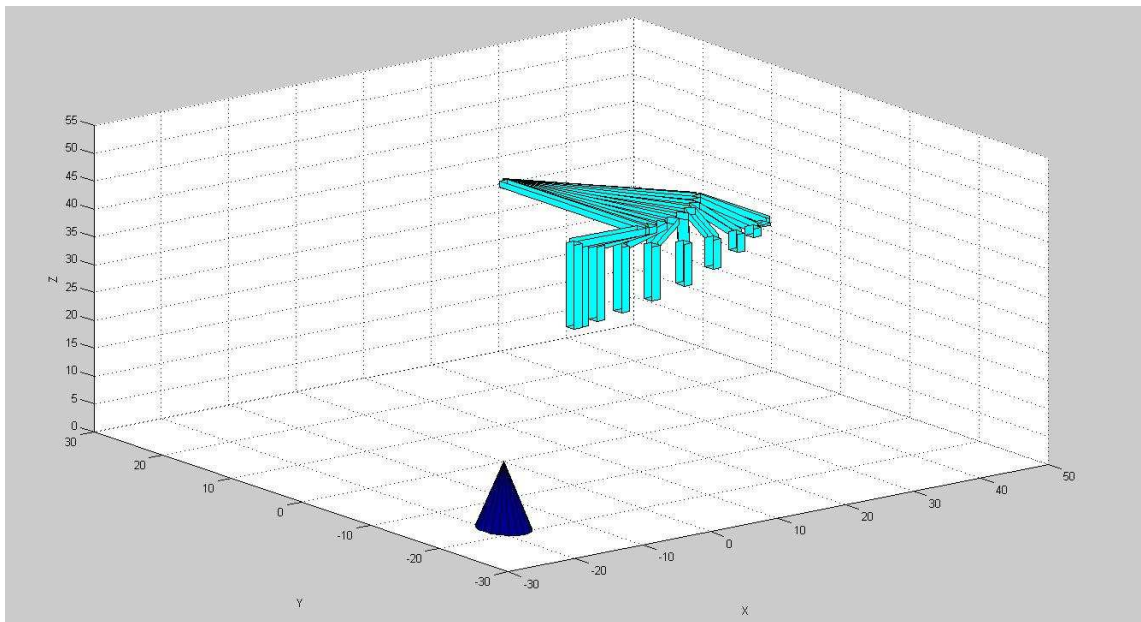


Trayectoria Polinómica de grado 3

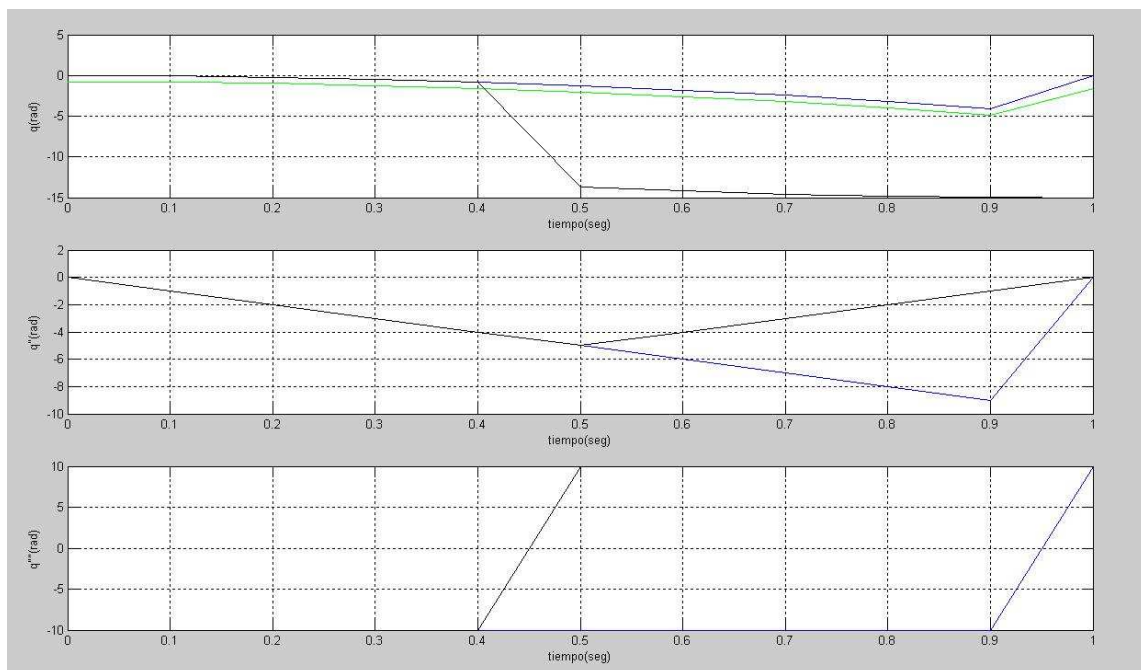


Curvas de Poligámica de grado 3

## Funciones Lineales con Enlaces parabólicos



Trayectoria de Curvas Lineales con Enlaces parabólicos



Curvas de Trayectorias con Enlaces parabólicos

Como se ve finalmente, las curvas de posición, velocidad y aceleración en función del tiempo por cada uno de los ejes, de acuerdo al tipo de trayectoria elegida previamente. La elección de cada una de estas dependerá de las prestaciones que se quiera dar al usuario, como velocidad en alcanzar el punto, menor golpes mecánicos, etc. Luego esto tendrá suma importancia en el control

dinámico, ya que todo esto se traducirá en picos de corriente, torques máximos, niveles de tensión, etc.

## Simulación del Manipulador por Cinemática Inversa

Ahora realizaremos la misma experiencia, pero implementado los cálculos de la cinemática inversa, que se han planteado y obtenido anteriormente. La idea será partir de un punto (x,y,z) inicial, para terminar en otro punto (x,y,z) final. En este caso, habrá que calcular las articulaciones iniciales y finales, y con la ayuda del Hemero del MatLab se simulará la trayectoria y se verá cómo irán variando las articulaciones del manipulador.

### Ejemplo 1:

Partiremos de las características del manipulador:

$$l_1 = 50; \quad l_2 = 20; \quad l_3 = 10$$

Se planteará partir desde el punto inicial  $p_i = (x_i, y_i, z_i) = (30, 0, 45)$ , para que el brazo consiga llegar finalmente al punto espacial final  $p_f = (x_f, y_f, z_f) = (19, 24, 20)$

En base a lo intencionado, se implementará el siguiente script en MatLab.

```
%*****
% Analisis cinematico inverso del manipulador SCARA
% *****
L1 = 50;
L2 = 20;
L3 = 10;
Xini = 30;
Xfin = 19;
Yini = 0;
Yfin = 24;
Zini = 45;
Zfin = 20;

UVW = [Xini;Yini;Zini;1];

% Se plotean ambos puntos origen y destino
plot3(X,Y,Z,'ok');
grid on;
xlabel('Eje X');
ylabel('Eje Y');
zlabel('Eje Z');
```

```
axis ([-30 30 -30 30 0 50]);
```

```
% Generando una trayectoria.
```

```
q3 = L1 - Zini;
```

```
q2 = acos((Xini^2 + Yini^2 - L2^2 - L3^2)/(2*L2*L3));
```

```
q1 = asin(Yini/sqrt(Xini^2 + Yini^2)) - acos((Xini^2 + Yini^2 + L2^2 - L3^2)/(2*L2*sqrt(Xini^2 + Yini^2)));
```

```
% -----
```

```
% Usando la libreria HEMERO
```

```
% -----
```

```
% Punto de inicial en coordenadas articulares
```

```
%q3 = 0;
```

```
%q2 = 0;
```

```
%q1 = 0;
```

```
A1 = [cos(q1) -sin(q1) 0 L2*cos(q1); sin(q1) cos(q1) 0 L2*sin(q1); 0 0 1 L1; 0 0 0 1];
```

```
A2 = [cos(q2) -sin(q2) 0 L3*cos(q2); sin(q2) cos(q2) 0 L3*sin(q2); 0 0 1 0; 0 0 0 1];
```

```
A3 = [1 0 0 0; 0 1 0 0; 0 0 1 -q3; 0 0 0 1];
```

```
T = A1 * A2 * A3;
```

```
TG(1, :, 1) = T(1, :);
```

```
TG(2, :, 1) = T(2, :);
```

```
TG(3, :, 1) = T(3, :);
```

```
% Punto destino en coordenadas articulares
```

```
%q3 = 15;
```

```
%q2 = pi/4;
```

```
%q1 = pi/4;
```

```
q3 = L1 - Zfin;
```

```
q2 = acos((Xfin^2 + Yfin^2 - L2^2 - L3^2)/(2*L2*L3));
```

```
q1 = asin(Yfin/sqrt(Xfin^2 + Yfin^2)) - acos((Xfin^2 + Yfin^2 + L2^2 - L3^2)/(2*L2*sqrt(Xfin^2 + Yfin^2)));
```

```
A1 = [cos(q1) -sin(q1) 0 L2*cos(q1); sin(q1) cos(q1) 0 L2*sin(q1); 0 0 1 L1; 0 0 0 1];
```

```
A2 = [cos(q2) -sin(q2) 0 L3*cos(q2); sin(q2) cos(q2) 0 L3*sin(q2); 0 0 1 0; 0 0 0 1];
```

```
A3 = [1 0 0 0; 0 1 0 0; 0 0 1 -q3; 0 0 0 1];
```

```

T = A1 * A2 * A3;
TG(1, :, 2) = T(1, :);
TG(2, :, 2) = T(2, :);
TG(3, :, 2) = T(3, :);

SCARA = [0 0 q1 L1 0; 0 L2 0 0 0; 0 L3 q2 0 0; 0 0 0 -q3 1];
%SCARA = [0 L2 q1 L1 0; 0 L3 q2 0 0; 0 0 0 -q3 1];
%Q = [q1 q2 0 -q3];

stol = 0.1; ilimit = 1000;

% TG contiene dos matrices 3x3 de traslacion, una con las coordenadas del punto de inicio
% y la siguiente matriz de traslacion contiene las coordenadas del punto destino de brazo robot.
Q = IKINE(SCARA, stol, ilimit, TG)

q = transpose (Q);
qd = [0 0; 0 0; 0 0; 0 0];
qdd = [0 0; 0 0; 0 0; 0 0];
to = 0;
tij = 1;

[Qcoef, time, q_t, qd_t, qdd_t] = pol3(q, qd, to, tij, 0.1);
puntos = transpose(q_t);

figure(1);
PLOTBOT(SCARA, puntos, 'l');
axis ([-30 50 -30 30 0 55]);

figure(2);
subplot(3,1,1);
plot(time,q_t(1,:), 'r');
hold on;
plot(time,q_t(2,:), 'g');
plot(time,q_t(3,:), 'b');
plot(time,q_t(4,:), 'k');
xlabel('tiempo(seg)'), ylabel('q(rad)'), grid;
hold off;

subplot(3,1,2);
plot(time,qd_t(1,:), 'r');
hold on;
plot(time,qd_t(2,:), 'g');

```

```
plot(time,qd_t(3,:),'b');
plot(time,qd_t(4,:),'k');
xlabel('tiempo(seg)'), ylabel('q"(rad)'), grid;
hold off;

subplot(3,1,3);
plot(time,qdd_t(1,:),'r');
hold on;
plot(time,qdd_t(2,:),'g');
plot(time,qdd_t(3,:),'b');
plot(time,qdd_t(4,:),'k'),
xlabel('tiempo(seg)'), ylabel('q""(rad)'), grid;
hold off;

[Qcoef, time, q_t, qd_t, qdd_t] = pol5(q, qd, qdd, to, tij, 0.1);
puntos2 = transpose(q_t);

figure(3);
PLOTBOT(SCARA, puntos2, 'l');
axis ([-30 50 -30 30 0 55]);

figure(4);
subplot(3,1,1);
plot(time,q_t(1,:),'r');
hold on;
plot(time,q_t(2,:),'g');
plot(time,q_t(3,:),'b');
plot(time,q_t(4,:),'k');
xlabel('tiempo(seg)'), ylabel('q(rad)'), grid;
hold off;

subplot(3,1,2);
plot(time,qd_t(1,:),'r');
hold on;
plot(time,qd_t(2,:),'g');
plot(time,qd_t(3,:),'b');
plot(time,qd_t(4,:),'k');
xlabel('tiempo(seg)'), ylabel('q"(rad)'), grid;
hold off;

subplot(3,1,3);
plot(time,qdd_t(1,:),'r');
```

```

hold on;
plot(time,qdd_t(2,:),'g');
plot(time,qdd_t(3,:),'b');
plot(time,qdd_t(4,:),'k'),
xlabel('tiempo(seg)'), ylabel('q""(rad)'), grid;
hold off;

qdd = [10 10; 10 10; 10 10; 10 10];
[Qcoef, time, q_t, qd_t, qdd_t, intervals] = parab(q, qdd, to, tij, 0.1);
puntos3 = transpose(q_t);

figure(5);
PLOTBOT(SCARA, puntos2, 'l');
axis ([-30 50 -30 30 0 55]);

figure(6);
subplot(3,1,1);
plot(time,q_t(1,:),'r');
hold on;
plot(time,q_t(2,:),'g');
plot(time,q_t(3,:),'b');
plot(time,q_t(4,:),'k');
xlabel('tiempo(seg)'), ylabel('q(rad)'), grid;
hold off;

subplot(3,1,2);
plot(time,qd_t(1,:),'r');
hold on;
plot(time,qd_t(2,:),'g');
plot(time,qd_t(3,:),'b');
plot(time,qd_t(4,:),'k');
xlabel('tiempo(seg)'), ylabel('q'(rad)'), grid;
hold off;

subplot(3,1,3);
plot(time,qdd_t(1,:),'r');
hold on;
plot(time,qdd_t(2,:),'g');
plot(time,qdd_t(3,:),'b');
plot(time,qdd_t(4,:),'k'),
xlabel('tiempo(seg)'), ylabel('q""(rad)'), grid;
hold off;

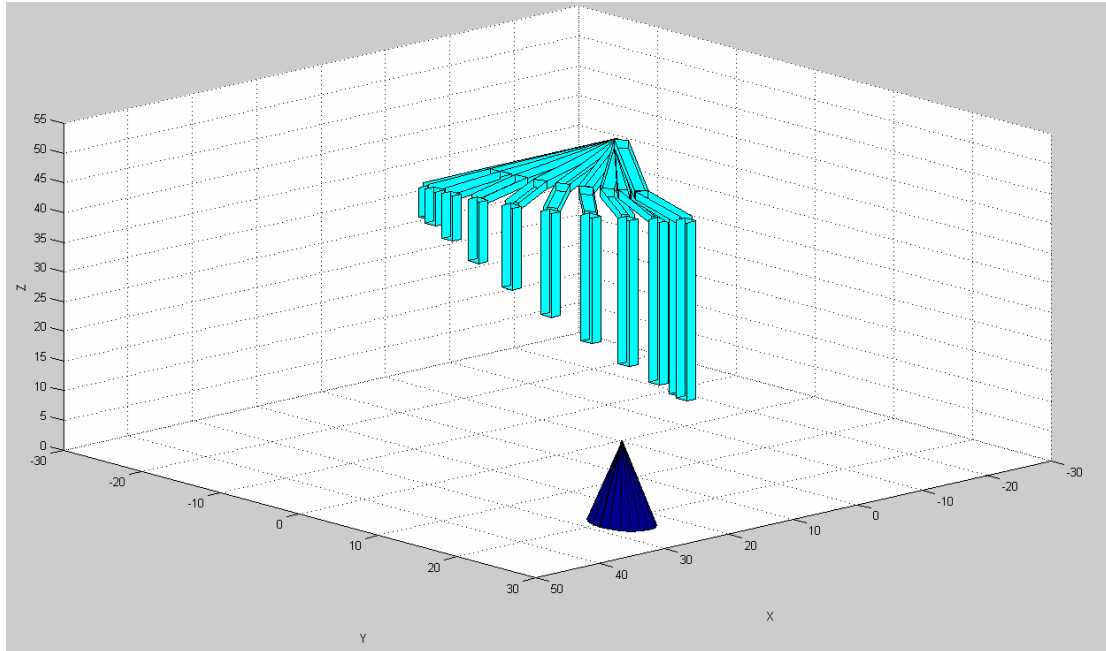
%*****
% Fin

```

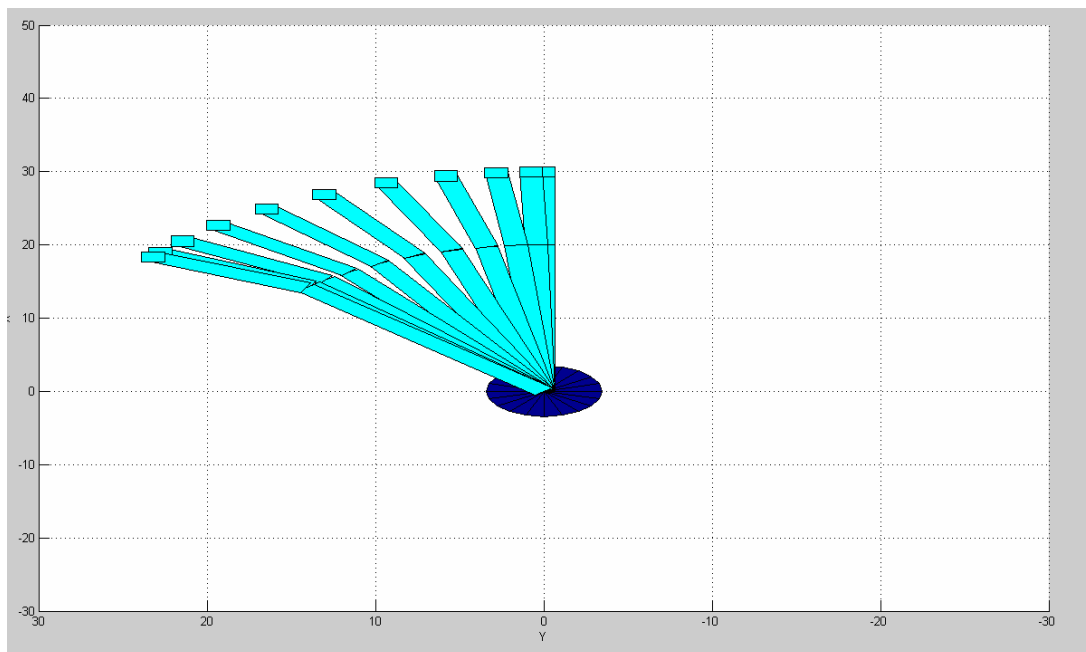
% \*\*\*\*\*

En este caso solamente analizaremos la trayectoria polinómica de grado 3.

Luego, ejecutando el mismo se obtienen los siguientes resultados:

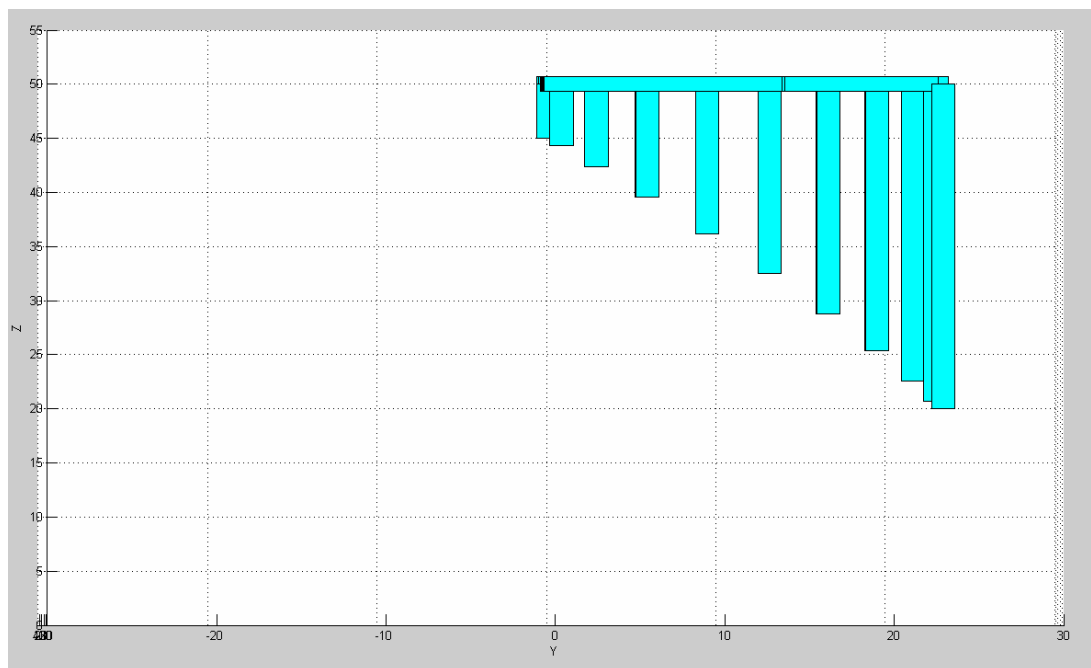


Vista 1. Trayectoria Polinómica de Grado 3 por Cinemática Inversa

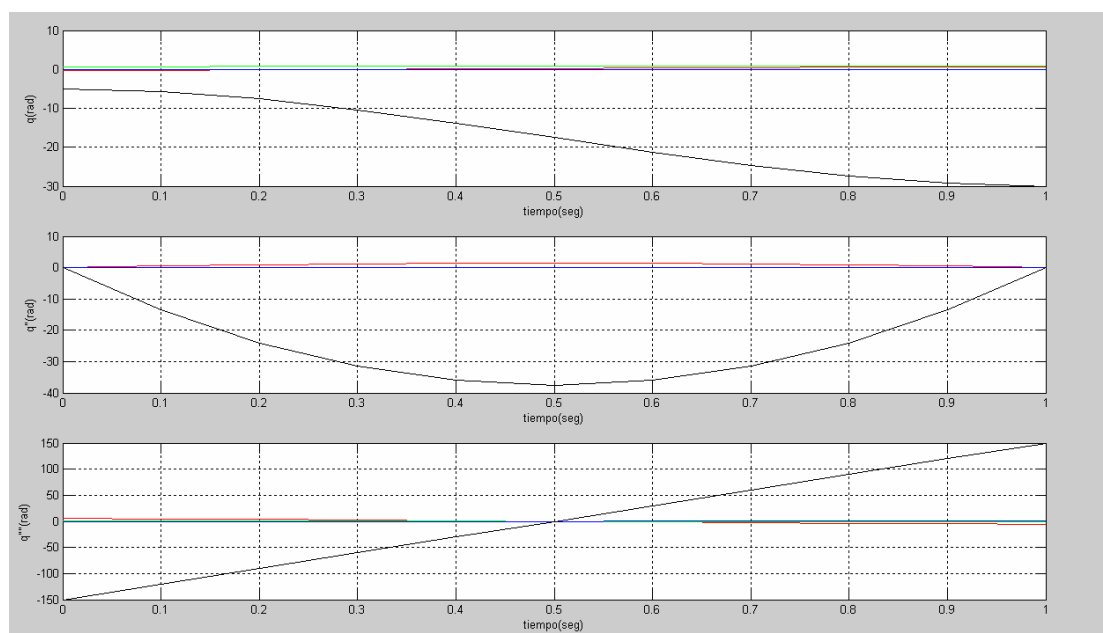


Vista 2. Vista Superior Trayectoria Polinómica de Grado 3 por Cinemática Inversa





Vista 3. Vista Lateral Trayectoria Polinómica de Grado 3 por Cinemática Inversa



Curvas Trayectoria Polinómica de Grado 3 por Cinemática Inversa

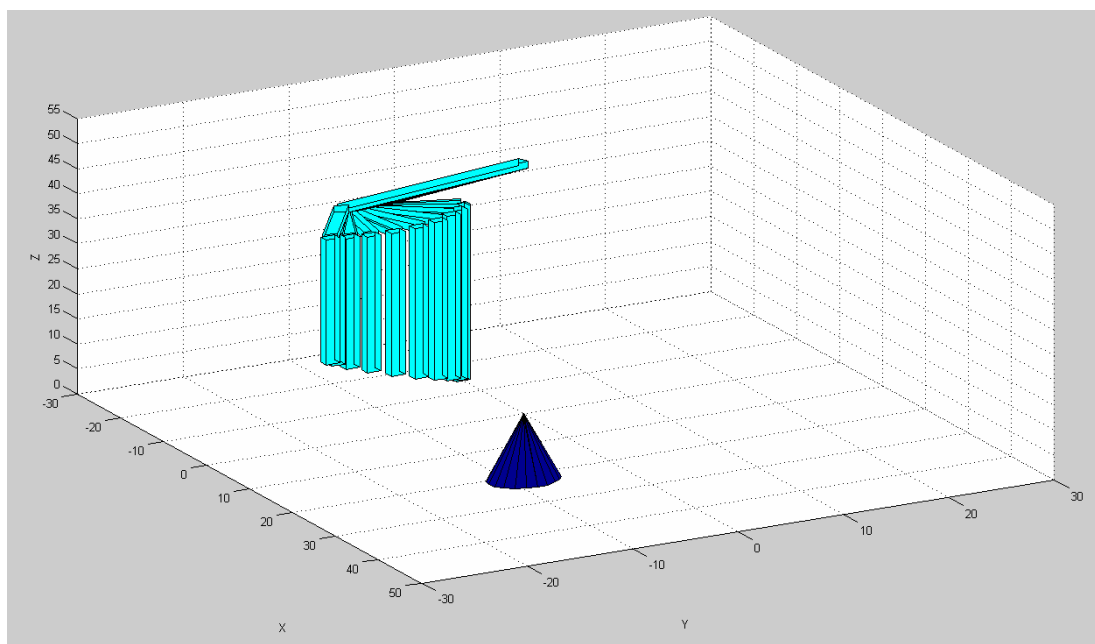
Ejemplo 2:

Partiremos de las características del manipulador:

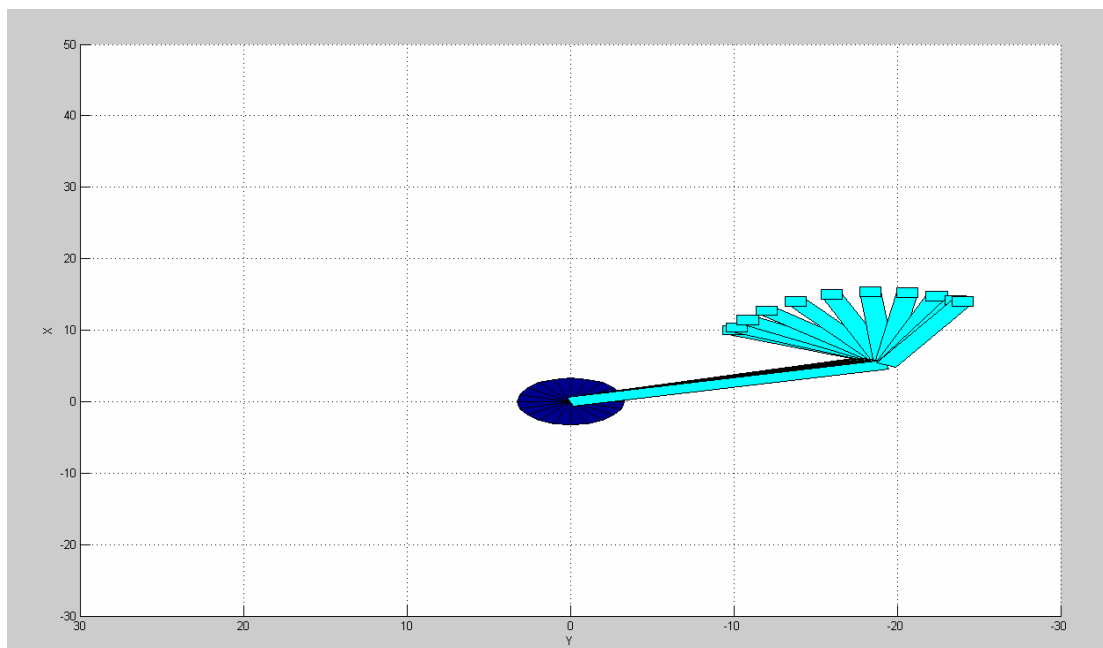
$$l_1 = 50; \quad l_2 = 20; \quad l_3 = 10$$

Se planteará partir desde el punto inicial  $p_i = (x_i, y_i, z_i) = (10, -10, 15)$ , para que el brazo consiga llegar finalmente al punto espacial final  $p_f = (x_f, y_f, z_f) = (14, -24, 25)$

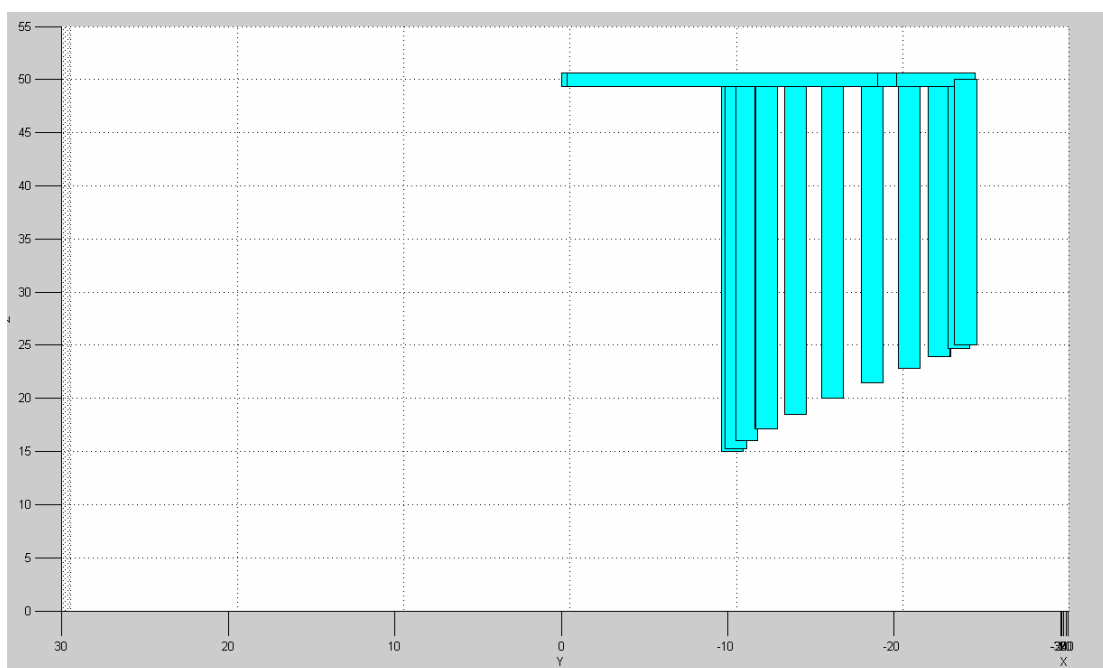
En base a lo intencionado, corrigiendo y ejecutando el script en MatLab obtenemos.



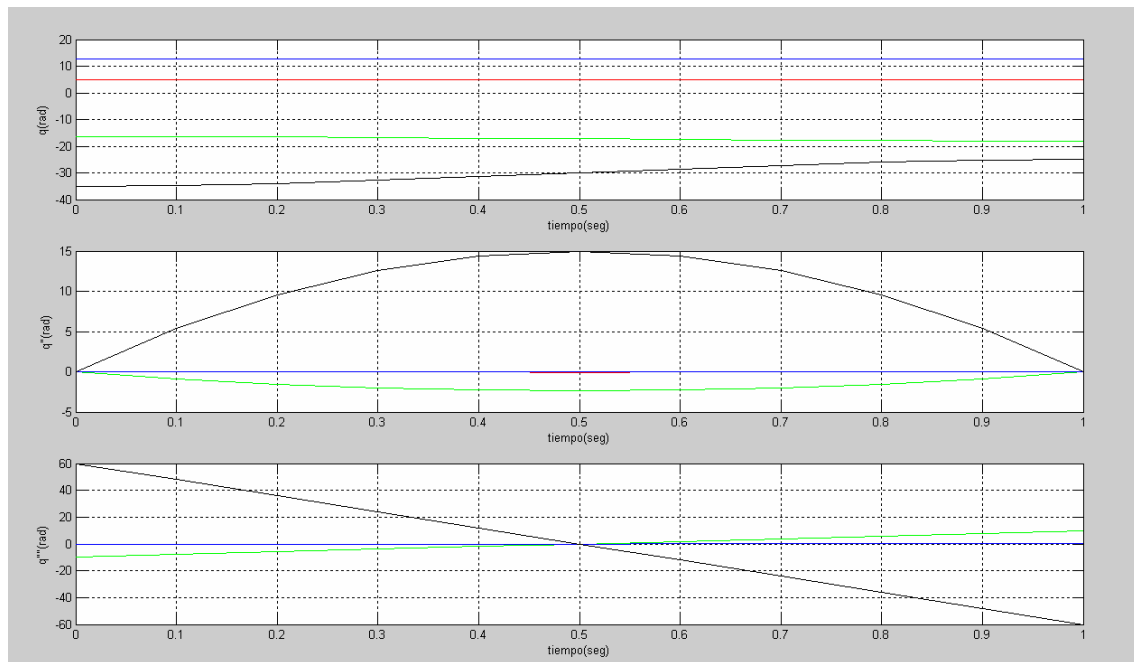
Vista 4. Trayectoria Polinómica de Grado 3 por Cinemática Inversa



Vista 5. Vista Superior Trayectoria Polinómica de Grado 3 por Cinemática Inversa



Vista 6. Vista Lateral Trayectoria Polinómica de Grado 3 por Cinemática Inversa



Curvas Trayectoria Polinómica de Grado 3 por Cinemática Inversa

Se concluye entonces finalmente que dándole al manipulador las coordenadas iniciales y las coordenadas finales, punto que debe alcanzar el extremo del mismo, éste recorrerá puntos intermedios, de acuerdo al tipo de trayectoria seleccionada con anterioridad, calculando los valores de cada articulación intermedia, para finalmente posicionarse en el punto espacial deseado.