Práctica 2

Cinemática de Robots

2.1.- Introducción

Los robots clásicos presentan una arquitectura antropomórfica serial, semejante al brazo humano. Consisten de una serie de barras rígidas unidas entre sí a través de articulaciones de un grado de libertad del tipo rotacional o prismática. En general cada articulación logra su movimiento a través de un accionamiento de potencia e incluye otros dispositivos como reductores de velocidad, frenos y sensores de posición o velocidad.

Aunque al definir las relaciones cinemáticas de un robot no se suelen consideran los aspectos dinámicos, nada más alejado de la realidad cuando se quiere diseñar un robot ya que existe una inevitable relación causa-efecto entre la cinemática y la dinámica. Nada más claro resulta que al pensar en las dimensiones de un robot, la longitud de un brazo afecta al cuadrado la inercia de los eslabones y por lo tanto el peso del robot y la potencia requerida en los actuadores.

Las arquitecturas de los robots clásicos presentan una serie de propiedades dinámicas y estructurales caracterizadas por una gran rigidez estructural, repetibilidad y elevado peso propio. El elevado peso propio de los robots clásicos limita la capacidad carga útil y las velocidades de trabajo, las cuales usualmente están en torno a los 60 grados/seg. para las primeras tres articulaciones de los robots industriales (robots de soldadura) y 250 grados/seg. para los robots pequeños de altas prestaciones como el STÄUBLI RX90.

Un aspecto importante que refleja las relaciones dinámicas del robot respecto de la carga útil que pueden manipular, puede estudiarse en la siguiente tabla.

Robot	Peso	Carga útil	Repetibilidad	Carga útil/peso
ABB IRB 2000	370	10	0.100	0.0270
ABB IRB 4400	940	45	0.100	0.0047
ABB IRB 6400/3.0	1450	75	0.100	0.0510
STÄUBLI RX 90	120	6	0.002	0.0500
GMF S 10	200	10	0.200	0.0417
Hitachi M6100	410	10	0.100	0.0243
Puma 550	63	4	0.100	0.0063
SCARA Adept 3	205	25	0.025	0.1220
SCARA GMF A-600	120	6	0.013	0.0500

Al calcular la cinemática de los robots clásicos debe considerarse que dependiendo de las dimensiones de sus primeras articulaciones, el peso de los robots de tipo industrial oscila en torno a valores que tienen una relación en el mejor de los casos de 0.150 (Carga útil/peso). Por lo cual, por ejemplo un robot industrial con un alcance de 3.0 metros con capacidad para mover cargas de 75 kg puede tener un peso de 1450 kg (ABB IRB 6400).

Las siguientes son algunas recomendaciones que deben tenerse en cuenta al definir la cinemática de un robot, la cual debe hacerse en consideración de la dinámica que imponen las dimensiones de las barras que lo forman:

- ✓ El espacio de trabajo del robot debe ser cuidadosamente estudiado para definir el volumen justo de trabajo del robot
- ✓ En un robot de seis grados de libertad rotacional, las primeras tres barras son las que aportan la mayor dinámica debido a su peso. A menudo es posible localizar los primeros tres accionamientos de potencia en la base del robot, pero para lograr esto se debe ser cuidadoso en el uso de mecanismos de cuatro barras que mueven el brazo mas alejado (robot ABB IRB2400).
- ✓ En un robot de seis grados de libertad, las tres primeras articulaciones del robot deben dar las condiciones de posición y las tres últimas articulaciones del extremo del robot deben concentrar en un punto de la mano, los tres grados de libertad de orientación.

En esta práctica se van a presentar las herramientas necesarias para resolver los dos problemas fundamentales en el estudio de la cinemática del robot. El primero de ellos, consiste en determinar la posición y orientación del extremo final de la cadena cinemática conocidos los valores de las coordenadas articulares y las características geométricas del robot, y es conocido como problema cinemático directo. La solución del problema inverso permite hallar las variables articulares conocida la posición y orientación del extremo de la cadena cinemática. Para la resolución de estos problemas se utiliza la representación de Denavit-Hartenberg y las matrices de transformación homogénea.

De acuerdo con la estructura del libro, todos los apartados presentan herramientas desarrolladas en MatLab® para la solución de los problemas planteados.

2.2- Cinemática directa del brazo de un robot manipulador

Las técnicas que se estudian aquí, se aplican a un manipulador mecánico de cadena abierta y tratan el estudio analítico y el modelado en MatLab® de la geometría del movimiento de un robot con respecto a un sistema de referencia fijo como una función del tiempo sin considerar la dinámica.

2.2.1 El problema cinemático directo

El problema cinemático directo se plantea en términos de encontrar una matriz de transformación que relaciona el sistema de coordenadas ligado al cuerpo en movimiento respecto a un sistema de coordenadas que se toma como referencia. Para lograr esta representación se usan las matrices de transformación homogénea 4x4, la cual incluye las operaciones de traslación y la orientación.

La matriz de transformación homogénea es una matriz de 4x4 que transforma un vector expresado en coordenadas homogéneas desde un sistema de coordenadas hasta otro sistema de coordenadas. Para una descripción más amplia acerca de las bases algebraicas de las transformaciones homogéneas se recomienda estudiar las referencias: [1] y [2].

La matriz de transformación homogénea tiene la siguiente estructura:

$$\mathbf{T} = \begin{bmatrix} \textbf{matriz de rotación} & \textbf{vector de posición} \\ \mathbf{f}_{1x3} & \textbf{escalado} \end{bmatrix} = \begin{bmatrix} \mathbf{n_x} & \mathbf{s_x} & \mathbf{a_x} & \mathbf{p_x} \\ \mathbf{n_y} & \mathbf{s_y} & \mathbf{a_y} & \mathbf{p_y} \\ \mathbf{n_z} & \mathbf{s_z} & \mathbf{a_z} & \mathbf{p_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{n} & \mathbf{s} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde los vectores **n, s, a**, son vectores ortogonales unitarios y **p** es un vector que describe la posición **x, y, z** del origen del sistema actual respecto del sistema de referencia.

Para entender las propiedades de la matriz de transformación homogénea nos fijamos en el siguiente gráfico.

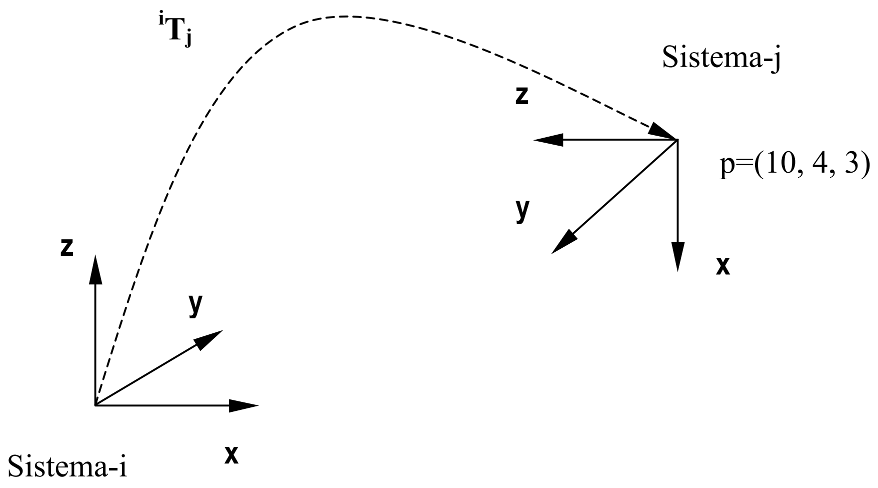


Figura-2.1. Interpretación geométrica de la matriz de transformación homogénea

$$
{}^{i}\mathbf{T}_{j} = \begin{bmatrix} 0 & 0 & -1 & 10 \\ 0 & -1 & 0 & 4 \\ -1 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Al analizar las columnas de la submatriz de rotación de la matriz de transformación homogénea ${}^{i}\mathbf{T}_{j}$, un observador localizado en el origen de **sitema-i**, puede ver como están orientados los ejes **x, y, z** del **sistema-j**, además también observa como se ha desplazado en coordenadas cartesianas el origen del **sistema-j** respecto del origen del sistema de referencia con la información del vector de posición.

2.2.2 La representación de Denavit-Hartenberg

La representación de D-H, se aplica a robots de cadena cinemática abierta y consiste en una serie de reglas para colocar los sistemas de referencia de cada eslabón del robot. Antes de aplicar el método de D-H es importante tener en cuenta los siguientes comentarios:

- Se parte de una configuración cualesquiera del robot, si bien es aconsejable colocarlo en una posición sencilla de analizar.
- Se numeran los eslabones, asignando el *0* para la base y *n-1* para el último eslabón, siendo *n* el número de grados de libertad del robot.
- El sistema de coordenadas ortonormal dextrógiro de la base **(x_0, y_0, z_0)** se establece con el eje z_0 localizado a lo largo del eje de movimiento de la articulación 1 y apuntando hacia fuera del hombro del brazo del robot.
- El sistema de referencia de cada eslabón se coloca al final del mismo, en el extremo de la articulación a la cual esta conectado el eslabón siguiente.
- El ángulo ó desplazamiento de cada eslabón siempre se mide tomando como base el sistema de referencia del eslabón anterior.

- Al colocar el sistema de referencia del **eslabón-i**, se deben seguir las siguientes reglas:
  - El eje $\mathbf{z_i}$ del sistema de referencia debe quedar alineado a lo largo de la articulación
  - El eje $\mathbf{x_i}$ debe colocarse con orientación normal al plano formado por los ejes $\mathbf{z_{i-1}}$ y $\mathbf{z_i}$
- Al establecer los sistemas de coordenadas de la mano se debe tener en cuenta el principio de Pieper's en el cual se establece que los tres últimos sistemas de referencia se intercepten en un punto, lo cual permite obtener una solución para el problema cinemático inverso de forma cerrada para estas articulaciones.

Además de las reglas anteriores la convención de D-H establece las siguientes condiciones para los demás parámetros geométricos, de acuerdo a la figura-2.2.

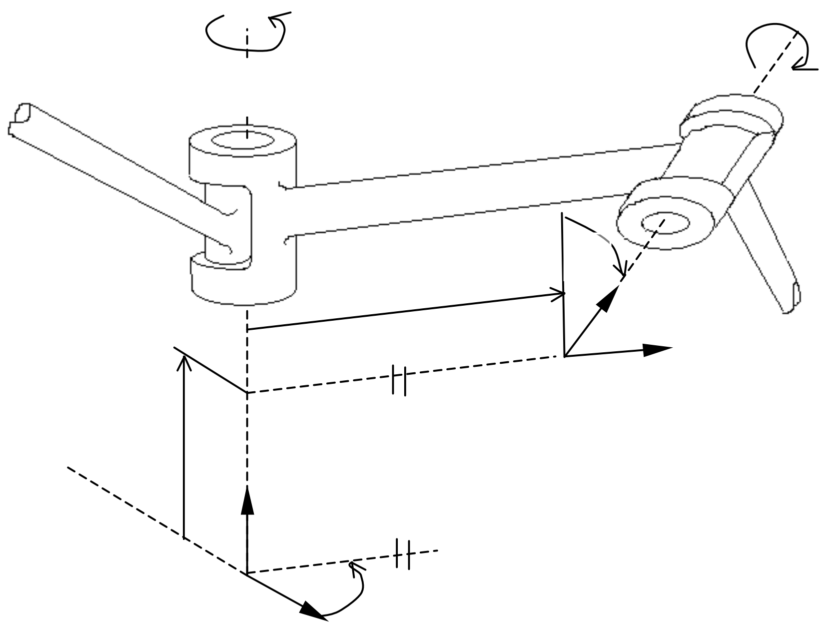


Figura-2.2. Sistemas de coordenadas para la convención de D-H.

Cada sistema de coordenadas se establece sobre las siguientes reglas.

θ_i: Es el ángulo de la articulación desde el eje $\mathbf{x_{i-1}}$ hasta el eje $\mathbf{x_i}$, medido respecto del eje $\mathbf{z_{i-1}}$, usando la regla de la mano derecha.
d_i: Es la distancia medida desde el origen del sistema **i-1**, a lo largo del eje $\mathbf{z_{i-1}}$ hasta la intersección del eje $\mathbf{z_{i-1}}$ con el eje $\mathbf{x_i}$.
a_i: Es la distancia de separación entre los orígenes de los sistemas de referencia **i-1** e **i**, medida a lo largo del eje $\mathbf{x_i}$ hasta la intersección con el eje $\mathbf{z_{i-1}}$. (o la distancia más corta entre los ejes $\mathbf{z_{i-1}}$ y $\mathbf{z_i}$, cuando estos no se interceptan)
α_i: Es el ángulo que separa los ejes $\mathbf{z_i}$ y $\mathbf{z_{i-1}}$, medido respecto del eje $\mathbf{x_i}$

Con base en la figura-2.2 y de acuerdo a las reglas de D-H, se determina la siguiente matriz de transformación homogénea:

$$
{}^{i-1}\mathbf{A}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathbf{d}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{c\theta}_i & -\mathbf{s\theta}_i & 0 & 0 \\ \mathbf{s\theta}_i & \mathbf{c\theta}_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \mathbf{a}_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \mathbf{c\alpha}_i & -\mathbf{s\alpha}_i & 0 \\ 0 & \mathbf{s\alpha}_i & \mathbf{c\alpha}_i & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
$$

$$
{}^{i-1}\mathbf{A}_i = \begin{bmatrix} \mathbf{c\theta}_i & -\mathbf{c\alpha}_i\mathbf{s\theta}_i & \mathbf{s\alpha}_i\mathbf{s\theta}_i & \mathbf{a}_i\mathbf{c\theta}_i \\ \mathbf{s\theta}_i & \mathbf{c\alpha}_i\mathbf{c\theta}_i & -\mathbf{s\alpha}_i\mathbf{c\theta}_i & \mathbf{a}_i\mathbf{s\theta}_i \\ 0 & \mathbf{s\alpha}_i & \mathbf{c\alpha}_i & \mathbf{d}_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Código en Matlab®. La función DENAVIT realiza los cálculos anteriores devolviendo la matriz de transformación homogénea

```
% DENAVIT      Matriz de transformación homogénea.
%  DH = DENAVIT(TETA, D, A, ALFA) devuelve la matriz de transformacion
%  homogénea 4 x 4 a partir de los parametros de Denavit-Hartenberg
%  D, ALFA, A y TETA.
%
%  See also DIRECTKINEMATIC.

function dh=denavit(teta, d, a, alfa)
dh=[cos(teta)   -cos(alfa)*sin(teta)    sin(alfa)*sin(teta)    a*cos(teta);
    sin(teta)    cos(alfa)*cos(teta)   -sin(alfa)*cos(teta)    a*sin(teta);
           0               sin(alfa)              cos(alfa)              d;
           0                       0                      0             1];
```

2.2.3 Representación de la cinemática directa de robots manipuladores

En esta sección se explican algunas arquitecturas de robots y como construir la tabla de los parámetros de D-H. Para una información más detallada sobre este tema, se recomienda estudiar las referencias [1] y [2].

Ejemplo 2.1

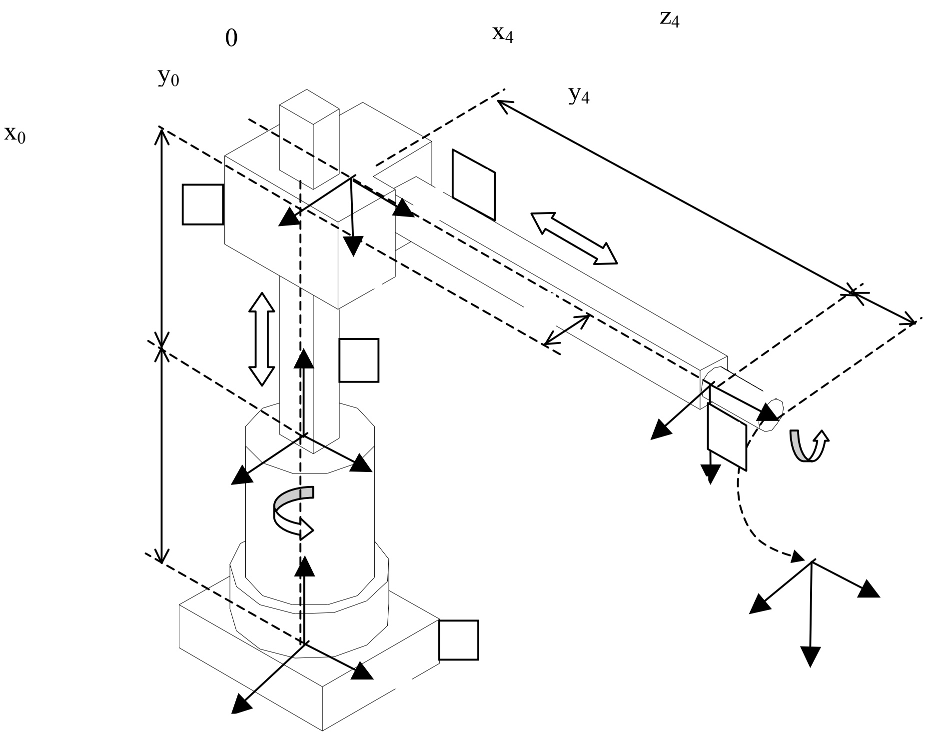


Figura-2.3 Parámetros de D-H para un robot cilíndrico

Eslabón	θ_i	d_i	a_i	α_i
1	θ_1	l_1	0	0
2	0	d_2	$- a_2$	$-\pi/2$
3	0	d_3	0	0
4	θ_4	l_4	0	0

Tabla 2.1 Parámetros de D-H para el robot cilíndrico de la figura-2.3

- Note el lector el signo negativo del parámetro a_2 así como la localización del origen del sistema de coordenadas **(x_2, y_2, z_2)**
- Las variables articulares son en este caso θ_1, d_2, d_3, θ_4

Ejemplo 2.2

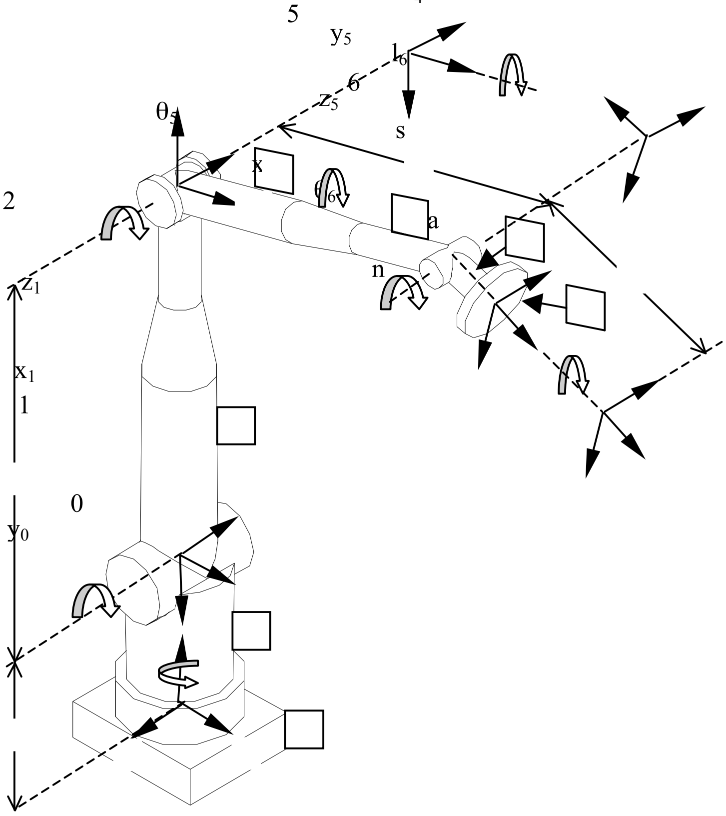


Figura-2.4 Parámetros de D-H para un robot rotacional

Tabla 2.2 Parámetros de D-H para el robot cilíndrico de la figura-2.4

Eslabón	θ_i	d_i	a_i	α_i
1	$\theta_1+\pi/2$	l_1	0	$-\pi/2$
2	$\theta_2-\pi/2$	0	a_2	0
3	$\theta_3+\pi$	0	0	$\pi/2$
4	θ_4	l_4	0	$-\pi/2$
5	θ_5	0	0	$\pi/2$
6	θ_6	l_6	0	0

- Note el lector que en este caso todas las variables articulares corresponden a los giros en las articulaciones.
- Obsérvese la rotación en θ_4 entre los eslabones 3 y 4.
- Obsérvese que el sistema de referencia **(x_3, y_3, z_3)** se ha trasladado al principio del eslabón 3 cumpliendo con la representación de D-H.

***Código en Matlab®*.** La función DIRECTKINEMATIC resuelve la cinemática directa de robots de cadena abierta. Para utilizarla debe invocarse junto con el vector de coordenadas articulares *Q* de *n* componentes siendo *n* el número de eslabones del robot.

A continuación se muestra el código fuente utilizado para cada uno de los robots que se están utilizando como ejemplos. El lector debe notar las modificaciones realizadas en el código, debido al diferente número de eslabones de cada robot y a los parámetros D-H de cada robot. Se recomienda que se realicen estos ejemplos como práctica de los conocimientos adquiridos.

Ejemplo 2.3

Código fuente de la función DIRECTKINEMATIC para el robot cilíndrico de 4 grados de libertad del ejemplo 2.1

Debe notarse que el vector de coordenadas articulares Q representa los 4 grados de libertad del robot, 2 rotacionales y 2 prismáticas y se introduce en los parámetros D-H como variables.

Las dimensiones del robot, introducidas en los parámetros de D-H, son las indicadas en el capítulo 1.

```
% DIRECTKINEMATIC4      Direct Kinematic.
%  A04 = DIRECTKINEMATIC4(Q) devuelve la matriz de transformación del
%  primer sistema de coordenadas al último en función del vector Q
%  de variables articulares.
%
%  See also DENAVIT.

function A04 = directkinematic4(q)

%  Parámetros Denavit-Hartenberg del robot
teta = [q(1)    0       0      q(4)];
d    = [0.4     q(2)    q(3)   0.2 ];
a    = [0      -0.1     0      0   ];
alfa = [0      -pi/2    0      0   ];

%  Matrices de transformación homogénea entre sistemas de coordenadas
consecutivos
A01 = denavit(teta(1), d(1), a(1), alfa(1));
A12 = denavit(teta(2), d(2), a(2), alfa(2));
A23 = denavit(teta(3), d(3), a(3), alfa(3));
A34 = denavit(teta(4), d(4), a(4), alfa(4));

%  Matriz de transformación del primer al último sistema de
coordenadas
A04  = A01 * A12 * A23 * A34;
```

A continuación se presenta la matriz obtenida para una configuración en la que todas las coordenadas articulares tienen valor nulo:

```
» q=zeros(4,1)

q =

     0
     0
     0
     0

» T=directkinematic4(q)

T =

    1.0000  0           0       -0.1000
         0  0.0000      1.0000   0.2000
         0 -1.0000      0.0000   0.4000
         0  0           0        1.0000

»
```

⇒ Se recomienda analizar la matriz homogénea obtenida tal y como se indicó en el apartado 2.2.1, comprobando que tanto la posición como la orientación del efector final coinciden con lo esperado cuando se introducen las coordenadas articulares del ejemplo.

Ejemplo 2.4

Código fuente de la función DIRECTKINEMATIC para el robot rotacional de 6 grados de libertad del ejemplo 2.2

Debe notarse que en el caso particular de un robot enteramente rotacional, las variables articulares representan los parámetros teta de D-H, como se estudió en el apartado 2.2.3

Las dimensiones del robot, introducidas en los parámetros de D-H, son las indicadas en la práctica 1.

```
% DIRECTKINEMATIC6      Direct Kinematic.
%  A06 = DIRECTKINEMATIC6(Q) devuelve la matriz de transformación del
%  primer sistema de coordenadas al último en función del vector Q
%  de variables articulares.
%
%  See also DENAVIT.

function A06 = directkinematic6(q)

%  Parámetros Denavit-Hartenberg del robot
teta = q;
d    = [0.315  0      0      0.5     0     0.08];
a    = [0      0.45   0      0       0     0];
alfa = [-pi/2  0    pi/2  -pi/2   pi/2   0];

%  Matrices de transformación homogénea entre sistemas de coordenadas
consecutivos
A01 = denavit(teta(1), d(1), a(1), alfa(1));
A12 = denavit(teta(2), d(2), a(2), alfa(2));
A23 = denavit(teta(3), d(3), a(3), alfa(3));
A34 = denavit(teta(4), d(4), a(4), alfa(4));
A45 = denavit(teta(5), d(5), a(5), alfa(5));
A56 = denavit(teta(6), d(6), a(6), alfa(6));

%  Matriz de transformación del primer al último sistema de
coordenadas
A06  = A01 * A12 * A23 * A34 * A45 * A56;
```

A continuación se presenta la matriz obtenida para una configuración en la que todas las coordenadas articulares tienen valor nulo:

```
» q=zeros(6,1)

q =

     0
     0
     0
     0
     0
     0

» T=directkinematic6(q)

T =

    1.0000         0         0    0.4500
         0    1.0000         0         0
         0         0    1.0000    0.8950
         0         0         0    1.0000
```

2.3.- Cinemática inversa del brazo de un robot manipulador

La cinemática inversa consiste en hallar los valores de las coordenadas articulares del robot $q = [q_1, q_2, ..., q_n]^T$ conocida la posición y orientación del extremo del robot.

A pesar de que en la literatura [1] y [2] se pueden encontrar diversos métodos genéricos para la resolución de la cinemática inversa que pueden ser implementados en computadora, suele ser habitual la resolución por medio de métodos geométricos. La mayor parte de los robots suelen tener cadenas cinemáticas relativamente sencillas, que facilitan la utilización de los métodos geométricos. Para muchos robots, si se consideran sólo los tres primeros grados de libertad, se tiene una estructura planar. Este hecho facilita la resolución del problema. Asimismo los últimos tres grados de libertad suelen usarse para la orientación de la herramienta, lo cual permite una resolución geométrica desacoplada de la posición de la muñeca del robot y de la orientación de la herramienta.

En esta sección se va a resolver el problema cinemático inverso para los dos robots anteriores, utilizando el método geométrico e implementándolo en ***Matlab®*.**

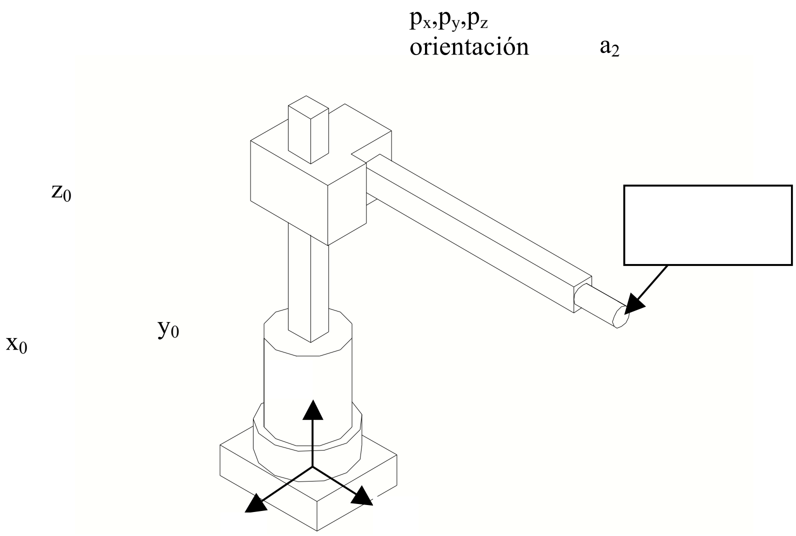
Ejemplo 2.5

Solución del robot cilíndrico de 4 grados de libertad.

En este caso particular, la solución geométrica es inmediata. Se parte de que la posición del extremo del robot es conocida **(p_x, p_y, p_z)** y se va a calcular los valores de las coordenadas articulares.

Articulación 1

Para obtener el valor de θ_1 (TETA1 en el código de ***Matlab®*.**) se proyecta el punto del extremo del robot **(p_x, p_y, p_z)** sobre el plano **(x_0, y_0, z_0)** obteniendo una sencilla relación angular. Sabiendo que θ_1 es el ángulo entre $\mathbf{x_0}$ y $\mathbf{x_1}$, se obtienen las siguientes gráficas.



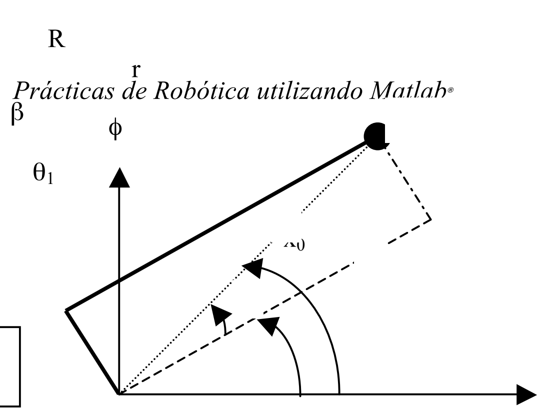


Figura-2.5 Cinemática inversa del robot de 4 gdl.

De las que se deducen las siguientes relaciones:

$$R = \sqrt{p_x^2 + p_y^2}$$

$$r = \sqrt{R^2 - a_2^2} = d_3 + l_4$$

$$sin\phi = \frac{p_y}{R} \qquad \cos\phi = \frac{p_x}{R}$$

$$sin\beta = \frac{a_2}{R} \qquad \cos\beta = \frac{r}{R}$$

utilizando la función atan2 de ***Matlab***® se calculan los valores de los ángulos:

$$\phi = atan2(sin\phi, \cos\phi)$$
$$\beta = atan2(sin\beta, \cos\beta)$$

que permiten el cálculo de θ_1 como ϕ-β.

Articulación 2

De la figura 2.3 se obtiene la siguiente fórmula:

$$l_1 + d_2 = p_z \Rightarrow d_2 = p_z - l_1$$

Articulación 3

De la figura 2.5:

$$r = \sqrt{R^2 - a_2^2} = d_3 + l_4$$

y_3

donde **n** y **s** son vectores de orientación del extremo del robot.

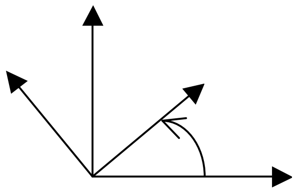
x_4

θ_4

x_3

Articulación 4

Para calcular la última articulación se necesita el cálculo previo del sistema de referencia **(x_3, y_3, z_3),** que se resolverá mediante la cinemática directa explicada en el ejemplo 2.3. Dado que el vector **a** de aproximación es necesariamente paralelo a $\mathbf{z_4}$ se deben cumplir las siguientes relaciones:



$$\left.\begin{aligned} sin\theta_4 &= n \cdot y_3 \\ \cos\theta_4 &= s \cdot y_3 \end{aligned}\right\} \quad \theta_4 = atan2(sin\theta_4, \cos\theta_4)$$

Código en Matlab®. La función INVERSEKINEMATIC4 resuelve la cinemática inversa del robot cilíndrico de 4 gdl. Para ello toma como parámetros la matriz homogénea T, que representa la orientación y posición del extremo del robot y devuelve el vector de coordenadas articulares.

```
%  Q = INVERSEKINEMATIC4(T) devuelve el vector de coordenadas
%  articulares correspondiente a la solución cinemática inversa de
%  la mano del manipulador en la posición y orientación expresadas
%  en la matriz T.
%
%  See also DIRECTKINEMATIC4, DENAVIT.

function q = inversekinematic4(T)

p = T(1:3,4);  % Posición de la mano del manipulador

% Inicialización de las variables articulares a calcular
q1 = 0;
q2 = 0;
q3 = 0;
q4 = 0;

%  Parámetros Denavit-Hartenberg del robot
teta = [q1    0     0     q4 ];
d    = [0.4   q2    q3    0.2];
a    = [0    -0.1   0     0  ];
alfa = [0    -pi/2  0     0  ];

% Solución de la primera articulación: q1
R = sqrt(p(1)^2+p(2)^2);
r = sqrt(R^2-a(2)^2);

sphi = -p(1)/R;
cphi = p(2)/R;
phi  = atan2(sphi, cphi);
```

```
sbeta = -a(2)/R;
cbeta = r/R;
beta  = atan2(sbeta, cbeta);

q1 = phi - beta;

% Solución de la segunda articulación: q2
q2 = p(3) - d(1);

% Solución de la tercera articulación: q3
q3 = r - d(4);

% Solución de la cuarta articulación: q4

%  Cálculo de la matriz de transformación A03
A01 = denavit(q1, d(1), a(1), alfa(1));
A12 = denavit(teta(2), q2, a(2), alfa(2));
A23 = denavit(teta(3), q3, a(3), alfa(3));
A03 = A01 * A12 * A23;

y3 = A03(1:3,2);

sq4 = dot(T(1:3,1), y3);   % Vector orientación n: T(1:3,1)
cq4 = dot(T(1:3,2), y3);   % Vector orientación s: T(1:3,2)
q4  = atan2(sq4, cq4);

% Vector de variables articulares
q  = [q1 q2 q3 q4]';
```

⇒ Se observa como la cinemática directa está incluida en los cálculos necesarios para obtener la matriz ${}^{0}A_{3}$.

En el ejemplo mostrado a continuación se puede comprobar como después de asignar un vector de coordenadas articulares aleatorio, y obtener la matriz homogénea del extremo de robot correspondiente a este vector, si sobre esta matriz se aplica la función INVERSEKINEMATIC4 se obtiene el vector q original.

```
» q=rand(4,1)

q =

    0.8913
    0.7621
    0.4565
    0.0185

» T=directkinematic4(q)

T =

    0.6283   -0.0116   -0.7779   -0.5735
    0.7778   -0.0144    0.6284    0.3347
   -0.0185   -0.9998    0.0000    1.1621
         0         0         0    1.0000

» inversekinematic4(T)

ans =

    0.8913
    0.7621
    0.4565
    0.0185
```

Ejemplo 2.6

Solución del robot rotacional de 6 grados de libertad.

En contraste con el ejemplo anterior, la solución de la cinemática inversa de un robot de 6 grados de libertad requiere del lector un mayor esfuerzo de comprensión. En este ejemplo se va a resolver mediante el método geométrico. Para ello, en virtud del principio de Pieper's, es necesario separar el cálculo de la posición y orientación del extremo del robot.

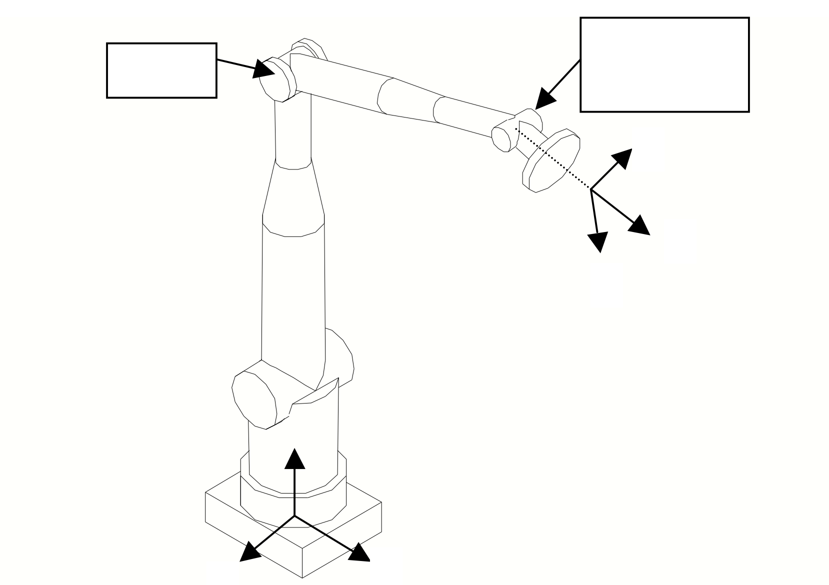
En primer lugar se calcula la posición del punto de intersección de los 3 últimos grados de libertad, que se conoce como MUÑECA del robot. Con las coordenadas **(p_x, p_y, p_z)** de la muñeca se resuelve el problema de la posición, de manera similar al ejemplo anterior.

Los 3 últimos grados de libertad se utilizan para orientar la herramienta colocada en el extremo del robot, y se resuelven en un paso posterior.

Como se observa en la figura 2.6, la articulación 3 (θ_3), en adelante CODO del robot, puede tener dos valores distintos, conocidos como configuración CODO ARRIBA y CODO ABAJO, y que representan los casos en que la articulación 3 está situada por debajo o por encima de la articulación 2, para lograr una misma posición de la muñeca. Asimismo, la MUÑECA puede tomar también dos configuraciones distintas, MUÑECA ARRIBA y MUÑECA ABAJO, para obtener una misma orientación del efector final.

Debido a las anteriores consideraciones, es necesario introducir en la resolución del problema cinemático inverso dos parámetros, llamados por similitud CODO y MUÑECA, que representen las cuatro posibles configuraciones que puede adoptar el robot cuando se le solicita alcanzar una posición y orientación determinadas.

Para una descripción más amplia acerca de la resolución del problema cinemático inverso se recomienda estudiar las referencias: [1] y [2].

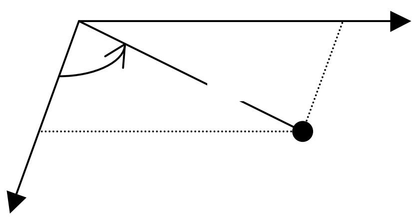


A continuación se muestra la solución geométrica del robot estudiado y su implementación en ***Matlab®***.

⇒ Se recomienda seguir las explicaciones con el código fuente de la función INVERSEKINEMATIC6.

Problema de posición. Cálculo de las tres primeras articulaciones.

Articulación 1



$$R = \sqrt{p_x^2 + p_y^2}$$

$$sin\theta_1 = \frac{p_y}{R} \qquad \cos\theta_1 = \frac{p_x}{R}$$

$$\theta_1 = atan2(sin\theta_1, \cos\theta_1)$$

x_2 p_x, p_y, p_z

l_3 p_x, p_y, p_z r l_3

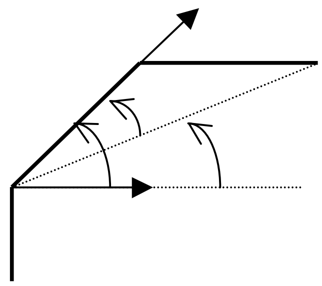
β r β x_2

l_2 θ_2 α l_2 θ_2 α

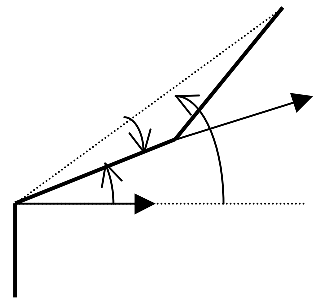
x_1 l_1 x_1

Articulación 2

Para la resolución de la segunda articulación, se ha de tener en cuenta la posibilidad de CODO ARRIBA y CODO ABAJO. En cada caso el ángulo θ_2 se calculará como la suma o la resta de los ángulos α y β calculados.



Configuración CODO ARRIBA



Configuración CODO ABAJO

Identificando las relaciones geométricas presentes en los anteriores esquemas:

$$r = \sqrt{p_x^2 + p_y^2 + (p_z - l_1)^2}$$

$$l_3^2 = r^2 + l_2^2 - 2rl_2 \cos\beta$$

$$\frac{l_3^2 - r^2 + l_2^2}{-2rl_2} = \cos\beta$$

$$sin\beta = \pm\sqrt{1 - \cos^2\beta}$$

$$sin\alpha = \frac{l_1 - p_z}{r}$$

$$\cos\alpha = \frac{R}{r}$$

$$\theta_2 = \alpha + \beta$$

Configuración CODO ARRIBA

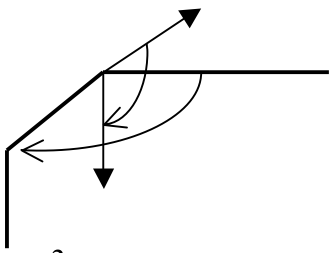
$$\theta_2 = \alpha - \beta$$

Configuración CODO ABAJO

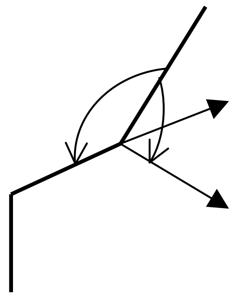
x_2 l_3 l_3 l_2 θ_3 β β x_2 θ_3 l_2 x_3 l_1 x_3

Articulación 3

Al igual que en la articulación 2, se debe tener en cuenta las dos posibles configuraciones del codo.



$$\theta_3 = \frac{3\pi}{2} - \beta$$



$$\theta_3 = \beta - \frac{\pi}{2}$$

Problema de orientación. Cálculo de las tres últimas articulaciones.

Conocidos los tres primeros ángulos θ_1, θ_2 y θ_3 se resuelve la cinemática directa para los tres primeros eslabones, obteniéndose la matriz ${}^0A_3$ necesaria para la resolución de las últimas tres articulaciones. Siguiendo la referencia [1], para resolver el problema de la orientación se han de conseguir que las tres últimas articulaciones cumplan los criterios siguientes:

1. Establecer la articulación 4 de forma tal que una rotación respecto de la rotación 5 alineará el eje de movimiento de la articulación 6 con el vector de aproximación dado (**a**).
2. La articulación 5 alineará el eje de movimiento de la articulación 6 con el vector de aproximación.
3. Fijar la articulación 6 para alinear el vector de orientación dado (s) (o el de deslizamiento (n)) y el vector normal.

Matemáticamente, estos criterios significan:

1. $z_4 = \dfrac{\pm(z_3 x a)}{\|z_3 x a\|}$
2. $\mathbf{a} = z_5$
3. $\mathbf{s} = y_6$

Ambas orientaciones de la muñeca (ARRIBA y ABAJO) se definen observando la orientación del sistema de coordenadas de la mano **(n,s,a)** con respecto al sistema de coordenadas **(x_5, y_5, z_5).** Para analizar las configuraciones de MUÑECA ARRIBA y MUÑECA ABAJO, se utiliza un parámetro de orientación Ω (omega en el código de ***Matlab®***) que hace referencia a la orientación del vector unitario **n** (o **s**) con respecto al vector unitario $\mathbf{x_5}$ (o $\mathbf{y_5}$) y que viene definido en la referencia Fu [1] como:

$$\Omega = \begin{cases} \mathbf{0} & \text{si se está en el caso degenerado} \\ \mathbf{s \cdot y_5} & \text{si } \mathbf{s \cdot y_5} \neq 0 \\ \mathbf{n \cdot y_5} & \text{si } \mathbf{s \cdot y_5} = 0 \end{cases}$$

s z_4 y_4
θ_6 θ_4 n y_5 θ_5 y_3
a
x_4
θ_6 θ_4 n x_4
x_5 x_3

Articulación 4

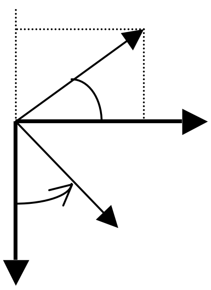
Para conocer el signo de la ecuación $z_4 = \frac{\pm(z_3 xa)}{\|z_3 xa\|}$ se determina conociendo la orientación Ω y la configuración de MUÑECA. Ω se calcula según su definición, y MUÑECA es un parámetro del robot. Se parte de la suposición de que el signo es positivo, y esta hipótesis se corrige con el parámetro M que se calcula como combinación de la configuración de la muñeca y la orientación requeridas.

Utilizando los parámetros Ω y MUÑECA se construye la siguiente tabla:

Configuración	MUÑECA	Ω	M=MUÑECA*sign(Ω)
ABAJO	+1	≥0	+1
ABAJO	+1	<0	-1
ARRIBA	-1	≥0	-1
ARRIBA	-1	<0	+1

que permite obtener M como parámetro para calcular la rotación θ_4 en cualquiera de las configuraciones posibles.

De la figura 2.4 se puede demostrar las siguientes relaciones:



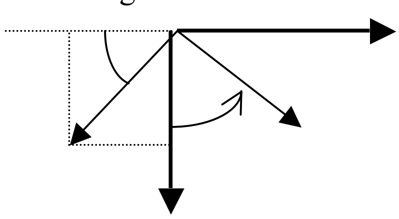
$$sin\theta_4 = -M(z_4 \bullet x_3)$$

$$\cos\theta_4 = M(z_4 \bullet y_3)$$

$$\theta_4 = atan2(sin\theta_4, \cos\theta_4)$$

Articulación 5

De la figura 2.4 se obtiene:

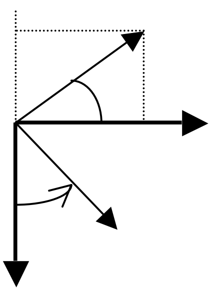


$$sin\theta_5 = a \bullet x_4$$

$$\cos\theta_5 = -(a \bullet y_4)$$

$$\theta_5 = atan2(sin\theta_5, \cos\theta_5)$$

Articulación 6



$$sin\theta_6 = n \bullet y_5$$

$$\cos\theta_6 = s \bullet y_5$$

$$\theta_6 = atan2(sin\theta_6, \cos\theta_6)$$

Código en Matlab®.La función INVERSEKINEMATIC6 resuelve la cinemática inversa del robot rotacional de 6 gdl. Para ello toma como parámetros la matriz homogénea T, y los parámetros CODO y MUÑECA para definir las posibles configuraciones. Ha de tenerse en consideración que determinados puntos y orientaciones que pertenezcan al espacio de trabajo del robot, no podrán alcanzarse con algunas configuraciones. Por ello se recomienda al lector que trate de experimentar con estos puntos extremos con la herramienta que a continuación se presenta, y analice los resultados.

```
% INVERSEKINEMATIC6  Inverse Kinematic
%   Q = INVERSEKINEMATIC6(T, CODO, MUNECA) devuelve el vector de coordenadas
%   articulares correspondiente a la solución cinemática inversa de la mano
%   del manipulador en la posición y orientación expresadas en la matriz T.
%   CODO = 1 indica codo del robot arriba, es decir, que la articulación 3 se
%   sitúa por encima de la articulación 2,  mientras que CODO = -1 indica codo
%   abajo, es decir que la articulación 2 se sitúa por encima de la 3.
%   MUNECA = 1 indica que la muñeca del robot se sitúa por debajo de la
coordenada
%   expresada en T, mientras que MUNECA = -1 significa que la muñeca se sitúa
%   por arriba.
%
%   See also DIRECTKINEMATIC6, DENAVIT.

function q = inversekinematic6(T,codo,muneca)

%  Parámetros Denavit-Hartenberg del robot
d    = [0.315  0      0     0.5     0    0.08];
a    = [0      0.45   0     0       0    0];
alfa = [-pi/2  0    pi/2  -pi/2   pi/2   0];

% Posición de la mano del manipulador
p  = T(1:3,4)-d(6)*T(1:3,3);

% Solución de la primera articulación: q1
R = sqrt(p(1)^2+p(2)^2);
sq1=p(2)/R;
cq1=p(1)/R;
q1 = atan2(sq1,cq1);

% Solución de la segunda articulación: q2
r = sqrt(R^2+(p(3)-d(1))^2);
salfa = (d(1)-p(3))/r;
calfa = R/r;

cbeta = (r^2+a(2)^2-d(4)^2)/(2*r*a(2));
sbeta = sqrt(1-cbeta^2);

if codo == -1  % Codo abajo
   sq2 = salfa*cbeta+sbeta*calfa;
   cq2 = calfa*cbeta-salfa*sbeta;
else           % Codo arriba
   sq2 = salfa*cbeta-sbeta*calfa;
   cq2 = calfa*cbeta+salfa*sbeta;
end

q2 = atan2(sq2,cq2);

% Solución de la tercera articulación: q3
cbeta=(a(2)^2+d(4)^2-r^2)/(2*a(2)*d(4));
sbeta=sqrt(1-cbeta^2);
beta=atan2(sbeta,cbeta);
```

```
if codo == 1   % Codo arriba
   q3 = 3*pi/2-beta;
else           % Codo abajo
   q3 = beta - pi/2;
end

% Solución de la cuarta articulación: q4
%  Cálculo de la matriz de transformación A03
A01 = denavit(q1, d(1), a(1), alfa(1));
A12 = denavit(q2, d(2), a(2), alfa(2));
A23 = denavit(q3, d(3), a(3), alfa(3));
A03 = A01 * A12 * A23;
x3 = A03(1:3,1);
y3 = A03(1:3,2);
z3 = A03(1:3,3);
z4 = cross(z3,T(1:3,3));   % Vector orientación a: T(1:3,3)

% Determinación del indicador de orientación omega
aux = dot(T(1:3,2),z4); % Vector orientación s: T(1:3,2)
if aux ~= 0
   omega = aux;
else
   aux=dot(T(1:3,1),z4);   % Vector orientación n: T(1:3,1)
   if aux ~=0
      omega=aux;
   else
      omega=0;
   end
end

M   = muneca*sign(omega);

sq4 = -M*dot(z4,x3);
cq4 = M*dot(z4,y3);
q4  = atan2(sq4,cq4);

% Solución de la quinta articulación: q5
z5  = T(1:3,3);      % Vector de orientación a: T(1:3,3)
A34 = denavit(q4, d(4), a(4), alfa(4));
A04 = A03 * A34;
x4  = A04(1:3,1);
y4  = A04(1:3,2);

sq5 = dot(T(1:3,3),x4); % Vector de orientación a: T(1:3,3)
cq5 = -dot(T(1:3,3),y4);   % Vector de orientación a: T(1:3,3)
q5  = atan2(sq5,cq5);

% Solución de la sexta articulación: q6
y6  = T(1:3,2);   % Vector de orientación s: T(1:3,2)

A45 = denavit(q5, d(5), a(5), alfa(5));
A05 = A04 * A45;
y5  = A05(1:3,2);

sq6 = dot(T(1:3,1),y5); % Vector de orientación n: T(1:3,1)
cq6 = dot(T(1:3,2),y5); % Vector de orientación s: T(1:3,2)
q6  = atan2(sq6,cq6);

% Vector de coordenadas articulares
q  = [q1 q2 q3 q4 q5 q6]';
```

En el ejemplo mostrado a continuación se puede comprobar como después de asignar un vector de coordenadas articulares aleatorio, lo cual ya incluye una determinada configuración de CODO y MUÑECA y obtener la matriz homogénea del extremo de robot correspondiente a este vector, si sobre esta matriz se aplica la función INVERSEKINEMATIC6 con los valores correctos de CODO y MUÑECA para el vector de coordenadas articulares, se obtiene el vector q original.

⇒ Se recomienda experimentar la función INVERSEKINEMATIC6 con vectores de coordenadas articulares sencillos de analizar.

```
» q=rand(6,1)

q =

    0.6721
    0.8381
    0.0196
    0.6813
    0.3795
    0.8318

» T=directkinematic6(q)

T =

   -0.7400   -0.3846    0.5518    0.5756
    0.6484   -0.1900    0.7372    0.4819
   -0.1787    0.9033    0.3900    0.3387
         0         0         0    1.0000

» inversekinematic6(T,-1,1)

ans =

    0.6721
    0.8381
    0.0196
    0.6813
    0.3795
    0.8318
```

2.4.- Representación gráfica en MatLab® usando alambres.

En este apartado se van a presentar una sencilla herramienta gráfica válida para comprobar los resultados anteriores utilizando las funciones ya estudiadas. Se recomienda al lector que realice estos ejemplos modificando el código de ***Matlab®***. Para comprobar los resultados se ha utilizado una nueva función nombrada como DRAWROBOT3D, que permite una sencilla representación 3D utilizando líneas de la configuración del manipulador. Ha de considerarse que esta función realiza el trazado del robot conforme a su representación D-H, por lo que los sistemas de referencia que han sido desplazados dan lugar a eslabones de longitud nula, dando la impresión óptica de que han desaparecido. Para comprobar que el efecto de estas rotaciones si es tomado en consideración se sugiere al lector que coloque un pequeño sistema de referencia en el extremo del robot.

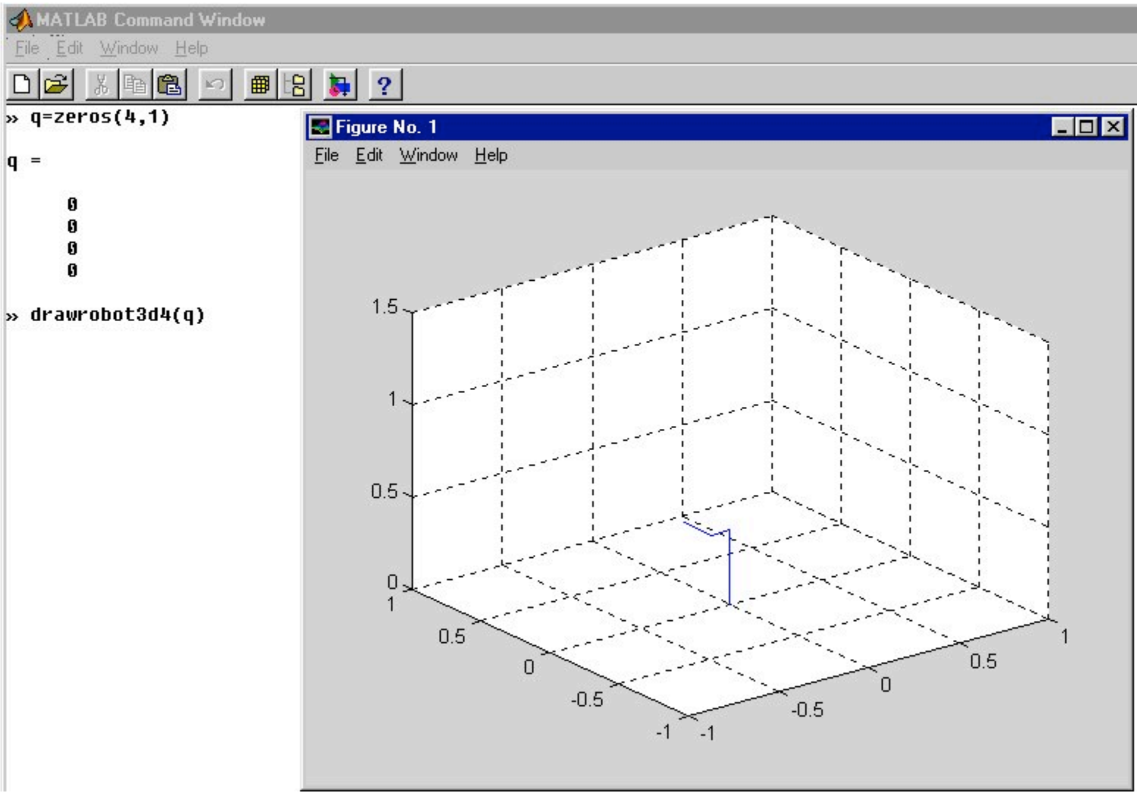
Ejemplo 2.7

Código en Matlab®. La función DRAWROBOT3D realiza una representación 3D de un robot en función del vector de variables articulares Q

Particularizando DRAWROBOT3D para los ejemplos anteriores se obtiene las funciones que se han llamado DRAWROBOT3D4 y DRAWROBOT3D6.

```
% DRAWROBOT3D4 Representación 3D de un robot.
%  DRAWROBOT3D4(Q) realiza una representación 3D de un robot
%  en función del vector de variables articulares Q.
%
%  See also DENAVIT, DIRECTKINEMATIC4.

function drawrobot3d4(q)

%  Parámetros Denavit-Hartenberg del robot
teta = [q(1)  0      0     q(4)];
d    = [0.4   q(2)   q(3)  0.2 ];
a    = [0    -0.1    0     0   ];
alfa = [0    -pi/2   0     0   ];

%  Matrices de transformación homogénea entre sistemas de coordenadas
consecutivos
A01 = denavit(teta(1), d(1), a(1), alfa(1));
A12 = denavit(teta(2), d(2), a(2), alfa(2));
A23 = denavit(teta(3), d(3), a(3), alfa(3));
A34 = denavit(teta(4), d(4), a(4), alfa(4));

%  Matrices de transformación del primer sistema al correspondiente
A02 = A01 * A12;
A03 = A02 * A23;
A04 = A03 * A34;

%  Vector de posicion (x, y, z) de cada sistema de coordenadas
x0 = 0;            y0 = 0;            z0 = 0;
x1 = A01(1,4);     y1 = A01(2,4);     z1 = A01(3,4);
xi = x1;           yi = y1;           zi = z1 + d(2);
x2 = A02(1,4);     y2 = A02(2,4);     z2 = A02(3,4);
x3 = A03(1,4);     y3 = A03(2,4);     z3 = A03(3,4);
x4 = A04(1,4);     y4 = A04(2,4);     z4 = A04(3,4);

%  Se dibuja el robot
x = [x0 x1 xi x2 x3 x4];
y = [y0 y1 yi y2 y3 y4];
z = [z0 z1 zi z2 z3 z4];
plot3(x,y,z);
% Se coloca una rejilla a los ejes
grid;
% Se establecen los límites de los ejes
axis([-1.5 1.5 -1.5 1.5 0 1.5]);
```

Si se representa la configuración en la que todas las coordenadas articulares son nulas, se obtiene el siguiente dibujo:



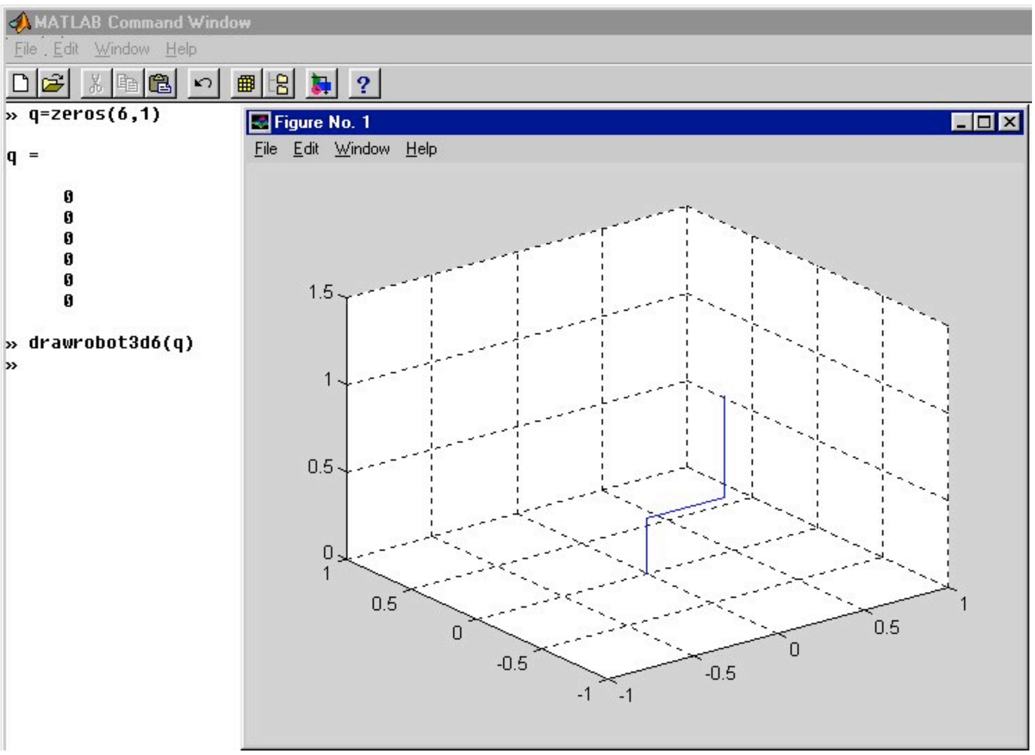
Ejemplo 2.8

Se presenta ahora el código y la representación gráfica del robot rotacional de 6 grados de libertad.

```
% DRAWROBOT3D6 Representación 3D de un robot.
%  DRAWROBOT3D6(Q) realiza una representación 3D de un robot
%  en función del vector de variables articulares Q.
%
%  See also DENAVIT, DIRECTKINEMATIC6.


function drawrobot3d6(q)

%  Parámetros Denavit-Hartenberg del robot
teta = q;
d    = [0.315  0      0     0.5     0    0.08];
a    = [0      0.45   0     0       0    0   ];
alfa = [-pi/2  0    pi/2  -pi/2   pi/2   0   ];

%  Matrices de transformación homogénea entre sistemas de coordenadas
consecutivos
A01 = denavit(teta(1), d(1), a(1), alfa(1));
A12 = denavit(teta(2), d(2), a(2), alfa(2));
A23 = denavit(teta(3), d(3), a(3), alfa(3));
A34 = denavit(teta(4), d(4), a(4), alfa(4));
A45 = denavit(teta(5), d(5), a(5), alfa(5));
A56 = denavit(teta(6), d(6), a(6), alfa(6));
```

```
%  Matrices de transformación del primer sistema al correspondiente
A02 = A01 * A12;
A03 = A02 * A23;
A04 = A03 * A34;
A05 = A04 * A45;
A06 = A05 * A56;

%  Vector de posicion (x, y, z) de cada sistema de coordenadas
x0 = 0;            y0 = 0;            z0 = 0;
x1 = A01(1,4);     y1 = A01(2,4);     z1 = A01(3,4);
x2 = A02(1,4);     y2 = A02(2,4);     z2 = A02(3,4);
x3 = A03(1,4);     y3 = A03(2,4);     z3 = A03(3,4);
x4 = A04(1,4);     y4 = A04(2,4);     z4 = A04(3,4);
x5 = A05(1,4);     y5 = A05(2,4);     z5 = A05(3,4);
x6 = A06(1,4);     y6 = A06(2,4);     z6 = A06(3,4);

%  Se dibuja el robot
x = [x0 x1 x2 x3 x4 x5 x6];
y = [y0 y1 y2 y3 y4 y5 y6];
z = [z0 z1 z2 z3 z4 z5 z6];
plot3(x,y,z);
% Se coloca una rejilla a los ejes
grid;
% Se establecen los límites de los ejes
axis([-1 1 -1 1 0 1.5]);
```

Si se representa la configuración en la que todas las coordenadas articulares son nulas, se obtiene el siguiente dibujo:



2.5.- PRACTICA. Animación de los robots.

En este apartado se va a realizar un sencillo ejemplo de utilización de las funciones estudiadas hasta ahora. Se trata de colocar al robot en dos posiciones distintas y animar una trayectoria recta entre esas dos configuraciones. Para ello se deberán generar varias posiciones intermedias.

Se utiliza la función planifica(p1,p2,ni) en la que se introducen las coordenadas cartesianas de los puntos inicial y final y el número de puntos intermedios.

Ejemplo 2.9

En este ejemplo se utiliza la cinemática inversa del robot rotacional de 6 gdl para trazar una línea recta entre un punto p1 inicial y un punto p2 final. El número de puntos intermedios es variable.

Se ha utilizado la función PLANIFICA6(P1,P2,N,S,A,CODO,MUÑECA,NPUNTOS) en el que se introduce las coordenadas cartesianas de los puntos inicial y final, la orientación (n,s,a) del punto final, los parámetros CODO y MUÑECA para seleccionar la configuración del robot y el número de puntos intermedio. Esta función proporciona una matriz de (npuntos+2) columnas por 6 filas que se utilizará por la función ANIMACION6(MAT_Q) para dibujar la trayectoria entre los dos puntos.

Código en Matlab®. La función PLANIFICA6(P1,P2,N,S,A,CODO,MUÑECA, NPUNTOS) calcula la matriz de coordenadas articulares utilizada para graficar el movimiento del robot.

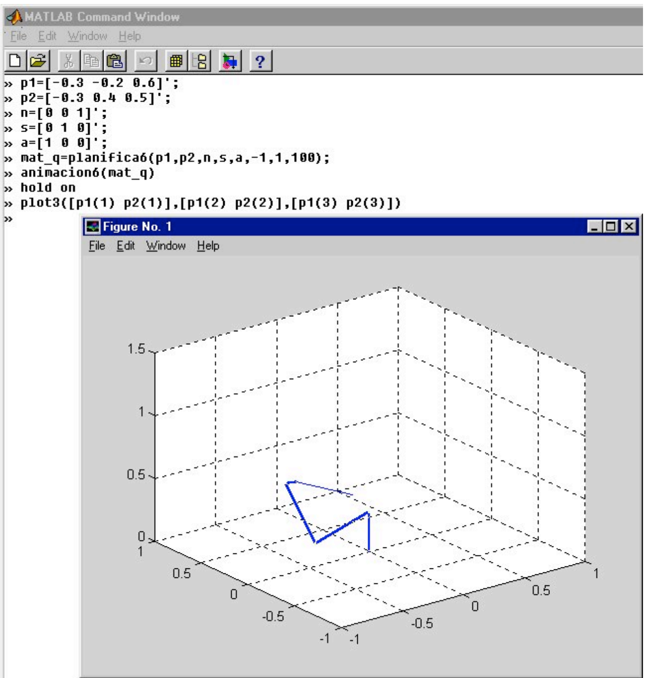
```
% PLANIFICA6  Planificación de trayectorias
%   MAT_Q = PLANIFICA6(P1, P2, N, S, A, CODO, MUNECA, NPUNTOS) realiza
%   una planificación de trayectoria en línea recta desde la coordenada
%   cartesiana P1 hasta la P2, de manera que la mano del manipulador
%   posee la orientación expresada por [N S A]. CODO = 1 indica codo del
%   robot arriba, es decir, que la articulación 3 se sitúa por encima de
%   la articulación 2, mientras que CODO = -1 indica codo abajo, es decir
%   que la articulación 2 se sitúa por encima de la 3. MUNECA = 1 indica que
%   la muñeca del robot se sitúa por debajo de la coordenada cartesiana,
%   mientras que MUNECA = -1 significa que la muñeca se sitúa por arriba.
%   NPUNTOS indica el número de puntos en los que se divide la trayectoria.
%   En MAT_Q se devuelve las coordenadas articulares, almacenadas por
%   columnas, correspondientes a cada uno de los puntos cartesianos en los
%   que se divide la trayectoria. MAT_Q es una matriz de NPUNTOS + 2 columnas
%   y 6 filas.
%
%   See also INVERSEKINEMATIC6.

function mat_q = planifica6(p1, p2, n, s, a, codo, muneca, npuntos)

% Cálculo del vector unitario
u  = p2-p1;
mu = sqrt(u(1)^2+u(2)^2+u(3)^2);
u  = (1/mu)*u;
% Cálculo de la distancia entre puntos
d = mu/(npuntos+1);
```

```
for i=0:(npuntos+1)
   % Cálculo de la posición cartesiana actual de la mano del
manipulador
   p = p1+(i*d)*u;
   T = [n s a p];
   % Cálculo de las coordenadas articulares
   q  = inversekinematic6(T,codo,muneca);
   mat_q(:,i+1) = q;
end
```

Utilizando ahora la función ANIMACION6(MAT_Q) se presenta el movimiento del robot entre los 2 puntos especificados.

```
%   ANIMACION6   Animación de la trayectoria de un robot
%      ANIMACION(MAT_Q) realiza la animación de la trayectoria, expresada
%      en la matriz MAT_Q, de un brazo robot de 6 GDL. MAT_Q contiene 6 filas
%      y una columna para cada disposición del robot.
%
%      See also PLANIFICA6, DRAWROBOT3D6.

function animacion6(mat_q)

%  Parámetros Denavit-Hartenberg del robot
d    = [0.315  0      0      0.5     0     0.08];
a    = [0      0.45   0      0       0     0   ];
alfa = [-pi/2  0     pi/2  -pi/2   pi/2   0   ];

% Vector de posicion (x, y, z) del sistema de coordenadas de
referencia
x0 = 0;        y0 = 0;        z0 = 0;

% Se dibuja el sistema de coordenadas de referencia. Se asigna el modo
XOR para borrar
% sólo el robot dibujado anteriormente. Se utiliza un grosor de línea
de 2 unidades
p = plot3(x0,y0,z0,'EraseMode','xor','LineWidth',2);
% Se asigna una rejilla a los ejes
grid;
% Se establecen los límites de los ejes
axis([-1 1 -1 1 0 1.5]);
% Mantiene el gráfico actual
hold on;

% Número de columnas de la matriz
n = size(mat_q,2);

% Se dibuja la disposición del robot correspondiente a cada columna
for i=1:n

   % Variables articulares del brazo robot
   teta1 = mat_q(1,i);
   teta2 = mat_q(2,i);
   teta3 = mat_q(3,i);
   teta4 = mat_q(4,i);
   teta5 = mat_q(5,i);
   teta6 = mat_q(6,i);
```

```
%  Matrices de transformación homogénea entre sistemas de coordenadas
consecutivos
   A01 = denavit(teta1, d(1), a(1), alfa(1));
   A12 = denavit(teta2, d(2), a(2), alfa(2));
   A23 = denavit(teta3, d(3), a(3), alfa(3));
   A34 = denavit(teta4, d(4), a(4), alfa(4));
   A45 = denavit(teta5, d(5), a(5), alfa(5));
   A56 = denavit(teta6, d(6), a(6), alfa(6));

   %  Matrices de transformación del primer sistema al correspondiente
   A02 = A01 * A12;
   A03 = A02 * A23;
   A04 = A03 * A34;
   A05 = A04 * A45;
   A06 = A05 * A56;

   %  Vector de posicion (x, y, z) de cada sistema de coordenadas
   x1 = A01(1,4);     y1 = A01(2,4);     z1 = A01(3,4);
   x2 = A02(1,4);     y2 = A02(2,4);     z2 = A02(3,4);
   x3 = A03(1,4);     y3 = A03(2,4);     z3 = A03(3,4);
   x4 = A04(1,4);     y4 = A04(2,4);     z4 = A04(3,4);
   x5 = A05(1,4);     y5 = A05(2,4);     z5 = A05(3,4);
   x6 = A06(1,4);     y6 = A06(2,4);     z6 = A06(3,4);

   %  Se dibuja el robot
   x = [x0 x1 x2 x3 x4 x5 x6];
   y = [y0 y1 y2 y3 y4 y5 y6];
   z = [z0 z1 z2 z3 z4 z5 z6];
   set(p,'XData',x,'YData',y,'ZData',z);
   % Se fuerza a MATLAB a actualizar la pantalla
   drawnow;
end
```



⇒ Notar que la función plot3 ha permitido dibujar sobre la animación la trayectoria seguida por el extremo del robot.

EJERCICIOS PROPUESTOS

Se pide implementar las funciones PLANIFICA4 y ANIMACIÓN4 para el ejemplo con el robot prismático de 4 gdl y realizar una animación entre dos puntos del espacio de trabajo del robot.