

Práctica 2

Cinemática inversa del robot 4 gdl

Ejemplo 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 ($\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z$) 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 ($\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z$) sobre el plano ($\mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0$) 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.

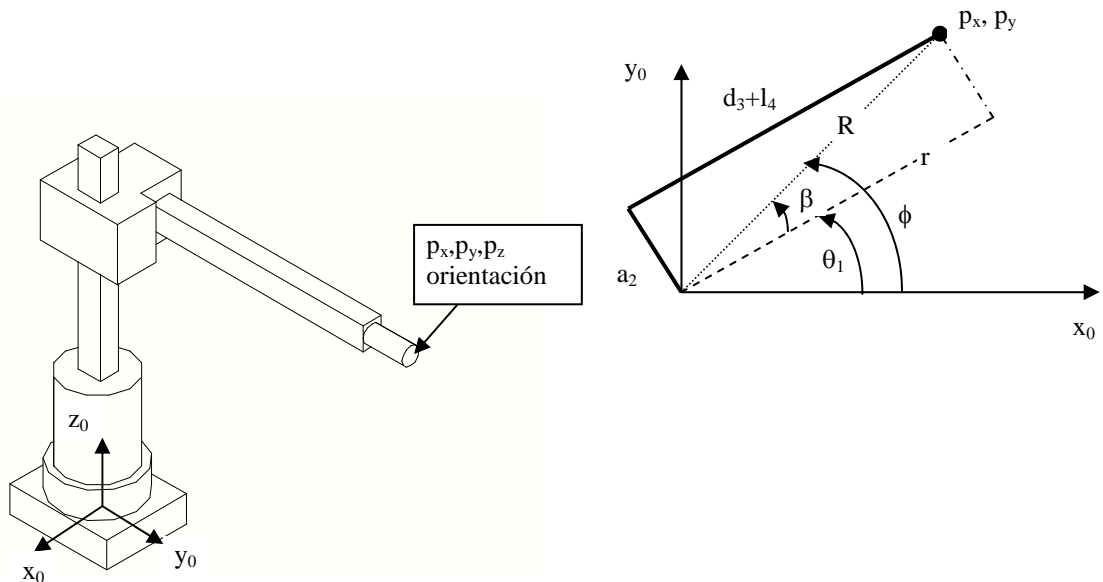


Figura-2.5 Cinemática inversa del robot de 4 gdl.

De las que se deducen las siguientes relaciones:

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

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

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

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

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

$$\phi = \text{atan2}(\sin \phi, \cos \phi)$$

$$\beta = \text{atan2}(\sin \beta, \cos \beta)$$

que permiten el cálculo de θ_1 como $\phi - \beta$.

Articulación 2

De la figura 2.3 se obtiene la siguiente fórmula:

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

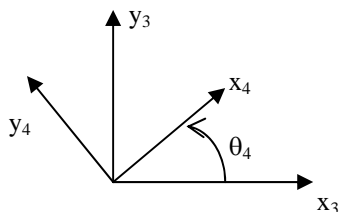
Articulación 3

De la figura 2.5:

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

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 **z₄** se deben cumplir las siguientes relaciones:



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

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

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 0A_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 en un robot 6R (R-R-R-R-R-R)

⇒ 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 =

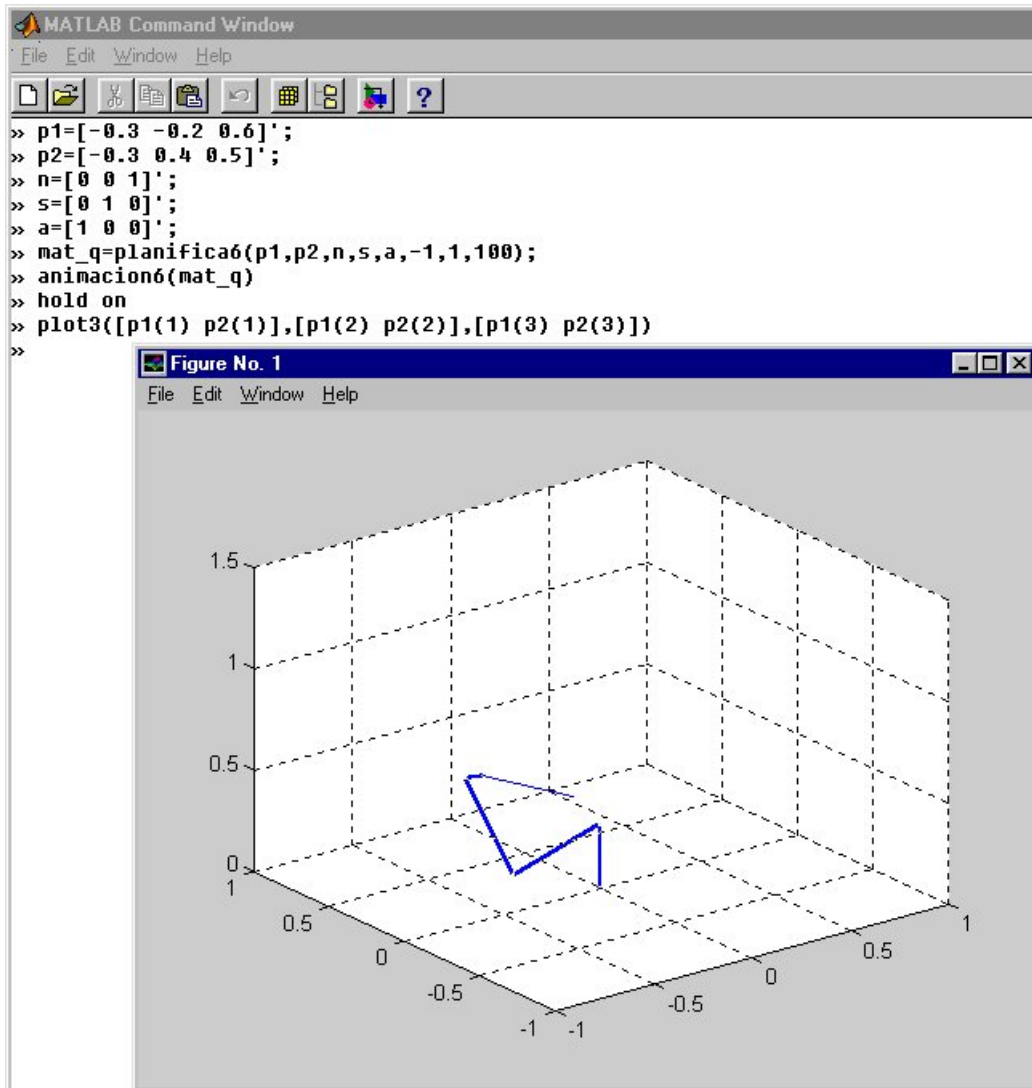
    0.6721
    0.8381
    0.0196
    0.6813
    0.3795
    0.8318

```

PRACTICA. Animación de los robots.Ejemplo con el robot 6R

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.



⇒ 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.