Actividad 5 | Robot 2 (6 GDL)

Oscar Ortiz Torres A01769292

```
clc
clear all
close all
```

Para encontrar la cinemática directa, se establecen los diferentes sistemas de coordenadas para cada articulación, así como las variables para los ángulos de giros y de las medidas de los eslabones.

```
% Declaración de las variables simbólicas para los ángulos y medidas de los
eslabones del robot
syms th0(t) th1(t) th2(t) th3(t) th4(t) th5(t) l0 l1 l2 l3 l4 l5 t

% Configuración del robot, de cada GDL
RP = [0 0 0 0 0 0];

% Creación del vector de coordenadas articulares
Q = [th0 th1 th2 th3 th4 th5];

% Creación del vector de velocidades generalizadas
Qp = diff(Q, t);

% Número de grados de libertad
GDL = size(RP, 2);
GDL_str = num2str(GDL);
```

Posteriormente, se determinan los sistemas de coordenadas, haciendo que el eje Z siempre concuerde con el eje de rotación de la junta. De esta forma, se analizan las rotaciones necesarias para pasar de un sistema de coordenadas a otro de manera secuencial, siempre considerando cada articulación como independiente y siguiendo la regla de la mano derecha.

Articulación 1

La medida l0 de este eslabón gira sobre el eje Z, haciendo que no exista variación en su posición, por lo que se mantiene como constante en este eje en su vector de posición.

```
% Posición de la articulación 0 a 1
P(:,:,1) = [
            0;
            0;
            l0
          ];
```

Para poder pasar del sistema de coordenadas 0 al 1, se deben aplicar dos tranformaciones de rotación, una de 90° en el eje Y y otra de -90° en el eje Z (respecto al sistema creado con la primera rotación). Los signos se deben a la regla de la mano derecha.

```
% Matriz de rotación de la junta 0 a 1
R(:,:,1) = [
            cos(th0)   -sin(th0)     0;
            sin(th0)    cos(th0)     0;
            0           0            1
            ]*rotY(90)*rotZ(-90);
```

Articulación 2

La rotación de esta articulación afecta directamente a la posición final del eslabón, por lo que se debe descomponer en sus componentes X y Y para el vector de posición.

```
% Posición de la articulación 1 a 2
P(:,:,2) = [
            l1*cos(th1);
            l1*sin(th1);
            0
          ];
```

La matriz de rotación se mantiene únicamente con la rotación en el eje Z por el movimiento natural de la articulación.

```
% Matriz de rotación de la junta 1 a 2
R(:,:,2) = [
            cos(th1)   -sin(th1)     0;
            sin(th1)    cos(th1)     0;
            0           0            1
            ];
```

Articulación 3

Como en el caso anterior, la rotación de esta articulación afecta directamente a la posición final del eslabón, así que se descompone en sus componentes X y Y para el vector de posición.

```
% Posición de la articulación 2 a 3
P(:,:,3) = [
            l2*cos(th2);
            l2*sin(th2);
            0
          ];
```

Para ir del sistema de coordenadas 2 al 3 es necesario aplicar una rotación de 90° en el eje Y, siendo positiva por la regla de la mano derecha.

```
% Matriz de rotación de la junta 2 a 3
```

```
R(:,:,3) = [
             cos(th2)   -sin(th2)     0;
             sin(th2)    cos(th2)     0;
             0           0            1
             ]*rotY(90);
```

Articulación 4

El parámetro l3 de este eslabón gira sobre el eje Z, haciendo que no exista variación en su posición, por lo que se mantiene como constante en este eje en su vector de posición.

```
% Posición de la articulación 3 a 4
P(:,:,4) = [
             0;
             0;
             l3
           ];
```

Para ir del sistema de coordenadas 3 al 4 es necesario aplicar una rotación de -90° en el eje Y, siendo negativa por la regla de la mano derecha.

```
% Matriz de rotación de la junta 3 a 4
R(:,:,4) = [
             cos(th3)   -sin(th3)     0;
             sin(th3)    cos(th3)     0;
             0           0            1
             ]*rotY(-90);
```

Articulación 5

La rotación de esta articulación afecta directamente a la posición final del eslabón, así que se descompone en sus componentes X y Y para el vector de posición.

```
% Posición de la articulación 4 a 5
P(:,:,5) = [
             l4*cos(th4);
             l4*sin(th4);
             0
           ];
```

Para poder convertir el sistema de coordenadas 3 al 4 es necesario aplicar una rotación de 90° en el eje X, siendo positiva por la regla de la mano derecha.

```
% Matriz de rotación de la junta 4 a 5
R(:,:,5) = [
             cos(th4)   -sin(th4)     0;
             sin(th4)    cos(th4)     0;
             0           0            1
             ]*rotX(90);
```

Articulación 6

Esta última articulación se interpretó como la junta que conecta al brazo con el efector final, por lo que el valor de l5 es la distancia que hay entre estos dos cuerpos. Se mantiene constante en el vector de posición debido a que la rotación en Z no afecta la posición final del eslabón.

```
% Posición de la articulación 5 a 6
P(:,:,6) = [
             0;
             0;
             l5
           ];
```

Se interpreta que el efector final tiene el mismo sistema de coordenadas que el anterior, por lo que no hay que aplicar ninguna rotación.

```
% Matriz de rotación de la junta 5 a 6
R(:,:,6) = [
             cos(th5)   -sin(th5)     0;
             sin(th5)    cos(th5)     0;
             0           0            1
             ];
```

Matrices

Una vez definidas los vectores de posición y las matrices de rotación de cada articulación, se arman las 6 matrices de tranformación homogenea locales y, al multiplicarlas entre ellas, se obtiene la matriz de tranformación homogenea global.

```
% Creación de vector de ceros
vector_zeros = zeros(1,3);

% Inicialización de las matrices de Transformación Homogenea locales
A(:, :, GDL) = simplify([ ...
                        R(:,:,GDL)      P(:,:,GDL); ...
                        vector_zeros    1 ...
                        ]);

% Inicialización de las matrices de tranformación Homogenea globales
T(:,:,GDL) = simplify([ ...
                        R(:,:,GDL)      P(:,:,GDL); ...
                        vector_zeros    1 ...
                        ]);

% Inicialización de los vectores de posición vistos desde el marco de
% referencia inercial
PO(:,:,GDL) = P(:,:,GDL);

% Inicialización de las matrices de rotación vistas desde el marco de
```

```
% referencia inercial
RO(:,:,GDL) = R(:,:,GDL);

for i = 1:GDL
    i_str = num2str(i);

    % Locales
    disp(strcat('Matriz de Transformación local A', i_str));
    A(:,:,i) = simplify([ ...
                         R(:,:,i)       P(:,:,i); ...
                         vector_zeros   1 ...
                        ]); A(:,:,i)

    % Globales
    try
        T(:,:,i) = T(:,:,i-1)*A(:,:,i);
    catch
        T(:,:,i) = A(:,:,i); % Caso específico cuando i=1 nos marcaría error en try
    end

    %disp(strcat('Matruz de Transformación global T', i_str));
    T(:,:,i) = simplify(T(:,:,i));

    % Obtención de la matriz de rotación "RO" y el vector de traslación PO
    % de la matriz de transformación homogenea global T(:,:, GDL)
    RO(:,:,i) = T(1:3,1:3,i);
    PO(:,:,i) = T(1:3,4,i);
end
```

```
Matriz de Transformación local A1
ans =
```

$$\begin{pmatrix} \sin(\mathrm{th}_0(t)) & 0 & \cos(\mathrm{th}_0(t)) & 0 \\ -\cos(\mathrm{th}_0(t)) & 0 & \sin(\mathrm{th}_0(t)) & 0 \\ 0 & -1 & 0 & l_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matriz de Transformación local A2
ans =
```

$$\begin{pmatrix} \cos(\mathrm{th}_1(t)) & -\sin(\mathrm{th}_1(t)) & 0 & l_1\cos(\mathrm{th}_1(t)) \\ \sin(\mathrm{th}_1(t)) & \cos(\mathrm{th}_1(t)) & 0 & l_1\sin(\mathrm{th}_1(t)) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matriz de Transformación local A3
ans =
```

$$\begin{pmatrix} 0 & -\sin(\mathrm{th}_2(t)) & \cos(\mathrm{th}_2(t)) & l_2\cos(\mathrm{th}_2(t)) \\ 0 & \cos(\mathrm{th}_2(t)) & \sin(\mathrm{th}_2(t)) & l_2\sin(\mathrm{th}_2(t)) \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matriz de Transformación local A4
ans =
```

$$\begin{pmatrix} 0 & -\sin(\mathrm{th}_3(t)) & -\cos(\mathrm{th}_3(t)) & 0 \\ 0 & \cos(\mathrm{th}_3(t)) & -\sin(\mathrm{th}_3(t)) & 0 \\ 1 & 0 & 0 & l_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matriz de Transformación local A5
ans =
```

$$\begin{pmatrix} \cos(\mathrm{th}_4(t)) & 0 & \sin(\mathrm{th}_4(t)) & l_4\cos(\mathrm{th}_4(t)) \\ \sin(\mathrm{th}_4(t)) & 0 & -\cos(\mathrm{th}_4(t)) & l_4\sin(\mathrm{th}_4(t)) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matriz de Transformación local A6
ans =
```

$$\begin{pmatrix} \cos(\mathrm{th}_5(t)) & -\sin(\mathrm{th}_5(t)) & 0 & 0 \\ \sin(\mathrm{th}_5(t)) & \cos(\mathrm{th}_5(t)) & 0 & 0 \\ 0 & 0 & 1 & l_5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La matriz de transformación homogenea global contiene las tranformaciones de traslación y rotación necesarias para poder encontrar exactamente donde se encuentra el efector final respecto al origen del robot, para de igual manera poder manipularlo de manera exacta y eficiente.

```
disp('Matriz de Transformación global'); T(:,:,GDL)
```

```
Matriz de Transformación global
```

Para poder calcular los vectores de velocidad linear y angular, es necesario calcula el Jacobiano, en este caso se determina de forma analitica.

```
% Inicialización de jacobianos analíticos (lineal y angular)
Jv_a(:,GDL) = PO(:,:,GDL);
Jw_a(:,GDL) = PO(:,:,GDL);

for k = 1:GDL
    if(RP(k) == 0)
        % Para las articulaciones rotacionales
        try
            Jv_a(:,k) = cross(RO(:,3,k-1), PO(:,:,GDL) - PO(:,:,k-1));
            Jw_a(:,k) = RO(:,3,k-1);
        catch
            Jv_a(:,k) = cross([0 0 1], PO(:,:,GDL)); % Matriz de rotación de 0 con
respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a(:,k) = [0 0 1]; % Si no hay matriz de rotación previa se obtiene
la matriz identidad
        end
    elseif(RP(k) == 1)
        % Para las articulaciones prismáticas
```

```
        try
            Jv_a(:,k) = RO(:,3,k-1);
        catch
            Jv_a(:,k) = [0,0,1]; % Si no hay matriz de rotación previa se obtiene
la matriz identidad
        end
            Jw_a(:,k) = [0 0 0];
    end
end
```

Despliegue

El vector de velocidad lineal representa el movimiento que hay en cada eje, y debido a la propagación de las diferentes rotaciones, multiples eslabones se mueven en el mismo eje respecto a el sistema de coordenadas inicial.

```
disp('Velocidad lineal obtenida mediante el Jacobiano lineal'); V = simplify(Jv_a *
Qp')
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal
V(t) =
```

$$\begin{pmatrix} l_1\,\sigma_4\cos(\mathrm{th}_0(t))\cos(\mathrm{th}_1(t)) - l_1\,\sigma_2\sin(\mathrm{th}_0(t))\sin(\mathrm{th}_1(t)) + l_2\,\sigma_4\cos(\mathrm{th}_0(t))\cos(\sigma_8) + l_3\,\sigma_4\cos(\mathrm{th}_0(t))\,\mathrm{cc} \\ l_1\,\sigma_4\cos(\mathrm{th}_1(t))\sin(\mathrm{th}_0(t)) + l_1\,\sigma_2\cos(\mathrm{th}_0(t))\sin(\mathrm{th}_1(t)) + l_2\,\sigma_4\sin(\mathrm{th}_0(t))\cos(\sigma_8) + l_2\,\sigma_2\cos(\mathrm{th}_0(t))\sin(\mathrm{th}_1(t)) \\ l_2\,\sigma_2\sin(\mathrm{th}_1(t))\sin(\mathrm{th}_2(t)) - l_2\,\sigma_2\cos(\mathrm{th}_1(t))\cos(\mathrm{th}_2(t)) - l_2\,\sigma_3\cos(\mathrm{th}_1(t))\cos(\mathrm{th}_2(t)) - l_3\,\sigma_2\cos(\mathrm{th}_1(t))\cos(\mathrm{t} \end{pmatrix}$$

where

$$\sigma_1 = \overline{\frac{\partial}{\partial t}\,\mathrm{th}_4(t)}$$

$$\sigma_2 = \overline{\frac{\partial}{\partial t}\,\mathrm{th}_1(t)}$$

$$\sigma_3 = \overline{\frac{\partial}{\partial t}\,\mathrm{th}_2(t)}$$

$$\sigma_4 = \overline{\frac{\partial}{\partial t}\,\mathrm{th}_0(t)}$$

$$\sigma_5 = \overline{\frac{\partial}{\partial t}\,\mathrm{th}_3(t)}$$

$$\sigma_6 = \sin(\mathrm{th}_3(t))^2$$

$$\sigma_7 = \cos(\mathrm{th}_3(t))^2$$

$$\sigma_8 = \mathrm{th}_1(t) + \mathrm{th}_2(t)$$

El vector de velocidad angular representa en qué ejes existe rotación y, similar al vector anterior, debido a las rotaciones involucradas en el sistema, existe movimiento rotativo en los 3 ejes del sistema inicial del robot.

```
disp('Velocidad angular obtenida mediante el Jacobiano angular'); W = simplify(Jw_a
* Qp')
```

```
Velocidad angular obtenida mediante el Jacobiano angular
W(t) =
```

$$\begin{pmatrix} \sigma_2\ (\cos(\mathrm{th}_0(t))\cos(\mathrm{th}_3(t)) + \sin(\mathrm{th}_0(t))\sin(\mathrm{th}_3(t))\sin(\sigma_4)) - \sigma_1\ (\cos(\mathrm{th}_4(t))\ (\cos(\mathrm{th}_0(t))\sin(\mathrm{th}_3(t)) - \cos(\mathrm{th}_3 \\ \sigma_2\ (\cos(\mathrm{th}_3(t))\sin(\mathrm{th}_0(t)) - \cos(\mathrm{th}_0(t))\sin(\mathrm{th}_3(t))\sin(\sigma_4)) - \sigma_1\ (\cos(\mathrm{th}_4(t))\ (\sin(\mathrm{th}_0(t))\sin(\mathrm{th}_3(t)) + \cos(\mathrm{th}_0(\\ \overline{\frac{\partial}{\partial t}\ \mathrm{th}_0(t)} - \sigma_3\sin(\sigma_4) - \sigma_1\ (\sin(\mathrm{th}_4(t))\sin(\sigma_4) - \mathrm{c}(\end{pmatrix}$$

where

$$\sigma_1 = \overline{\frac{\partial}{\partial t}\ \mathrm{th}_5(t)}$$

$$\sigma_2 = \overline{\frac{\partial}{\partial t}\ \mathrm{th}_4(t)}$$

$$\sigma_3 = \overline{\frac{\partial}{\partial t}\ \mathrm{th}_3(t)}$$

$$\sigma_4 = \mathrm{th}_1(t) + \mathrm{th}_2(t)$$

$$\sigma_5 = \overline{\frac{\partial}{\partial t}\ \mathrm{th}_2(t)}$$

$$\sigma_6 = \overline{\frac{\partial}{\partial t}\ \mathrm{th}_1(t)}$$

Funciones de rotacion

```
function r_x = rotX(th)
    r_x = [1 0 0; 0 cosd(th) -sind(th); 0 sind(th) cosd(th)];
end

function r_y = rotY(th)
    r_y = [cosd(th) 0 sind(th); 0 1 0; -sind(th) 0 cosd(th)];
end

function r_z = rotZ(th)
    r_z = [cosd(th) -sind(th) 0; sind(th) cosd(th) 0; 0 0 1];
end
```