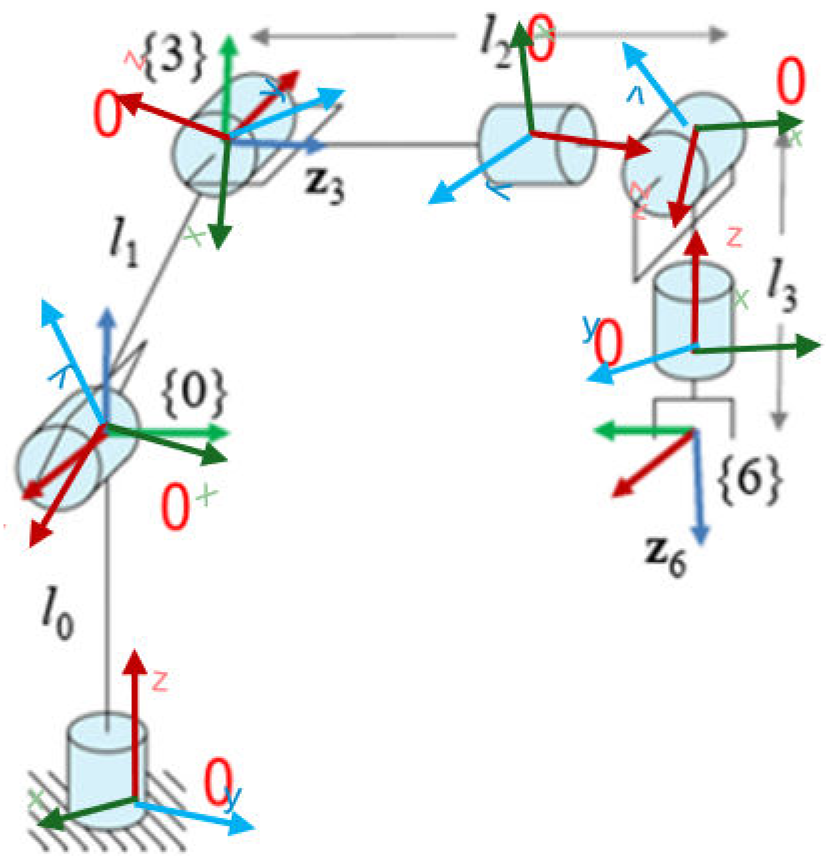
Actividad 3 | Sistema de interacción

Ana Itzel Hernández García A01737526

Obtener la **cinemática directa** del robot:

- **Obtener** el vector de velocidades lineales y el vector de velocidades angulares para los siguientes robots manipuladores



Se limpian las variables

```
clear all
close all
clc
```

Delcaración de variables simbólicas (**6 grados libertad**)

```
syms th1(t) th2(t) th3(t) th4(t) th5(t) th6(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 = [th1, th2, th3, th4, th5, th6];
```

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

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

Articulación 1

Posición de la articulación 1 a 2

Debido a que cuenta con una traslación en el eje z se toma en cuenta la longitud que se encuentra en ese eje, es decir l0

```
P(:,:,1) = [
             0;
             0;
             l0
            ];
```

Matriz de rotación de la junta 1 a 2

A la matriz de rotación Z se le realiza una transfromación con una rotación de 180° positivos en Y

```
R(:,:,1) = [
             cos(th1)   -sin(th1)     0;
             sin(th1)    cos(th1)     0;
             0           0            1
             ]*rotY(180);
```

Articulación 2

Posición de la articulación 2 a 3

Como la rotación afecta la posición de la articulación 2 entonces usamos la forma: longitud * cos(angulo) en X , longitud * sen(angulo) en Y

```
P(:,:,2) = [
            l1*cos(th2);
            l1*sin(th2);
            0
            ];
```

Matriz de rotación 2 a 3

A la matriz de rotación Z se le realiza una transfromación con una rotación de 90° negativos en Y

```
R(:,:,2) = [
             cos(th2)   -sin(th2)     0;
             sin(th2)    cos(th2)     0;
             0           0            1
             ]*rotY(-90);
```

Articulación 3

Posición de la articulación 3 respecto a la 4

Como la rotación afecta la posición de la articulación 2 entonces usamos la forma: longitud * cos(angulo) en X , longitud * sen(angulo) en Y

```
P(:,:,3) = [
             l2*cos(th3);
             l2*sin(th3);
             0
           ];
```

Matriz de rotación de la junta 3 respecto a 2 0°

A la matriz de rotación Z se le realiza una transfromación con una rotación de 180° negativos en Y

```
R(:,:,3) = [
             cos(th3)   -sin(th3)     0;
             sin(th3)    cos(th3)     0;
             0           0            1
             ]*rotY(-180);
```

Articulación 4

Posición de la articulación 4 respecto a la 5

Como la rotación afecta la posición de la articulación 4 entonces usamos la forma: longitud * cos(angulo) en X , longitud * sen(angulo) en Y

```
% Posición de la articulación 4 a 5
P(:,:,3) = [
             l3*sin(th4);
             l3*cos(th4);
             0
           ];
```

Matriz de rotación de la junta 4 respecto a 5

A la matriz de rotación Z se le realiza una transfromación con una rotación de 90° positivos en Y

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

Articulación 5

Posición de la articulación 5 respecto a la 6

Como la rotación afecta la posición de la articulación 4 entonces usamos la forma: longitud * cos(angulo) en X , longitud * sen(angulo) en Y

```
% Posición de la articulación 5 a 6
P(:,:,5) = [
            l4*sin(th5);
            l4*cos(th5);
            0
           ];
```

Matriz de rotación de la junta 5 respecto a 6

A la matriz de rotación Z se le realiza una transfromación con una rotación de 90° negativos en X

```

% Matriz de rotación de la junta 5 a 6
R(:,:,5) = [
             cos(th5)   -sin(th5)     0;
             sin(th5)    cos(th5)     0;
             0           0            1
             ]*rotX(-90);
```

Articulación 6

Posición de la articulación 6

Debido a que cuenta con una traslación en el eje z se toma en cuenta la longitud que se encuentra en ese eje, es decir l5

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

Matriz de rotación de la junta 6

A la matriz de rotación Z se mantiene debido a que no hay ninguna transformación

```

% Matriz de rotación de la junta 6
R(:,:,6) = [
             cos(th6)   -sin(th6)     0;
             sin(th6)    cos(th6)     0;
             0           0            1
             ];
```

Matrices

```
% 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)); 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 =
```

$$\begin{pmatrix} -\cos(\mathrm{th}_1(t)) & -\sin(\mathrm{th}_1(t)) & 0 & 0 \\ -\sin(\mathrm{th}_1(t)) & \cos(\mathrm{th}_1(t)) & 0 & 0 \\ 0 & 0 & -1 & l_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matruz de Transformación global T1
ans =
```

$$\begin{pmatrix} -\cos(\mathrm{th}_1(t)) & -\sin(\mathrm{th}_1(t)) & 0 & 0 \\ -\sin(\mathrm{th}_1(t)) & \cos(\mathrm{th}_1(t)) & 0 & 0 \\ 0 & 0 & -1 & l_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

$$\begin{pmatrix} 0 & -\sin(\mathrm{th}_2(t)) & -\cos(\mathrm{th}_2(t)) & l_1\cos(\mathrm{th}_2(t)) \\ 0 & \cos(\mathrm{th}_2(t)) & -\sin(\mathrm{th}_2(t)) & l_1\sin(\mathrm{th}_2(t)) \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matruz de Transformación global T2
ans =
```

$$\begin{pmatrix} 0 & -\sin(\mathrm{th}_1(t)-\mathrm{th}_2(t)) & \cos(\mathrm{th}_1(t)-\mathrm{th}_2(t)) & -l_1\cos(\mathrm{th}_1(t)-\mathrm{th}_2(t)) \\ 0 & \cos(\mathrm{th}_1(t)-\mathrm{th}_2(t)) & \sin(\mathrm{th}_1(t)-\mathrm{th}_2(t)) & -l_1\sin(\mathrm{th}_1(t)-\mathrm{th}_2(t)) \\ -1 & 0 & 0 & l_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

$$\begin{pmatrix} -\cos(\mathrm{th}_3(t)) & -\sin(\mathrm{th}_3(t)) & 0 & l_3\sin(\mathrm{th}_4(t)) \\ -\sin(\mathrm{th}_3(t)) & \cos(\mathrm{th}_3(t)) & 0 & l_3\cos(\mathrm{th}_4(t)) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matruz de Transformación global T3
ans =
```

$$\begin{pmatrix} \sin(\sigma_1)\sin(\mathrm{th}_3(t)) & -\sin(\sigma_1)\cos(\mathrm{th}_3(t)) & -\cos(\sigma_1) & -l_1\cos(\sigma_1)-l_3\sin(\sigma_1)\cos(\mathrm{th}_4(t)) \\ -\sin(\mathrm{th}_3(t))\cos(\sigma_1) & \cos(\mathrm{th}_3(t))\cos(\sigma_1) & -\sin(\sigma_1) & l_3\cos(\mathrm{th}_4(t))\cos(\sigma_1)-l_1\sin(\sigma_1) \\ \cos(\mathrm{th}_3(t)) & \sin(\mathrm{th}_3(t)) & 0 & l_0-l_3\sin(\mathrm{th}_4(t)) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = \mathrm{th}_1(t)-\mathrm{th}_2(t)$$

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

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

Matruz de Transformación global T4
ans =

$$\begin{pmatrix} \sigma_2 & -\sigma_1\,\sigma_4 & \sigma_1\,\sigma_3 & -l_1\,\sigma_2 - l_3\,\sigma_1\cos(\mathrm{th}_4(t)) \\ \sigma_1 & \sigma_2\,\sigma_4 & -\sigma_3\,\sigma_2 & l_3\cos(\mathrm{th}_4(t))\,\sigma_2 - l_1\,\sigma_1 \\ 0 & \sigma_3 & \sigma_4 & l_0 - l_3\sin(\mathrm{th}_4(t)) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = \sin(\mathrm{th}_1(t) - \mathrm{th}_2(t))$$

$$\sigma_2 = \cos(\mathrm{th}_1(t) - \mathrm{th}_2(t))$$

$$\sigma_3 = \sin(\mathrm{th}_3(t) - \mathrm{th}_4(t))$$

$$\sigma_4 = \cos(\mathrm{th}_3(t) - \mathrm{th}_4(t))$$

Matriz de Transformación local A5
ans =

$$\begin{pmatrix} \cos(\mathrm{th}_5(t)) & 0 & -\sin(\mathrm{th}_5(t)) & l_4\cos(\mathrm{th}_5(t)) \\ \sin(\mathrm{th}_5(t)) & 0 & \cos(\mathrm{th}_5(t)) & l_4\sin(\mathrm{th}_5(t)) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matruz de Transformación global T5
ans =

$$\begin{pmatrix} \cos(\mathrm{th}_5(t))\cos(\sigma_2) - \sin(\sigma_2)\sin(\mathrm{th}_5(t))\cos(\sigma_1) & -\sin(\sigma_2)\sin(\sigma_1) & -\sin(\mathrm{th}_5(t))\cos(\sigma_2) - \sin(\sigma_2)\cos(\mathrm{th}_5(t)) \\ \sin(\sigma_2)\cos(\mathrm{th}_5(t)) + \sin(\mathrm{th}_5(t))\cos(\sigma_2)\cos(\sigma_1) & \sin(\sigma_1)\cos(\sigma_2) & \cos(\mathrm{th}_5(t))\cos(\sigma_2)\cos(\sigma_1) - \sin(\sigma_2)\sin(\\ \sin(\sigma_1)\sin(\mathrm{th}_5(t)) & -\cos(\sigma_1) & \sin(\sigma_1)\cos(\mathrm{th}_5(t)) \\ 0 & 0 & 0 \end{pmatrix}$$

where

$$\sigma_1 = \mathrm{th}_3(t) - \mathrm{th}_4(t)$$

$$\sigma_2 = \mathrm{th}_1(t) - \mathrm{th}_2(t)$$

Matriz de Transformación local A6
ans =

$$\begin{pmatrix} \cos(\mathrm{th}_6(t)) & -\sin(\mathrm{th}_6(t)) & 0 & 0 \\ \sin(\mathrm{th}_6(t)) & \cos(\mathrm{th}_6(t)) & 0 & 0 \\ 0 & 0 & 1 & l_5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matruz de Transformación global T6
ans =
```

$$\left(\begin{matrix} \cos(\mathrm{th}_6(t))\,\sigma_1 - \sin(\sigma_8)\sin(\sigma_7)\sin(\mathrm{th}_6(t)) & -\sin(\mathrm{th}_6(t))\,\sigma_1 - \sin(\sigma_8)\sin(\sigma_7)\cos(\mathrm{th}_6(t)) \\ \cos(\mathrm{th}_6(t))\,\sigma_2 + \sin(\sigma_7)\sin(\mathrm{th}_6(t))\cos(\sigma_8) & \sin(\sigma_7)\cos(\mathrm{th}_6(t))\cos(\sigma_8) - \sin(\mathrm{th}_6(t))\,\sigma_2 \\ \sin(\sigma_7)\cos(\mathrm{th}_6(t))\sin(\mathrm{th}_5(t)) - \sin(\mathrm{th}_6(t))\cos(\sigma_7) & -\cos(\mathrm{th}_6(t))\cos(\sigma_7) - \sin(\sigma_7)\sin(\mathrm{th}_5(t))\sin(\mathrm{th}_6(t)) \quad \sin \\ 0 & 0 \end{matrix} \right.$$

where

$$\sigma_1 = \cos(\mathrm{th}_5(t))\cos(\sigma_8) - \sin(\sigma_8)\sin(\mathrm{th}_5(t))\cos(\sigma_7)$$

$$\sigma_2 = \sin(\sigma_8)\cos(\mathrm{th}_5(t)) + \sin(\mathrm{th}_5(t))\cos(\sigma_8)\cos(\sigma_7)$$

$$\sigma_3 = \cos(\mathrm{th}_5(t))\cos(\sigma_8)\cos(\sigma_7)$$

$$\sigma_4 = \sin(\sigma_8)\cos(\mathrm{th}_5(t))\cos(\sigma_7)$$

$$\sigma_5 = \sin(\mathrm{th}_5(t))\cos(\sigma_8)$$

$$\sigma_6 = \sin(\sigma_8)\sin(\mathrm{th}_5(t))$$

$$\sigma_7 = \mathrm{th}_3(t) - \mathrm{th}_4(t)$$

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

En las matrices locales podemos observar que efectivamente en la ultima columna se respeta el vector de posción, mientas que en global concentra las pocisiones de las articulaciones.

De igual manera las matrices locales contienen cada matriz de rotación 3x3 de cada rotación mientras la global contiene la general del sistema

```
% 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

```
disp('Jacobiano lineal obtenido de forma analítica'); Jv_a = simplify(Jv_a);
```

```
Jacobiano lineal obtenido de forma analítica
```

```
disp('Jacobiano angular obtenido de forma ananlítica'); Jw_a = simplify(Jw_a);
```

```
Jacobiano angular obtenido de forma ananlítica
```

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

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

$$\left(\begin{array}{c} \sigma_5\,\sigma_7 - \sigma_6\,\sigma_7 - \sigma_2\,\left(l_5\cos(\mathrm{th}_5(t))\cos(\sigma_{10}) + l_4\sin(\mathrm{th}_5(t))\cos(\sigma_{10}) + l_4\sin(\sigma_{10})\,\mathrm{co} \right.\\ \sigma_5\,\sigma_8 - \sigma_6\,\sigma_8 - \sigma_2\,\left(l_5\sin(\sigma_{10})\cos(\mathrm{th}_5(t)) + l_4\sin(\sigma_{10})\sin(\mathrm{th}_5(t)) + l_5\sin(\mathrm{th}_5(t))\,\mathrm{c} \right.\\ \sigma_4\,\left(l_3\cos(\mathrm{th}_4(t)) + l_5\cos(\mathrm{th}_3(t))\cos(\mathrm{th}_4(t))\cos(\mathrm{th}_5(t)) + l_4\cos(\mathrm{th}_3(t))\cos(\mathrm{th}_4(t))\sin(\mathrm{th}_5(t)) + l_5\cos(\mathrm{th}_5(t)) \right. \end{array}\right.$$

where

$$\sigma_1 = l_5\cos(\mathrm{th}_5(t)) + l_4\sin(\mathrm{th}_5(t))$$

$$\sigma_2 = \overline{\frac{\partial}{\partial t}\,\mathrm{th}_5(t)}$$

$$\sigma_3 = \overline{\frac{\partial}{\partial t}\,\mathrm{th}_4(t)}$$

$$\sigma_4 = \overline{\frac{\partial}{\partial t}\,\mathrm{th}_3(t)}$$

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

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

$$\sigma_7 = l_3\cos(\mathrm{th}_4(t))\cos(\sigma_{10}) - l_1\sin(\sigma_{10}) - l_5\,\left(\sin(\sigma_{10})\sin(\mathrm{th}_5(t)) - \cos(\mathrm{th}_5(t))\cos(\sigma_{10})\cos(\sigma_{11})\right) + l_4\sin(\sigma_{10}$$

$$\sigma_8 = l_5\,\left(\sin(\mathrm{th}_5(t))\cos(\sigma_{10}) + \sin(\sigma_{10})\cos(\mathrm{th}_5(t))\cos(\sigma_{11})\right) + l_1\cos(\sigma_{10}) - l_4\cos(\mathrm{th}_5(t))\cos(\sigma_{10}) + l_3\sin(\sigma_{10}$$

$$\sigma_9 = l_5\sin(\sigma_{11})\cos(\mathrm{th}_5(t)) - l_3\sin(\mathrm{th}_4(t)) + l_4\sin(\sigma_{11})\sin(\mathrm{th}_5(t))$$

$$\sigma_{10} = \mathrm{th}_1(t) - \mathrm{th}_2(t)$$

$$\sigma_{11} = \mathrm{th}_3(t) - \mathrm{th}_4(t)$$

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

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

$$\begin{pmatrix} \sigma_6 \cos(\sigma_4) - \sigma_1 \; (\sin(\mathrm{th}_5(t)) \cos(\sigma_4) + \sin(\sigma_4) \cos(\mathrm{th}_5(t)) \cos(\sigma_3)) - \sigma_5 \cos(\sigma_4) + \sin(\sigma_4) \sin(\sigma_3)\, \sigma_2 \\ \sin(\sigma_4)\, \sigma_6 - \sigma_1 \; (\sin(\sigma_4) \sin(\mathrm{th}_5(t)) - \cos(\mathrm{th}_5(t)) \cos(\sigma_4) \cos(\sigma_3)) - \sin(\sigma_4)\, \sigma_5 - \sin(\sigma_3)\, \sigma_2 \cos(\sigma_4) \\ \overline{\frac{\partial}{\partial t}\, \mathrm{th}_1(t)} - \overline{\frac{\partial}{\partial t}\, \mathrm{th}_2(t)} + \sigma_2 \cos(\sigma_3) + \sin(\sigma_3)\, \sigma_1 \cos(\mathrm{th}_5(t)) \end{pmatrix}$$

where

$$\sigma_1 = \overline{\frac{\partial}{\partial t}\, \mathrm{th}_6(t)}$$

$$\sigma_2 = \overline{\frac{\partial}{\partial t}\, \mathrm{th}_5(t)}$$

$$\sigma_3 = \mathrm{th}_3(t) - \mathrm{th}_4(t)$$

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

$$\sigma_5 = \overline{\frac{\partial}{\partial t}\, \mathrm{th}_4(t)}$$

$$\sigma_6 = \overline{\frac{\partial}{\partial t}\, \mathrm{th}_3(t)}$$

Funciones de rotacion

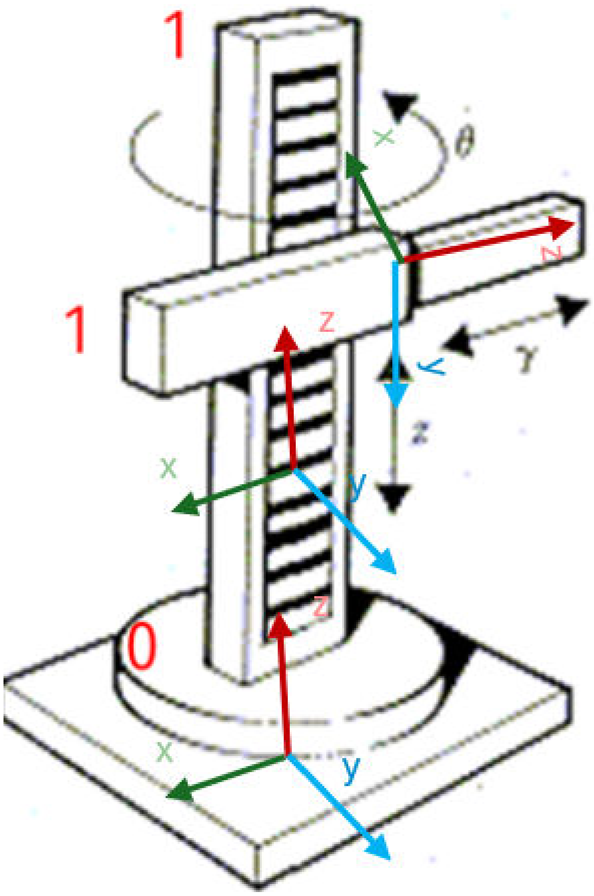
```
function r_x = rotX(th)
    r_x = [1 0 0; 0 cosd(th) -sind(th); 0 sind(th) cosd(th)];
end

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
```

Evaluación

Ana Itzel Hernández García A01737526

Obtener la **cinemática directa** del robot:

- **Obtener** el vector de velocidades lineales y el vector de velocidades angulares para los siguientes robots manipuladores



Limpieza de pantalla

```
clear all
close all
clc
```

Declaración de variables simbolicas (**3 grados de libertad**)

Donde hay una rotacional y 2 prismaticas

```
syms th(t) l1(t) l2(t) l3(t) t

% configuración del robot, 0 para la rotacional y 1 para las juntas prismáticas
RP = [0 1 1];

% Creación del vector de coordenadas articulares
```

```
Q = [th l2 l3];

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

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

Articulación 1

Posición de la articulación 1 respecto a 0

Debido a que cuenta con una traslación en el eje z se toma en cuenta la longitud que se encuentra en ese eje, es decir l1

```
P(:,:,1) = [
            0;
            0;
            l1
           ];
```

Matriz de rotación de la junta 1 respecto a la 0

A la matriz de rotación Z se mantiene debido a que no hay ninguna rotación en sus ejes

```
R(:,:,1) = [
             cos(th)   -sin(th)     0;
             sin(th)    cos(th)     0;
             0           0            1
             ];
```

Articulacion 2

Posición de la articulación 2 respecto a 1

Debido a que cuenta con una traslación en el eje z se toma en cuenta la longitud que se encuentra en ese eje, es decir l2

```
P(:,:,2) = [
            0;
            0;
            l2
           ];
```

Matriz de rotación de la junta 2 respecto a 1

Para esta matriz se realiza una doble rotación: en Y(+90)

```
R(:,:,2) = rotY(90);
R(:,:,2)
```

ans =

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

Articulacion 3

Posición de la articulación 3 respecto a la 2

Debido a que cuenta con una traslación en el eje z se toma en cuenta la longitud que se encuentra en ese eje, es decir l3

```
P(:,:,3) = [
             0;
             0;
             l3
            ];
```

Matriz de rotación de la junta 3 respecto a la 2 (0°)

Como no hay trasnformación se evalua la rotación Z en 0°

```
R(:,:,3) = rotZ(0);
R(:,:,3)
```

ans =

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matrices

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

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

% Inicialización de las amtrices de transformación 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)); 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 =
```

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

```
Matruz de Transformación global T1
ans =
```

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

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

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & l_2(t) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matruz de Transformación global T2
ans =
```

$$\begin{pmatrix} 0 & -\sin(\mathrm{th}(t)) & \cos(\mathrm{th}(t)) & 0 \\ 0 & \cos(\mathrm{th}(t)) & \sin(\mathrm{th}(t)) & 0 \\ -1 & 0 & 0 & l_1(t)+l_2(t) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_3(t) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
Matruz de Transformación global T3
ans =
```

$$\begin{pmatrix} 0 & -\sin(\mathrm{th}(t)) & \cos(\mathrm{th}(t)) & \cos(\mathrm{th}(t))\, l_3(t) \\ 0 & \cos(\mathrm{th}(t)) & \sin(\mathrm{th}(t)) & \sin(\mathrm{th}(t))\, l_3(t) \\ -1 & 0 & 0 & l_1(t)+l_2(t) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

En las matrices locales podemos observar que efectivamente en la ultima columna se respeta el vector de posción, mientas que en global concentra las pocisiones de las articulaciones.

De igual manera las matrices locales contienen cada matriz de rotación 3x3 de cada rotación mientras la global contiene la general del sistema

Jacobiano lineal y angular de forma analítica

```
% Inicialización de jacobianos analíticos
Jv_a(:,GDL) = PO(:,:,GDL); % Lineal
Jw_a(:,GDL) = PO(:,:,GDL); % Angular

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

```
disp('Jacobiano lineal obtenido de forma analítica'); Jv_a = simplify(Jv_a)
```

```
Jacobiano lineal obtenido de forma analítica
Jv_a =
```

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

```
disp('Jacobiano angular obtenido de forma ananlítica'); Jw_a = simplify(Jw_a)
```

```
Jacobiano angular obtenido de forma ananlítica
Jw_a =
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Aquí, en el vecto de velocidad líneal podemos observar el movimiento de la articulación angular y la prismatica 3 interactuan entres si, mientras que la articulación primatica 2 se mueve en el eje z sin afectar a las otras

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

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

$$\begin{pmatrix} \overline{\frac{\partial}{\partial t}\, l_3(t)}\, \cos(\mathrm{th}(t)) - \overline{\frac{\partial}{\partial t}\, \mathrm{th}(t)}\, \sin(\mathrm{th}(t))\, l_3(t) \\ \overline{\frac{\partial}{\partial t}\, l_3(t)}\, \sin(\mathrm{th}(t)) + \overline{\frac{\partial}{\partial t}\, \mathrm{th}(t)}\, \cos(\mathrm{th}(t))\, l_3(t) \\ \overline{\frac{\partial}{\partial t}\, l_2(t)} \end{pmatrix}$$

Como tenemos movimientos líneales, solo contamos con 1 ángulo y por lo tanto solo se mueve la articulación 1

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

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

$$\begin{pmatrix} 0 \\ 0 \\ \dfrac{\partial}{\partial t}\,\mathrm{th}(t) \end{pmatrix}$$

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
```