Presentación Final | Cinemática diferencial de ejercicio 2

Oscar Ortiz Torres A01769292

```
clc
clear all
close all

% Declaración de las variables simbólicas para los ángulos y medidas de los
eslabones del robot
syms th1(t) th2(t) l1 l2 t

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

% Creación del vector de coordenadas articulares
Q = [th1 th2];

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

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

P = sym(zeros(3,1,GDL));
```

Articulación 1

```
% Posición de la articulación 0 a 1
P(:,:,1) = [
             l1;
             0;
             0;
           ];

% Matriz de rotación de la junta 0 a 1
R(:,:,1) = [
             cos(th1)   -sin(th1)     0;
             sin(th1)    cos(th1)     0;
             0           0            1
             ]*rotz(180);
```

Articulación 2

```
% Posición de la articulación 1 a 2
P(:,:,2) = [
             0;
             0;
             l2
```

```
          ];

% Matriz de rotación de la junta 1 a 2
R(:,:,2) = [
             cos(th2)   -sin(th2)     0;
             sin(th2)    cos(th2)     0;
             0           0            1
             ]*rotY(90)*rotX(150);
```

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));

    % 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 =
```

$$\begin{pmatrix} \sigma_1 & \sigma_2 & 0 & l_1 \\ -\sigma_2 & \sigma_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = -\frac{\sqrt{81129638414606672444637910133645}\ \cos\left(\mathrm{th}_1(t) + \mathrm{atan}\left(\frac{7216141423515539}{5390449088010182}\right)\right)}{9007199254740992}$$

$$\sigma_2 = \frac{\sqrt{81129638414606672444637910133645}\ \cos\left(\mathrm{th}_1(t) - \mathrm{atan}\left(\frac{5390449088010182}{7216141423515539}\right)\right)}{9007199254740992}$$

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

$$\begin{pmatrix} 0 & \cos\left(\frac{\pi}{3} - \mathrm{th}_2(t)\right) & -\cos\left(\frac{\pi}{6} + \mathrm{th}_2(t)\right) & 0 \\ 0 & -\cos\left(\frac{\pi}{6} + \mathrm{th}_2(t)\right) & -\cos\left(\frac{\pi}{3} - \mathrm{th}_2(t)\right) & 0 \\ -1 & 0 & 0 & l_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

```
Matriz de Transformación global
ans =
```

$$\begin{pmatrix} 0 & -\sigma_2 & \sigma_1 & l_1 \\ 0 & \sigma_1 & \sigma_2 & 0 \\ -1 & 0 & 0 & l_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where

$$\sigma_1 = -\frac{\sqrt{81129638414606672444637910133645}\ \sin\left(\operatorname{atan}\left(\frac{81129638414606672444637910133645\ \sqrt{3}}{127161149762100805937001187708439}\right.\right.}{9007199254740992}$$

$$\sigma_2 = \frac{\sqrt{81129638414606672444637910133645}\ \cos\left(\operatorname{atan}\left(\frac{81129638414606672444637910133645\ \sqrt{3}}{127161149762100805937001187708439}\right.\right.}{9007199254740992}$$

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

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

$$\begin{pmatrix} 0 \\ \overline{l_1 \frac{\partial}{\partial t} \mathrm{th}_1(t)} \\ 0 \end{pmatrix}$$

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

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

$$\begin{pmatrix} 0 \\ 0 \\ \overline{\frac{\partial}{\partial t} \mathrm{th}_1(t)} + \overline{\frac{\partial}{\partial t} \mathrm{th}_2(t)} \end{pmatrix}$$

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
```