Actividad 6 | Robot pendulo 1 GDL

Oscar Ortiz Torres A01769292

```
% Limpieza de pantalla
clear all
close all
clc

tic
% Declaración de variables sinbólicas
syms th1(t) t   % Ángulos de cada articulación
syms m1 Ixx1 Iyy1 Izz1  % Masas y matrices de Inercia
syms l1 lc1     % l = longitud de eslabones y lc = distancia al centro de masa de
cada eslabón
syms pi g a cero

% Vector de coordenadas articulares
Q = [th1];

% Vector de velocidades articulares
Qp = diff(Q, t);

% Vector de aceleraciones articulares
Qpp = diff(Qp, t);

% Configuración del robot
RP = [0];

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

Articulación 1

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

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

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

    % Globales
    try
       T(:,:,i)= T(:,:,i-1)*A(:,:,i);
    catch
       T(:,:,i)= A(:,:,i);
    end
    % disp(strcat('Matriz de Transformación global T', i_str));
    T(:,:,i)= simplify(T(:,:,i));
    % pretty(T(:,:,i))

    RO(:,:,i)= T(1:3,1:3,i);
    PO(:,:,i)= T(1:3,4,i);
    % pretty(RO(:,:,i));
    % pretty(PO(:,:,i));
end
```

Cálculo del jacobiano lineal de forma analítica

```
Jv_a(:,GDL)=PO(:,:,GDL);
Jw_a(:,GDL)=PO(:,:,GDL);

for k= 1:GDL
    if RP(k)==0
```

```
        % Para las juntas de revolución
         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 juntas prismáticas
        try
            Jv_a(:,k)= RO(:,3,k-1);
        catch
            Jv_a(:,k)=[0,0,1];
        end
            Jw_a(:,k)=[0,0,0];
     end
 end
```

SubMatrices de Jacobianos

```
Jv_a= simplify (Jv_a);
Jw_a= simplify (Jw_a);

%disp('Jacobiano lineal obtenido de forma analítica');
%pretty (Jv_a);
%disp('Jacobiano ángular obtenido de forma analítica');
%pretty (Jw_a);

%Matriz de Jacobiano Completa
%disp('Matriz de Jacobiano');
Jac= [Jv_a;
      Jw_a];
Jacobiano= simplify(Jac); %pretty(Jacobiano);
```

Vectores de Velocidades Lineales y Angulares

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

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

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

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

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

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

Energía Cinética

```
% Distancia del origen del eslabón a su centro de masa
% Vectores de posición respecto al centro de masa
P01=subs(P(:,:,1)/2, l1, lc1) % La función subs sustituye l1 por lc1 en la
expresión P(:,:,1)/2
```

```
P01 =
```

$$\begin{pmatrix} \frac{\mathrm{lc}_1 \cos(\mathrm{th}_1(t))}{2} \\ \frac{\mathrm{lc}_1 \sin(\mathrm{th}_1(t))}{2} \\ 0 \end{pmatrix}$$

```
% Matrices de inercia para cada eslabón
I1=[Ixx1 0 0;
    0 Iyy1 0;
    0 0 Izz1];
```

Función de energía cinética

```
% Velocidades lineales en cada eje
V = V(t);
Vx = V(1,1);
Vy = V(2,1);
Vz = V(3,1);

% Velocidad angular en cada ángulo de Euler
W = W(t);
W_pitch = W(1,1);
W_roll = W(2,1);
W_yaw = W(3,1);
```

Calculamos las velocidades para cada eslabón

```
% Eslabón 1

% Ya lo calculamos previamente al multiplicar la matriz jacobiana por Qp
```

Calculamos la energía cinética para cada uno de los eslabones

```
% Eslabón 1
V1_Total = V + cross(W, P01); % Se suma la velocidad lineal producida por la velocidad angular producida en el punto P01

K1 = (1/2*m1*(V1_Total))'*(V1_Total) + (1/2*W)'*(I1*W);
disp('Energía Cinética en el Eslabón 1'); K1= simplify (K1) %pretty (K1);
```

```
Energía Cinética en el Eslabón 1
K1 =
```

$$\frac{\mathrm{Izz}_1 \left|\frac{\partial}{\partial t}\ \mathrm{th}_1(t)\right|^2}{2} + \frac{\left|\frac{\partial}{\partial t}\ \mathrm{th}_1(t)\right|^2 \cos\left(\overline{\mathrm{th}_1(t)} - \mathrm{th}_1(t)\right) \overline{m_1}\ \left(2\,\mathrm{lc}_1\,|l_1|^2 + l_1\,|\mathrm{lc}_1|^2\right)\ (2\,l_1 + \mathrm{lc}_1)}{8\,l_1\,\mathrm{lc}_1}$$

```
K_Total= simplify (K1);
```

Energía Potencial

```
% Obtenemos las alturas respecto a la gravedad
h1= P01(2); %Tomo la altura paralela al eje y

U1=m1*g*h1
```

```
U1 =
```

$$\frac{g\,\mathrm{lc}_1\,m_1 \sin(\mathrm{th}_1(t))}{2}$$

```
% Calculamos la energía potencial total
U_Total= U1;

% Obtenemos el Lagrangiano
Lagrangiano= simplify (K_Total-U_Total) %pretty (Lagrangiano);
```

```
Lagrangiano =
```

$$\frac{\mathrm{Izz}_1 \left|\frac{\partial}{\partial t}\ \mathrm{th}_1(t)\right|^2}{2} - \frac{g\,\mathrm{lc}_1\,m_1 \sin(\mathrm{th}_1(t))}{2} + \frac{\left|\frac{\partial}{\partial t}\ \mathrm{th}_1(t)\right|^2 \cos\left(\overline{\mathrm{th}_1(t)} - \mathrm{th}_1(t)\right) \overline{m_1}\ \left(2\,\mathrm{lc}_1\,|l_1|^2 + l_1\,|\mathrm{lc}_1|^2\right)\ (2\,l_1 + \mathrm{lc}_1)}{8\,l_1\,\mathrm{lc}_1}$$

```
% Modelo de Energía
H = simplify (K_Total+U_Total) %pretty (H)
```

```
H =
```

$$\frac{\mathrm{Izz}_1 \left|\frac{\partial}{\partial t}\ \mathrm{th}_1(t)\right|^2}{2} + \frac{g\,\mathrm{lc}_1\,m_1 \sin(\mathrm{th}_1(t))}{2} + \frac{\left|\frac{\partial}{\partial t}\ \mathrm{th}_1(t)\right|^2 \cos\left(\overline{\mathrm{th}_1(t)} - \mathrm{th}_1(t)\right) \overline{m_1}\ \left(2\,\mathrm{lc}_1\,|l_1|^2 + l_1\,|\mathrm{lc}_1|^2\right)\ (2\,l_1 + \mathrm{lc}_1)}{8\,l_1\,\mathrm{lc}_1}$$

```
toc
```

```
Elapsed time is 2.275973 seconds.
```