Actividad 6: Modelado de Energía Cinética

Ana Itzel Hernández Garía A01737526

Obtener el modelo de la energía cinética total para cada una de las siguientes configuraciones de robots manipuladores

Figura 4.10 Péndulo robot.

Robot Péndulo (1gdl)

Péndulo

Limpieza de pantalla

```
clear all
close all
clc
```

Declaración de variables simbólicas

```
syms th1(t)  t  %Angulos 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);
```

Creamos el vector de aceleraciones articulares

```
Qpp= diff(Qp, t);
```

Configuración del robot (0 para junta rotacional, 1 para junta prismática)

```
RP=[0];
```

Número de grado 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);
```

Matrices

Inicializamos las matrices de transformación Homogénea locales

```
A(:,:,GDL)=simplify([R(:,:,GDL) P(:,:,GDL); Vector_Zeros 1]);
```

Inicializamos las matrices de transformación Homogénea globales

```
T(:,:,GDL)=simplify([R(:,:,GDL) P(:,:,GDL); Vector_Zeros 1]);
```

Inicializamos las posiciones vistas desde el marco de referencia inercial

```
PO(:,:,GDL)= P(:,:,GDL);
```

Inicializamos las matrices de rotación vistas desde el marco de referencia inercial

```
RO(:,:,GDL)= R(:,:,GDL);


for i = 1:GDL
```

```
    i_str= num2str(i);
    A(:,:,i)=simplify([R(:,:,i) P(:,:,i); Vector_Zeros 1]);

   %Globales
    try
       T(:,:,i)= T(:,:,i-1)*A(:,:,i);
    catch
       T(:,:,i)= A(:,:,i);
    end
    T(:,:,i)= simplify(T(:,:,i));

    RO(:,:,i)= T(1:3,1:3,i);
    PO(:,:,i)= T(1:3,4,i);

end
```

Calculamos el jacobiano y angular lineal de forma analítica

Inicializamos jacobianos analíticos

```
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

Jacobiano lineal obtenido de forma analítica

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

Jacobiano ángular obtenido de forma analítica

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

Matriz de Jacobiano Completa

```
Jac= [Jv_a;
      Jw_a];
Jacobiano= simplify(Jac)
```

```
Jacobiano =
```

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

Vectores de Velocidades Lineales y Angulares

Velocidad lineal

```
disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 3');
V=simplify (Jv_a*Qp)
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 3
V(t) =
```

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

Velocidad angular

```
disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 3');
W=simplify (Jw_a*Qp)
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 3
W(t) =
```

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

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
```

```
P01 =
```

$$\begin{pmatrix} \frac{lc_1 \cos(th_1(t))}{2} \\ \frac{lc_1 \sin(th_1(t))}{2} \\ 0 \end{pmatrix}$$

```
                                %la expresión P(:,:,1)/2
```

Matrices de inercia para cada eslabón

Eslabón 1

```
I1=[Ixx1 0 0;
    0 Iyy1 0;
    0 0 Izz1];
```

Función de energía cinética

Extraemos las velocidades lineales del efector final en cada eje

```
V=V(t);
Vx= V(1,1);
Vy= V(2,1);
Vz= V(3,1);
```

Extraemos las velocidades angular del efector final en cada ángulo de Euler

```
W=W(t);
W_pitch= W(1,1);
W_roll= W(2,1);
W_yaw= W(3,1);
```

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');
```

```
Energía Cinética en el Eslabón 1
```

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

```
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)
```

K_Total =

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

Energía potencial

Obtenemos las alturas respecto a la gravedad

```
h1= P01(2); %Tomo la altura paralela al eje y
```

Energía potencial de cada eslabón

```
U_Total=m1*g*h1
```

U_Total =

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

Obtenemos el Lagrangiano

```
Lagrangiano= simplify (K_Total-U_Total);
```

Modelo de Energía

```
H= simplify (K_Total+U_Total)
```

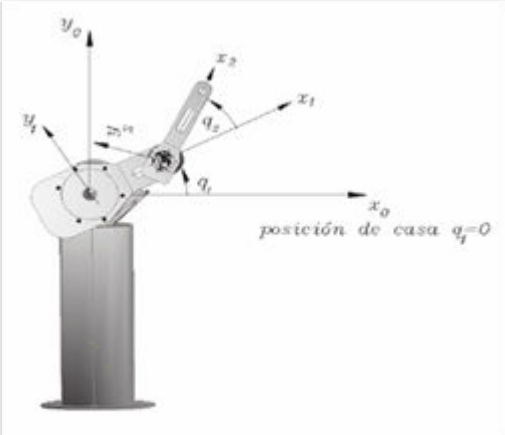
H =

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

Actividad 6: Modelado de Energía Cinética

Ana Itzel Hernández Garía A01737526

Obtener el modelo de la energía cinética total para cada una de las siguientes configuraciones de robots manipuladores



Robot Rotacional (2gdl)

Rotacional

Limpieza de pantalla

```
clear all
close all
clc
```

Declaración de variables simbólicas

```
syms th1(t) th2(t)  t   %Angulos de cada articulación
syms th1p(t) th2p(t)    %Velocidades de cada articulación
syms th1pp(t) th2pp(t)  %Aceleraciones de cada articulación
syms m1 m2 Ixx1 Iyy1 Izz1 Ixx2 Iyy2 Izz2   %Masas y matrices de Inercia
syms l1 l2 lc1 lc2 %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; th2];
```

Vector de velocidades articulares

```
Qp= [th1p; th2p];
Qp=Qp(t);
```

Creamos el vector de aceleraciones articulares

```
Qpp= [th1pp; th2pp];
```

Configuración del robot (0 para junta rotacional, 1 para junta prismática)

```
RP=[0 0];
```

Número de grado 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];
```

Articulación 2

Posición de la articulación 2 respecto a 1

```
P(:,:,2)= [l2*cos(th2); l2*sin(th2);0];
```

Matriz de rotación de la junta 1 respecto a 0

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

Vector de ceros

```
Vector_Zeros= zeros(1, 3);
```

Matrices

Inicializamos las matrices de transformación Homogénea locales

```
A(:,:,GDL)=simplify([R(:,:,GDL) P(:,:,GDL); Vector_Zeros 1]);
```

Inicializamos las matrices de transformación Homogénea globales

```
T(:,:,GDL)=simplify([R(:,:,GDL) P(:,:,GDL); Vector_Zeros 1]);
```

Inicializamos las posiciones vistas desde el marco de referencia inercial

```
PO(:,:,GDL)= P(:,:,GDL);
```

Inicializamos las matrices de rotación vistas desde el marco de referencia inercial

```
RO(:,:,GDL)= R(:,:,GDL);


for i = 1:GDL
    i_str= num2str(i);
    A(:,:,i)=simplify([R(:,:,i) P(:,:,i); Vector_Zeros 1]);

   %Globales
    try
       T(:,:,i)= T(:,:,i-1)*A(:,:,i);
    catch
       T(:,:,i)= A(:,:,i);
    end
    T(:,:,i)= simplify(T(:,:,i));

    RO(:,:,i)= T(1:3,1:3,i);
    PO(:,:,i)= T(1:3,4,i);

end
```

Velocidades para el eslabón 2

Calculamos el jacobiano y angular lineal de forma analítica

Inicializamos jacobianos analíticos

```
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

Jacobiano lineal obtenido de forma analítica

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

Jacobiano ángular obtenido de forma analítica

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

Matriz de Jacobiano Completa

```
Jac= [Jv_a;
      Jw_a];
Jacobiano= simplify(Jac)
```

```
Jacobiano =
```

$$\begin{pmatrix} -l_1 \sin(\mathrm{th}_1(t)) - l_2 \sin(\mathrm{th}_1(t) + \mathrm{th}_2(t)) & -l_2 \sin(\mathrm{th}_1(t) + \mathrm{th}_2(t)) \\ l_1 \cos(\mathrm{th}_1(t)) + l_2 \cos(\mathrm{th}_1(t) + \mathrm{th}_2(t)) & l_2 \cos(\mathrm{th}_1(t) + \mathrm{th}_2(t)) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$$

Vectores de Velocidades Lineales y Angulares

Velocidad lineal

```
disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 2');
V2=simplify (Jv_a*Qp); V2
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 2
V2 =
```

$$\begin{pmatrix} -\mathrm{th1p}(t)\ (l_1 \sin(\mathrm{th}_1(t)) + l_2\,\sigma_1) - l_2\,\sigma_1\,\mathrm{th2p}(t) \\ \mathrm{th1p}(t)\ (l_1 \cos(\mathrm{th}_1(t)) + l_2\,\sigma_2) + l_2\,\sigma_2\,\mathrm{th2p}(t) \\ 0 \end{pmatrix}$$

where

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

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

Velocidad angular

```
disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 2');
W2=simplify (Jw_a*Qp); W2
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 2
W2 =
```

$$\begin{pmatrix} 0 \\ 0 \\ \mathrm{th1p}(t) + \mathrm{th2p}(t) \end{pmatrix}$$

Velocidades para eslabón 1

Calculamos el jacobiano lineal y angular de forma analítica

Inicializamos jacobianos analíticos

```
Jv_a1(:,GDL-1)=PO(:,:,GDL-1);
Jw_a1(:,GDL-1)=PO(:,:,GDL-1);

for k= 1:GDL-1
    if RP(k)==0
       %Para las juntas de revolución
        try
            Jv_a1(:,k)= cross(RO(:,3,k-1), PO(:,:,GDL-1)-PO(:,:,k-1));
            Jw_a1(:,k)= RO(:,3,k-1);
        catch
            Jv_a1(:,k)= cross([0,0,1], PO(:,:,GDL-1));%Matriz de rotación de 0 con
respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a1(:,k)=[0,0,1];%Si no hay matriz de rotación previa se obtiene la
Matriz identidad
         end
     else
%         %Para las juntas prismáticas
        try
            Jv_a1(:,k)= RO(:,3,k-1);
        catch
            Jv_a1(:,k)=[0,0,1];
        end
            Jw_a1(:,k)=[0,0,0];
     end
 end
```

SubMatrices de Jacobianos

Jacobiano lineal obtenido de forma analítica

```
Jv_a1= simplify (Jv_a1);
```

Jacobiano ángular obtenido de forma analítica

```
Jw_a1= simplify (Jw_a1);
```

Matriz de Jacobiano Completa

```
Jac1= [Jv_a1;
      Jw_a1];
Jacobiano1= simplify(Jac1);
```

Vectores de velocidades lineales y angulares para el eslabón 2

Velocidad lineal

```
disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 1');
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 1
```

```
V1=simplify (Jv_a1*Qp(1)); V1
```

```
V1 =
```

$$\begin{pmatrix} -l_1 \sin(\mathrm{th}_1(t))\,\mathrm{th1p}(t) \\ l_1 \cos(\mathrm{th}_1(t))\,\mathrm{th1p}(t) \\ 0 \end{pmatrix}$$

Velocidad angular

```
disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 1');
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 1
```

```
W1=simplify (Jw_a1*Qp(1)); W1
```

```
W1 =
```

$$\begin{pmatrix} 0 \\ 0 \\ \mathrm{th1p}(t) \end{pmatrix}$$

Energía Cinética

Distancia del origen del eslabón a su centro de masa

Vectores de posición respecto al centro de masa

```
P01=subs(P(:,:,1), l1, lc1);%La función subs sustituye l1 por lc1 en
P12=subs(P(:,:,2), l2, lc2); %la expresión P(:,:,1)/2
```

Matrices de inercia para cada eslabón

Eslabón 1

```
I1=[Ixx1 0 0;
    0 Iyy1 0;
    0 0 Izz1];
```

Eslabón 2

```
I2=[Ixx2 0 0;
    0 Iyy2 0;
    0 0 Izz2];
```

Función de energía cinética

Extraemos las velocidades lineales del efector final en cada eje

```
Vx= V2(1,1);
Vy= V2(2,1);
Vz= V2(3,1);
```

Extraemos las velocidades angular del efector final en cada ángulo de Euler

```
W_pitch= W2(1,1);
W_roll= W2(2,1);
W_yaw= W2(3,1);
```

Energía cinética para cada uno de los eslabones

Eslabón 1

```
V1_Total= V1+cross(W1,P01);
K1= (1/2*m1*(V1_Total))'*((V1_Total)) + (1/2*W1)'*(I1*W1);
disp('Energía Cinética en el Eslabón 1');
```

```
Energía Cinética en el Eslabón 1
```

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

Eslabón 2

```
V2_Total= V2+cross(W2,P12);
K2= (1/2*m2*(V2_Total))'*((V2_Total)) + (1/2*W2)'*(I2*W2);
disp('Energía Cinética en el Eslabón 2');
```

```
Energía Cinética en el Eslabón 2
```

```
K2= simplify (K2);
K_Total= simplify (K1+K2);
disp('Energía Cinética Total');
```

```
Energía Cinética Total
```

```
K_Total
```

```
K_Total =
```

$$\frac{\mathrm{Izz}_1\,\sigma_5}{2}+\frac{\overline{m_2}\;(\mathrm{th1p}(t)\;(l_1\sin(\mathrm{th}_1(t))+l_2\sin(\sigma_4))+l_2\sin(\sigma_4)\,\mathrm{th2p}(t)+\mathrm{lc}_2\sin(\mathrm{th}_2(t))\,\sigma_1)\;\left(\overline{\mathrm{th1p}(t)}\;\left(\sin(\sigma_3)\,\overline{l_2}+\right.\right.}{2}$$

where

$$\sigma_1=\mathrm{th1p}(t)+\mathrm{th2p}(t)$$

$$\sigma_2=\overline{\mathrm{th1p}(t)}+\overline{\mathrm{th2p}(t)}$$

$$\sigma_3=\overline{\mathrm{th}_1(t)}+\overline{\mathrm{th}_2(t)}$$

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

$$\sigma_5=|\mathrm{th1p}(t)|^2$$

Energía potencial

Obtenemos las alturas respecto a la gravedad

```
h1= P01(2); %Tomo la altura paralela al eje y
h2= P12(2); %Tomo la altura paralela al eje y
```

Energía potencial de cada eslabón

```
U1=m1*g*h1
```

U1 = $g\,\mathrm{lc}_1\,m_1\sin(\mathrm{th}_1(t))$

```
U2=m2*g*h2
```

U2 = $g\,\mathrm{lc}_2\,m_2\sin(\mathrm{th}_2(t))$

Calculamos la energía potencial total

```
U_Total= U1 + U2;
```

Obtenemos el Lagrangiano

```
Lagrangiano= simplify (K_Total-U_Total);
```

Modelo de Energía

```
H= simplify (K_Total+U_Total)
```

H =

$$\frac{\mathrm{Izz}_1\,\sigma_5}{2}+\frac{\overline{m_2}\;(\mathrm{th1p}(t)\;(l_1\sin(\mathrm{th}_1(t))+l_2\sin(\sigma_4))+l_2\sin(\sigma_4)\,\mathrm{th2p}(t)+\mathrm{lc}_2\sin(\mathrm{th}_2(t))\,\sigma_1)\;\left(\overline{\mathrm{th1p}(t)}\;\left(\sin(\sigma_3)\,\overline{l_2}+\right.\right.}{2}$$

where

$$\sigma_1=\mathrm{th1p}(t)+\mathrm{th2p}(t)$$

$$\sigma_2=\overline{\mathrm{th1p}(t)}+\overline{\mathrm{th2p}(t)}$$

$$\sigma_3=\overline{\mathrm{th}_1(t)}+\overline{\mathrm{th}_2(t)}$$

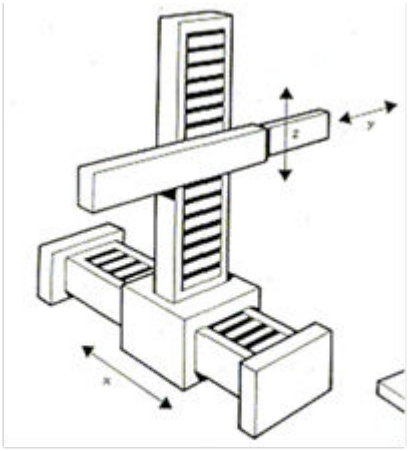
$$\sigma_4=\mathrm{th}_1(t)+\mathrm{th}_2(t)$$

$$\sigma_5=|\mathrm{th1p}(t)|^2$$

Actividad 6: Modelado de Energía Cinética

Ana Itzel Hernández Garía A01737526

Obtener el modelo de la energía cinética total para cada una de las siguientes configuraciones de robots manipuladores



Cartesiano

Limpieza de pantalla

```
clear all
close all
clc
```

Declaración de variables simbólicas

```
syms l1(t) l2(t) l3(t) t   %Angulos de cada articulación
syms l1p(t) l2p(t)  l3p(t)  %Velocidades de cada articulación
syms l1pp(t) l2pp(t) l3pp(t) %Aceleraciones de cada articulación
syms m1 m2 m3 Ixx1 Iyy1 Izz1 Ixx2 Iyy2 Izz2 Ixx3 Iyy3 Izz3  %Masas y matrices de
Inercia
syms lc1 lc2 lc3 %l=longitud de eslabones y lc=distancia al centro de masa de cada
eslabón
syms pi g a cero
```

Vector de coordenadas articulares

```
Q= [l1; l2; l3];
```

Vector de velocidades articulares

```
Qp= [l1p; l2p; l3p];
Qp=Qp(t);
```

Creamos el vector de aceleraciones articulares

```
Qpp= [l1pp; l2pp; l3pp];
```

Configuración del robot (0 para junta rotacional, 1 para junta prismática)

```
RP=[1 1 1];
```

Número de grado de libertad del robot

```
GDL= size(RP,2);
GDL_str= num2str(GDL);
```

Articulación 1

Posición de la articulación 1 respecto a 0

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

Matriz de rotación de la junta 1 respecto a 0

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

Articulación 2

Posición de la articulación 2 respecto a 1

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

Matriz de rotación de la junta 1 respecto a 0

```
R(:,:,2)= [1  0   0;
           0  0  -1;
           0  1   0];
```

Articulación 3

Posición de la articulación 3 respecto a 2

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

Matriz de rotación de la junta 1 respecto a 0

```
R(:,:,3)=   [1  0  0;
             0  1  0;
             0  0  1];
```

Vector de ceros

```
Vector_Zeros= zeros(1, 3);
```

Matrices

Inicializamos las matrices de transformación Homogénea locales

```
A(:,:,GDL)=simplify([R(:,:,GDL) P(:,:,GDL); Vector_Zeros 1]);
```

Inicializamos las matrices de transformación Homogénea globales

```
T(:,:,GDL)=simplify([R(:,:,GDL) P(:,:,GDL); Vector_Zeros 1]);
```

Inicializamos las posiciones vistas desde el marco de referencia inercial

```
PO(:,:,GDL)= P(:,:,GDL);
```

Inicializamos las matrices de rotación vistas desde el marco de referencia inercial

```
RO(:,:,GDL)= R(:,:,GDL);


for i = 1:GDL
    i_str= num2str(i);
    A(:,:,i)=simplify([R(:,:,i) P(:,:,i); Vector_Zeros 1]);

   %Globales
    try
       T(:,:,i)= T(:,:,i-1)*A(:,:,i);
    catch
       T(:,:,i)= A(:,:,i);
    end
    T(:,:,i)= simplify(T(:,:,i));

    RO(:,:,i)= T(1:3,1:3,i);
    PO(:,:,i)= T(1:3,4,i);

end
```

Velocidades para el eslabón 3

Calculamos el jacobiano y angular lineal de forma analítica

Inicializamos jacobianos analíticos

```
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

Jacobiano lineal obtenido de forma analítica

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

Jacobiano ángular obtenido de forma analítica

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

Matriz de Jacobiano Completa

```
Jac= [Jv_a;
      Jw_a];
Jacobiano= simplify(Jac)
```

```
Jacobiano =
```

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

Vectores de Velocidades Lineales y Angulares

Velocidad lineal

```
disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 3');
V3=simplify (Jv_a*Qp)
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 3
V3 =
```

$$\begin{pmatrix} l2\mathrm{p}(t) \\ -l3\mathrm{p}(t) \\ l1\mathrm{p}(t) \end{pmatrix}$$

Velocidad angular

```
disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 3');
W3=simplify (Jw_a*Qp)
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 3
W3 =
```

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

Velocidades para eslabón 2

Calculamos el jacobiano lineal y angular de forma analítica

Inicializamos jacobianos analíticos

```
Jv_a2(:,GDL-1)=PO(:,:,GDL-1);
Jw_a2(:,GDL-1)=PO(:,:,GDL-1);

for k= 1:GDL-1
    if RP(k)==0
       %Para las juntas de revolución
        try
            Jv_a2(:,k)= cross(RO(:,3,k-1), PO(:,:,GDL-1)-PO(:,:,k-1));
            Jw_a2(:,k)= RO(:,3,k-1);
        catch
            Jv_a2(:,k)= cross([0,0,1], PO(:,:,GDL-1));%Matriz de rotación de 0 con
respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a2(:,k)=[0,0,1];%Si no hay matriz de rotación previa se obtiene la
Matriz identidad
         end
     else
%         %Para las juntas prismáticas
        try
            Jv_a2(:,k)= RO(:,3,k-1);
        catch
            Jv_a2(:,k)=[0,0,1];
```

```
        end
            Jw_a2(:,k)=[0,0,0];
    end
end
```

SubMatrices de Jacobianos

Jacobiano lineal obtenido de forma analítica

```
Jv_a2= simplify (Jv_a2);
```

Jacobiano ángular obtenido de forma analítica

```
Jw_a2= simplify (Jw_a2);
```

Matriz de Jacobiano Completa

```
Jac2= [Jv_a2;Jw_a2];
Jacobiano2 = simplify(Jac2);
```

Vectores de velocidades lineales y angulares para el eslabón 2

Velocidad lineal

```
disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 2');
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 2
```

```
V2=simplify (Jv_a2*Qp(1:2)); V2
```

```
V2 =
```

$$\begin{pmatrix} \mathrm{l2p}(t) \\ 0 \\ \mathrm{l1p}(t) \end{pmatrix}$$

Velocidad angular

```
disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 2');
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 2
```

```
W2=simplify (Jw_a2*Qp(1:2)); W2
```

```
W2 =
```

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

Velocidades para eslabón 1

Calculamos el jacobiano lineal y angular de forma analítica

```
Jv_a1(:,GDL-2)=PO(:,:,GDL-2);
```

```
Jw_a1(:,GDL-2)=PO(:,:,GDL-2);

for k= 1:GDL-2
    if RP(k)==0
       %Para las juntas de revolución
        try
            Jv_a1(:,k)= cross(RO(:,3,k-1), PO(:,:,GDL-2)-PO(:,:,k-1));
            Jw_a1(:,k)= RO(:,3,k-1);
        catch
            Jv_a1(:,k)= cross([0,0,1], PO(:,:,GDL-2));%Matriz de rotación de 0 con
respecto a 0 es la Matriz Identidad, la posición previa tambien será 0
            Jw_a1(:,k)=[0,0,1];%Si no hay matriz de rotación previa se obtiene la
Matriz identidad
         end
     else
%         %Para las juntas prismáticas
        try
            Jv_a1(:,k)= RO(:,3,k-1);
        catch
            Jv_a1(:,k)=[0,0,1];
        end
            Jw_a1(:,k)=[0,0,0];
     end
 end
```

SubMatrices de Jacobianos

Jacobiano lineal obtenido de forma analítica

```
Jv_a1= simplify (Jv_a1);
```

Jacobiano ángular obtenido de forma analítica

```
Jw_a1= simplify (Jw_a1);
```

Matriz de Jacobiano Completa

```
Jac1= [Jv_a1;
      Jw_a1];
Jacobiano1= simplify(Jac1);
```

Vectores de velocidades lineales y angulares para el eslabón 1

Velocidad lineal

```
disp('Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 1');
```

```
Velocidad lineal obtenida mediante el Jacobiano lineal del Eslabón 1
```

```
V1=simplify (Jv_a1*Qp(1)); V1
```

```
V1 =
```

$$\begin{pmatrix} 0 \\ 0 \\ \mathrm{l1p}(t) \end{pmatrix}$$

Velocidad angular

```
disp('Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 1');
```

```
Velocidad angular obtenida mediante el Jacobiano angular del Eslabón 1
```

```
W1=simplify (Jw_a1*Qp(1)); W1
```

```
W1 =
```

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

Energía Cinética

Distancia del origen del eslabón a su centro de masa

Vectores de posición respecto al centro de masa

```
P01=subs(P(:,:,1), l1, lc1);%La función subs sustituye l1 por lc1 en
P12=subs(P(:,:,2), l2, lc2); %la expresión P(:,:,1)/2
P23=subs(P(:,:,3), l3, lc3);
```

Matrices de inercia para cada eslabón

Eslabón 1

```
I1=[Ixx1 0 0;
    0 Iyy1 0;
    0 0 Izz1];
```

Eslabón 2

```
I2=[Ixx2 0 0;
    0 Iyy2 0;
    0 0 Izz2];
```

Eslabón 3

```
I3=[Ixx3 0 0;
    0 Iyy3 0;
    0 0 Izz3];
```

Función de energía cinética

Extraemos las velocidades lineales del efector final en cada eje

```
Vx= V3(1,1);
Vy= V3(2,1);
```

```
Vz= V3(3,1);
```

Extraemos las velocidades angular del efector final en cada ángulo de Euler

```
W_pitch= W3(1,1);
W_roll= W3(2,1);
W_yaw= W3(3,1);
```

Energía cinética para cada uno de los eslabones

Eslabón 1

```
V1_Total= V1+cross(W1,P01);
K1= (1/2*m1*(V1_Total))'*((V1_Total)) + (1/2*W1)'*(I1*W1);
disp('Energía Cinética en el Eslabón 1');
```

```
Energía Cinética en el Eslabón 1
```

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

Eslabón 2

```
V2_Total= V2+cross(W2,P12);
K2= (1/2*m2*(V2_Total))'*((V2_Total)) + (1/2*W2)'*(I2*W2);
disp('Energía Cinética en el Eslabón 2');
```

```
Energía Cinética en el Eslabón 2
```

```
K2= simplify (K2);
```

Eslabón 3

```
V3_Total= V3+cross(W3,P23);
K3= (1/2*m3*(V3_Total))'*((V3_Total)) + (1/2*W3)'*(I3*W3);
disp('Energía Cinética en el Eslabón 2');
```

```
Energía Cinética en el Eslabón 2
```

```
K3= simplify (K3);

K_Total= simplify (K1+K2+K3);
disp('Energía Cinética Total');
```

```
Energía Cinética Total
```

```
K_Total
```

```
K_Total =
```

$$\frac{\overline{m_3}\ (|\mathrm{l1p}(t)|^2+|\mathrm{l2p}(t)|^2+|\mathrm{l3p}(t)|^2)}{2}+\frac{|\mathrm{l1p}(t)|^2\,\overline{m_1}}{2}+\frac{\overline{m_2}\ (|\mathrm{l1p}(t)|^2+|\mathrm{l2p}(t)|^2)}{2}$$

Energía potencial

Obtenemos las alturas respecto a la gravedad

```
h1= P01(2); %Tomo la altura paralela al eje y
h2= P12(3); %Tomo la altura paralela al eje z
h3= P23(1); %Tomo la altura paralela al eje x
```

Energía potencial de cada eslabón

```
U1=m1*g*h1
```

U1 = 0

```
U2=m2*g*h2
```

U2 = $g\,\mathrm{lc}_2\,m_2$

```
U3=m3*g*h3
```

U3 = 0

Calculamos la energía potencial total

```
U_Total= U1 + U2 +U3;
```

Obtenemos el Lagrangiano

```
Lagrangiano= simplify (K_Total-U_Total);
```

Modelo de Energía

```
H= simplify (K_Total+U_Total)
```

H =

$$\frac{\overline{m_3}\,(|\mathrm{l1p}(t)|^2+|\mathrm{l2p}(t)|^2+|\mathrm{l3p}(t)|^2)}{2}+\frac{|\mathrm{l1p}(t)|^2\,\overline{m_1}}{2}+\frac{\overline{m_2}\,(|\mathrm{l1p}(t)|^2+|\mathrm{l2p}(t)|^2)}{2}+g\,\mathrm{lc}_2\,m_2$$