Actividad 1 (Velocidades Lineales y angulares)

Erik García Cruz, A01732440

Escuela de Ingeniería y Ciencias, Instituto Tecnológico y de Estudios Superiores de Monterrey

TE3001B.101: Fundamentación de robótica (Gpo 101)

19 de febrero, 2025

Introducción

Planteamiento

Con el código realizado en clase que modela la matriz homogénea para el péndulo que contiene dos brazos robóticos.

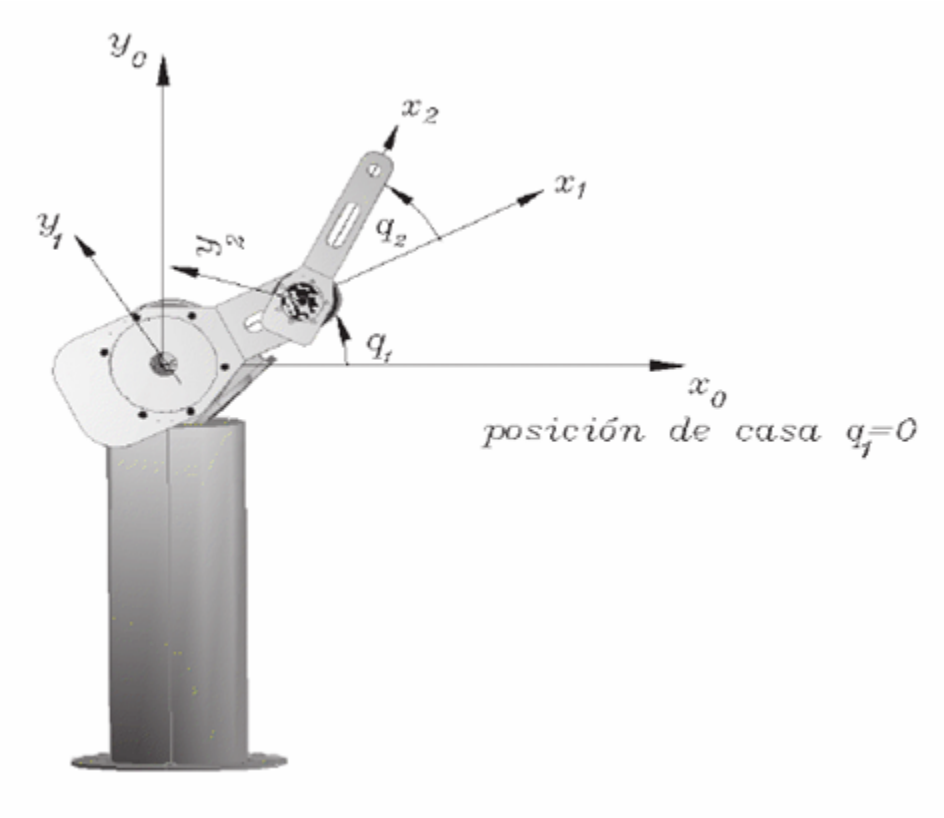

Imagen 1: péndulo de dos articulaciones a calcular

Desarrollo

Primeramente hay que declarar las variables simbólicas a utilizar:

```
%% Declaración de variables simbólicas
syms th1(t) th2(t) l1 l2 t
```

Declarar las Q y Qp (velocidad) a emplear

```
%% Vector de coordenadas articulares
Q = [th1; th2];
disp('Coordenadas articulares:');
pretty(Q);
%% Vector de velocidades articulares
Qp = diff(Q, t);
disp('Velocidades articulares:');
pretty(Qp);
```

Se declaran los grados de libertad: dos debido a que el péndulo tiene dos articulaciones

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

Declaración de las posiciones de las juntas o uniones de los brazos en el péndulo. Para el segundo brazo se suman las posiciones de th1 y th2

```
%% Posición de las juntas respecto al origen
% Posición de la primera junta (extremo del primer péndulo)
P(:,:,1) = [l1*cos(th1);
            l1*sin(th1);
            0];

% Posición de la segunda junta (extremo del segundo péndulo)
P(:,:,2) = P(:,:,1) + [l2*cos(th1 + th2);
                       l2*sin(th1 + th2);
                       0];
```

Se declaran las matrices de rotación para ambos brazos. Para el segundo brazo se emplea la matriz del primero para encontrar la posición respecto al mundo

```
%% Matrices de rotación para cada junta
```

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

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

Vector de posición y rotación. A para locales y T para globales.

```
%% Vectores de posición y rotación globales
Vector_Zeros = zeros(1,3);
for i = 1:GDL
   A(:,:,i) = simplify([R(:,:,i) P(:,:,i); Vector_Zeros 1]);
   try
       T(:,:,i) = T(:,:,i-1) * A(:,:,i);
   catch
       T(:,:,i) = A(:,:,i);
   end
   RO(:,:,i) = T(1:3,1:3,i);
   PO(:,:,i) = T(1:3,4,i);
end
```

Obtenemos el Jacobiano de forma diferencial

```
%% Jacobiano lineal de forma diferencial
Jv = sym(zeros(3,GDL));
for j = 1:GDL
   Jv(:,j) = diff(PO(1:3,1,GDL), Q(j));
end
disp('Jacobiano lineal diferencial:');
pretty(simplify(Jv));
```

Obtenemos el Jacobiano de forma analitica

```
disp('Jacobiano lineal diferencial:');
pretty(simplify(Jv));
%% Jacobiano de forma analítica
Jv_a = sym(zeros(3,GDL));
Jw_a = sym(zeros(3,GDL));
for k = 1:GDL
   if (RP(k) == 0)
       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));
           Jw_a(:,k) = [0;0;1];
       end
   end
end
Jv_a = simplify(Jv_a);
Jw_a = simplify(Jw_a);
disp('Jacobiano lineal analítico:');
pretty(Jv_a);
disp('Jacobiano angular analítico:');
pretty(Jw_a);
```

Se observa que obtenido de forma diferencial y analítica da el mismo resultado.

Se calculan las velocidades Ambos Jacobianos se multiplican por Qp para obtener las velocidades

```
%% Cálculo de velocidades
V = simplify(Jv_a * Qp);
disp('Velocidad lineal:');
pretty(V);
W = simplify(Jw_a * Qp);
disp('Velocidad angular:');
pretty(W);
```

Velocidad lineal:

$$\begin{pmatrix} -(\#2 + l2\sin(\#1))\,\frac{d}{dt}\,th2(t) - \frac{d}{dt}\,th1(t)\,(l1\sin(th1(t)) + \#2 + l2\sin(\#1)) \\ (\#3 + l2\cos(\#1))\,\frac{d}{dt}\,th2(t) + \frac{d}{dt}\,th1(t)\,(l1\cos(th1(t)) + \#3 + l2\cos(\#1)) \\ 0 \end{pmatrix}$$

Velocidad angular:

$$\begin{pmatrix} 0 \\ 0 \\ \frac{d}{dt}\,th1(t) + \frac{d}{dt}\,th2(t) \end{pmatrix}$$