



Instituto Tecnológico y de Estudios Superiores de Monterrey

TE3001B Fundamentos de Robótica(Gpo 101)

Actividad 1 (Velocidades Lineales y angulares)

Autores:

Yestli Darinka Santos Sánchez // A01736992

Profesores:

Juan Manuel Ahuactzin Larios

Rigoberto Cerino Jiménez

Alfredo García Suárez

Miercoles 19 de Febrero de 2025 Semestre (5) Feb-Jun 2025

Campus Puebla

Introducción

El objetivo de este reporte es determinar los vectores de velocidad lineal y angular de un robot manipulador con dos grados de libertad (GDL) en un espacio bidimensional. Se utiliza el método del Jacobiano para calcular la relación entre las velocidades articulares y las velocidades del extremo del robot.

Definición de coordenadas articulares y velocidades

El código define las variables articulares $q1 = \theta1$ y $q2 = \theta2$ como funciones del tiempo.

Se calculan sus derivadas $\dot{q1}(dt)$ y $\dot{q2}(dt)$, que representan las velocidades articulares.

Cálculo de las posiciones de las juntas

Se determinan las posiciones de las juntas usando las coordenadas polares $(l1, \theta1)$ y $(l2, \theta1 + \theta2)$, lo que coincide con la estructura de un robot manipulador de dos eslabones rotacionales.

Matrices de transformación homogénea

Se construyen las matrices T para transformar coordenadas desde el sistema base hasta el extremo del manipulador.

Cálculo del Jacobiano

Se construyen las matrices del Jacobiano lineal J_v y angular J_w , para obtener las velocidades del extremo.

Cálculo de velocidades

Se obtienen los vectores de velocidad lineal V y angular W multiplicando el Jacobiano por el vector de velocidades articulares.

Código

```

1 clear all
2 close all
3 clc
4
5 % Declaración de variables simbólicas
6 syms th1(t) th2(t) l1 l2 t
7
8 % Configuración del robot (0: rotacional, 1: prismático)
9 RP = [0 0];
10
11 % Vector de coordenadas articulares
12 Q = [th1; th2];
13 disp('Coordenadas articulares');
14 pretty(Q);
15
16 % Vector de velocidades articulares
17 Qp = diff(Q, t);
18 disp('Velocidades articulares');
19 pretty(Qp);
20
21 % Número de grados de libertad del robot
22 GDL = size(RP, 2);
23
24 % Posiciones de las juntas
25 P(:,1) = [l1*cos(th1); l1*sin(th1); 0];
26 P(:,2) = P(:,1) + [l2*cos(th1 + th2); l2*sin(th1 + th2); 0];
27
28 % Matrices de rotación
29 R(:,1) = [cos(th1) -sin(th1) 0;
30           sin(th1) cos(th1) 0;
31           0 0 1];
32 R(:,2) = R(:,1) * [cos(th2) -sin(th2) 0;
33                   sin(th2) cos(th2) 0;
34                   0 0 1];
35
36 % Vector de ceros
37 Vector_Zeros = zeros(1, 3);
38
39 % Inicialización de matrices de transformación homogénea
40 for i = 1:GDL
41     A(:,i) = simplify([R(:,i) P(:,i); Vector_Zeros 1]);
42     try
43         T(:,i) = T(:,i-1) * A(:,i);
44     catch
45         T(:,i) = A(:,i);
46     end
47     PO(:,i) = T(1:3,4,i);
48     RO(:,i) = T(1:3,1:3,i);
49 end
50
51 % Cálculo del Jacobiano
52 Jv_a = sym(zeros(3, GDL));
53 Jw_a = sym(zeros(3, GDL));
54
55 for k = 1:GDL
56     if RP(k) == 0
57         try
58             Jv_a(:,k) = cross(RO(:,3,k-1), PO(:,1:GDL) - PO(:,1:k-1));
59             Jw_a(:,k) = RO(:,3,k-1);
60         catch
61             Jv_a(:,k) = cross([0;0;1], PO(:,1:GDL));
62             Jw_a(:,k) = [0;0;1];
63         end
64     else
65         try
66             Jv_a(:,k) = RO(:,3,k-1);
67         catch
68             Jv_a(:,k) = [0;0;1];
69         end
70         Jw_a(:,k) = [0;0;0];
71     end
72 end
73
74 % Cálculo de velocidades
75 V = simplify(Jv_a * Qp);
76 W = simplify(Jw_a * Qp);
77
78 disp('Velocidad lineal obtenida mediante el Jacobiano:');
79 pretty(V);
80 disp('Velocidad angular obtenida mediante el Jacobiano:');
81 pretty(W);

```

Resultados

```

Coordenadas articulares
/ th1(t) \
|         |
\ th2(t) /

Velocidades articulares
/ d      \
| -- th1(t) |
| dt        |
|         |
| d      |
| -- th2(t) |
| dt        |

Velocidad lineal obtenida mediante el Jacobiano:
/          d          d          d          d          \
| - l1 #2 -- th1(t) - l1 #2 -- th2(t) - l2 sin(#1) -- th1(t) - l2 sin(#1) -- th2(t) - l1 sin(th1(t)) -- th1(t) |
|          dt          dt          dt          dt          dt          |
|          |          |          |          |          |          |
|          d          d          d          d          d          |
| l1 #3 -- th1(t) + l1 #3 -- th2(t) + l2 cos(#1) -- th1(t) + l2 cos(#1) -- th2(t) + l1 cos(th1(t)) -- th1(t) |
|          dt          dt          dt          dt          dt          |
|          |          |          |          |          |          |
\          0          /

where

#1 == 2 th1(t) + th2(t)

#2 == sin(2 th1(t))

#3 == cos(2 th1(t))

```

Velocidad angular obtenida mediante el Jacobiano:

$$\begin{bmatrix} 0 \\ 0 \\ \frac{d}{dt} \theta_1(t) + \frac{d}{dt} \theta_2(t) \end{bmatrix}$$