



Tecnológico de Monterrey

Actividad 1

Emiliano Olguin Ortega A01737561

19 de Febrero del 2025

Fundamentación Robótica

1. Declaración de Variables Simbólicas

- Se definen las variables simbólicas necesarias para representar las coordenadas articulares y los parámetros del robot:
- $\theta_1(t)$, $\theta_2(t)$: Ángulos de las articulaciones en función del tiempo.
- L_1 , L_2 : Longitudes de los eslabones del robot.
- t : Variable de tiempo.

2. Configuración del Robot

- Se define el tipo de articulaciones del robot:
 - $\text{Robot_Type} = [0 \ 0]$: Indica que ambas articulaciones son rotacionales (0 para rotacional, 1 para prismático).

3. Definición de Coordenadas Articulares

- Se crea el vector de coordenadas articulares:
 - $\text{Joint_Coordinates} = [\theta_1; \theta_2]$: Representa las posiciones angulares de las articulaciones.

4. Cálculo de Velocidades Articulares

- Se calculan las velocidades articulares derivando las coordenadas articulares con respecto al tiempo:

- `Joint_Velocities = diff(Joint_Coordinates, t)`: Representa las velocidades angulares de las articulaciones.

5. Determinación del Número de Grados de Libertad (DOF)

- Se calcula el número de grados de libertad del robot:

- `DOF = size(Robot_Type, 2)`: En este caso, el robot tiene 2 grados de libertad.

6. Cálculo de Posiciones de las Juntas

- Se calculan las posiciones de las juntas en el espacio cartesiano:

- `Positions(:,1) = [L1*cos(theta1); L1*sin(theta1); 0]`: Posición de la primera articulación.

- `Positions(:,2) = Positions(:,1) + [L2*cos(theta1 + theta2); L2*sin(theta1 + theta2); 0]`: Posición de la segunda articulación.

7. Cálculo de Matrices de Rotación

- Se calculan las matrices de rotación para cada articulación:

- `Rotation_Matrices(:,1)`: Matriz de rotación para la primera articulación.

- `Rotation_Matrices(:,2)`: Matriz de rotación para la segunda articulación, considerando la rotación acumulada.

8. Construcción de Matrices de Transformación Homogénea

- Se construyen las matrices de transformación homogénea para cada articulación:
 - Homogeneous_Matrix(:, :, i): Matriz de transformación homogénea para la articulación i.
 - Transformation_Matrix(:, :, i): Matriz de transformación acumulada hasta la articulación i.
 - End_Effector_Position(:, :, i): Posición del efector final en el sistema de coordenadas global.
 - End_Effector_Rotation(:, :, i): Orientación del efector final en el sistema de coordenadas global.

9. Cálculo del Jacobiano

- Se calculan las matrices del Jacobiano para la velocidad lineal y angular:
 - Jacobiano Lineal (Linear_Jacobian):
 - Para articulaciones rotacionales: $\text{cross}(\text{End_Effector_Rotation}(:, 3, k-1), \text{End_Effector_Position}(:, :, \text{DOF}) - \text{End_Effector_Position}(:, :, k-1))$.
 - Para articulaciones prismáticas: $\text{End_Effector_Rotation}(:, 3, k-1)$.
 - Jacobiano Angular (Angular_Jacobian):
 - Para articulaciones rotacionales: $\text{End_Effector_Rotation}(:, 3, k-1)$.
 - Para articulaciones prismáticas: $[0; 0; 0]$.

10. Cálculo de Velocidades Lineal y Angular

- Se calculan las velocidades lineal y angular del efector final:

- $\text{Linear_Velocity} = \text{simplify}(\text{Linear_Jacobian} * \text{Joint_Velocities})$: Velocidad lineal del efector final.

- $\text{Angular_Velocity} = \text{simplify}(\text{Angular_Jacobian} * \text{Joint_Velocities})$: Velocidad angular del efector final.

11. Visualización de Resultados

- Se muestran los resultados de las velocidades lineal y angular:

- $\text{pretty}(\text{Linear_Velocity})$: Muestra la velocidad lineal de forma simplificada.

- $\text{pretty}(\text{Angular_Velocity})$: Muestra la velocidad angular de forma simplificada.