Universidad del Valle de Guatemala
Departamento de Ingeniería Mecatrónica
MT3005 - Robótica 1
MSc. Miguel Zea

Laboratorio 6

Cinemática diferencial numérica de manipuladores seriales

Objetivos

- Implementar funciones que generen la cinemática directa numérica de un manipulador serial mediante sus parámetros Denavit-Hartenberg.
- Implementar un algoritmo que determine el jacobiano de un manipulador serial general de forma numérica.
- Analizar la cinemática del manipulador myCobot 280.

Procedimiento

En la práctica de esta semana usted desarrollará una función en MATLAB que sea capaz de calcular el jacobiano de cualquier tipo de manipulador serial, bajo el supuesto que se conoce (o puede calcularse) su cinemática directa $\mathcal{K}(\mathbf{q})$. Para ello, deberá implementar un algoritmo basado en la aproximación por diferencias finitas de la derivada

$$\frac{df(x)}{dx} \approx \frac{f(x+\delta x) - f(x)}{\delta x}.$$

El algoritmo a implementar empleará esto para calcular el jacobiano del manipulador columna por columna, empleando solamente información obtenida a partir de la cinemática directa del mismo. La serie de pasos para determinar la columna j del jacobiano se detalla a continuación:

(i) Calcule la cinemática directa $\mathcal{K}(\mathbf{q})$ del manipulador para la configuración $\mathbf{q}$ dada.

(ii) Extraiga y almacene la matriz de rotación $\mathbf{R}(\mathbf{q})$ obtenida en la cinemática directa.

(iii) Calcule el cambio en la cinemática directa debido a un cambio infinitesimal del parámetro q_j empleando la aproximación de la primera derivada, de la forma

$$\frac{\partial \mathcal{K}(\mathbf{q})}{\partial q_j} \approx \frac{\mathcal{K}(\mathbf{q}+\delta q_j \mathbf{e}_j) - \mathcal{K}(\mathbf{q})}{\delta q_j}.$$

El vector $\delta q_j \mathbf{e}_j$ corresponde a un cambio "infinitesimal" en la coordenada q_j tal que, por ejemplo, si estuviésemos analizando un manipulador con configuración $\mathbf{q} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix}^\top$, un cambio diferencial en la segunda coordenada q_2 puede representarse mediante el vector $\delta q_2 \mathbf{e}_2 = \begin{bmatrix} 0 & \delta q_2 & 0 \end{bmatrix}^\top = \begin{bmatrix} 0 & 0.01 & 0 \end{bmatrix}^\top$. Nótese que mientras más pequeño sea el cambio "infinitesimal" mejor será la aproximación para la derivada.

(IV) Extraiga y almacene el cambio en la posición del efector final respecto al cambio diferencial en el parámetro q_j, dado por $\dfrac{\partial \mathbf{o}(\mathbf{q})}{\partial q_j}$. Sabemos que la cinemática directa **completa** de un manipulador no es más que la pose del efector final con respecto a la base del mismo, es decir, es simplemente una matriz de transformación homogénea tal que

$$\mathcal{K}(\mathbf{q}) = \mathbf{T}(\mathbf{q}) = \begin{bmatrix} \mathbf{R}(\mathbf{q}) & \mathbf{o}(\mathbf{q}) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix},$$

con $\mathbf{R} \in \mathbb{R}^{3\times 3}$ y $\mathbf{o} \in \mathbb{R}^3$. Claramente el cambio en la cinemática directa con respecto a la coordenada q_j *también* es una matriz de transformación homogénea, sólo que está conformada por

$$\frac{\partial \mathcal{K}(\mathbf{q})}{\partial q_j} = \frac{\partial \mathbf{T}(\mathbf{q})}{\partial q_j} = \begin{bmatrix} \dfrac{\partial \mathbf{R}(\mathbf{q})}{\partial q_j} & \dfrac{\partial \mathbf{o}(\mathbf{q})}{\partial q_j} \\ \mathbf{0}_{1\times 3} & 0 \end{bmatrix}.$$

(V) Extraiga y almacene el cambio en la orientación del efector final respecto al cambio diferencial en el parámetro q_j, dado por $\dfrac{\partial \mathbf{R}(\mathbf{q})}{\partial q_j}$.

(VI) Determine la matriz anti-simétrica $\mathcal{S}\left(\boldsymbol{\omega}_{q_j}\right)$, que contiene la información sobre el cambio en la orientación del efector final con respecto al cambio diferencial en la coordenada q_j. Para ello, recuerde la relación:

$$\dot{\mathbf{R}}(\mathbf{q}) = \mathcal{S}(\boldsymbol{\omega}_{q_j})\mathbf{R}(\mathbf{q}) \qquad \Rightarrow \mathcal{S}(\boldsymbol{\omega}_{q_j}) = \dot{\mathbf{R}}(\mathbf{q})\mathbf{R}^\top(\mathbf{q}).$$

(VII) Emplee la matriz anti-simétrica para determinar/extraer la derivada parcial de la orientación del efector final con respecto a un cambio en la coordenada q_j, si se sabe que

$$\boldsymbol{\omega}_{q_j} = \frac{\partial \boldsymbol{\theta}(\mathbf{q})}{\partial q_j}.$$

(VIII) Forme la j-ésima columna del jacobiano de la forma

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \cdots & \cdots & \dfrac{\partial \mathbf{o}(\mathbf{q})}{\partial q_j} & \cdots \\ \cdots & \cdots & \dfrac{\partial \boldsymbol{\theta}(\mathbf{q})}{\partial q_j} & \cdots \end{bmatrix}.$$

(IX) Repita el procedimiento para cada una de los n parámetros que conforman la configuración del manipulador. Recuerde que el jacobiano debe ser una matriz de $6 \times n$.

Con el fin de evaluar el funcionamiento correcto del algoritmo, se analizará el caso de uno de los manipuladores myCobot 280 que posee el Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica de la UVG, el cual se muestra en la Figura 1. Para ello, efectúe lo siguiente:

1. Descargue de Canvas el archivo `mt3005lab6.zip` y extraiga sus contenidos dentro de una carpeta en una ubicación de su preferencia. Cambie el folder actual de MATLAB para que coincida con esta carpeta.

θ_j	d_j	a_j	α_j
q_1	131.22	0	$\pi/2$
$q_2 - \pi/2$	0	−110.4	0
q_3	0	−96	0
$q_4 - \pi/2$	63.4	0	$\pi/2$
$q_5 + \pi/2$	75.05	0	$-\pi/2$
q_6	45.6	0	0

Figura 1: Parámetros DH para el manipulador myCobot de 6 GDL.

2. Emplee los parámetros cinemáticos del robot de la Figura 1 para completar la función `robot_def`. Note que la función permite la definición de las transformaciones de base y herramienta, aunque en este caso no se emplearán.

3. Complete el cálculo de la cinemática directa (numérica) del manipulador en la función `robot_fkine`. Recuerde que, bajo la convención de Denavit Hartenberg, la cinemática directa de un manipulador serial está dada por
$$\mathcal{K}(\mathbf{q}) = \mathbf{A}_1\mathbf{A}_2 ... \mathbf{A}_{N-1}\mathbf{A}_N.$$
Tome en consideración que ya se le proveen las funciones para calcular las transformaciones de junta a junta
$$\mathbf{A}_j =^{j-1} \mathbf{T}_j = \mathbf{Rot}_z(\theta_j)\mathbf{Transl}_z(d_j)\mathbf{Transl}_x(a_j)\mathbf{Rot}_x(\alpha_j).$$

4. Para el inciso anterior y posteriores, puede emplear el siguiente fragmento de código para definir el robot myCobot empleando las funciones de la *Robotics Toolbox*. Recuerde que puede emplear `myCobot.fkine(q)` y `myCobot.jacob0(q)` para calcular la cinemática directa y la cinemática diferencial del robot respectivamente, dada la configuración como un vector fila.

```
% Dimensiones del robot (en mm)
d1 = 131.22; a1 = 0; alpha1 = pi/2; offset1 = 0;
d2 = 0; a2 = -110.4; alpha2 = 0; offset2 = - pi/2;
d3 = 0; a3 = -96; alpha3 = 0; offset3 = 0;
d4 = 63.4; a4 = 0; alpha4 = pi/2; offset4 = - pi/2;
d5 = 75.05; a5 = 0; alpha5 = -pi/2; offset5 = pi/2;
d6 = 45.6; a6 = 0; alpha6 = 0; offset6 = 0;

% Definimos el robot como un objeto SerialLink
J1 = Revolute('d', d1, 'a', a1, 'alpha', alpha1, 'offset', offset1);
J2 = Revolute('d', d2, 'a', a2, 'alpha', alpha2, 'offset', offset2);
```

```
J3 = Revolute('d', d3, 'a', a3, 'alpha', alpha3, 'offset', offset3);
J4 = Revolute('d', d4, 'a', a4, 'alpha', alpha4, 'offset', offset4);
J5 = Revolute('d', d5, 'a', a5, 'alpha', alpha5, 'offset', offset5);
J6 = Revolute('d', d6, 'a', a6, 'alpha', alpha6, 'offset', offset6);
myCobot = SerialLink([J1, J2, J3, J4, J5, J6], 'name', 'myCobot')
```

5. Implemente el algoritmo detallado al inicio de la guía en la función `robot_jacobian(q)`, la cual calculará el jacobiano del manipulador para cierta configuración especificada $\mathbf{q} \in \mathbb{R}^n$ (en forma de vector columna). Tome en cuenta que la función `robot_fkine(q)` que completó corresponde a $\mathcal{K}(\mathbf{q})$, es decir, permite calcular la cinemática directa para el robot myCobot dada cierta configuración.

6. La función auxiliar `robot_ellipsoid(q, tipo)` genera el elipsoide de manipulabilidad (de velocidad lineal cuando `tipo = 'v'` o de fuerza cuando `tipo = 'f'`) para el manipulador en una configuración $\mathbf{q} \in \mathbb{R}^n$ dada (como vector columna), junto con el respectivo diagrama de eslabones del robot. Utilice esta función para analizar qué ocurre con el movimiento del robot cercano a sus singularidades. Presente, durante la calificación, las gráficas correspondientes a dos singularidades para el robot myCobot 280.

Para verificar si sus soluciones están correctas, corra la sentencia `calificar('laboratorio6')` en la línea de comando. Cuando esté satisfecho con los resultados, presénteselos al profesor del laboratorio o al auxiliar. Recuerde que entregas tardías representan una penalización del 25 % por semana.