Universidad de Ingeniería y Tecnología

Departamento de Ingeniería Mecatrónica

Fundamentos de Robótica

Profesores: Oscar E. Ramos, Alexander López

Laboratorio 5

Cinemática Inversa del robot Scorbot IX

Lima - Perú

2024-2

Laboratorio 5
Cinemática Inversa del robot Scorbot IX

Objetivos

- Modelar, calcular y verificar la cinemática directa de un robot manipulador de cinco grados de libertad (Scorbot IX) usando la convención de Denavit-Hartenberg estándar.
- Implementar cinematica inversa para el robot Scorbot IX con base en métodos numéricos (método de Newton).
- Mover el robot a ciertas posiciones, utilizando la cinemática inversa desarrollada.

Equipo y Materiales

Descripción	Cantidad
Computadora con Internet	1

Indicaciones del Laboratorio

1. Todos los laboratorios son evaluados, por tanto se debe leer la guía del laboratorio antes de asistir a cada laboratorio.
2. El laboratorio comienza a más tardar 5 minutos después de la hora de inicio.
3. A los 5 minutos de iniciar el laboratorio se empieza con la prueba de entrada, la cual se debe desarrollar de manera personal. Esta prueba de entrada cuenta como parte de la nota del laboratorio, la cual evalúa que el alumn@ haya leído la guía de laboratorio. La prueba tiene una duración de 3 minutos.
4. Los alumnos que lleguen 20 minutos después de iniciar el laboratorio son libres de asistir, pero no tendrán nota en dicho laboratorio.
5. El laboratorio se desarrolla en pareja de 2 personas (solo será de 3 personas en caso falte un alumn@ que no tenga pareja).
6. El laboratorio también será evaluado por medio de un cuestionario colgado en canvas, el cual será resuelto por ambos integrantes del grupo.
7. La parte práctica será revisada por el profesor durante el laboratorio, es posible que se haga preguntas de los códigos desarrollados.
8. Está prohibido copiar, en caso de copia, los alumnos involucrados serán presentados al comité de ética de la universidad.

Resumen del laboratorio

Se presenta un resumen de los puntos a tratar en el siguiente laboratorio.

1. Prueba de entrada (4 puntos).
2. Descargar los paquetes para este laboratorio.
3. Cinematica inversa de Scorbot IX (9 puntos).
4. Evaluación en Canvas (5 puntos).
5. Implementación (2 puntos).

1. Paquetes del laboratorio

Para trabajar en este laboratorio, es necesario descargar el repositorio del curso en el espacio de trabajo “lab_ws”:

$ git clone https://github.com/utecrobotics/fundrobotica-20242 frlabs

En caso que el espacio de trabajo “lab_ws” haya sido borrado, el alumn@ debe volver a crearlo y hacer que el sistema lo reconozca como un espacio de trabajo de ROS. También se debe clonar los repositorios (de github) que contienen toda la información y los drivers del robot Scorbot IX, así como su descripción de su modelo (URDF). Los comandos a utilizar son los siguientes:

$ git clone -b noetic-devel https://github.com/alexwbots/scorbot_er.git

Para ambos casos, compilar los paquetes descargados:
$ cd ..
$ catkin_make

2. Cinemática Inversa del Scorbot IX

El problema de la cinemática inversa de un robot manipulador consiste en encontrar la configuración articular (q) que encuentra la posición ($\boldsymbol{x}$) y orientación (w, $\boldsymbol{\epsilon}$) deseada del efector final. En este laboratorio solamente se trabajará con la posición. Debido a que el robot Scorbot IX posee 5 grados de libertad, el cálculo analítico de su cinemática inversa es complejo. En lugar de un enfoque analítico, en este caso se utilizará un algoritmo numérico, como el método de Newton. Este método, al igual que el método del descenso del gradiente, requiere primero el cálculo de la matriz Jacobiana.

2.1. El Jacobiano

Antes de utilizar el método de Newton, se requiere el Jacobiano del robot. Debido a que el robot tiene seis grados de libertad, la cinemática directa de manera simbólica es una expresión bastante

grande, y por tanto el Jacobiano simbólico es un tanto prohibitivo. En este caso, se optará por el cálculo numérico del Jacobiano. La posición se representa mediante coordenadas cartesianas como $\boldsymbol{x} = [x\ y\ z]^T$. Usando esta posición, el Jacobiano analítico con respecto a las variaciones articulares, para un robot de 5 grados de libertad, es:

$$J = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_6} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_6} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \cdots & \frac{\partial z}{\partial q_6} \end{bmatrix} = \begin{bmatrix} \frac{\partial \boldsymbol{x}}{\partial q_1} & \frac{\partial \boldsymbol{x}}{\partial q_2} & \cdots & \frac{\partial \boldsymbol{x}}{\partial q_6} \end{bmatrix}$$

cada elemento puede ser calculado utilizando diferencias finitas. Notar que, debido a que cada columna toma la derivada con respecto a la misma articulación, se puede utilizar el mismo incremento para toda la columna. Por ejemplo, la primera columna de esta matriz puede ser calculada como:

$$\frac{\partial \boldsymbol{x}}{\partial q_1} \approx \frac{\boldsymbol{x}(\boldsymbol{q} + \delta q_1) - \boldsymbol{x}(\boldsymbol{q})}{\delta q_1}$$

donde ***x(q)*** es la posición en términos de la variable articular ***q***, y δq_1 es un incremento en la articulación 1. Dado que ***x(q)*** representa la posición, puede ser obtenido de la cuarta columna de la matriz de transformación homogénea que relaciona el efector final con respecto a la base del robot, como se muestra en la Figura 1.

$${}^0T_e = \left[\begin{array}{cccc} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Figura 1: La transformación homogénea indica el cambio de posición y orientación con respecto a una articulación y la siguiente en una misma cadena cinemática.

2.2. Metodo de Newton

El método de Newton es un método iterativo popular utilizado en muchos campos de las matemáticas, incluida la cinemática inversa. En cinemática inversa, el método de Newton se puede utilizar para resolver los ángulos de articulación necesarios para lograr la posición y orientación deseadas del efector-final. La idea básica del método de Newton es actualizar iterativamente una estimación inicial de la solución hasta que converja a la solución real. En

cinemática inversa, la estimación inicial suele ser los ángulos de las articulaciones actuales del brazo del robot.

Para aplicar el método de Newton en cinemática inversa, primero debemos definir una función de costo que mida el error entre la posición actual del efector final y la posición deseada. Esta función de coste suele ser una medida de distancia, como la distancia euclidiana o la distancia euclidiana al cuadrado. Luego calculamos la matriz jacobiana del brazo del robot, que relaciona los cambios en los ángulos de las articulaciones con los cambios en la posición y orientación del efector-final. La matriz jacobiana es un componente crucial del método de Newton, ya que nos permite estimar cómo deben actualizarse los ángulos de las articulaciones para acercar el efector final a la posición deseada. Luego, la fórmula de actualización de Newton se aplica iterativamente para actualizar los ángulos de las articulaciones hasta que se minimiza la función de costo o se alcanza la precisión deseada. La fórmula de actualización implica calcular la inversa de la matriz jacobiana y multiplicarla por el vector de error entre la posición actual del efector final y la posición deseada.

2.3. Método Gradiente

Este es otro método iterativo popular utilizado en cinemática inversa para hallar los ángulos de unión necesarios para lograr la posición y orientación deseadas del efector final. Es un método de optimización de primer orden que usa el gradiente de la función de costo para actualizar los ángulos de las articulaciones en la dirección del descenso más pronunciado. La idea básica del método del gradiente es comenzar con una suposición inicial de los ángulos de las articulaciones y luego actualizarlos iterativamente en la dirección del gradiente negativo de la función de costo. El gradiente es un vector que apunta en la dirección del mayor incremento en la función de costo, y tomando su negativo da la dirección del descenso más pronunciado.

Para aplicar el método del gradiente en cinemática inversa, primero debemos definir una función de costo que mida el error entre la posición actual del efector final y la posición deseada. Esta función de coste suele ser una medida de distancia, como la distancia euclidiana o la distancia euclidiana al cuadrado. Luego, se calcula la gradiente de la función de costo con respecto a los ángulos de articulación. Esto implica calcular las derivadas parciales de la función de costo con respecto a cada ángulo de articulación. La gradiente es un vector cuyas componentes son las derivadas parciales de la función de coste. Por último, los ángulos de las articulaciones se actualizan iterativamente utilizando el tamaño de paso o la tasa de aprendizaje, y la derivada parcial de la función de coste con respecto a cada ángulo de articulación.

El método de gradiente en cinemática inversa es computacionalmente eficiente y relativamente fácil de implementar. Sin embargo, puede converger a un mínimo local en lugar del mínimo global de la función de costo, y la elección del tamaño del paso puede ser un problema desafiante. También hay varias variaciones y mejoras del método de gradiente, como el método

de gradiente conjugado y el método de descenso de gradiente estocástico, que tienen como objetivo abordar estos problemas.

2.4. Cinemática Inversa

Actividad 2

2.1. (1 punto) Crear un lanzador con el nombre "scorbot_display_sliders.launch" y "scorbot_display.launch" (este sin sliders) con los nodos necesarios para que se pueda mostrar la visualización del robot en RViz.

2.2. (3 punto) Crear un "test_fkine", en base al usado en el laboratorio 4, el cual pueda trabajar con este robot. A partir de este código se puede probar el modelo denavit-hatenberg del robot. Completar las funciones del archivo "lab5functions.py".

2.3. (1 punto) En el archivo "lab5functions.py" implementar la función "ikine", la cual calcula la cinemática inversa del robot. Se puede realizar esto utilizando el método de Newton. Algunas variables han sido ya añadidas (epsilon, max iter, delta), pero sus valores pueden cambiar según los resultados que se obtengan.

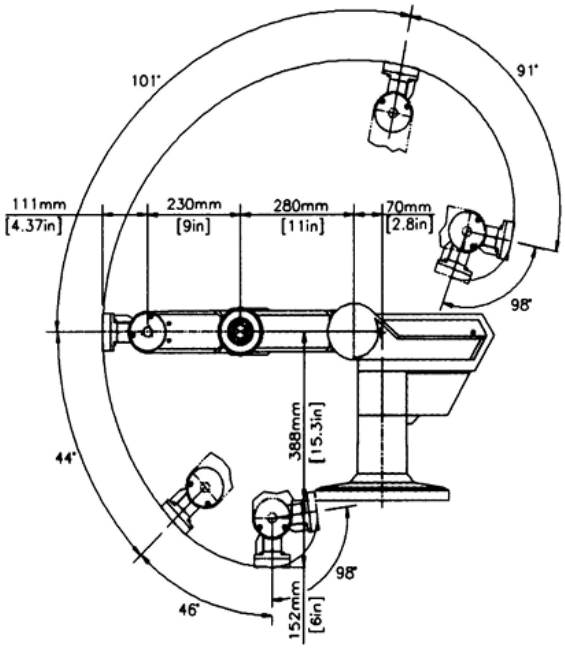

Figura 2: Medidas del robot Scorbot IX.

Los parámetros Denavit-Hartenberg se pueden calcular a partir de las siguientes medidas que se muestran en la Figura 2. En esta figura no se muestran todas las medidas, por tanto el alumn@

tendrá que aproximar o ajustar ciertas medidas, para que calce con los ejes de rotación seleccionados.

Notar que para terminar el bucle principal antes del máximo número de iteraciones, se debe añadir una condición de término al final. Se puede considerar que cuando la normal del error es menor que un determinado epsilon, la solución obtenida con el algoritmo es muy cercana a la solución real y es seguro salir del bucle.

Para probar el correcto funcionamiento de la cinemática inversa, se puede correr la prueba llamada "test_ikine". Primero, dar una mirada a este archivo y asegurarse de que se comprende cada línea de código que contiene. Para visualizar el robot, utilizar la siguiente línea de código:

```
$ roslaunch lab5 scorbot_display.launch
```

Para ambos casos, ejecutar el siguiente nodo:

```
$ rosrun lab5 test_ikine
```

Una bola verde indica la posición deseada en espacio, y una bola roja (obtenida usando la cinemática directa) muestra la posición alcanzada por el efector final usando la cinemática inversa. Probar con diferentes valores de posiciones cartesianas deseadas.

Es importante mencionar que el código "test_ikine" sólo funciona para robots de la marca Universal Robots.

2.4. (1 punto) Por tanto, se debe adecuar el código de "test_ikine", como se hizo con el código de test_fkine.

2.5. Movimiento del Robot

En esta parte del laboratorio se busca controlar la posición de la simulación del robot Scorbot IX utilizando un nodo.

2.5.1. Movimiento en Gazebo

Antes de poder utilizar el robot Scorbot IX en Gazebo, para realizar la prueba, se utilizará el archivo de tipo launch llamado "gazebo.launch" del paquete "lab5", el cual abre el simulador gazebo y invoca al robot Scorbot IX sobre un soporte. Luego, se ejecutará un nodo de ejemplo, llamado "test_gazebo", el cual modifica el valor de una articulación del robot Scorbot IX y lo envía al simulador Gazebo. Ejecutar la simulación por medio del siguiente comando:

```
$ roslaunch scorbot_er_description gazebo.launch
```

Para ejecutar el nodo, usar la siguiente línea de código:

```
$ rosrun lab5 test_gazebo
```

Actividad 3

3.1. (3 puntos) Mover el eje final del robot de forma que dibuje un cuadrado en el plano XY, siendo este plano paralelo al piso. El cuadrado debe tener 8 cm de lado.

3. Implementación

Actividad 4

4.1. (2 punto) Implementación en el robot UR5 usando Polyscope.