

Universidad de Ingeniería y Tecnología



FUNDAMENTOS DE ROBÓTICA

Prof. Oscar E. Ramos Ponce

Guía de Laboratorio 2:
Cinemática de un Robot Manipulador

Lima - Perú

2017 - 1

Laboratorio 2: Cinemática de un Robot Manipulador

1. Objetivos

- Modelar cinemáticamente el brazo manipulador del robot Fetch.
- Implementar computacionalmente la cinemática directa del robot Fetch en ROS usando los parámetros de Denavit-Hartenberg.
- Implementar la cinemática inversa de este robot en ROS, usando Python, con base en el método de Newton.
- Desplazar objetos de una mesa a otra usando el manipulador, y la cinemática implementada, en un ambiente simulado (en Gazebo).

2. Equipos, Materiales y Otros

No	Descripción	Cantidad
1	Computadora con ROS y Gazebo en Linux	1

3. Orientaciones de Seguridad

- Respetar las recomendaciones del docente así como las recomendaciones indicadas en los carteles de señalización.
- Usar los EPP durante el laboratorio.
- Utilizar los equipos de acuerdo con las recomendaciones.
- Respetar las recomendaciones para el uso del ambiente.

4. Cinemática Directa del Robot Fetch

El robot Fetch es un manipulador móvil desarrollado por la compañía *Fetch Robotics*, fundada en el año 2014. Consta de sensores, una cabeza con dos grados de libertad, una base móvil y un brazo de siete grados de libertad. En este laboratorio solamente la parte correspondiente al manipulador será de importancia (no se utilizará ni la parte móvil ni el grado de libertad prismático vertical del robot). Las dimensiones del brazo de este robot

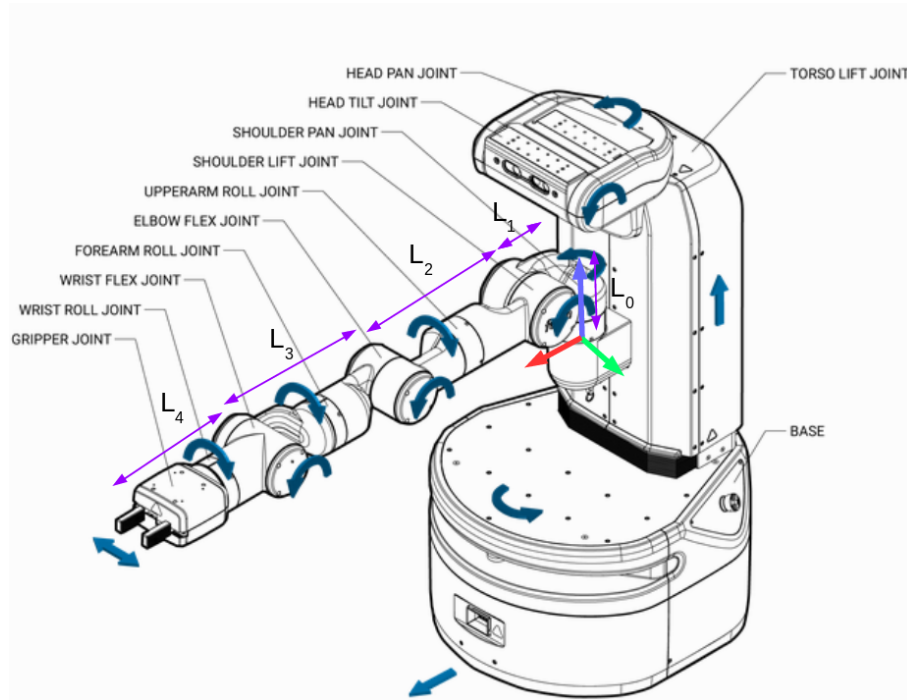


Figura 1: Robot Fetch: articulaciones y dimensiones [1]

pueden observarse en la Figura 1 y son las siguientes: $L_0 = 0,06$, $L_1 = 0,117$, $L_2 = 0,352$, $L_3 = 0,3215$, $L_4 = 0,30495$.

El controlador del robot así como los modelos del mismo pueden ser encontrados en el repositorio de Github <https://github.com/fetchrobotics>. En particular, los paquetes necesarios para este laboratorio son los denominados `fetch_gazebo` y `fetch_ros`. Estos paquetes ya han sido descargados así que no es necesario la configuración de los mismos. Sin embargo, es necesario descargar el archivo comprimido `lab2.zip` y extraerlo dentro de `lab_ws/src` para que pueda ser encontrado por ROS.

El robot puede ser visualizado usando RViz como en el laboratorio anterior. Para la visualización se puede ejecutar el archivo `fetch_sliders.launch` como

```
$ roslaunch lab2 fetch_sliders.launch
```

obteniéndose una salida semejante a la de la Figura 2

Problema Obtener la cinemática del manipulador del robot Fetch usando el modelamiento de Denavit-Hartenberg:

1. Asignar un sistema adecuado de coordenadas para cada articulación.
2. Obtener una tabla con los parámetros DH

4.1. Implementación de la cinemática directa

Una vez obtenidos los parámetros DH, es necesario introducirlos al programa. Una función llamada `dh` ha sido implementada en Python (en el archivo `helpers.py`). Notar que en Python para el uso de matrices se utiliza la librería llamada `numpy`. Usando esta función,

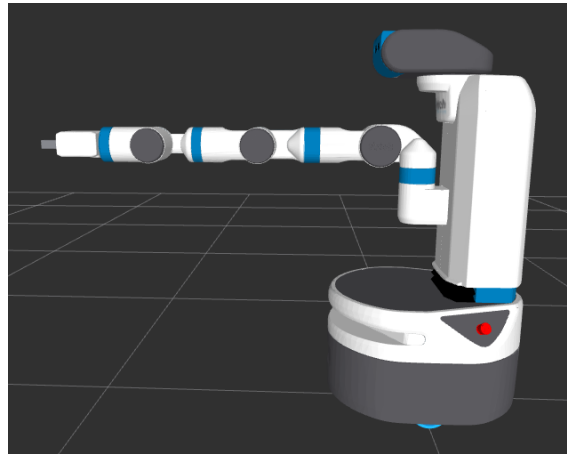


Figura 2: Vista del robot Fetch desde RViz

obtener las matrices y el producto de las mismas, las cuales definen la cinemática directa del robot. Esta implementación se realizará en la función `forward_kinematics` que se encuentra en el archivo `lab2/src/fkine`.

Verificación Se puede verificar la validez del procedimiento anterior con RViz. Lanzar el modelo del robot en RViz como:

```
$ roslaunch lab2 fetch_rviz.launch
```

Luego de esto completar el archivo `lab2/src/fkine` para que la posición y orientación obtenidas estén con respecto al sistema de la base del robot (en el suelo). Una vez completado esto, al ejecutar el programa como

```
$ rosrunc lab2 fkine
```

Se observará un marcador rojo que indica la posición del efector final calculada mediante cinemática inversa. Para diferentes configuraciones articulares, verificar que dicho marcador sea consistente con lo esperado.

5. Cinemática Inversa del Robot Fetch

En el archivo llamado `lab2/src/ikine` completar la cinemática inversa utilizando el método de Newton y el cálculo numérico del Jacobiano. En este laboratorio, por simplicidad, solamente se considerará la cinemática inversa de la posición del robot y no de la orientación.

Verificación en RViz Se puede verificar la validez del cálculo de cinemática inversa usando los marcadores. Se observará un marcador rojo que indica la posición obtenida con cinemática directa y un marcador verde que indica la posición deseada. Verificar que ambos marcadores estén superpuestos.

6. Manipulación de Objetos

Luego de implementar la cinemática inversa, realizar la manipulación de objetos en gazebo como se muestra en el ejemplo dado en `lab2/src/manip`. Notar que este ejemplo solamente funcionará cuando se tenga las funciones de cinemática directa e inversa.

Para observar el resultado, lanzar Gazebo como

```
$ roslaunch lab2 fetch_gazebo.launch
```

El objetivo de esta parte consiste en hacer que el manipulador desplace los tres cubos (mostrados en la Figura 3) de una mesa a la otra. Para ayudarse en el movimiento del efector final se utilizará la cinemática inversa anteriormente desarrollada.

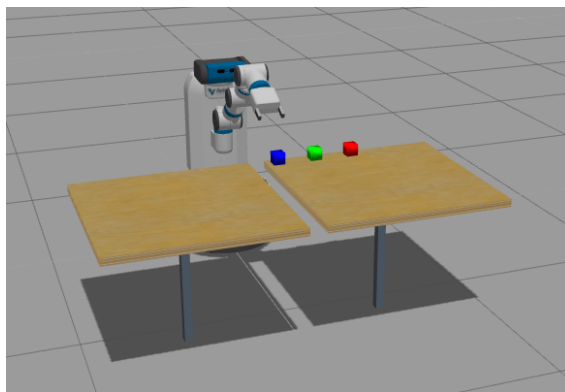


Figura 3: Robot Fetch en Gazebo con 3 cubos a manipular

7. Conclusiones

Elaborar algunas conclusiones sobre el laboratorio desarrollado e indicarlas en el informe del laboratorio.

Referencias

- Fetch Robotics. *Fetch Robot Documentation*. http://docs.fetchrobotics.com/robot_hardware.html
- Tutoriales de ROS: <http://wiki.ros.org/ROS/Tutorials>