

Lab. 4 - Cinemática Directa - Phantom X - ROS

1. Objetivos

- Crear todos los *Joint Controllers* con ROS para manipular servomotores Dynamixel AX-12 del robot *Phantom X Pincher*.
- Manipular los tópicos de estado y comando para todos los *Joint Controllers* del robot *Phantom X Pincher*.
- Manipular los servicios para todos los *Joint Controllers* del robot *Phantom X Pincher*.
- Conectar el robot *Phantom X Pincher* con MATLAB o Python usando ROS.

2. Requisitos

- Ubuntu versión 20.xx preferible 20.04 LTS con ROS.
- Espacio de trabajo para *catkin* correctamente configurado.
- Paquetes de Dynamixel Workbench. https://github.com/fegonzalez7/rob_unal_clase3
- Paquete del robot Phantom X: https://github.com/felipeg17/px_robot.
- Python o MATLAB 2015b o superior instalado en el equipo.
- Robotics toolbox de Mathworks (Disponible desde la versión 2015 en adelante).
- Toolbox de robótica de *Peter Corke*.
- Un (1) manipulador Phantom X Pincher con su base en madera.



Figura 1: Robot PhantomX

3. Ejercicio en el laboratorio

Cada grupo tendrá a cargo un robot **Phantom X Pincher**. Siga las indicaciones del laboratorista y profesores para hacer buen uso de los robots del laboratorio.

Mediciones:

- Establezca las longitudes de eslabón para cada articulación del robot *Phantom X Pincher*, para este proceso apóyese en un **CALIBRADOR**. Recuerde que la longitud de eslabón es la mínima distancia que conecta dos juntas consecutivas. Genere un diagrama como el presentado en la figura 2 con los datos medidos.

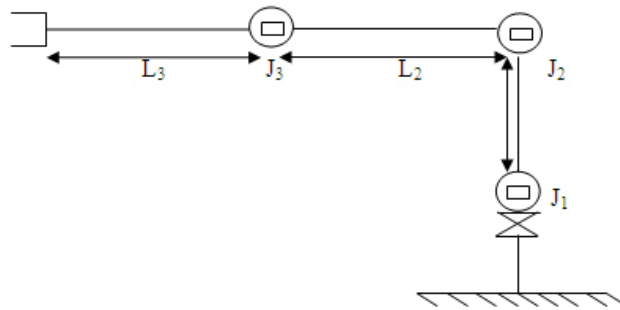


Figura 2: Ejemplo de diagrama.

Análisis:

- Con las dimensiones medidas obtenga los parámetros DH del robot *Phantom X Pincher*.
- Genere diagrama del robot con las tablas de parámetros articulares (Utilice algún software de ilustración).

ROS:

- Con base en la documentación de los motores Dynamixel en ROS, cree un script que publique a los tópicos y llame a los servicios correspondientes para realizar el movimiento de cada una de las articulaciones del manipulador (*waist*, *shoulder*, *elbow*, *wrist*). La lógica del script puede ser la siguiente:
 - Se debe realizar el movimiento entre dos posiciones angulares características: una de home y otra objetivo.
 - El movimiento de las articulaciones debe realizarse de forma secuencial iniciando por la articulación de la base, agregue una pequeña espera entre cada movimiento para facilitar la grabación de videos de demostración.

Toolbox:

- Utilice el comando `SerialLink` para crear el robot con los parámetros de su tabla DH.
- Obtenga la matriz de transformación homogénea desde la base hasta el efector final, la idea es realizar el análisis en el TCP, este punto puede ser elegido en la mitad de la pinza (Cinemática directa del robot).
- Grafique varias posiciones del robot incluyendo la de HOME utilizando las funciones del toolbox (`SerialLink.plot`).
- **Importante:** Para la correcta orientación del marco de coordenadas del efector final, revise la propiedad `.tool` del robot creado mediante `SerialLink`.

Conexión con Matlab o Python:

- Cree un *script* que permita publicar en cada tópico de controlador de junta, se deben validar los límites articulares de cada junta.

- Cree un *script* que permita suscribirse a cada tópico de controlador de junta, el *script* debe retornar la configuración de 5 ángulos en grados.

Matlab o Python + ROS + Toolbox:

- Cree un código en Matlab que envíe la posición en ángulos deseada a cada articulación del robot utilizando las herramientas de ROS + Dynamixel, el programa deberá graficar la configuración del robot usando las herramientas del *toolbox*, esta configuración deberá coincidir con la obtenida en el robot real.
- Pruebe las siguientes poses generadas a partir de los valores articulares de q_1, q_2, q_3, q_4, q_5 (Recuerde que los valores de configuración se toman respecto a home, para el cual todos los valores articulares son cero):
 1. 0, 0, 0, 0, 0.
 2. -25, 15, -20, 20, 0.
 3. 35, -35, 30, -30, 0.
 4. -85, 20, -55, 17, 0.
 5. -80, 35, -55, 45, 0.

Cuide que las poses no interfieran con los límites articulares o algún objeto en el espacio de trabajo, puede ajustar el valor de los ángulos de las poses para esquivar algún obstáculo.

- Cree una interfase de usuario o HMI donde se muestre al usuario:
 1. Nombres, logos y datos de los integrantes del grupo.
 2. Imagen perspectiva de la posición actual del manipulador con la última posición enviada.
 3. Opción para seleccionar 1 de las 5 poses y enviarlas al manipulador.
 4. Valores de los valores articulares reales de cada motor.
 5. Imagen perspectiva de la posición actual del manipulador con los valores articulares.

Entrega



1. **Forma de trabajo:** Grupal de 2 personas. **Importante:** Cada integrante deberá poner la URL del repo creado.
2. **Entregables:** Se deberá crear un repositorio en GitHub con:
 - Código del script utilizado para el desarrollo de la práctica.
 - Vídeo del brazo alcanzando cada posición solicitada.
 - Vídeo demostración de uso de la interfase de usuario.
 - Gráfica digital de las poses comparándola con la fotografía del brazo real en la misma configuración.
 - Descripción de la solución planteada.
3. **Fecha de entrega:** según actividad en Moodle.