

Robótica

Cinemática Directa - Phantom X - ROS No.4

Integrantes:

Daniel Felipe Segura Rincon - dsegurar@unal.edu.co

Juan Andres Barrera Rodriguez - jbarreraro@unal.edu.co

Universidad Nacional de Colombia

2023-2

Metodología:

1. **Introducción:** Este informe aborda un estudio detallado del manipulador Phantom X Pincher en el campo de la robótica. Exploramos su cinemática, integración con ROS, y el uso de herramientas como MATLAB y Python. Además, analizamos las limitaciones y desafíos encontrados, así como las conclusiones derivadas del estudio.

2. **Materiales y equipos:**

- Ubuntu preferible 20.04 LTS.
- MATLAB 2015b o superior.
- Robotics toolbox de matworks.
- Un manipulador Phantom X Pincher con su base en madera.
- Calibrador pie de rey.

3. **Procedimiento:**

a. **Análisis de Cinemática directa:**

Utilizando el calibrador se miden las longitudes de todos los eslabones del robot y se plantea el siguiente modelo de cinemática directa para el Phantom X Pincher:

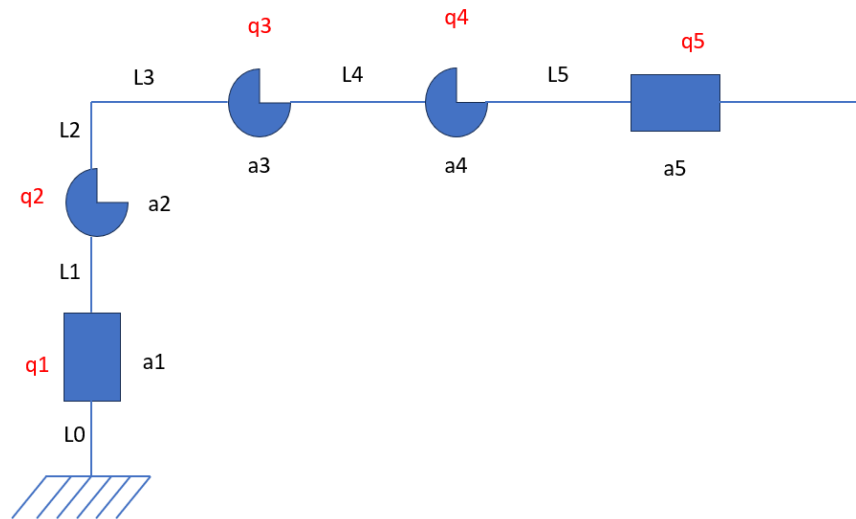


Figura 1: Representación robot Phantom X

Donde:

- $L1 = 45 \text{ mm}$
- $L2 = 100 \text{ mm}$
- $L3 = 35.5 \text{ mm}$
- $L4 = 99 \text{ mm}$
- $L5 = 45 \text{ mm}$

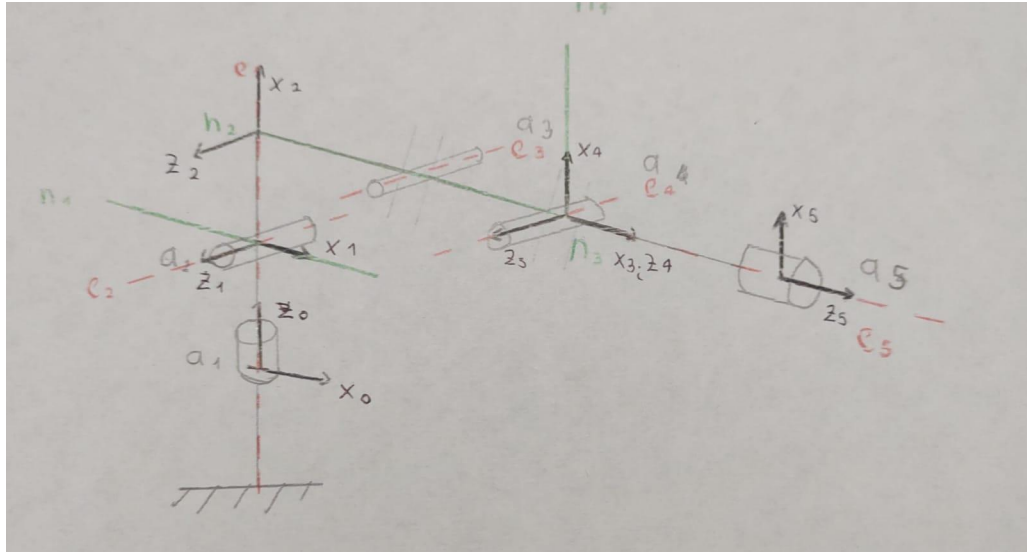
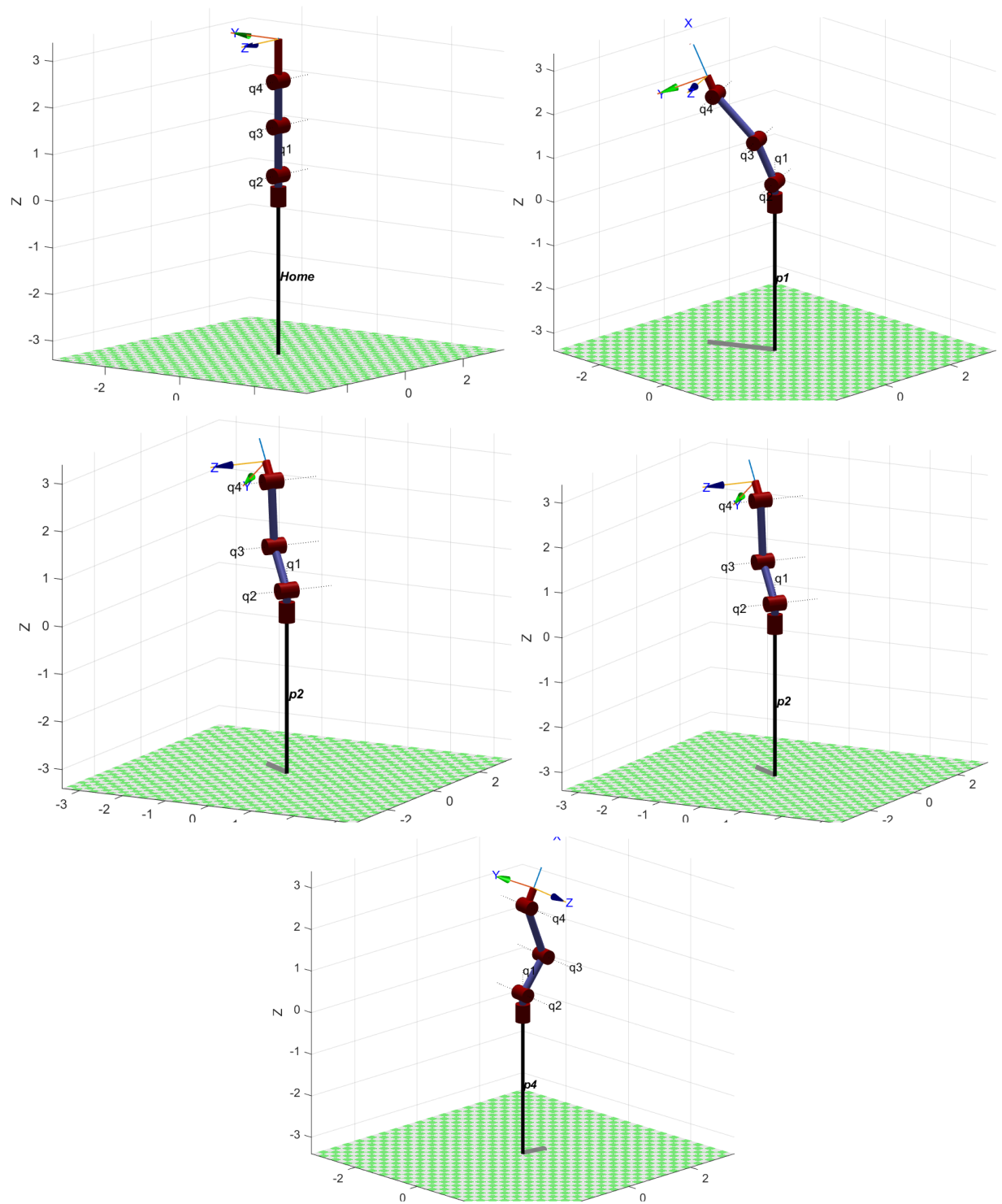


Figura 2: Análisis Directo robot Phantom X

Con este modelo se puede determinar la siguiente tabla con los parámetros DH del robot:

| i | Θ_i | d_i | a_i | α_i |
|---|-------------|-------|----------------|------------|
| 1 | q_1 | L_1 | 0 | $-\pi/2$ |
| 2 | $q_2 - q_1$ | 0 | L_2 | 0 |
| 3 | $-q_3$ | 0 | $-(L_3 + L_4)$ | 0 |
| 4 | $q_4 - q_3$ | 0 | 0 | $\pi/2$ |
| 5 | q_5 | L_5 | 0 | 0 |

- b. **ROS:** Basándose en la documentación de los motores Dynamixel en ROS, se crea un script que publique en los tópicos adecuados y llame a los servicios correspondientes para controlar el movimiento de las articulaciones del manipulador, como la cintura (waist), el hombro (shoulder), el codo (elbow) y la muñeca (wrist).
- c. **Toolbox:** Usando el comando SerialLink, se crea un modelo de robot utilizando los parámetros de la tabla de Denavit-Hartenberg (DH). A continuación, se obtiene la matriz de transformación homogénea desde la base hasta el efector final, centrándose en el punto de interés en la mitad de la pinza (para el análisis de la cinemática directa del robot). Además, se grafican varias posiciones del robot, incluyendo la posición de "HOME" utilizando las funciones del Toolbox SerialLink.



El script de Matlab se puede encontrar en el repositorio de GitHub.

- d. **Conexión con Matlab o Python:** Se realiza la conexión con python por medio de un script que permita publicar en cada tópico de controlador de articulación, y otro que que permita suscribirse a cada tópico de controlador de articulación.

- e. **Python + ROS + Toolbox:** Se desarrolla un código en python que utiliza las herramientas de ROS + Dynamixel para enviar las posiciones angulares deseadas a cada articulación del robot. El programa tiene la capacidad de representar gráficamente la configuración del robot utilizando el toolbox, y esta representación se debe corresponder con la configuración real del robot.
- Se realizaron pruebas con las siguientes poses generadas a partir de los valores articulares q_1 , q_2 , q_3 , q_4 y q_5 :

- 0, 0, 0, 0, 0.
- 25, 25, 20, -20, 0.
- -35, 35, -30, 30, 0.
- 85, -20, 55, 25, 0.
- 80, -35, 55, -45, 0.

Se asegura que estas poses no entren en conflicto con los límites articulares del robot ni con objetos en su espacio de trabajo.

El script de python creado, recibe el parámetro de posición en bits por lo que para poder determinar su equivalencia en grados se usó el software dynamixel y así se pudo determinar la posición en bits a la cual se debía mover el robot.

Adicionalmente, mediante este software también se determinaron los 'ceros' del robot ya que en había algunas articulaciones que tenían cierto desfase, los 'ceros' del robot son los siguientes [0 180 270 180 150]

Además, se ha creado una Interfaz de Usuario (HMI) que proporciona al usuario la siguiente información:

- Nombres, logotipos y detalles de los miembros del grupo de trabajo.
- Una vista en perspectiva de la posición actual del manipulador con la última posición enviada.
- Una opción para seleccionar una de las cinco poses mencionadas y enviarla al manipulador.
- Visualización de los valores articulares reales de cada motor.
- Una imagen en perspectiva de la posición actual del manipulador con los valores articulares actuales.

Resultados:

- El código de Python utilizado para desarrollar la práctica se encuentra en el repositorio.
- Los videos correspondientes al robot alcanzando las diferentes posiciones solicitadas en el laboratorio se encuentran en la carpeta de drive disponible en el repositorio.

Conclusiones:

- Se ha logrado comprender y aplicar con éxito los conceptos de cinemática directa en el robot Phantom X Pincher. Gracias a las mediciones de las longitudes de los eslabones y los parámetros DH calculados, hemos sido capaces de representar y comprender la cinemática del robot de manera efectiva.
- Se han adquirido habilidades para integrar el robot Phantom X Pincher con el sistema operativo robótico (ROS), permitiéndonos controlar las articulaciones del robot y adquirir información sobre su estado de manera eficiente. También hemos empleado herramientas como Python para lograr esta integración, lo que amplía las posibilidades de control y análisis.
- Se logró probar el robot en diferentes posiciones generadas a partir de valores articulares específicos y hemos garantizado que estas poses no entren en conflicto con los límites articulares del robot ni con objetos en su espacio de trabajo. Esto demuestra la capacidad de control y planificación de movimientos del Phantom X Pincher.