

## Laboratorio 5 - Cinemática Inversa - Robot Phantom X - ROS

### 1. Objetivos

- Determinar el modelo cinemático inverso del robot *Phantom X*.
- Generar trayectorias de trabajo a partir del modelo cinemático inverso del robot.
- Implementar el modelo cinemático inverso del robot en MATLAB o Python.
- Operar el robot usando ROS a partir de trayectorias generadas en MATLAB o Python.

### 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](https://github.com/fegonzalez7/rob_unal_clase3)
- Paquete del robot Phantom X: [https://github.com/felipeg17/px\\_robot](https://github.com/felipeg17/px_robot).
- Toolbox de robótica de *Peter Corke*.
- Marcadores borrables.



Figura 1: Robot Phantom X Pincher.

### 3. Descripción

#### 3.1. Cinemática Inversa

El problema cinemático inverso consiste en determinar la configuración articular de un manipulador, dadas la posición y orientación del efector final respecto a la base. Este problema puede resolverse mediante métodos geométricos, algebraicos o numéricos. En el caso particular del robot *Phantom X* el cual posee 4 GDL, el enfoque más práctico es combinar el método geométrico con el desacople de muñeca.

## 4. Ejercicio en el laboratorio

### MATLAB o Python + Toolbox:

- Siguiendo la metodología analizada en clase, desarrolle un modelo de cinemática inversa del manipulador *Phantom X* e implémtelo en MATLAB o Python. Se recomienda usar el *toolbox* para verificar la solución hallada.
- Esboce el espacio de trabajo del robot *Phantom X*. Debido a restricciones de control sobre los motores, las trayectorias deben ser creadas utilizando múltiples puntos intermedios (via-points), ajuste la cantidad y separación de estos puntos a cada uno de los trazos de las rutinas. Los trazos curvos pueden ser múltiples trazos rectos cortos pero ajuste la cantidad de puntos para una correcta aproximación.
- Consulte los métodos disponibles en el *toolbox* para determinar la cinemática inversa de un manipulador.

### ROS - Aplicación de escritura:

- Se tiene como objetivo implementar rutinas de escritura con el robot *Phantom X*, que consiste en lo siguiente:
  1. El robot deberá tomar un marcador borrable y dibujar figuras en la superficie plana de trabajo.
  2. El robot debe ser controlado por un único programa (script) y este programa debe ser controlado por el usuario.

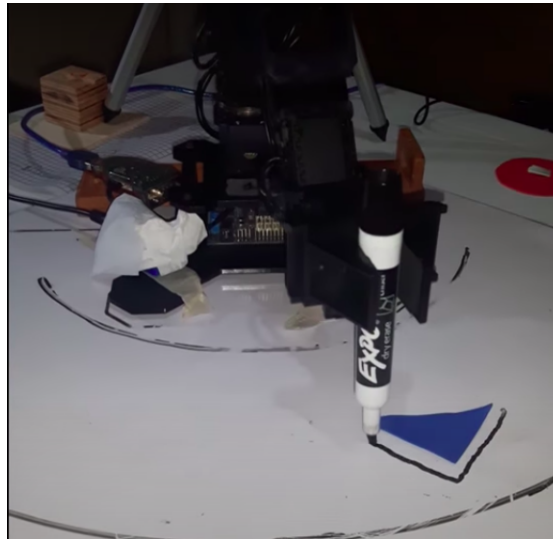


Figura 2: Proceso de escritura usando robot *Phantom X*.

- Tomando como base el trabajo realizado en el laboratorio de cinemática directa, y utilizando el modelo de cinemática inversa desarrollado, cree un *script* que ejecute las rutinas descritas. Puede realizarlo por medio de MATLAB o Python, según sea su preferencia.
- **Restricciones:**
  - Se debe utilizar las bases en madera y marcadores borrables para la practica, evite realizar escritura en otras superficies o con otros instrumentos.
  - Se debe configurar correctamente la apertura y cierre del gripper para sujetar el marcador, si es necesario agregue al marcador recubrimientos para asegurar la rigidez de la sujeción.
  - Se debe cuidar la integridad física y eléctrica del brazo y demás elementos de l sistema en todo momento.
  - El trazado de letras y figuras se puede hacer en cualquier posición dentro del área de trabajo. Configure la posición, tamaño y orientación de los trazos de forma que todas las rutinas puedan quedar marcadas sobre la superficie sin tener que borrar otros trazos.

**ROS - Rutinas de escritura:**

- Cree una base porta herramienta para el marcador de forma que cuando el brazo tome y suelte el marcador la orientación respecto a la herramienta sea valida para escribir sobre la superficie plana, esta base debe estar sujeta a la tabla de trabajo mediante un acople temporal, no dañe la superficie de la tabla.
- Rutinas:
  - Cargar herramienta: el brazo se desplaza a la base porta herramienta, sujeta el marcador y se ubica en una posición de espera.
  - Espacio de trabajo: el brazo dibuja dos arcos que representan los limites de espacio de trabajo diestro plano sobre la superficie y regresa a una posición de espera.
  - Dibujo de Iniciales: El brazo dibuja al menos dos letras, iniciales de los nombres de los estudiantes, sobre la superficie y retorna a una posición de espera.
  - Dibujo de figuras geométricas: Se dibuja sobre la superficie un triangulo equilátero, una circunferencia y 3 lineas rectas paralelas y regresa a una posición de espera.
  - Dibujo de puntos: El brazo dibuja 5 puntos equidistantes y regresa a una posición de espera.
  - Dibujo figura libre: Se dibuja una figura libre que utilice trazos rectos tanto curvos y regresa a una posición de espera.
  - Descarga de la herramienta: el brazo se desplaza a la base porta herramienta, suelta el marcador y se ubica en una posición de Home.
- Interacción con el usuario:
  - Se debe crear un programa (script) que permita seleccionar que rutina se desea ejecutar, el control de selección se debe hacer mediante comandos o teclas especificas del teclado del computador. Las rutinas de escritura no se deben poder ejecutar si no se ha cargado la herramienta (marcador).
  - Se deben mostrar mensajes al usuario en pantalla (consola) en cada una de las etapas de operación:
    - Estado de la herramienta (cargada o descargada).
    - Posición actual del brazo.
    - Rutina seleccionada.
    - Rutina finalizada.
    - Tiempo de ejecución de la ultima rutina.
- Las demostraciones deben iniciar con el brazo tomando el marcador y deben finalizar con el brazo descargando el marcador en la base.
- Verificación dimensional:
  - Se debe verificar las dimensiones, calidad de trazo, rectitud, radio y homogeneidad de todos los trazos.
  - Escoja e implemente una metodología para medir la precisión de los trazos.
  - Tome imágenes de los trazos y compárelas digitalmente con imágenes de las trayectorias ideales.
  - Analice si el trazo es igual haciendo múltiples rutinas de escritura descargando y cargando la herramienta.

## 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 en RAPID del módulo utilizado para el desarrollo de la práctica.
  - Modelos, fotografías y descripción de la base porta herramienta y el marcador utilizado.
  - Análisis y resultados de la verificación dimensional.
  - Vídeo que contenga el montaje del brazo con todos los elementos de la practica así como la ejecución de todas las rutinas. El vídeo debe describir cada una de las rutinas y se debe demostrar todo el conjunto de rutinas en el orden deseado, el vídeo debe iniciar con la superficie de escritura limpia. El vídeo debe incluir las pruebas de verificación dimensional.
  - Descripción de la solución planteada.
  - Los puntos que requieran implementación de funciones deberán tener comentarios de cómo se utilizan y adjuntar archivos **.m** y **.py** al repositorio. También se deberán adjuntar los demás archivos relevantes que hayan sido creados y utilizados, como **.launch**, **.yaml**, **.urdf**, **.rviz**, entre otros.
  - Se deberá incluir vídeos en los que se muestre el funcionamiento de los *scripts* generados en Matlab o Python para cada una de las aplicaciones propuestas con ROS.
3. **Fecha de entrega:** segun actividad en Moodle.