

# ROBÓTICA

## PRÁCTICA N° 1

### 1. TEMA

#### INTRODUCCIÓN A COPPELIASIM

### 2. OBJETIVOS

- 2.1. Determinar los principales elementos y aplicaciones del ambiente de programación del software CoppeliaSim.
- 2.2. Realizar la comunicación entre el software CoppeliaSim y el software Matlab para controlar los movimientos de un robot móvil.

### 3. MARCO TEÓRICO

CoppeliaSim es un software de simulación en 3D, que cuenta con un ambiente de desarrollo integrado, que se basa en una arquitectura de control distribuido, lo cual hace que este software sea una herramienta ideal para aplicaciones multi robot. CoppeliaSim es utilizado para el desarrollo rápido de algoritmos, simulaciones de automatización de procesos de fabricación, prototipado rápido y verificación, educación relacionada con robótica, entre otros.

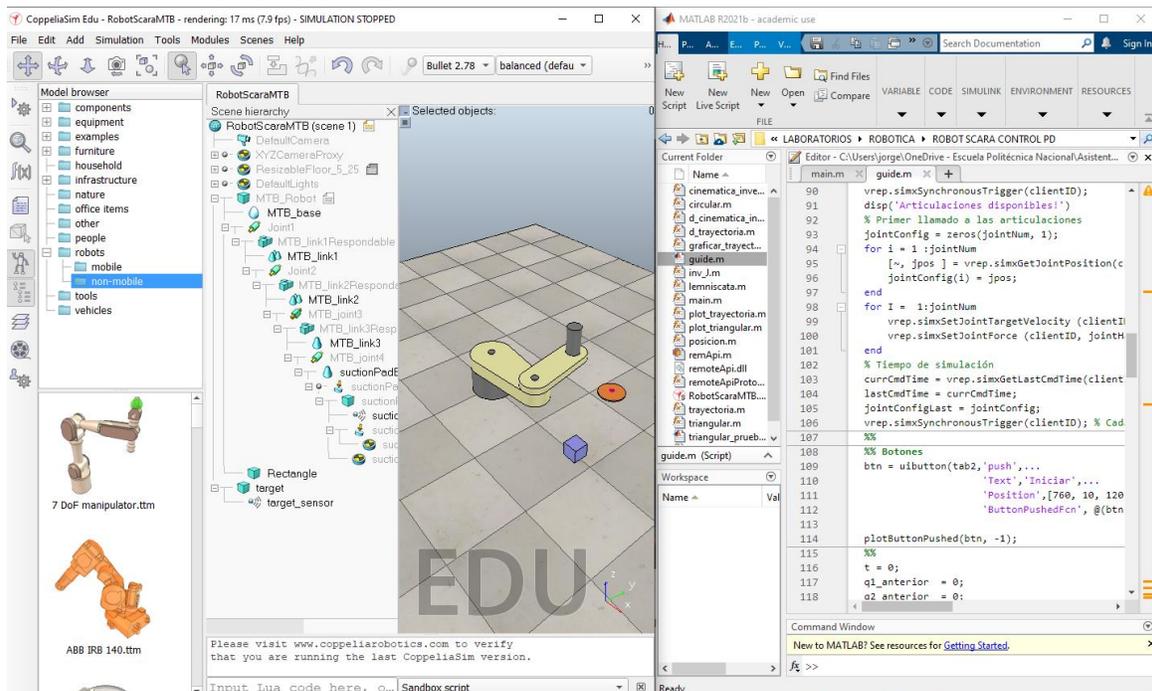


Fig. 3.1. Interfaz de CoppeliaSim y Matlab en comunicación por API remota.

Cada objeto o modelo de CoppeliaSim puede ser controlado de manera individual mediante un código embebido, un plugin, un nodo ROS, un cliente remoto API, o por una solución personalizada. Los controladores pueden ser escritos en C/C++, Python, Java, Lua, Matlab or Octave. Es posible crear modelos personalizados utilizando una amplia variedad de actuadores y sensores incorporados, articulaciones, formas y mallas, scripts, etc.

Como se mencionó anteriormente, los objetos de CoppeliaSim pueden ser controlados por una Interfaz de Programación de Aplicaciones (API). La API remota permite realizar la comunicación entre CoppeliaSim y una aplicación externa (es decir, una aplicación que se ejecuta en un proceso diferente, o en una máquina diferente) y es multiplataforma. Para manipular CoppeliaSim desde Matlab, es necesario que Matlab se conecte como un cliente remoto API.

**Nota:** en el MANUAL DE USO DE COPPELIASIM se describe el proceso de instalación de CoppeliaSim y la instalación de las librerías para la comunicación con Matlab.

## 4. TRABAJO PREPARATORIO

4.1. Consultar las características del robot Pioneer P3DX:

- Datos técnicos de sensores y actuadores
- Arquitectura y dimensiones.
- Comandos base para establecer la comunicación con matlab

4.2. Tomando en cuenta el MANUAL DE USO DE COPPELIASIM descrito en el MARCO TEÓRICO, crear un espacio de trabajo denominado "Coppelia\_workspace" e incluir las librerías descritas en el manual.

4.3. Consultar la estructura base del script en Matlab con los comandos que permiten establecer comunicación con CoppeliaSim

4.4. Para el robot Pioneer P3DX, consulte los comandos y funciones API remotas entre MATLAB y CoppeliaSim que permitan:

- Registrar y analizar sensores de posición, velocidad y orientación virtuales
- Enviar órdenes a los motores

4.5. Para el robot Pioneer P3DX, realice un script en Matlab con el uso de funciones API remotas que permitan:

- Iniciar la comunicación de forma sincrónica con CoppeliaSim.
  - Enviar comandos de velocidad lineal y angular a la rueda derecha y rueda izquierda
  - Leer y graficar la posición del robot
-

## 5. EQUIPO Y MATERIALES

### Equipo

- Computador

### Software

- CoppeliaSim
- Matlab

## 6. PROCEDIMIENTO

- 6.1. Exposición por parte del instructor sobre la configuración del entorno de CoppeliaSim, la configuración y el procedimiento para establecer conexión con Matlab.
- 6.2. Desde el explorador de modelos de CoppeliaSim, insertar el **robot Pioneer P3DX** en el área de simulación y modificar la posición de este mediante las propiedades de los objetos.
- 6.3. Para el **robot Pioneer P3DX**, programar un script en Matlab que permita:
  - Establecer comunicación de forma sincrónica con CoppeliaSim.
  - Modificar la velocidad de la rueda derecha y rueda izquierda
  - Leer el sensor de presencia del robot en CoppeliaSim mediante MATLAB.
  - Graficar las lecturas de velocidad y posición del robot.

## 7. INFORME

- 7.1. Realice un análisis detallado sobre cómo afectan los comandos de velocidad al comportamiento del robot en la simulación.
  - 7.2. Realice una explicación detallada de cómo controlar el tiempo de simulación del robot, es decir, como parametrizar el tiempo de movimiento del robot mediante bucles de lazo for en Matlab.
  - 7.3. Presentar las gráficas de posición y velocidad obtenidas en la simulación, analice el comportamiento de estas.
  - 7.4. Utilizando la información de los sensores implemente un control de posición en lazo cerrado desde el punto P(0,0) hasta el punto P(1,1). Presente el script y las gráficas de su solución propuesta.
  - 7.5. Conclusiones (Individuales)
  - 7.6. Recomendaciones (Individuales)
  - 7.7. Bibliografía
-

## 8. BIBLIOGRAFÍA

Coppelia Robotics. "Robot simulator CoppeliaSim: create, compose, simulate, any robot - Coppelia Robotics". Robot simulator CoppeliaSim: create, compose, simulate, any robot - Coppelia Robotics. <https://www.coppeliarobotics.com/>

Coppelia Robotics. "Remote API functions (Matlab)". Robot simulator CoppeliaSim: create, compose, simulate, any robot - Coppelia Robotics. <https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm>

**Elaborado por:** Ing. Jorge Luis Vega y MSc. Diego Maldonado

**Revisado por:** Dr. Paulo Leica y Dr. Danilo Chávez

---