

Nicolás Rincón Sánchez – 202021963

Álex Ariza Herrera – 201920733

David Mauricio Montealegre Rojas – 201915177

Juan Miguel Yate Alcalá – 201914901

IELE3338 – ROBÓTICA

Taller #1 – Introducción a ROS TurtleBot2

PARTE 1

En la figura 1, se presenta el grafo de ROS con la relación entre nodos y tópicos usados en la solución del punto 1 del taller. En este caso, solo son necesarios dos nodos, “turtle_bot_teleop” y “sim_ros2_interface” y un tópico “turtlebot_cmdVel”. El nodo “turtle_bot_teleop” publica la información de la velocidad lineal y angular (introducidas por parámetro preguntándole al usuario) y si hay alguna tecla presionada, indicar hacia dónde debe moverse el robot. Esta publicación se realiza mediante el tópico “turtlebot_cmdVel” y para que la información llegue al robot en la escena de Coppelia, el nodo “sim_ros2_interface” se suscribe al tópico. De esta manera se logra la comunicación entre los dos nodos y la transferencia de información.

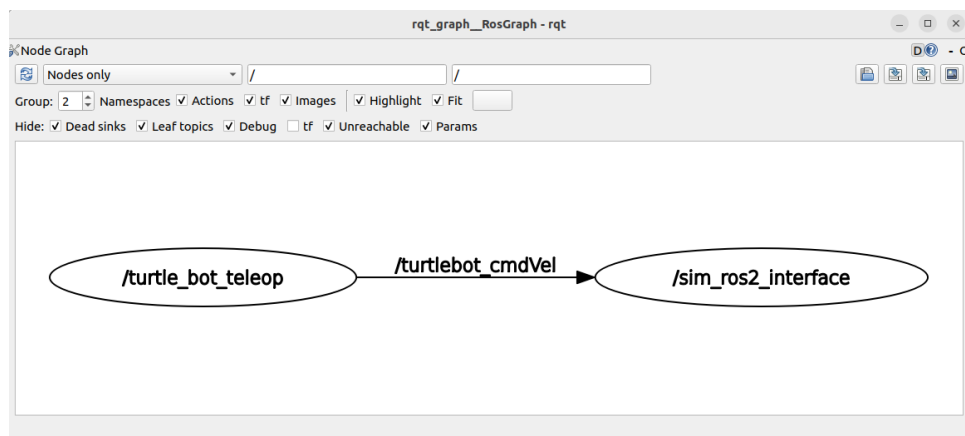


Figura 1. Grafo de ROS2 que muestra los nodos y tópicos usados para la solución del punto 1 del taller.

En la figura 2 se presenta un diagrama de flujo que explica resumidamente la implementación de la solución del punto 1 del taller. Cuando se realiza la ejecución inicial del paquete, en la terminal se introducen los valores de velocidad lineal y angular deseados. Luego, se pueden accionar las teclas para que el robot de la escena del taller en Coppelia se pueda mover en la dirección deseada. El nodo “turtle_bot_teleop” publica al tópico “turtlebot_cmdVel” y el nodo “sim_ros2_interface” se suscribe a este tópico.

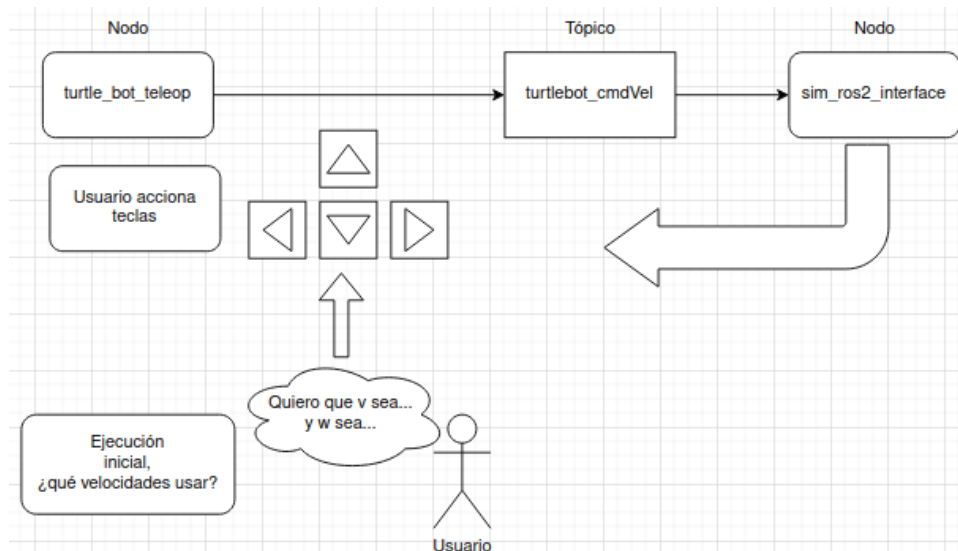


Figura 2. Diagrama de flujo que presenta la implementación de la solución del punto 1 del taller.

Enlace del vídeo de demostración: [EJECUCION PUNTO1](#)

Enlace de los grafos y diagrama de flujo: <https://youtu.be/GI5scRsCgIA>

PARTE 2

Para la Parte 2 del taller, se realizó la implementación de una interfaz gráfica para que interactúe con el usuario y Coppelia desde ROS. En la Figura 3, se puede apreciar el diagrama de bloques general de la implementación. En general, el usuario va a continuar a partir de lo que ha sido ejecutado en el Punto 1. Por medio de las teclas de las flechas va a moverse el robot en Coppelia. Sin embargo, ahora se le adiciona la implementación del nodo `turtle_bot_interface`. En particular, el nodo original `turtle_bot_teleop` se comunicará con Coppelia a través de la publicación en el tópico `turtlebot_cmdVel` y Coppelia a su vez publicará en el tópico `turtlebot_position`. Luego, el nodo `turtle_bot_interface` va a suscribirse al tópico `turtlebot_position` y así se podrá visualizar en la gráfica la posición en tiempo real.

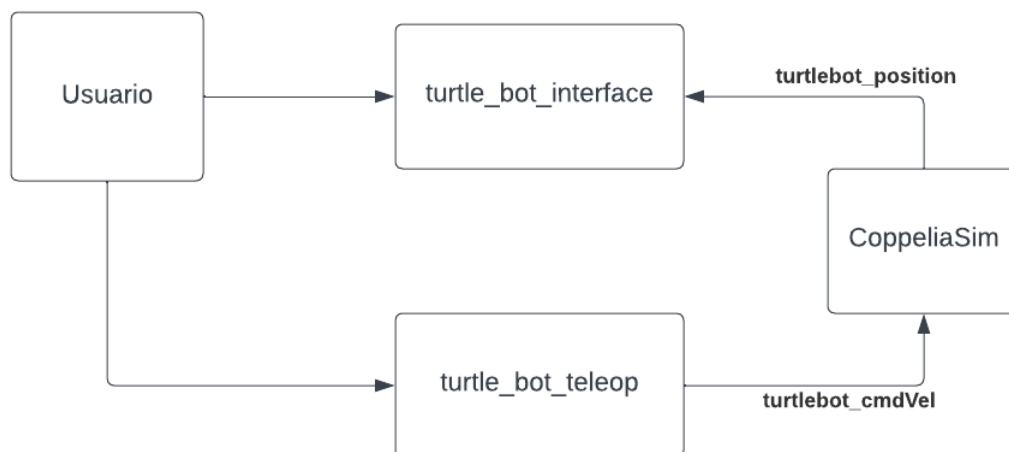


Figura 3. Diagrama de bloques que presenta la implementación del punto 2 del taller.

Por medio del comando `rqt_graph` se generó la visualización simplificada del nodo de ROS. De esta manera, se puede apreciar que para el punto 2 hay comunicación entre la interfaz de Coppelia y el nodo `turtle_bot_interface` a través de la publicación y suscripción al tópico de posición.

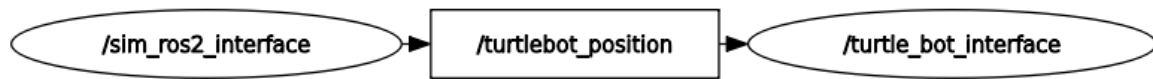


Figura 4. Grafo de ROS2 que muestra los nodos y tópicos usados para la solución del punto 2 del taller.

Enlace del video de demostración: <https://youtu.be/jWWGBa1Sw3A>

PARTE 3

Para la Parte 3 del taller, a partir de la implementación ya hecha anteriormente se le agregó la función al nodo interfaz de guardar la partida como archivo txt.

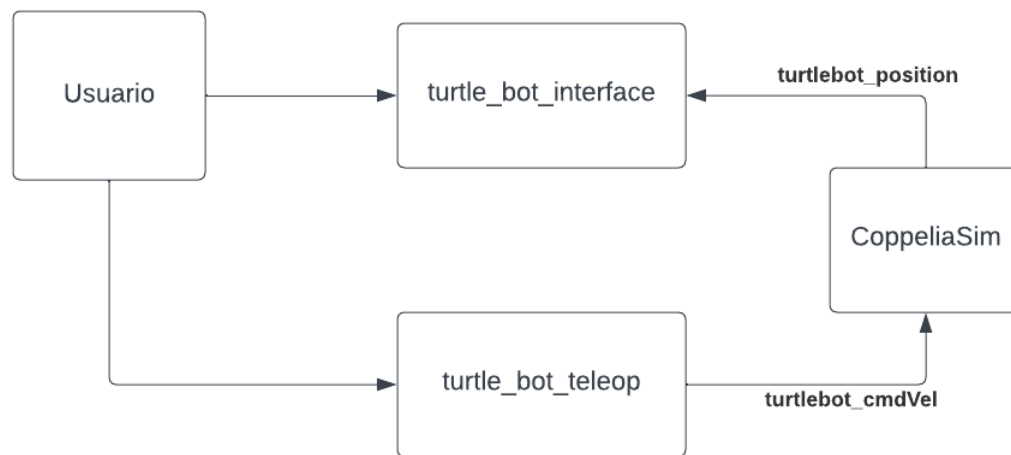


Figura 3. Diagrama de bloques que presenta la implementación del punto 2 del taller.

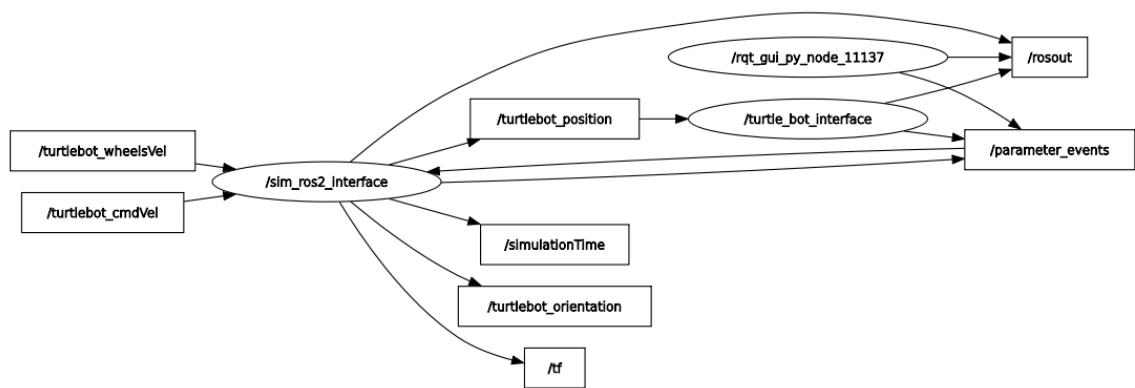


Figura 4. Grafo de ROS2 que muestra los nodos y tópicos usados para la solución del punto 2 del taller.

PARTE 4