



Práctica 4 - Creación de Mapas

Robótica de Servicios
Máster en Robótica e Inteligencia Artificial

Resumen

Esta práctica está asociada con la instalación y configuración de herramientas de navegación en ROS 2, específicamente el paquete Navigation 2 y el slam_toolbox [1] para la creación de mapas mediante SLAM. Los objetivos planteados son:

- Familiarizarse con la instalación y configuración de los paquetes de navegación en ROS 2.
- Configurar un entorno de simulación con Turtlebot3 en Gazebo.
- Explorar el entorno y generar un mapa utilizando el paquete slam_toolbox.
- Guardar el mapa generado en formato adecuado.

Requisitos

Será necesario disponer de un entorno de trabajo base. La recomendación por simplicidad es la siguiente:

- Robot Operating System 2 (ROS 2)
 - Ubuntu 22.04
 - ROS 2 Humble y Gazebo

Workspace en ROS

Utilizando el repositorio de ejemplos: https://github.com/fjrodl/navigation2_ignition_gazebo_turtlebot3 y la guía proporcionada, sigue los pasos indicados para completar esta práctica.

Ejercicios

1. Instalación de paquetes:

- Instala los paquetes Navigation 2 y slam_toolbox con los comandos:

```
1 sudo apt install ros-humble-navigation2 ros-humble-nav2-bringup
2 sudo apt install ros-humble-turtlebot3*
3 sudo apt install ros-humble-slam-toolbox
4
```

2. Inicio de la pila ROS 2 para Turtlebot3:

- Especifica el modelo de Turtlebot3 (ROS_LOCALHOST_ONLY=1 TURTLEBOT3_MODEL=waffle):

```
1 export TURTLEBOT3_MODEL=waffle
2
```

- Inicia la simulación en Gazebo:

```
1 ros2 launch turtlebot3 simulation.launch.py
2
```

3. Verificación del tópico /scan:

- Lista los tópicos activos para verificar la disponibilidad del tópico /scan:

```
1 ros2 topic list
2
```

- Comprueba el tipo del tópico /scan:

```
1 ros2 topic info /scan
2
```

4. Inicio de slam_toolbox y Navigation 2:

- Abre cuatro terminales e inicia los siguientes comandos:

- Terminal 1: Inicia la simulación en Gazebo:

```
1 ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py
2
```

- Terminal 2: Inicia el stack de Navigation 2:

```
1 ros2 launch nav2_bringup navigation_launch.py use_sim_time:=True
2
```

- Terminal 3: Inicia slam_toolbox:

```
1 ros2 launch slam_toolbox online_async_launch.py use_sim_time:=True
2
```

- Terminal 4: Inicia RViz con la configuración predefinida:

```
1 ros2 run rviz2 rviz2 -d /opt/ros/humble/share/nav2_bringup/rviz/
  nav2_default_view.rviz
2
```

5. Generación de un mapa con `slam_toolbox`:

- Controla el robot para que explore el entorno y genere el mapa en RViz.

```
1 ros2 run teleop_twist_keyboard teleop_twist_keyboard
2
```

6. Guardar el mapa generado:

- Guarda el mapa ejecutando:

```
1 ros2 run nav2_map_server map_saver_cli -f my_map
2
```

- Esto generará los archivos `my_map.yaml` (metadatos) y `my_map.pgm` (imagen del mapa).

Preguntas

- a) ¿Cuál es la función del paquete `slam_toolbox` en ROS 2 y cómo contribuye a la generación de mapas en un entorno simulado?
- b) Describe el propósito del comando `ros2 launch` y explica cómo se utiliza para iniciar la simulación de Turtlebot3 en Gazebo.
- c) ¿Qué tipo de información proporciona el tópico `/scan` y por qué es importante para el proceso de SLAM?
- d) ¿Cómo se guardan los mapas generados en ROS 2 y qué archivos se generan en el proceso? Explica el contenido de cada archivo.

Referencias

- [1] Steve Macenski and Ivona Jambrecic. Slam toolbox: Slam for the dynamic world. *Journal of Open Source Software*, 6(61):2783, 2021.