CS-475 Assignment 0
Introduction to ROS 2

Manos Papadakis
manospapad@ics.forth.gr

Release date: 12/02/25
Deadline: 26/02/25

1 Introduction

This assignment is a tutorial on how to set up and use the Robot Operating System **ROS 2 Humble**. You will install ROS 2 Humble and then add the TurtleBot3 robot to the built-in simulated environment, *Gazebo*. The goal is to get familiar with the core concepts and tools of ROS 2.

2 Installation

ROS 2 Humble is designed to run on Ubuntu 22.04 (Jammy Jellyfish). Therefore, you need to install Ubuntu 22.04 first. Choose one of the following options:

1. (Recommended) Install Ubuntu 22.04 as your main operating system (or dual boot). Follow these instructions:
   https://ubuntu.com/tutorials/install-ubuntu-desktop#1-overview
2. (OK) Dual boot your main operating system with Ubuntu. Note, this is a bit more challenging. Follow this tutorial: https://www.youtube.com/watch?v=QKn5U2esuRk
3. (OK) Install Ubuntu 22.04 in WSL (preferred) or a VM. Download the Desktop image (.iso) from https://releases.ubuntu.com/jammy/ and then follow a tutorial for VirtualBox or WSL setup.

Now that you have installed Ubuntu 22.04, follow the ROS instructions to install ROS 2 Humble:
https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html (Install the `ros-humble-desktop` version).

After the installation, it is highly suggested to practice on the beginner tutorials from: https://docs.ros.org/en/humble/Tutorials.html, especially the "Beginner: Client Libraries" section.

3 Setup TurtleBot3

Assuming that you have installed ROS 2 and followed some of the tutorials, you should now be familiar with the file structure and key concepts of ROS 2 (nodes, topics, interfaces, etc.).

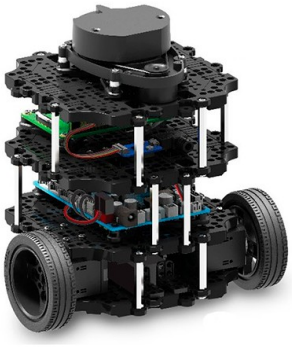

Figure 1: TurtleBot3

Follow the instructions below in order to create a fresh workspace and install the TurtleBot3 simulation packages.

Use a terminal to navigate to your home directory and then:

```
$ mkdir -p 475_ws/src
$ cd 475_ws/src
```

If you do not have *git* installed, you can use:

```
$ sudo apt install git
```

Clone the TurtleBot3 repositories to your workspace (using the Humble branch):

```
$ git clone -b humble
   https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
```

```
$ git clone -b humble
   https://github.com/ROBOTIS-GIT/turtlebot3.git
```

Install dependencies:
`$ sudo apt install ros-humble-turtlebot3-msgs ros-humble-gazebo-ros-pkgs`

Compile the project (Note: `catkin_make` is replaced by `colcon` in ROS 2):

```
$ cd ..
$ colcon build --symlink-install
```

In order for your system to see the new ROS workspace, you need to source the `install/setup.bash` file every time you open a new terminal window by using:
`$ source /home/<usr_name>/475_ws/install/setup.bash`

I suggest adding this line at the bottom of your `.bashrc` so you never have to deal with it again. Additionally, you **must** specify the robot model in ROS 2. The `.bashrc` file is located in your home directory; modify it using a text editor:

```
$ sudo gedit ~/.bashrc
```

Add the following lines at the end of the file:

```
source /opt/ros/humble/setup.bash
source /home/<usr_name>/475_ws/install/setup.bash
export TURTLEBOT3_MODEL=burger
```

Save and exit! Open a new terminal window. Now that everything is in place, you can spawn the robot inside the simulation environment using `ros2 launch`:

```
$ ros2 launch turtlebot3_gazebo empty_world.launch.py
```

You should be able to see the TurtleBot3 located at the origin (0,0,0) inside an empty world. In general, you can use the `ros2 launch` command to run launch files. The general syntax is:
`$ ros2 launch <package_name> <launch_file_name>`

I suggest navigating and testing different environments

(e.g., `turtlebot3_house.launch.py`). Note that the more complicated the environment, the longer you may have to wait for it to load.

4 Create a Package

Finally, the only thing that remains is for you to create a package and then a node that subscribes and publishes to a topic. Navigate to the `src/` folder of your workspace and create a Python package:

```
$ ros2 pkg create --build-type ament_python <package_name> --dependencies rclpy nav_msgs std
```

Navigate to your new package directory: `src/<package_name>/<package_name>/`. Create a file named `<name>.py` (where `<name>` is your registration number). This will be your node script.

In ROS 2 Python packages, we do not use `CMakeLists.txt` for Python scripts. Instead, we define entry points in `setup.py`. Open `setup.py` in your package root and locate the `entry_points` dictionary. Modify it to look like this:

```
entry_points={
    'console_scripts': [
        '<node_name> =
           <package_name>.<script_name>:main',
    ],
},
```

Replace `<node_name>` with the name you want to use to run the node, and `<script_name>` with the filename (without .py) you created earlier. Ensure your Python script has a `main` function.

5 Compute Distance

The purpose of your node is to extract the necessary information from the `/odom` topic and calculate the 2D distance between the robot and the world frame (0,0,0). Then, you will need to publish the calculated distance to another topic with a name of your choice.

After writing your code, go back to the root of your workspace, run `colcon build`, and source your workspace again. To run your node use:
`$ ros2 run <package_name> <node_name>`

Move the robot around by using the teleop node and then subscribe to your

created topic to see if you get the expected results. Open a new terminal and run:
`$ ros2 run turtlebot3_teleop teleop_keyboard`
(Follow the prompt instructions on how to move it)

6 Tips

1. Use: `ros2 topic type <topic_name>` to find out what type of interface a topic uses.
2. Run `$ ros2 interface show <type>` to easily see the data structure of the message (e.g., `ros2 interface show std_msgs/msg/Float32`).
3. To see the data being published on a topic in real-time, use:
   `ros2 topic echo <topic_name>`

7 Submission

Send your node script (`<registration number>.py`) attached via email to: **manospapad@ics.forth.gr** Don't forget to mention your name and registration number. The deadline is **26/02/25**.