Introduction to Mobile Robotics

Practical Work 4: Play with Gazebo

Zhi Yan

ENSTA, Institut Polytechnique de Paris

1 Objective

The goal of this session is to introduce Gazebo, the primary 3D robotics simulator used with ROS 2. You will learn how to launch a simulated world, spawn a robot, and interact with it using the ROS 2 command-line tools you are already familiar with. This is a crucial step up from the 2D Turtlesim.

2 Prerequisites

- A working ROS 2 Humble installation with Gazebo. The `ros-humble-desktop` install includes it.
- A decent graphics card is recommended, as Gazebo can be computationally intensive.
- We will use the standard TurtleBot3 robot model, a common platform for learning ROS. Let's install its simulation package. Open a terminal and run:

```
sudo apt update
sudo apt install ros-humble-turtlebot3-gazebo
```

3 What is Gazebo?

Gazebo is a powerful 3D rigid-body simulator. Unlike the simple 2D `turtlesim`, Gazebo provides:

- **Physics engine:** Simulates gravity, friction, collisions, and forces.
- **3D environments:** Allows you to build complex worlds from simple shapes or import detailed models (meshes).
- **Sensor simulation:** Can generate realistic data from simulated sensors like LiDAR, cameras, and IMUs.
- **Robot models:** Robots are defined in detailed URDF or SDF files, specifying their physical properties (links, joints, mass).

It is the standard tool for developing and testing robotics algorithms before deploying them on a real robot.

4 Launching a Robot in Gazebo

1. Open a new terminal. We will use a ROS 2 launch file from the package we just installed to start Gazebo and spawn a TurtleBot3.

```
# First, tell the system which TurtleBot3 model we want to use (burger
    is the simplest)
export TURTLEBOT3_MODEL=burger

# Now, launch the simulation
ros2 launch turtlebot3_gazebo empty_world.launch.py
```

   This may take some time to start up, especially the first time. A new window, the Gazebo client, should appear, showing a TurtleBot3 robot in an empty world.

2. Explore the Gazebo GUI:

   - **Left mouse button:** Orbit the camera.
   - **Mouse scroll wheel:** Zoom in and out.
   - **Shift + left mouse button:** Pan the camera.

   On the left panel, you can see the models currently in the world. On the top bar, you can find tools to insert simple shapes (spheres, cubes) or manipulate objects.

5 Controlling the Robot with ROS 2

Even though the simulator is more complex, the ROS 2 interaction principles remain the same.

1. **Open a new terminal (Terminal 2).** Remember to source your main ROS 2 setup file if it's not in your `.bashrc`:

```
source /opt/ros/humble/setup.bash
```

2. **Introspection:** Let's see what topics are available now that Gazebo is running with a robot. In Terminal 2, run:

```
ros2 topic list
```

   You will see many topics! The robot's LiDAR scanner publishes on `/scan`, its camera on `/camera/image_raw`, and, most importantly for us now, its velocity command topic is `/cmd_vel`.

3. **Publishing manually:** Just like with turtlesim, we can publish a command to make the robot move. Let's make it drive forward. In Terminal 2, run:

```
# Please first check the message type of your /cmd_vel.
ros2 topic pub --once /cmd_vel geometry_msgs/msg/TwistStamped "{twist:
    {linear: {x: 0.3, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z:
   0.0}}}"
# Or
ros2 topic pub --once /cmd_vel geometry_msgs/msg/Twist "{linear: {x:
   0.3, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
```

   You should be able to see the robot moving forward in the Gazebo window.

4. **Keyboard control:** Manually publishing is tedious. Let's use a teleoperation node. The TurtleBot3 package provides one. In Terminal 2, run:

```
# Make sure you have set the model name in this terminal too!
export TURTLEBOT3_MODEL=burger

ros2 run turtlebot3_teleop teleop_keyboard
```

   Make sure Terminal 2 is the active window. You can now use the 'w', 'a', 's', 'd', 'x' keys to drive the robot around in the simulator.

5. **Visualizing sensor data:** Let's "see" what the robot's laser scanner is seeing. Open a **new terminal (Terminal 3)** and echo the `/scan` topic:

```
ros2 topic echo /scan
```

   You will see a continuous stream of data representing the laser scan ranges. If you insert a cube in Gazebo and drive the robot near it, you will see the numbers in the 'ranges' array change.

6 Visualizing the Graph

Finally, let's see how our new system looks. In a new terminal (or in Terminal 3 after stopping the echo with Ctrl+C), run `rqt_graph`:

```
rqt_graph
```

You will now see a more complex graph. However, you should be able to identify the core connection: the `/teleop_keyboard` node is publishing `geometry_msgs/msg/TwistStamped` messages[1] on the `/cmd_vel` topic, and the Gazebo node (which contains the robot model plugin) is subscribing to it.

This concludes the practical session. You have successfully launched, controlled, and inspected a simulated robot in Gazebo!

[1] To actually view the message type, you need to run `rqt` instead of `rqt_graph`, then click `Plugins` → `Topics` → `Topic Monitor`.