

Robot Practical Course Bachelor Assignment #2

In this assignment, you are introduced to the robotic platform “turtlebot” which you will program in the rest of this course. You will get to know its sensors and their data visualization. In a second step you will write a program to move the Turtlebot and use the sensor data to stop before it collides with obstacles in the environment.

Real robots require to be handled with care.
Be prepared to rescue the robot from hitting obstacles.

Task 2.1 Starting the Turtle and ROS 2: We now try to get a connection to the turtlebot and start the ROS 2 network on it.

2.1.1: To start the turtlebot, place it onto the charging station and wait for it to boot up. Once booted you will hear a sound indicating that the turtlebot is ready to be used. This can take a few minutes, be patient.

2.1.2: Once the turtlebot is ready, you can connect your local ros network to the turtlebot’s ros network using the ros2 zenoh bridge. This will allow you to see the topics and services of the turtlebot on your local computer.

We have again prepared a shorthand command to do this for you. Open a terminal on your local computer and run the following command:

```
just connect <HOSTNAME>
```

Replace <HOSTNAME> with the hostname of the turtlebot you want to connect to (e.g. donny).

If everything works correctly, you should see a lot of topics from the turtlebot in your local ros network. You can check this by running the following command:

```
ros2 topic list
```

2.1.3: Start the visualization tool RViz with some basic configuration on your local computer. Run the following command in a terminal, change the fixed frame to `odom` and add the camera image topic to the display:

```
ros2 launch turtlebot4_viz view_robot.launch.py
```

2.1.4: Drive the turtlebot around using the teleop node and your keyboard. Open a new terminal and run the following command:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

Note: The robot can only be controlled the terminal is in focus. If you click on another window, the robot will stop moving.

2.1.5: Try to open a terminal on the robot. You can login to the robot via SSH with the username `ubuntu` and the password `turtlebot4` on the hosts `donny`, `leo`, `mikey`, and `raph` from any TAMS pool computer.

```
ssh ubuntu@<HOSTNAME>
```

Replace `<HOSTNAME>` with the hostname of the turtlebot you want to connect to (e.g. donny). Now you can try to run the teleoperation node on the turtlebot itself.

Task 2.2 Lidar Listener: Write a listener node to read the laser scan data

2.2.1: Find out the identifier and the message type of the Lidar topic.

2.2.2: Look up and read the ROS tutorial code for a simple subscriber. It is similar to the publisher tutorial, but you will use a subscriber instead of a publisher.

<https://docs.ros.org/en/humble/Tutorials/Beginner-Client-Libraries/Writing-A-Simple-Py-Publisher-And-Subscriber.html#write-the-subscriber-node>

2.2.3: We have prepared a command that copies your code to the turtlebot and builds it there. You can simply use it to deploy your code to the turtlebot. Open a terminal in the development environment and run the following command:

```
just deploy <HOSTNAME>
```

Replace `<HOSTNAME>` with the hostname of the turtlebot you want to deploy to (e.g. donny).

2.2.4: Use your new knowledge to write a listener node that reads the laser scan data from the turtlebot and prints the closest obstacle distance to the console. Use the deployment command to deploy your code to the turtlebot and run it by opening a terminal on the turtlebot using SSH to execute the familiar `ros run` command there.

Task 2.3 Simple Movement: Write a node to move the Turtlebot.

2.3.1: Extend your sensor node to move the Turtle bot a bit in each direction.

2.3.2: Test your node on the robot. Make sure there is enough space around the robot.

Never leave the robot unattended!

Task 2.4 Detect obstacles: Write a program to move the TurtleBot and stop in front of an obstacle.

2.4.1: Use your code from the previous two tasks and extract the distance to an object directly to the front of the robot.

2.4.2: Now let the robot move forward until it detects an obstacle closer than 1 m. Use a soft obstacle to test your code.

Task 2.5 Optional: Wall following: Write a program to move the TurtleBot along a wall to explore the environment.