Robot Practical Course Master

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 must be handled with care.
Be prepared to rescue the robot from hitting obstacles.

Task 2.1 Starting the Turtle and ROS: You can log into the robot via SSH with the username `pracm2025` on the hosts `donny`, `leo`, `mikey`, and `raph` from any TAMS pool computer. You will find a prebuilt ROS workspace `~/ros` on each robot that you should use to develop your own ROS packages.

2.1.1: Boot the turtle bot & its laptop. Log in to the system.

2.1.2: Start ROS and the nodes for a basic setup to use the mobile base and the camera. Good tutorials are available on the Turtlebot Wiki page:

```
http://wiki.ros.org/Robots/TurtleBot
```

To account for custom changes on TAMS' turtlebots (i.e. an additional laser scanner), you have to start TAMS-specific launch files which are modified for our setup. To bring up the turtle bot, don't run any launch file from the package `turtlebot_bringup`. Instead, run:

```
roslaunch tams_turtlebot_bringup tams_turtlebot.launch
```

before the previous command making sure to source the pre-build workspace:

```
source /home/demo/ws_ros/devel/setup.bash
```

You can also add this line to `~/.bashrc`.

2.1.3: Start the RViz visualization tool on your local computer. Try to get a live camera image and drive manually with the `turtlebot_teleop` node. Visualize the point cloud provided by the Kinect sensor. Use the 3D visualization tutorial to read the data provided by the Kinect sensor:

```
http://wiki.ros.org/turtlebot/Tutorials/indigo/3D%20Visualisation
```

Make sure to correctly export the `ROS_MASTER_URI` on your desktop computer before starting the RViz visualization tool.

Task 2.2 Kinect Listener: Write a listener node to read the Kinect data

2.2.1: Find the identifier and message type of the Kinect topic.

2.2.2: Look up the ROS tutorial for a simple subscriber. There are two for C/C++ and one for Python. You can choose the language you want. Make sure that you are looking for the right version of ROS. You used this tutorial the last time for the publisher.

```
C/C++: http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber(c++)
Python: http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber(python)
```

2.2.3: On the TAMS website you will find the `deploy.sh` script. Make sure to set the `ROS_MASTER_URI` and source your workspace. The script copies your package to the robot and runs `catkin build` on the robot.

```
./deploy.sh
```

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

2.3.1: Extend your sensor node to move the turtle bot.

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 turtle bot 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 Kinect.

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 turtle bot along a wall to explore the environment.