Task Sheet 8

Universität Konstanz

Creating a world model on the TurtleBot 4

Deadline 10:00am June 21, 2024

Review on June 25 & 26, 2024

Lecture: *Advanced Autonomous Robotics*, Summer Term 2024

Lecturer: Prof. Dr.-Ing. Heiko Hamann

Tutor: Jonas Kuckling & Paolo Leopardi

In this exercise sheet, you will learn how to create your own world model on the TurtleBot4.

Task 8.1 Collecting sensor data (*preparation at home*)

You will use the maze environment to collect LiDAR data from the TurtleBot. However, there is a small bug in the current `.sdf` file that needs to be fixed before the LiDAR will work.

1. Check the installation path of the `turtlebot4_ignition_bringup` package:
   ```
   $ ros2 pkg prefix turtlebot4_ignition_bringup
   ```
2. Change directory to the worlds directory in the installation of the package:
   ```
   $ cd [/path/to/package]/share/turtlebot4_ignition_bringup/worlds,
   ```
   where `[/path/to/package]` needs to be replaced with the path that the first command returned.
3. Edit the file `maze.sdf` and find the line `<plugin name=’ignition::gazebo::systems::Contact’ filename=’ignition-gazebo-contact-system’ />`.
4. Below that line add the following text:
   ```
   <plugin filename="libignition-gazebo-sensors-system.so" name="ignition::gazebo::systems::Sensors">
   <render_engine>ogre</render_engine>
   </plugin>
   ```
5. Save the file and exit the editor.
6. Run the simulation in Gazebo:
   ```
   $ ros2 launch turtlebot4_ignition_bringup turtlebot4_ignition.launch.py
   world:=maze
   ```

NOTE: Several students in the past have reported issues with the simulation and especially the LiDAR in simulation. While we strongly encourage you to collect the data in simulation, we will also provide a ROSbag with prerecorded LiDAR data that you can use for the next step if you cannot run the simulation.

Task 8.2 Building a line map (*preparation at home*)

Using the simulation environment, write a node that subscribes to the topic `/scan` to build a line map of the environment. Follow the workflow of building a world model (see Chapter 6, slide 27).

a) sensor calibration: The odometry and LiDAR are already pre-calibrated on the TurtleBot4 and in the simulation. However, think about what could go wrong if you were to build your own robot.

b) data collection: Move your robot to an interesting position in the environment and use the data from the `/scan` topic to collect the LiDAR data while the robot is stationary. Use this LiDAR scan to build a line map of the surroundings of the robot, following the procedure described on slide 11. **NOTE:** In order to limit the computational power that your node uses, you can decide to process the LiDAR messages only at a lower frequency than it is published (every second, every 2 seconds, or every 5 seconds, for example).

c) model validation: Based on a screenshot of the simulation and your corresponding line map, decide how accurate your world model is. Do you observe any artefacts in your map that are not present in the simulation? How do you handle discontinuities in the scan?

Upload your code, a documentation on how to execute it, as well as the results for the line map.

BONUS: How would you handle a moving robot? Can you update your map while moving continuously? For simplicity, you can use the `/odom` topic to get the movement of the robot. How would you handle the map updates if you did not have access to the `/odom` topic?

Task 8.3 Creating a world model (*practical tutorial*)

In the practical session, connect to your TurtleBot4 and run your node to build a world model in the real environment. Does it still work? What challenges are you facing?