ROBT502 – ROBOT PERCEPTION AND VISION

Laboratory Report 4

Z. Kuangaliyev, A. Amangeldi, K. Kantoreyeva

February, 2024

---

1. Introduction

This report details a lab session focused on Gazebo simulation environment and understanding the process of setting up a simulation, including creating and manipulating a custom labyrinth model for TurtleBot3.

2. Preparing Labyrinth Model

The initial step involved cloning the TurtleBot3 simulation package into our catkin workspace and compiling it using '**catkin_make**'. We then proceeded to create a custom world, which includes the labyrinth model created in our previous laboratory session. Inside the models directory, we have placed an '**.stl**' file and prepared a '**model.sdf**' file that will define the model's appearance and properties. The models' location as a path to the '**.stl**' file had to be included inside the '**mesh**' tag, which is a geometric object and inherits collision properties (see Code 1).

```
<geometry>
        <mesh>
                <uri> model://labyrinth_model/meshes/Labyrinth.stl </uri>
                <scale> 0.001 0.001 0.001 </scale>
        </mesh>
</geometry>
```

Code 1. Part of model.sdf code structure

Apart from '**model.stf**' we have created '**model.config**' that contains metadata like model name, model description, sdf version and model author.

3. Custom World and Launch Files

A custom world file, '**labyrinth.world**', was created to include the labyrinth model along with essential elements like lightning (sun) and ground plane for simulation environment. The physics engine "Open Dynamic Engine" is used for physical calculations in our simulation. The settings within '**physics**' tag, control how the interactions are simulated, affecting everything from motion to collisions within the virtual environment. Finally, using the '**include**' tag we have imported our labyrinth model to the environment.

```
<include>
    <uri> model://labyrinth_model/meshes/Labyrinth.stl </uri>
    <scale> 0.001 0.001 0.001 </scale>
</include>
```

Code 2. Part of labyrinth.world code structure

The simulation was launched using a custom launch file, '**labyrinth.launch**', to initialize Gazebo environment with our labyrinth world and TurtleBot3 model positioned at predefined coordinates.

4. Comparing Real and Simulation World SLAM Performance

The TurtleBot3 was controlled within the labyrinth using teleoperation commands, and simultaneous localization and mapping (SLAM) techniques were applied to map the environment.

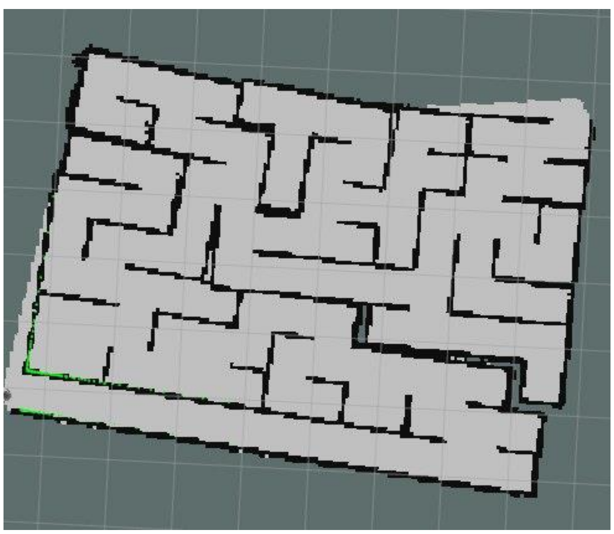

Fig. 1. Simulation world SLAM produced map

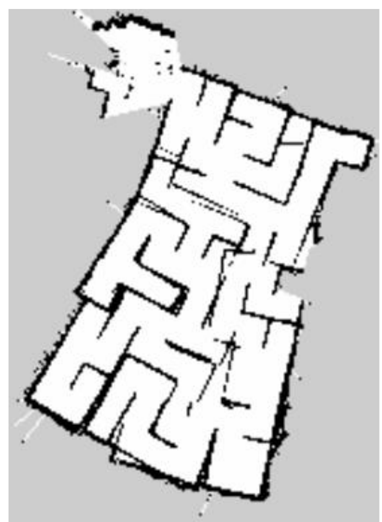

Fig. 2. Real world SLAM produced map

Comparing the two SLAM maps above we have figured out the following:

- The simulated map appears to be very clean and well-defined while in real-world map is less defined. Perhaps due to sensor noise and limitation in precision due to implication of odometry.
- The lines of simulated map are smooth and straight while in real-world map's clarity probably affected by environmental variations.