Grid-Based FastSLAM Demo using ROS and Gazebo

Mohamed Shawky Zaky AbdelAal Sabae
Section:2,BN:15

Remonda Talaat Eskarous
Section:1,BN:19

Ahmed Mohamed Zakaria ElKarashily
Section:1,BN:3

Mohamed Ahmed Mohamed Ahmed
Section:2,BN:10

***Abstract*— In this work, we show a complete demonstration of Grid-Based FastSLAM using ROS and Gazebo simulator. Also, we show the result maps on two different worlds using TurtleBot3 burger model.**

I. INTRODUCTION

Grid-Based FastSLAM is a method that uses *particle filters* and *scan matching*, in order to perform simultaneous localization and mapping on a grid-based map. In this work, we perform a practical demonstration of the method using *ROS* and *Gazebo* simulator and show the result maps on two different environments using *TurtleBot3 burger model*.

II. IMPLEMENTATION DETAILS

A. Algorithms

The main algorithm used is **Gird-Based FastSLAM** from *slam_gmapping* ROS package. It's implemented in the same way described in the lectures. It can be summarized as follows :

- Pre-correct short odometry sequences using scan matching.
- Use the corrected poses as an input to *FastSLAM*, which uses *particle filters* to update the grid map.

B. Robots

The used robot is **TurtleBot3** *burger* model. It contains *360 Laser Distance Sensor LDS-01* as a range sensor.

C. Environments

We use two environments *(worlds)* :

- First environment is a custom environment, completely built by us. It's used to perform the demo video.
- Second one is a pre-defined **TurtleBot3** environment, which we modified, in order to include the results of more complex environments.

D. Used Packages

We mainly used 3 packages in the demo :

- **turtlebot3_gazebo :** for *TurtleBot3* Gazebo simulation.
- **slam_gmapping :** for *Grid-Based FastSLAM* built on top of *ROS openslam_gmapping* SLAM library.
- **turtlebot3_teleop :** for controlling a robot through external peripherals *(used to navigate the robot during SLAM)*.

E. Integration

We used and edited **turtlebot3_slam** package to bridge between **turtlebot3_gazebo** simulation and **slam_gmapping**, as well as launching *Rvis* to visualize the output maps and save them. Moreover, our code has two modes of operation :

- First, we run different packages, mentioned above, in separate terminals.
- Second, we created a single launch file the launches the whole demo *(can be a bit slower than running each package separately)*.

III. DEMO RESULTS

In this section, we show a portion of the result map of each of the two worlds, mentioned above. A complete demo video can be found at https://drive.google.com/file/d/1pKTtOii-n0M5TY9ntGf92lQSl5re4SKz/view?usp=sharing

A. World One

This world is a custom environment, completely built by us. We include a view of the world 1, the robot 2 and the output occupancy grid map (OGM) using *Grid-Based FastSLAM* 3.

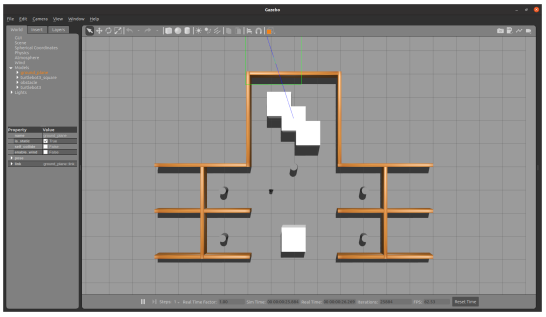

Fig. 1: First world in Gazebo simulator.

B. World Two

This world is a pre-built environment with some edits to include a more realistic and complex environment. We include a view of the world 4, the robot 5 and the output occupancy grid map (OGM) using *Grid-Based FastSLAM* 6.

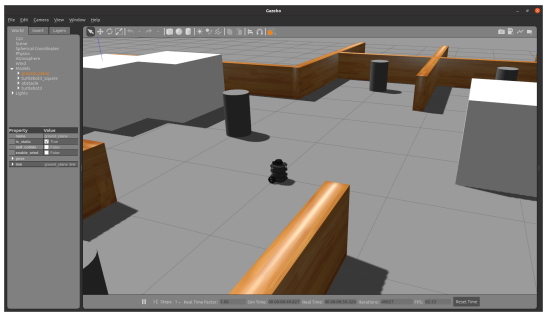

Fig. 2: A 3D view of TurtleBot3 in Gazebo simulator (First Wrold).

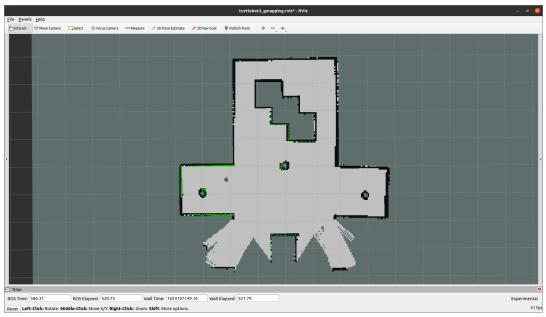

Fig. 3: A view of a portion of result occupancy grid map (OGM) in Rvis (First World).

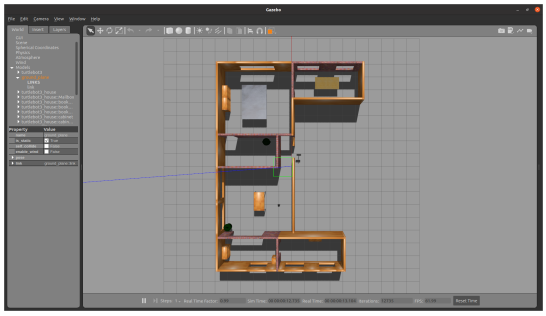

Fig. 4: Second world in Gazebo simulator.

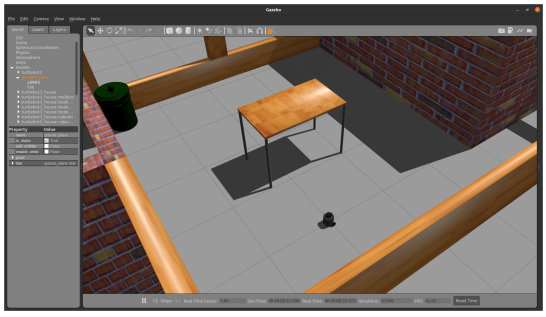

Fig. 5: A 3D view of TurtleBot3 in Gazebo simulator (Second World).

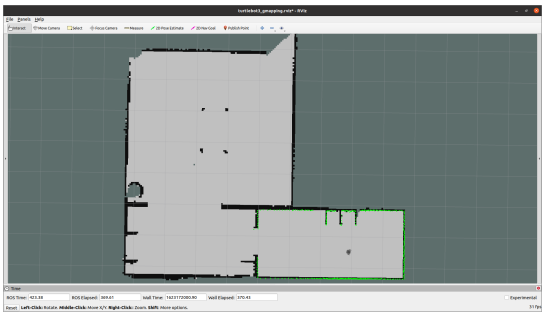

Fig. 6: A view of a portion of result occupancy grid map (OGM) in Rvis (Second World).

IV. WORKLOAD DIVISION

Member	Contribution
Mohamed Shawky	- Software integration & launch files. - Final report. - Packages understanding.
Remonda Talaat	- Software integration. - Packages search. - Packages understanding.
Ahmed Zakaria	- Custom environment. - Packages search. - Packages understanding.
Mohamed Ahmed	- Custom environment. - Packages search. - Packages understanding.

V. CONCLUSION

To sum up, we create a demo with *ROS* and *Gazebo* for *Grid-Based FastSLAM* using *TurtleBot3 burger* on a *custom* environment. We show the results and the accuracy of the method using *Rvis*.