Mobile Robot Localization and Mapping

Moldir Zabirova, Valeriya Kostyukova, Daryn Kenzhebek

I. Recording Map with SLAM

SLAM (simulataneous localization and mapping) means that while we move the robot around, we track its location while also saving all landmarks it sees in a map. We used SLAM to create a map of a new environment, that is the maze, in which TurtleBot3 must navigate. It's what humans do naturally when they enter a new room or environment.

Robot navigation means the robot's ability to determine its own position and then to plan a path towards some goal location.

For the TurtleBot3 mobile robot to find the exit from the maze, it needs to explore the entire labyrinth. In this task, we used a combination of two approaches: keyboard teleoperation and setting a navigation goal in RViz. The latter method, unfortunately, did not work so well for us due to the time synchronization problems. Thus, we went for the former approach and did the SLAM "manually" by controlling the robot movements with the keyboard.

To open RViz and perform SLAM navigation, we used the following commands:

```
roslaunch turtlebot3_slam turtlebot3_
    slam.launch
roslaunch turtlebot3_teleop turtlebot3_
    teleop_key.launch
```

The robot was exploring the whole maze and the resulting map of the environment has been saved as .pmg file together with corresponding .yaml fiel with

```
rosrun map_server map_saver -f ~/map
```

The resulting photo has been eddited in Picsart mobile application to correct the map. With black brush we drew walls, with white path, and gray is unknown area. The final version is depicted on Figure 1.

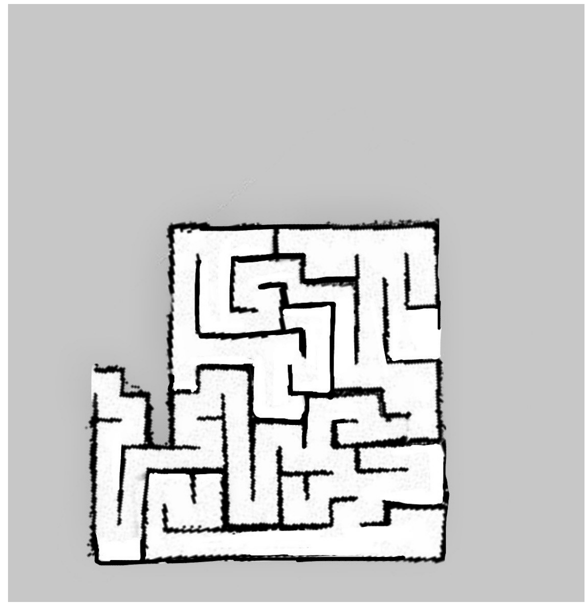

Fig. 1. Maze as a Map for the Robot Navigation

II. Navigation and Path Planning

To create local path for differential drive robot and try to follow a global path, we use DWA (Dynamic Window Approach) local planner algorithm. DWA produces a velocity pair (v, ω) representing a circular trajectory for a robot based on its local conditions. This involves exploring admissible velocities within the next time interval, ensuring the robot can stop before reaching the nearest obstacle along the trajectory. DWA maximizes an objective function considering progress to the target, clearance from obstacles, and forward velocity to generate the optimal velocity pair.

These parameters will be tuned in the next laboratory session. In this work, we launched navigation to try the generated map and set the navigation goals with default parameters (Fig. 2).

To upload the map we used the following command:

```
roslaunch turtlebot3_navigation
    turtlebot3_navigation.launch map_file:
    =$HOME/map.yaml
```

In the RViz toolbar, we first helped robot to understand its current pose in the map by using 2D Pose Estimate green arrow. Next, we set a navigation goal with 2D Nav Goal pink arrow and the robot planned the path to reach the goal position.

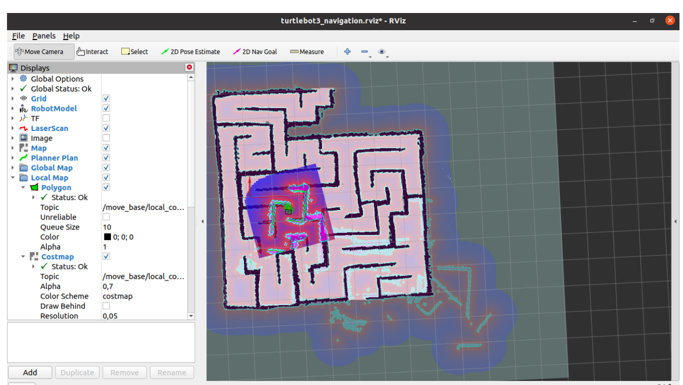

Fig. 2. Robot Navigation in Maze

Because parameters have not been tuned yet, the robot produces oscillations when being too close to a wall. Sometimes, it also stopped when an obstacle was near.

III. CONCLUSION

In this laboratory, we implemented SLAM for a TurtleBot3 robot to navigate through a maze. The SLAM process involved control using keyboard teleoperation. The resulting map was saved and further edited for clarity. Further refinement of parameters is planned for subsequent laboratory sessions to address oscillations and obstacle proximity challenges observed during initial navigation attempts.

IV. TEAM EVALUATION

All team members contributed equally in this laboratory session.