Lab Report 5 and 6 ROBT403

1st Almat Dalabekov
School of Engineering and Digital Sciences, ROBT 303
Nazarbayev University
Astana, Kazakhstan
almat.dalabekov@nu.edu.kz

2nd Dias Akimbay
School of Engineering and Digital Sciences, ROBT 303
Nazarbayev University
Astana, Kazakhstan
dias.akimbay@nu.edu.kz

3rd Bekmurat Amangeldiyev
School of Engineering and Digital Sciences, ROBT 303
Nazarbayev University
Astana, Kazakhstan
bekmurat.amangeldiyev@nu.edu.kz

I. Introduction

Mobile robots are becoming more common in various fields, from warehouse automation to navigating challenging environments. This report focuses on activities carried out in Labs 5 and 6 of the Robotics II course, which used the TurtleBot 3 Burger. This compact robot, designed for research and education, has a Raspberry Pi as its control unit, an OpenCR board for motor and sensor control, and a 360-degree LiDAR sensor for mapping and navigation. Its modular design and compatibility with the Robot Operating System (ROS) make it an excellent platform for learning robotics and automation.

LiDAR, which stands for Light Detection and Ranging, is an essential sensor in mobile robots. It works by emitting laser pulses and measuring their reflections, allowing robots to detect obstacles, map environments, and navigate effectively. During these labs, LiDAR was combined with ROS tools to implement Simultaneous Localization and Mapping (SLAM). SLAM allows robots to create a map of their surroundings while keeping track of their position, which is crucial for navigating unpredictable environments.

In Lab 5, the focus was on assembling and setting up the robot, including configuring the Raspberry Pi, installing ROS Noetic, and testing teleoperation controls. A custom-built labyrinth was used to evaluate the robot's navigation abilities, ensuring the walls were tall enough for the LiDAR to detect them. Teleoperation commands in ROS were used for initial exploration and testing.

Lab 6 expanded on this by introducing SLAM for real-time mapping and localization. The gmapping package was used to create a 2D map of the labyrinth, enabling the robot to navigate autonomously along predefined paths. Parameters like angular update rates and map resolution were fine-tuned for better performance. Additionally, algorithms like the Dynamic Window Approach (DWA) were used for generating smooth and safe paths based on the map.

These labs highlighted the complexity of integrating hardware, software, and algorithms to achieve autonomous operation. The following sections of the report will delve into the methods, experiments, and insights from these exercises.

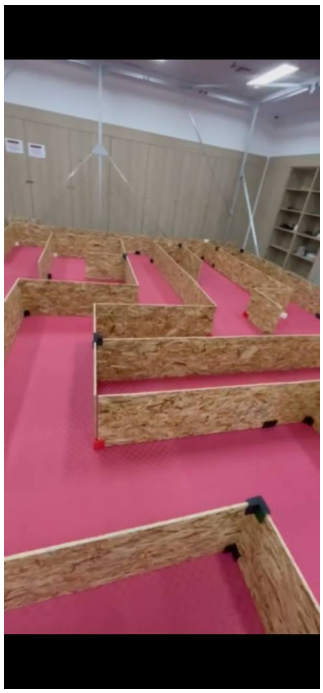

Fig. 1.

II. Methods

This section describes the approach and procedures undertaken in Labs 5 and 6. Lab 5 concentrated on constructing a physical labyrinth for navigation experiments, while Lab 6 transitioned to simulation-based activities for teleoperation, localization, and mapping using Gazebo and ROS.

A. Lab 5: Physical Labyrinth Construction

Objective:to design and build a physical labyrinth tailored for testing the TurtleBot 3 Burger's navigation abilities.

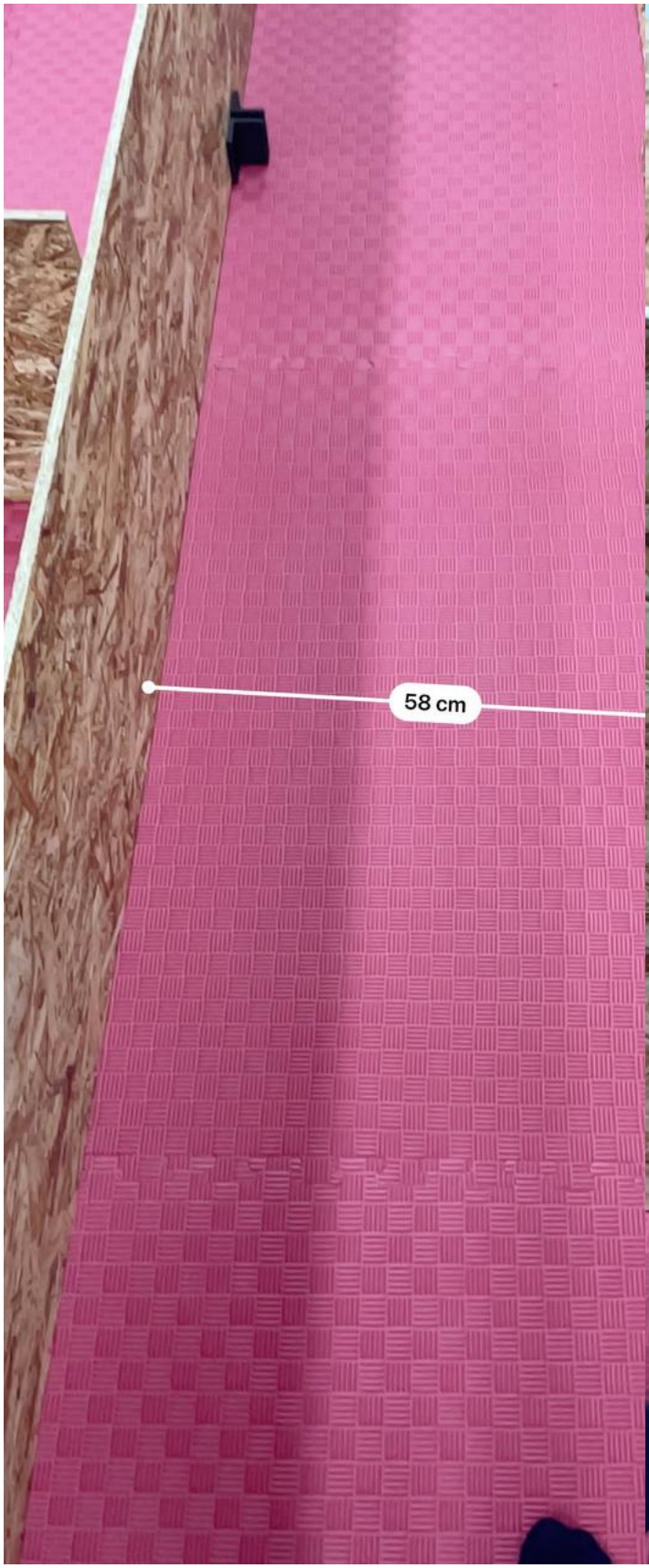

Fig. 2.

Fig. 3.

The labyrinth walls were designed to exceed the TurtleBot's LiDAR sensor height, set at a minimum of 10 cm. The corridors were designed to be wider than the robot's base width, ensuring smooth navigation (13.8 cm minimum).Figures 1-3 shows the real-world Labyrinth pictures with dimensions.

The CAD of the labyrinth almost the same, but height was reduced two times. Figure 4 illustrates the new world done by the user with addition of spawned model of TurtleBot3 of Burger Model.

B. Lab 6:Simulation-Based Localization and Mapping

Lab 6 aimed to simulate the TurtleBot 3 Burger's navigation capabilities in a virtual environment, implementing teleoperation, localization, and mapping.

The virtual environment was set up by creating a Gazebo simulation that mirrored the physical labyrinth constructed in Lab 5. This process began with installing ROS Noetic and Gazebo on an Ubuntu 20.04 system. The TurtleBot 3 Burger model was then loaded into the simulation, where its sensors and movement parameters were carefully configured. To ensure consistency with the physical environment, the labyrinth design was recreated in Gazebo, replicating its dimensions and obstacle placements with precision.

SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM)

Simultaneous Localization and Mapping (SLAM) was implemented within this virtual environment using the `gmapping` package. During the SLAM process, the robot autonomously explored the simulated labyrinth, generating a 2D occupancy grid map in real time. To improve the accuracy and resolution of the generated maps, key SLAM parameters such as `map_update_interval`, `linear_update`, and `angular_update` were fine-tuned. Once the mapping was complete, navigation goals were defined in RViz, and the robot used the Dynamic Window Approach (DWA) to plan paths that were smooth and collision-free, even in narrow corridors.

Before transitioning to autonomous navigation, the teleoperation functionality of the TurtleBot was tested using the `turtlebot3_teleop` package. This step ensured that the robot could move and be controlled precisely within the simulated labyrinth, validating its performance under manual commands.

However, due to an issue with the `turtlebot3_teleop` package, it was unable to send messages to the `cmd_vel` topic. To address this, a custom Python script named `custom_teleop.py` was developed. This script enabled the TurtleBot to be controlled using arrow keys, ensuring that manual teleoperation functionality was retained for testing and validation.

III. RESULTS

A. Lab5

B. Lab6

In Lab 5, the physical labyrinth was initially designed and modeled using SolidWorks to ensure compatibility with the dimensions and capabilities of the TurtleBot 3 Burger. The key considerations during the design process included maintaining a minimum wall height greater than the LiDAR sensor's height (10 cm) and setting corridor widths wider than the robot's base width (13.8 cm) to enable safe navigation. The SolidWorks model served as a precise visual representation of the labyrinth, allowing iterative adjustments and validation. After finalizing the design, the labyrinth was physically constructed using cardboard and wooden supports. The materials were provided by the lab coordinators, and the final construction

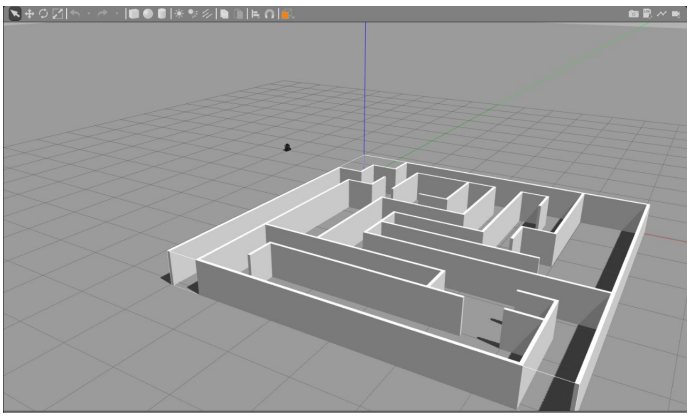

Fig. 4.

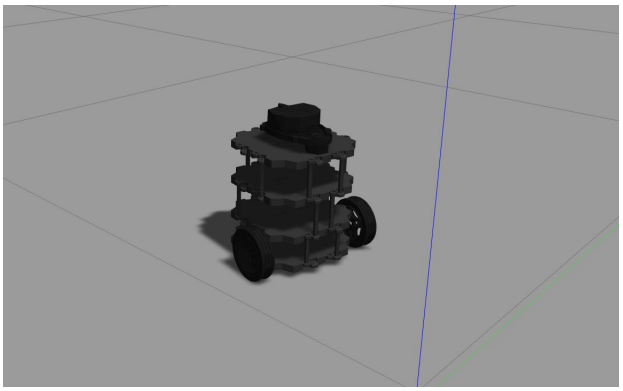

Fig. 5. Visualisation of TurtleBot Model Burger

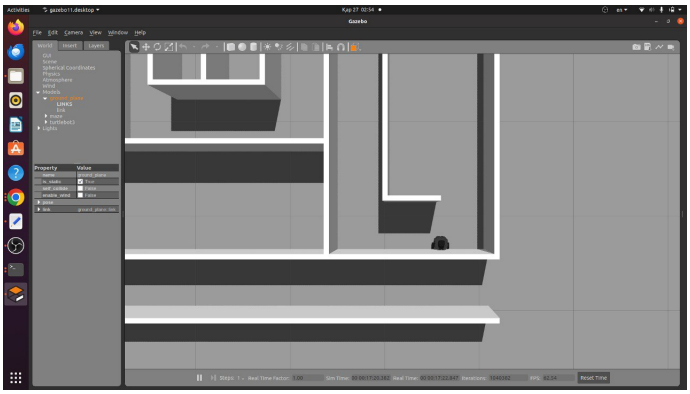

Fig. 6. Robot Inside Labyrinth

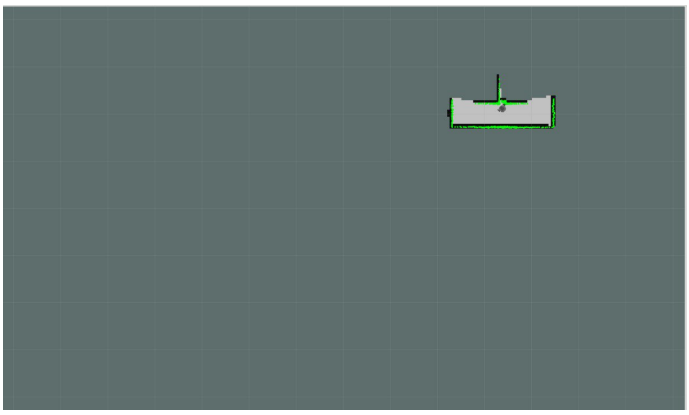

Fig. 7. start of SLAM generation

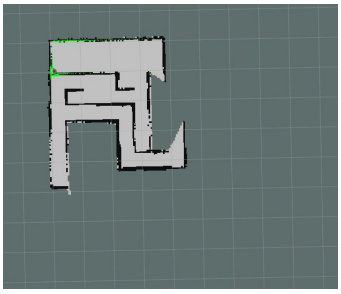

Fig. 8. Middle process of the SLAM

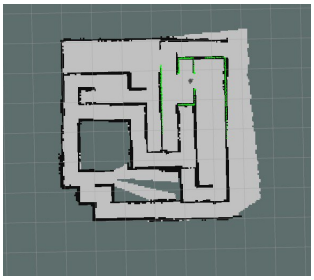

Fig. 9. Final processes

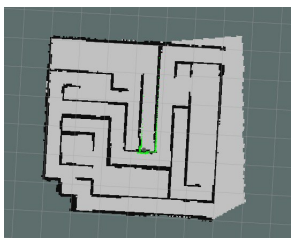

Fig. 10. Final processes

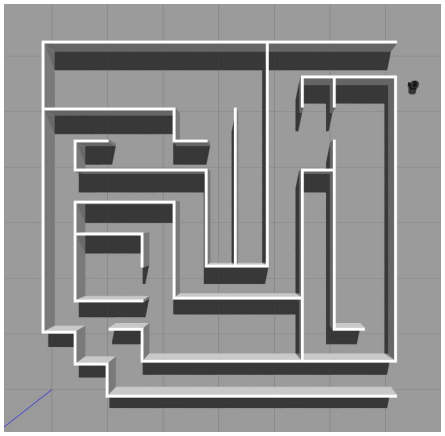

Fig. 11. Ready to for autonomous naviagtion

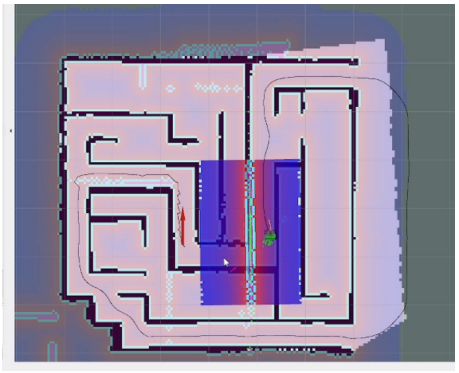

Fig. 12. Rviz visualisation

was verified for accuracy by comparing the physical structure with the digital model. This ensured consistency in dimensions and obstacle placement, providing a reliable test environment for navigation experiments.

Lab 6 focused on testing and simulating the TurtleBot 3 Burger's navigation and mapping capabilities using the physical and virtual labyrinth. The SolidWorks design from Lab 5 was exported as an STL file and imported into the Gazebo simulation environment to replicate the physical setup. The TurtleBot 3 Burger model, equipped with a simulated LiDAR sensor and differential drive, was loaded into the environment for experimentation. This virtual setup allowed for more flexibility and iterative improvements while maintaining alignment with the physical structure.

The first stage of Lab 6 involved mapping the labyrinth using Simultaneous Localization and Mapping (SLAM) techniques. The `gmapping` ROS package enabled the TurtleBot to create a detailed 2D occupancy grid map of the simulated environment. The SLAM process involved capturing LiDAR measurements and optimizing parameters such as `map_update_interval`, `linear_update`, and `angular_update` to improve mapping accuracy and resolution. Initially, the robot was manually controlled using the `turtlebot3_teleop` package to explore the labyrinth.

However, due to issues with the package not sending commands to the `cmd_vel` topic, a custom Python script, `custom_teleop.py`, was developed. This script allowed the robot to be controlled via arrow keys, ensuring the manual teleoperation phase could proceed as planned.

Once the SLAM-generated map was complete, the robot transitioned to autonomous navigation. Using the map generated during SLAM, the `move_base` ROS package facilitated both global and local path planning. The Dynamic Window Approach (DWA) planner was implemented to enable collision-free navigation by utilizing real-time sensor data for obstacle avoidance. Navigation goals were set and visualized in RViz, allowing the robot to traverse predefined paths smoothly. The simulation confirmed the effectiveness of the mapping and navigation strategies, as the robot successfully navigated the labyrinth and reached its target destinations without collisions.

The results demonstrated the successful integration of hardware, software, and algorithms in both the physical and simulated environments. The TurtleBot 3 Burger effectively combined sensor data and pre-mapped information to navigate complex environments, highlighting the reliability and precision of the implemented methods. These findings provide a solid foundation for future work involving autonomous navigation and mapping in real-world applications.

IV. Conclusion

In this lab, we successfully demonstrated the use of the TurtleBot 3 Burger to simulate navigation within a maze using Gazebo. The process began with designing a maze in SolidWorks, which was then converted into a compatible simulation environment. The design ensured that all key specifications, such as wall height and corridor width, matched the TurtleBot's capabilities, providing a realistic and reliable test scenario. Following the detailed instructions provided in the TurtleBot 3 e-manual, we established the necessary ROS environment, which included configuring essential packages and simulation tools, and tested the robot's performance within the virtual labyrinth.

This hands-on experience was instrumental in enhancing our understanding of key concepts in mobile robotics. By configuring the robot to implement teleoperation, mapping, and autonomous navigation, we gained insights into the intricate processes involved in SLAM techniques and path-planning algorithms. Additionally, the lab provided practical exposure to using tools like SolidWorks for modeling, Gazebo for simulation, and ROS for robot control and system integration. Each step reinforced fundamental principles such as localization, obstacle avoidance, and dynamic path planning, allowing us to bridge the gap between theoretical knowledge and real-world applications.

Moreover, the challenges encountered during the project, such as resolving issues with the `turtlebot3_teleop` package and developing the custom `custom_teleop.py` script, further enriched the learning experience. These challenges emphasized the importance of adaptability and problem-solving when working with complex robotic systems. Successfully addressing these issues underscored our ability to troubleshoot and implement alternative solutions effectively.

Overall, this lab served as a foundational step towards tackling more advanced tasks in robotics, such as autonomous navigation, multi-robot coordination, and complex path-planning in dynamic environments. The knowledge and skills acquired through these experiments have prepared us to undertake future projects with greater confidence, contributing to a deeper understanding of robotics and its potential applications in solving real-world problems.

References

[1] Google Drive. Available: https://drive.google.com/drive/folders/1am7mLfmPEWlRsywoXD1AVj7dC9fwXXDD?usp=sharing

[2] GitHub Repository. "ROBT403 Lab." Available: https://github.com/almatikx/robt403_lab

[3] Zhanat Kappasov, "ROBT 403 Lab 6 Manual," [Online]. Available through the ROBT 403 course resources.