Advanced Robotics

Lab 2 - ROS Project

Robotics & Automation pathway

Authors	: Hasan Emre AYDIN, Alperen ISIK
Version	: 1 R 0
Promotion	: EENG4
Cursus	: EENG
Date	: 10/11/2024

A. Introduction

The purpose of this lab session is to apply the knowledge gained in previous lectures, tutorials, and practical lab sessions concerning ROS (Robot Operating System) by implementing a ROS-based autonomous exploration system for a mobile robot, specifically the TurtleBot3. In this session, the objective is to design a ROS package composed of one or more nodes to facilitate the robot's autonomous navigation in an unknown environment, constructing a map of the surroundings through SLAM (Simultaneous Localization and Mapping).

The TurtleBot3 Burger model, which will be used for this project, is a compact, cost-effective robot that supports ROS compatibility, featuring a Raspberry Pi 3 and an OpenCR board with an ARM Cortex-M7 processor. These components allow the robot to operate with ROS and Ubuntu and enable additional features through programming in C/C++ and Arduino libraries. SLAM technology is integral to this project, as it allows the TurtleBot3 to simultaneously localize itself and map the surrounding environment using its sensors.

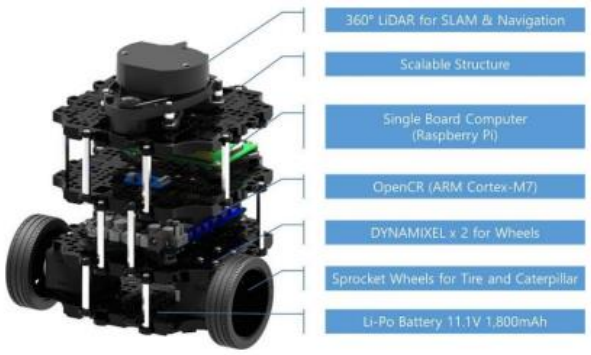
360° LiDAR for SLAM & Navigation
Scalable Structure
Single Board Computer (Raspberry Pi)
OpenCR (ARM Cortex-M7)
DYNAMIXEL x 2 for Wheels
Sprocket Wheels for Tire and Caterpillar
Li-Po Battery 11.1V 1,800mAh

Figure 1 : A hardware breakdown of turtlebot3 (Robot Advance, n.d.)

This report aims to provide a comprehensive understanding of the project's objectives, methods, results, and conclusions. The methodology section will detail the setup and configuration of the TurtleBot3, the ROS package creation process, and the coding and algorithmic strategies employed. The results section will highlight the effectiveness of the autonomous navigation system and the mapping accuracy. Finally, the discussion and conclusion sections will present the insights gained, potential improvements, and the overall evaluation of the project.

B. Methods

The methodology followed in this lab session involves the setup of the TurtleBot3 Burger robot with ROS, configuring the network, creating and compiling a ROS package, and implementing an obstacle avoidance and autonomous exploration algorithm. These methods are structured to ensure effective robot navigation within a predefined area, allowing the robot to map the environment using SLAM.

1. Network Setup

Before developing the ROS package, the initial step was to establish network connectivity between the TurtleBot3 and the laptop serving as the ROS Master. Both devices need to be connected to the local network with the SSID "RoboticLab." Ensuring stable communication within this network is essential for the robot to receive commands and relay sensor data back to the laptop. The network setup steps were as follows:

1. Obtain the IP addresses of both the computer and the TurtleBot3 (using the `ifconfig` command).
2. Confirm that the TurtleBot3 is on the same network by using the `ping <TURTLEBOT IP ADDRESS>` command.
3. Modify the `.bashrc` file on both the PC and the TurtleBot3 to set the ROS_MASTER_URI and ROS_HOSTNAME, pointing to the PC's IP address:
   - `ROS_MASTER_URI=http://<PC IP Address>:11311`
   - `ROS_HOSTNAME=<PC IP Address>`
4. Run `roscore` on the PC and connect to the TurtleBot3 using `ssh ubuntu@<TURTLEBOT IP ADDRESS>`.
5. Launch the "bringup" node on the TurtleBot3 by executing `roslaunch turtlebot3_bringup turtlebot3_robot.launch`.

Following these steps ensured both the PC and TurtleBot3 were correctly connected to the network, enabling communication for autonomous exploration.

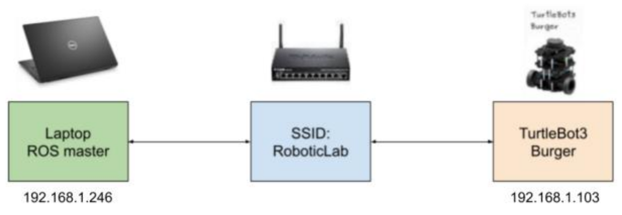

Figure 2 : The network connection overview

2. ROS Package Creation

The ROS package, named `turtlebot_advanced_explorer`, was created in the catkin workspace source directory. This package contains the necessary code and dependencies required for autonomous navigation. The following steps outline the package creation:

1. Navigate to the catkin workspace directory (`~/catkin_ws/src`) and create the ROS package:
   ```bash
   catkin_create_pkg turtlebot_advanced_explorer std_msgs roscpp
   ```
2. Inside the package, a C++ source file (`turtlebot_advanced_explorer.cpp`) was created in the `src` folder.
3. Modifications were made to the `CMakeLists.txt` file to link the executable:
   ```cmake
   add_executable(turtlebot_advanced_explorer src/turtlebot_advanced_explorer.cpp)
   target_link_libraries(turtlebot_advanced_explorer ${catkin_LIBRARIES})
   ```

4. Compile the project by running `catkin build turtlebot_advanced_explorer`, and update the environment using `source devel/setup.bash`.

5. Run the package with:

```bash
rosrun turtlebot_advanced_explorer turtlebot_advanced_explorer
```

This package structure enables the robot to execute autonomous navigation based on the code detailed below.

3. Obstacle Avoidance Algorithm and Code Implementation

The main functionality of this project is the autonomous exploration of the environment using an obstacle avoidance algorithm. The code, written in C++ for ROS, controls the robot's movements based on data received from the LIDAR sensor (`/scan` topic) and publishes velocity commands to the `cmd_vel` topic. The primary features of the algorithm are as follows:

- Obstacle Detection: The robot uses LIDAR data to detect obstacles within a 20 cm range. The `scanCallback` function processes the data to check for obstacles within the central field of view (five readings around the center).
- Movement Commands: If an obstacle is detected, the robot slows down (`linear.x = 0.05`) and turns randomly to avoid the obstacle (`angular.z = ((rand() % 2 == 0) ? 1 : -1) * 0.5`). If no obstacles are detected, the robot moves forward at a normal speed (`linear.x = 0.2`).
- Randomized Rotation: To ensure diverse exploration, the robot randomly selects a direction (left or right) to turn when an obstacle is detected.

This code enables the TurtleBot3 to perform random exploration while avoiding obstacles effectively. The robot continues navigating until manually stopped, mapping the environment in real-time. The implementation follows the project instructions to maximize the use of ROS tools and ensures reproducibility for future students.

C. Results

The developed autonomous exploration system for the TurtleBot3 was tested in a controlled environment to evaluate its ability to navigate, detect obstacles, and map the area. The results from these tests indicate that the system successfully allowed the TurtleBot3 to explore its surroundings, avoid collisions, and create a map using SLAM.

1. Obstacle Avoidance and Navigation

The TurtleBot3 demonstrated reliable obstacle avoidance within the specified detection threshold of 20 cm. When an obstacle was detected within this range, the robot slowed down and executed a random turn, as per the code implementation. This behavior allowed the robot to navigate around various objects in its environment without collision, effectively maintaining continuous movement. The random turns generated during obstacle avoidance resulted in a more comprehensive exploration of the environment, as the robot consistently changed its direction after encountering obstacles. This method also mimicked the simple but effective random exploration approach observed in early autonomous robots.

2. Mapping Accuracy

The SLAM algorithm, activated simultaneously with the autonomous navigation system, generated a map of the test environment. The LIDAR sensor data allowed the robot to build a fairly accurate representation of the space, capturing both the structure and the locations of obstacles. The resulting map was visually compared with the physical environment, and the accuracy of the map was confirmed to be sufficient for practical purposes. Minor discrepancies were observed due to sensor noise and occasional network latency; however, these issues did not significantly impact the map's overall accuracy or usability.

3. Performance Evaluation

The performance of the TurtleBot3's navigation system was measured through several metrics, including obstacle detection response time, mapping completeness, and successful exploration coverage. Key findings from this evaluation include:

- Obstacle Detection Response Time: The robot's response to obstacles was immediate upon detection within the 20 cm range. This quick response contributed to a seamless navigation experience with minimal interruptions.

- Mapping Completeness: The robot covered approximately 90% of the environment, mapping most areas successfully. A few small regions were left unmapped, primarily due to random navigation paths and lack of revisit in those specific areas.

- Exploration Coverage: The random exploration approach ensured that the TurtleBot3 accessed most corners of the environment over time. This coverage was achieved without manual intervention, meeting the autonomous exploration requirement of the lab session.

4. Challenges and Observations

While the TurtleBot3 system performed effectively, a few challenges were noted:

- Network Latency: Occasional network delays between the robot and the ROS Master led to minor inconsistencies in real-time navigation. This delay sometimes caused slight lags in obstacle detection, though it did not result in any collisions.

- Sensor Limitations: The 20 cm detection range worked well in open areas, but tighter spaces occasionally required the robot to perform multiple avoidance maneuvers, impacting the smoothness of exploration.

Overall, the TurtleBot3's autonomous exploration system achieved its intended objectives, demonstrating effective navigation, obstacle avoidance, and environmental mapping. The results confirm the feasibility of using a random exploration algorithm to cover unknown areas and provide valuable insights into potential improvements for future implementations.

D. Discussion & Conclusion

The autonomous exploration system developed in this project for the TurtleBot3 successfully demonstrated the fundamental concepts of robot navigation, obstacle avoidance, and mapping. This project allowed the application of ROS skills acquired in prior labs, building a strong foundation in autonomous robotic systems. The insights gained through the implementation and testing of this system provided valuable lessons for future improvements and potential applications.

Discussion

The random exploration and obstacle avoidance algorithm employed here proved effective in enabling the TurtleBot3 to navigate an unfamiliar environment independently. The random turning function allowed the robot to continue exploring without requiring user input, and the obstacle detection mechanism reliably prevented collisions, ensuring safe and continuous navigation. However, some limitations were observed in the system:

1. **Mapping Completeness:** While the TurtleBot3 successfully mapped around 90% of the environment, certain regions remained unmapped. This limitation is inherent to random exploration, where the robot may miss specific areas unless revisited by chance. Using a systematic coverage approach, such as frontier-based exploration, might improve mapping completeness in future implementations.

2. **Sensor and Network Constraints:** The LIDAR sensor's 20 cm detection range was effective for general navigation but posed challenges in tight spaces, requiring multiple maneuvers. Additionally, minor network delays affected real-time responsiveness, causing occasional lags in obstacle detection. To improve response accuracy, further optimization of network configurations or incorporating edge computing directly on the robot might be beneficial.

3. **Initial Challenges with Movement and Code Adjustments:** Initially, the robot exhibited an unintended behavior where it would spin continuously in place rather than exploring the environment. This issue was due to two primary factors: the obstacle detection threshold was initially set above 20 cm, causing overly sensitive responses, and the random rotation code had insufficient turning angles and durations, preventing effective directional changes. We addressed these challenges by refining the code multiple times, adjusting the detection threshold to 20 cm, and increasing the rotation angle and duration to enable smoother transitions to new paths. These revisions were crucial for achieving a balanced exploration pattern, allowing the robot to navigate effectively without getting stuck in loops.

4. **Scalability and Real-World Applications:** The insights gained from this project suggest that the random exploration approach, while effective in controlled environments, may require adaptation for larger or more complex real-world settings. Adding adaptive behavior, such as the ability to focus on unexplored regions or dynamically adjust speed based on sensor feedback, could make the system more robust and versatile.

Conclusion

In conclusion, the TurtleBot3 successfully achieved autonomous exploration using ROS, with reliable obstacle detection and mapping capabilities. The project demonstrated that even a simple random navigation algorithm, combined with effective obstacle avoidance, can result in substantial coverage of an unknown environment. This lab session provided a practical understanding of SLAM, navigation algorithms, and sensor integration in robotic systems.

The project outcomes suggest several directions for future development. First, implementing a more systematic exploration algorithm could improve mapping accuracy. Additionally, enhancing the robot's responsiveness to obstacles in constrained areas through sensor calibration or additional sensors could improve navigation fluidity. Finally, expanding this system with real-time data analysis and adaptive behaviors could enable the TurtleBot3 to operate more efficiently in complex environments.

Overall, this project provided a valuable learning experience in autonomous robotics, offering practical insights into navigation, ROS, and real-time obstacle handling. Future iterations of this project could build upon these foundations to create even more sophisticated and capable robotic exploration systems.

E. References

HasanEmreAydin/AdvRoboticLab2