CONTROL AND NAVIGATION OF MOBILE ROBOT IN ROS SIMULATION PLATFORM

Son Huynh Thanh[1*], Oanh Tran Thi Hoang[2]

[1]*Dong Nai Technology University*
[2]*Binh Duong Economics and Technology University*
*Corresponding author: *Son Huynh Thanh, huynhthanhson@dntu.edu.vn*

GENERAL INFORMATION

Received date: 06/03/2024

Revised date: 09/05/2024

Accepted date: 12/07/2024

KEYWORD

Autonomous mobile robot, controller;

LiDAR;

ROS;

RViz*;*

SLAM.

ABSTRACT

Control and navigation systems are two critical issues in autonomous mobile robot control. In this paper, the Robot Operating System (ROS) is studied. ROS is a free and open-source platform, supported by large communities. It is not only a platform for robot software development but also provides programs with access to hardware resources. Localization and mapping are performed on ROS using laser scanning data from Light Detection and Ranging (LiDAR) with the simultaneous localization and mapping (SLAM) method to control and navigate the mobile robot. The simulation is conducted in the Robot Visualization tool (RViz). The mapping and path planning are demonstrated with various obstacles, and the robot successfully reached its destination.

1. INTRODUCTION

Today, with the fast development of robotic, it has attracted many attentions. As a branch of robotics, mobile robot is a branch of robotics and widely used in many cases, such as industrial transportation, logistics, mobile operations, logistic, etc. (Arkin, et al., 1990; Köseoğlu, et al., 2017; Ochiai, et al.. 2014). Mobile robots operate in many different locations, different positions, so it is necessary to have a precise position measurement device for the robot (Siegwart, et al.. 2011). The autonomous navigation of the robot to its target location is simulated on Gazebo (Takaya, et al.. 2016; Pietrzik, et al., 2019). To ensure successfully navigation, SLAM localization was employed to locate the TurtleBot3 in the map (J. M. Santos, et al., 2019). The simulation results were visualized using RViz and were found to be satisfactory. This research not only creates value for the industry, but also opens up new opportunities for future creativity and innovation.

2. METHODOLOGY

The Robot Operating System (ROS) is introduce. In ROS, robot TurtleBot3 is used for simulation. LiDAR is the tool described. Additionally, the entire mapping, localization and navigation process is simulated and explained. TurtleBot3 can navigate different obstacle environments and reached goal.

2.1 TurtleBot3 Platform

TurtleBot3 is a compact, cost-effective mobile robot, designed for education, research, hobbies, and product development (Robotics, 2024). Its primary aim is significantly to reduce the cost while maintaining its functionality and quality, offering flexibility for expansion. The TurtleBot3 can be customized in various ways modifying the mechanical components and optional parts like sensors and embedded systems. Moreover. TurtleBot3 has been developed to be the cost-effective and small-sized mobile robot, making it is a suitable for robust embedded system (Orkan Murat Çelik1, Murat Köseoğlu, 2023).

2.2 Simultaneous Localization and Mapping platform

Simultaneous Localization and Mapping is a technique that enables a robot to create a map of environment around robot and location itself in the space. This technology permits establish a mobile map. Allowing the efficient digitization of large areas. SLAM systems collect data, mapping space around environments both indoor and outdoor (Takaya, et al.,2016).

Mathematically, SLAM can be described by following steps (Durrant-Whyte, et al., 2006): XT is the position of robot. T is the sample time.

$$XT = \{x0, x1, x2, \dots \quad xT\}$$

If UT is the robot's movement between times T-1 and T, assuming that this motion data is derived from encoder readings or motor control input, the robot's time–dependent can be written as:

$$UT = \{u0, u1, u2, \dots \quad uT\}$$

Accordingly, if it is assumed that the value of $m0$, $m1$, $m2$… is presented objects around the robot, M is written as:

$$MT = \{m0, m1, m2, \dots \ mn-1\}$$

If each sensor on the robot captures only one measure at a time, the complete set of measurements ZT:

$$ZT = \{z0, z1, z2, \dots \ zT\}$$

The data obtained from position(XT), odometer (UT) and observations (ZT) robot can mapping the path. Two primary algorithms are full SLAM and online SLAM that are used by AMRs. AMR can navigate various areas composed of successive point on the map to reach a target. The probability of the next joint to be followed by AMR along the full SLAM XT, depends on various factors, and this relationship is represented by the following equation:

$$p\{XT, M \vee ZT, UT\}$$

A difference between the full SLAM and online SLAM: In online SLAM, XT indicate the next most probability path along M for the next point. As seen, full SLAM aims to estimate the entire path (XT) of the robot, while online focus solely on the current trajectory.

2.3 Robotic Operating System (ROS)

The Robot Operating System (ROS) is an open-source framework based on Linux, designed for controlling robots (ROS Wiki, 2024). It facilitates peer-to-peer communication, where executable programs known as nodes interact with each other during runtime. These nodes are registered with the ROS master, which is essential for them to be aware of one another. Instead of direct communication between nodes, they exchange information by publishing or subscribing to messages on specific topics. For instance, if a node requires data, it subscribes to the relevant topic, while nodes generating data publish their messages to that topic. This setup exemplifies a decoupled system, allowing different parts of the robot to perform distinct functions independently, ensuring that a failure in one function does not disrupt the entire robot's operation. Another benefit of ROS is the reusability of code, which allows researchers to build upon existing codebases rather than starting from scratch, thereby accelerating the development process.

2.4. Simulation world

Gazebo is the platform to simulate and the leader of in robot simulation. At figure 1, there is a Hexagon wall. There are 9 obstacles inside.

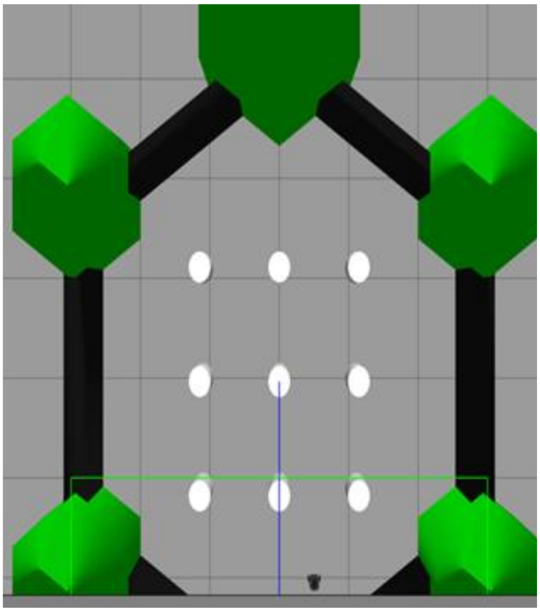

Figure1. Robot simulation environment

2.5 Simultaneous Mapping and Localization

For an autonomous robot to successfully navigate within an unfamiliar environment, it must be able to both map its surroundings and determine its own location within that environment. This is why Simultaneous Localization and Mapping (SLAM) plays a crucial role.

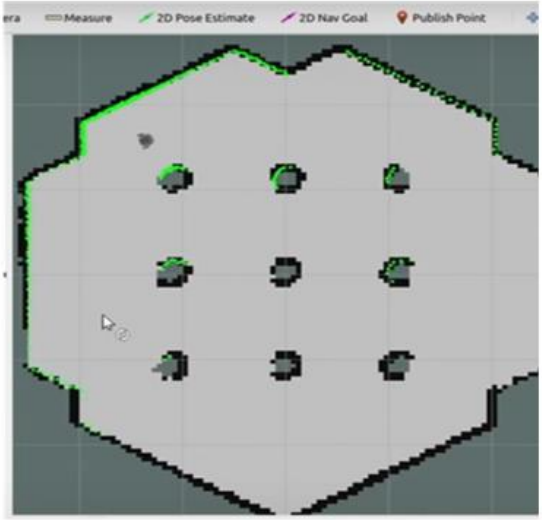

Figure 2. Robot maps the enviroment

3. FINDINGS AND DISCUSSION

3.1. Mapping-localizing

To visualize the simulation results, I utilized RViz, a 3D visualization tool compatible with ROS. RViz allows for the visualization of various elements such as 3D robot models, sensor data, and camera inputs. Figure 3 displays the projection map as rendered in RViz. This figure illustrates the displacement between the local and global maps, highlighting the necessity of localization.

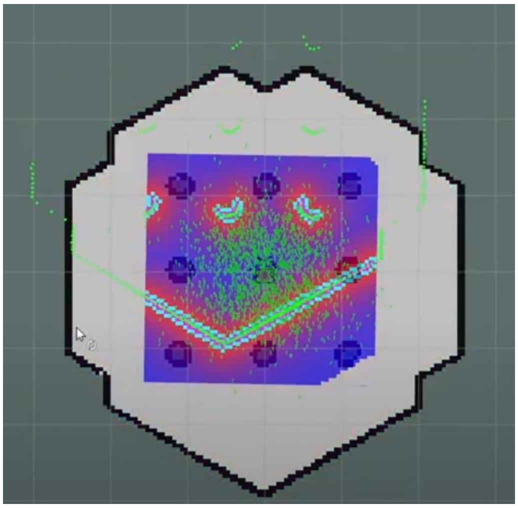

Figure 3. Global map and local map is not match

The TurtleBot3's movement is managed by the move-base package, which includes a ROS node named move-base. This package maintains both local and global cost maps to facilitate local and global planning. Figures 4 and 5 depict these cost maps. The cost maps record obstacle information, with the global cost map used for overall environment planning and the local cost map utilized for short-range planning and obstacle avoidance. The planner assists the TurtleBot3 in navigating according to its global plan and setting preferences for movement.

3.2 Navigation without obstacle

Simulation showing that the robot is chosen the line to the goal. And the destination is satisfied.

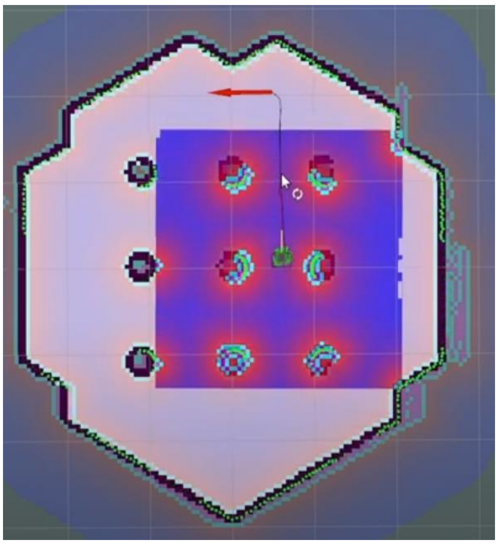

Figure 4. Robot is running to the goal without obstacle

3.3 Navigation with obstacle

From figure 4 and figure 5, we can easily see that the goal is the same but from the figure 5, we see the different path planning.

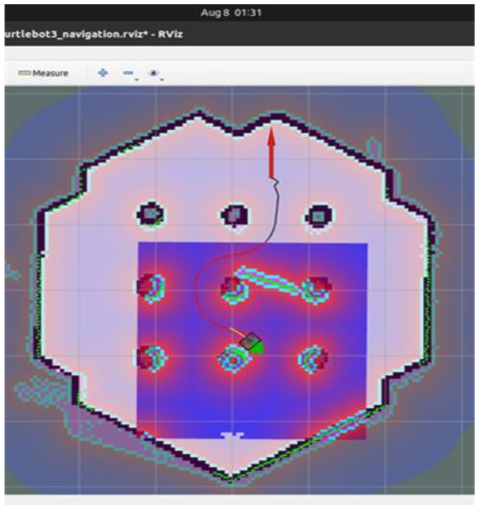

Figure 5: Robot is running to the goal in horizontal obstacle

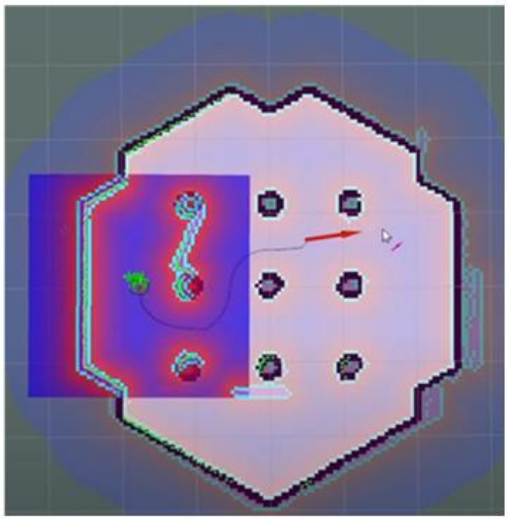

Figure 6. Robot is running to the goal with vertical obstacle

Figure 5 and figure 6 is simulated with different angles to check the path planning. In environments with multiple obstacles, the robot's ability to navigate effectively depends on the accuracy of the sensors and the responsiveness of the planning algorithms. The quality of the local cost map is critical, as it needs to quickly and accurately reflect changes in the environment for the robot to make the best decisions.

4. CONCLUSION

The objective of our work was to simulate a low-cost autonomous mobile robot in simulation environment. The mobile robot can capable map an unknown environment using an inexpensive LiDAR. Additionally, the robot model successfully localized itself within the map generated by the SLAM platform and navigated to the designated point. The entire navigation process was visualized using RViz. The robot effectively reached its point with different simulation environments.

Effective navigation in the presence of obstacles is a critical aspect of autonomous robotics, requiring the integration of robust planning algorithms, real-time sensor data, and adaptive strategies. By continuously updating its cost maps and adjusting its path in response to obstacles.

REFERENCES

Arkin, R. C., & Murphy, R. R. (1990). Autonomous navigation in a manufacturing environment. *IEEE Transactions on Robotics and Automation*, 6(4), 445-454.

https://doi.org/10.1109/70.59355.

Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: part I. *IEEE Robotics Automation Magazine*, 13(2), 99–110.

https://doi.org/10.1109/MRA.2006.1638022.

J. M. Santos, D. Portugal and R. P. Rocha. (2019). An evaluation of 2D SLAM techniques available in Robot Operating System. *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Linkoping, 2013, pp. 1-6.ing.

Köseoğlu, M., Çelik, O. M., & Pektaş, Ö. (2017). Design of an autonomous mobile robot based on ROS. *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*,1-5. https://doi.org/10.1109/IDAP.2017.8090199.

Ochiai, Y., Takemura, K., Ikeda, A., Takamatsu, J., & Ogasawara, T. (2014). Remote control system for multiple mobile robots using touch panel interface and autonomous mobility. *International International Conference on Intelligent Robots and Systems, 3272–3277*. https://doi.org/10.1109/IROS.2014.6943017.

Orkan Murat Çelik1, Murat Köseoğlu (2023). A Modified Dijkstra Algorithm for ROS Based for autonomous Mobile Robots. *Journal of Advanced Research in Natural and Applied Sciences*, 3272-3277.

https:// doi.org/10.28979/jarnas.1119957

Pietrzik, S., and B. Chandrasekaran (2019). "Setting up and Using ROS Kinetic and Gazebo for Educational Robotic Projects and Learning." *In Journal of Physics: Conference Series*, vol. 1207, no. 1, p. 012019. IOP Publish.

Robotics. (2024). TurtleBot3 simulation. Retrieved July 28, 2024, from https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/

ROS Wiki. (2024). Documentation. http://wiki.ros.org/Documentation.

Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011*). Introduction to Autonomous Mobile Robots*. (Second Edi, Vol. 5). The MIT Press.

Takaya, Kenta, Toshinori Asai, Valeri Kroumov, and Florentin Smarandache (2016). "Simulation environment for mobile robots testing using ROS and Gazebo." *20th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 96-101. IEEE, 2016. 3272–3277