Transactions on Machine Intelligence 8(4) (2025) 196–206

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/

Gazebo Simulation and ROS Implementation to Examine G-mapping SLAM Algorithm to Navigate Turtlebot-3 Robot

M.R. Sayyed Noorani [1,*], F. Aghdam Shahryar [1]

[1] Department of Mechatronics Engineering, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Article History:
Received 24 May 2025
Received in revised form 23 August 2025
Accepted 20 October 2025
Available online 1 December 2025

Keywords:
SLAM, G-mapping, Robotic Operating System (ROS), Gazebo, Turtlebot-3 Robot.

ABSTRACT

This study investigates both Gazebo-based simulations and real-world implementations of the Gmapping Simultaneous Localization and Mapping (SLAM) algorithm using the Robot Operating System (ROS) framework for autonomous navigation of a TurtleBot3 mobile robot. The primary objective is to evaluate the performance, robustness, and adaptability of Gmapping under different navigation strategies and environmental conditions. To this end, the SLAM_Gmapping package, a widely adopted ROS-based implementation of grid-based SLAM, is employed in combination with a two-dimensional Light Detection and Ranging (LiDAR) sensor to estimate the robot pose and construct an occupancy-grid representation of the surrounding environment. The performance of the proposed navigation framework is assessed in both simulated and physical indoor environments. Two navigation strategies are considered: manual teleoperation and autonomous navigation based on the Dynamic Window Approach (DWA). Their performance is comparatively examined in the presence of both static and dynamic obstacles in order to investigate the influence of navigation behavior on mapping accuracy, computational efficiency, and mapping time. The experimental results reveal noticeable differences between simulation and real-world operation. In the Gazebo environment, manual navigation in scenarios containing static obstacles produces more accurate maps within a shorter mapping time. In contrast, during real-world experiments, autonomous navigation using DWA provides more consistent and reliable mapping performance, although a longer period is required to complete the mapping process. Furthermore, a parametric evaluation of the Gmapping algorithm under different environmental and navigation conditions demonstrates that its performance is strongly affected by robot motion, obstacle configuration, and navigation strategy. Despite these variations, the algorithm exhibits satisfactory adaptability in indoor environments and is capable of generating usable maps for TurtleBot3 navigation. The findings highlight the importance of selecting an appropriate navigation strategy when deploying SLAM algorithms and demonstrate the practical potential of the ROS-based Gmapping framework for indoor mobile robotic mapping and autonomous navigation applications.

[*] Corresponding Author: smrs.noorani@tabrizu.ac.ir
Department of Mechatronics Engineering, University of Tabriz, Tabriz, Iran

http://dx.doi.org/10.22034/tmi.2025.244865

1. INTRODUCTION

The field of robotics has witnessed significant advancements, particularly in the development and utilization of wheeled robots, over the past two decades. These robots, through innovative navigation and environmental exploration methods, have opened new horizons in industrial and research applications. A major challenge in this domain is accurate environmental localization and mapping, especially in areas with lacking GPS signals. The simultaneous localization and mapping (SLAM) strategy plays a crucial role in enhancing the robots' movement precision and efficiency by enabling them to simultaneously create an accurate map and determine their position on this map with high accuracy in real-time [1]. The SLAM strategy has thus become a pivotal research area in the navigation of mobile robots and generally any type of autonomous vehicle, with ongoing development in both theoretical and practical versions [2].

The Robotic Operating System (ROS) is a Linux-based (Ubuntu distribution) software that provides an integrated framework for programming and managing robots. ROS includes libraries, software packages, and various simulation and monitoring environments. The Gazebo dynamic simulation environment under ROS offers realistic simulations of robots and their environments. Turtlebot-3 is a differential drive robot equipped with various sensors, including LIDAR, depth cameras, and advanced control boards, making it an ideal platform for developing and evaluating navigation algorithms under ROS [3].

In this paper we address to the implementation and examination of the SLAM_Gmapping navigation strategy under simulation in the ROS-Gazebo as well as real-world on the Turtlebot3 robot. The SLAM_Gmapping is used for fast and precise mapping of unknown environments and is distinguished by its simplicity and efficiency, especially in indoor environments with relatively fixed and limited obstacles, while can exploit diverse sensory information from the robot [4-7].

2. RELATED WORK

Aydemir et al. (2021) examined the impact of object geometry on the accuracy of the SLAM_Gmapping algorithm under simulations in ROS-Gazebo environment, using the Turtlebot3 model but lacking real-world testing [4]. Spina et al. (2021) analyzed the SLAM_Gmapping navigation under Gazebo simulation across multiple autonomous robots managed by ROS. Various tests involved collaborative mapping of a large environment by multiple robots [5]. Nurzam et al. (2019) studied the impact of tuning parameters of the SLAM_Gmapping algorithm for indoor mapping using a differential drive robot, utilizing both Gazebo simulation and real-world testing with a LIDAR-equipped robot [6]. Rojas-Fernandez et al. (2018) conducted a comprehensive study of different 2D SLAM methods under ROS, identifying G-mapping as the most efficient algorithm across all scenarios [7]. Yang (2021) simulated a service robot under Gazebo, analyzing SLAM performance with LIDAR sensor data [8]. Chen et al. (2022) simulated all stages of localization, mapping, and path planning with an image-based SLAM algorithm on a quadrotor robot model in Gazebo [9]. Ali et al. (2021) studied Turtlebot2 navigation under the Cartographer algorithm in an indoor environment via Gazebo simulation, using LIDAR as the environment monitor [10]. Singandhupe and La (2019) reviewed various SLAM methods and their applications in autonomous driving safety, suggesting SLAM as an excellent solution in environments lacking external positioning systems, e.g. GPS [11]. Fan et al. (2021) compared G-mapping, Karto, and Cartographer algorithms for 2D indoor environment mapping under Gazebo simulation [12].

3. SPECIFICATIONS OF TURTLEBOT3 (WAFFLE)

The Turtlebot3 (Waffle) is a standard differential drive mobile robot by ROBOTICS, offering advanced control and navigation experiments in diverse environments. This robot is equipped with multi-sensor arrays including LIDAR (Light Detection and Ranging), depth cameras, and IMU, which making the Turtlebot3 to accurately gather environmental data. It is compatible with the distribution of ROS Kinetic Kame, installed on Ubuntu 16.04, enabling the use of the Turtlebot3 package and its libraries, which include SLAM-based algorithms utilized in this study [3]. Specifications of the Turtlebot3 robot is listed in Table 1.

4. ROBOTIC OPERATING SYSTEM (ROS)

ROS is an open-source framework for developing robotic software and managing information flow between hardware and software components. In this study, several ROS tools and software were utilized, each playing a specific role in the development and execution of SLAM algorithms. Gazebo offers a graphical environment for simulating robot movements based on kinematic and dynamic analyses. Rviz provides 3D visualization for sensor data, robot status, and surrounding environment in real-time. Rqt_graph and rqt_tf_tree are tools for displaying information flow and troubleshooting processes in ROS, showing node communications via topics and services, and the structure of transformations between frames respectively. These tools facilitate precise evaluation, effective simulation, and better visualization of results [3].

Table 1: Dimensional and Technical Specifications of the Turtlebot3 (Waffle)

Dimensions: length x width x height	**281mm x 306mm x 141mm**
total weight	**1.8 kg**
Maximum transmission speed	**0.26 m/s**
Maximum rotational speed	**1.82 rad/s (104.27 deg/s)**
Maximum load that can be carried	**30 kg**
Dynamixel servo motor	**XM430-W210-R**
Depth detection stereo camera	**Intel® Realsense R200**
Lidar sensor with 360 deg. scanner	**HLS-LFCD2**
Control board	**OpenCR** **32-bit ARM® Cortex®-M7**
Mini computer	**Intel® Joule**

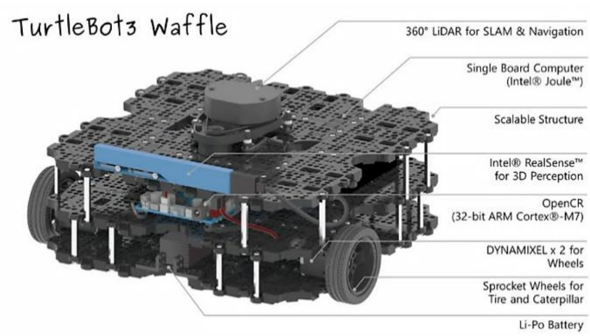


Fig. 1. Turtlebot3 robot (Waffle)

5. SLAM_GMAPPING ALGORITHM

This study focuses on using and evaluating the Gmapping algorithm, selected for its simplicity, efficiency, and compatibility with robots like Turtlebot3 that have limited hardware resources. Gmapping, based on the Rao-Blackwellized particle filter, uses raw laser scan data and odometry to create a 2D grid map of occupied spaces in the environment. It encodes the probability of each cell being occupied based on a Bayesian approach. The theoretical details of the algorithm is available in [13, 14].

Factors like wheel slippage, robot speed, and map update frequency can introduce noise in odometry, potentially causing discontinuous and corrupted maps. The Gmapping SLAM algorithm corrects the available map by continuously integrating measurements from previous environmental points and makes an updated map in output.

6. RESULTS OF SIMULATION AND IMPLEMENTATION

In this section, the results of various experiments conducted to evaluate the SLAM Gmapping algorithm implemented on the Turtlebot3 robot in both simulated and real-world environments are presented. The experiments are initiated by running the turtlebot3_slam software package and configuring it to use the Gmapping method. For this purpose, after calling the ROS core, the following commands are executed in three separate terminals:

1. Estabilishing wireless connection between the laptop-robot and launching the Gazebo simulated environment:

```
$ ssh turtle@192.168.1.0
$ roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

2. Launching the SLAM_Gmapping algorithm:

```
$ export TURTLEBOT3_MODEL=waffle
$ roslaunch turtlebot3_slam turtlebot3_slam.launch
```

3. Launching manual guidance to move the Turtlebot3 robot in the simulated environment:

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

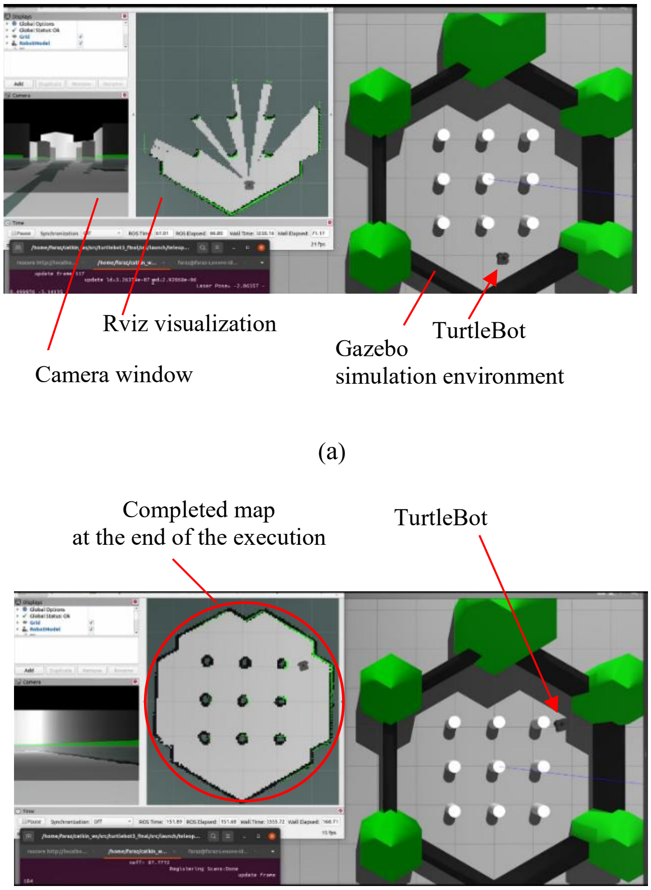
The simulated environment in ROS-Gazebo is a regular hexagon enclosed by six walls, with each pair of adjacent walls connected to a thick column, and it contains nine evenly spaced circular columns, as shown in Figures 2–4. However, the real-world area is a square room connected to a rectangular hallway, forming an L-shaped environment, as depicted in Figures 6 and 7. In this environment, there are office items such as desks and chairs acting as obstacles.

For mapping, the robot needs to move around the environment without a specific goal, using the Lidar sensor to emit laser beams and receive their reflections. This helps the robot gather information about obstacles and boundaries to draw and complete the map of the environment. There are two ways to guidance the robot's movement: manually, by running the node of "turtlebot3_teleop_key", and automatically, using the DWA algorithm.

The experiments were conducted using both methods in both simulated and real-world environments, and the results are presented in Table 2.

Table 2. The performance of the Gmapping Algorithm to build a map

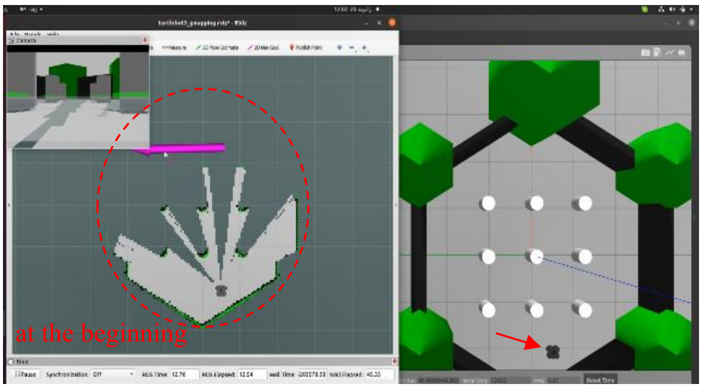
Ultimate details were gained at the end (in %)	Total time of Execution (in sec.)	Details (in %) has been gained	Time (in sec.) to build a map with most details	area (m^2)	The presence of obstacles	Guidance type
Simulation in ROS-Gazebo environment						
100	105	85	45	20	**static**	**manual**
100	380	84	150	20	**static**	**DWA**
90	2640	70	150	20	**dynamic**	**DWA**
Implementation in real-world environment						
95	325	79	150	13	**static**	**manual**
100	525	85	230	13	**static**	**DWA**



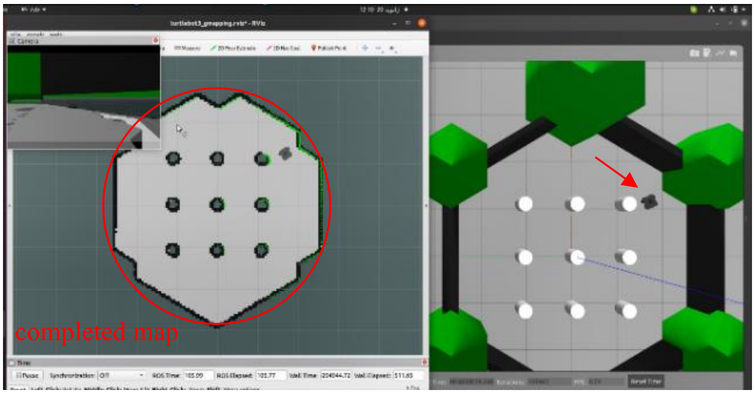
(a)

(b)

Fig. 2. Gazebo simulation for mapping the environment with manual guidance, a) at the beginning, b) at the end of execution. Duration for completing the map was 105 sec.



(a)



(b)

Fig. 3. Gazebo simulation for mapping the environment with automatic guidance, a) at the biginning, b) at the end of execution. Duration for completing the map was 380 sec.

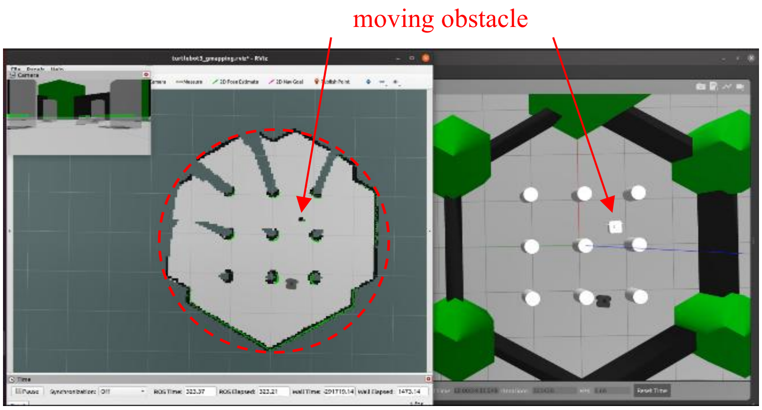
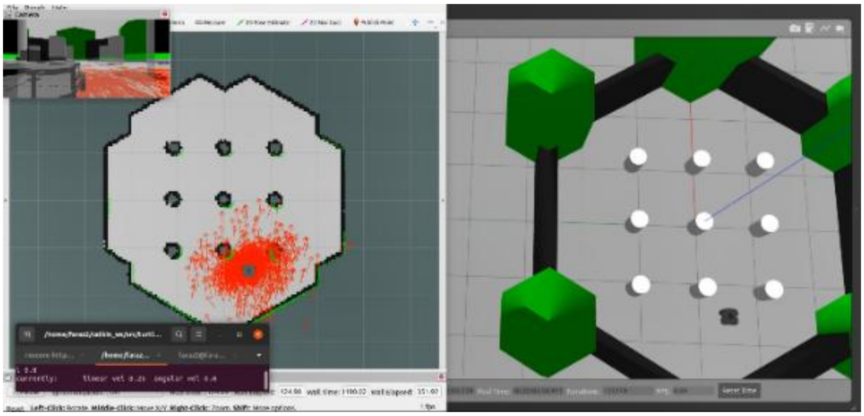


Fig. 4. Gazebo simulation for mapping the environment with automatic guidance, and presence a moving obstacle (white Turtlebot). Duration for completing the map was 44 min. with 90% details.

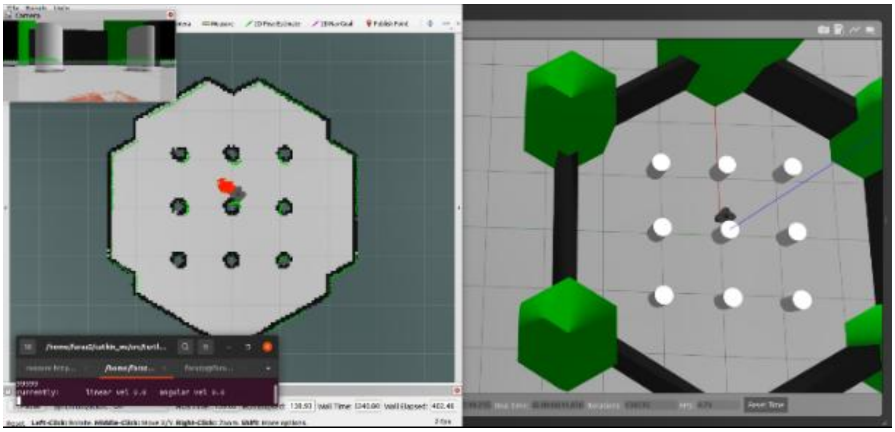
In a special experiment conducted only in the simulated environment, a moving obstacle was added. The obstacle was a white wheeled robot, as shown in Figure 4, which was also controlled using the DWA algorithm. The main robot was required to avoid collisions with this moving obstacle.

After mapping the environment, some experiments were conducted to localize the robot in both the simulated and real environments. The goal of these experiments was for the robot to estimate its current position based on the created map and move towards a designated target. The movement commands were executed both manually and automatically (under the DWA algorithm). This process involved sampling points in the environment as positions and then filtering them using the algorithm of AMCL (Adaptive Monte-Carlo Localization), which the sampled points have been depected in red in Figures of 6 (simulated enviornment), and 9-10 (real-world experiments).

Similarly, mapping and localization experiments were conducted in a real-world environment using the TurtleBot3. The mapping results are shown in Figures 6 and 7, whereas the localization results are presented in Figures 8 and 9.



(a)



(b)

Fig. 5. Sampling points used for robot localization in the simulated environment: (a) sparse sampling points representing possible robot positions and (b) increased density of sampling points after applying the AMCL algorithm.

Additionally, Table 3 contains the quantitative results of the localization experiments in both of the simulated and real-world environments.

Table 3: Results of localization experiments via Gmapping algorithm

best credibility has been gained (in %)	Total time till to locate with the best credibility	good credibility has been gained (in %)	Time (in sec.) to locate with a good credibility	area (m^2)	The presence of obstacles	Guidance type
Simulation in ROS-Gazebo environment						
90	45	70	31	20	**static**	**manual**
95	289	60	44	20	**static**	**DWA**
Implementation in real-world environment						
90	42	70	28	13	**static**	**manual**
95	33	70	11	13	**static**	**DWA**

It can be observed that in both the simulated environment (shown in Figure 5) and the real-world environment (shown in Figures 8 and 9), the performance of the TurtleBot3 was satisfactory for mapping the environment within a reasonable time and for achieving localization under the SLAM_Gmapping framework. By comparing the results given in Tables 2 and 3, however, it can be seen that mapping in the real-world environment took more time, whereas the opposite was true for localization, meaning that the robot reached the target point faster in the real-world environment than in the simulated environment. Furthermore, in all cases, manual guidance performed the tasks in less time than automatic guidance because the operator was aware of the environment being explored. On the other hand, automatic guidance provided more extensive environmental information and generated more detailed maps than manual guidance.

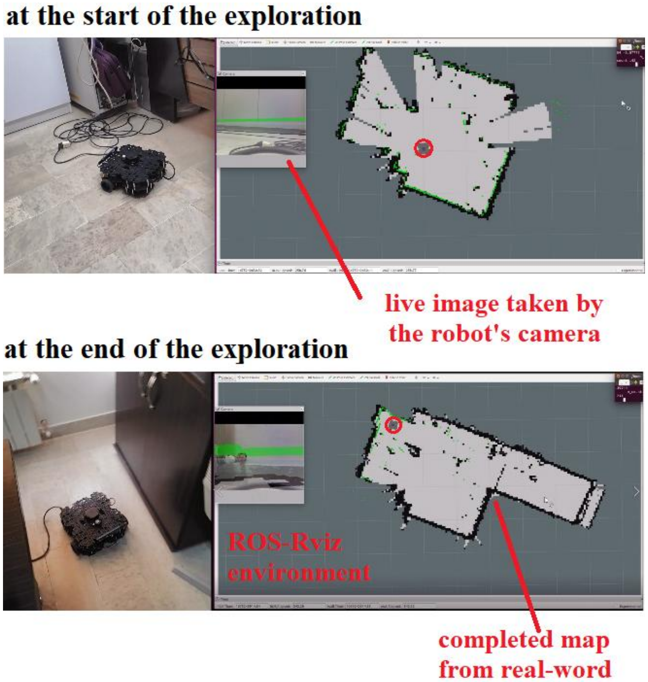


Fig. 6. Mapping under manual guidance in the real-world environment.

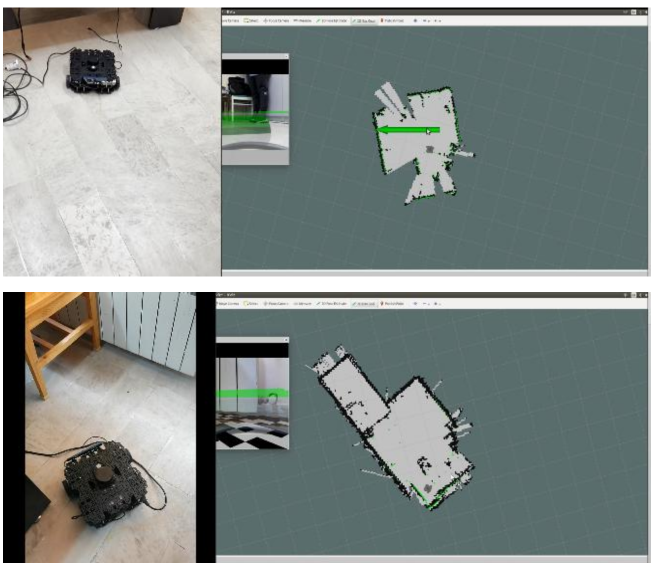


Fig. 7. Mapping under automatic guidance in the real-world environment.

7. CONCLUSIONS

In this paper, we addressed the implementation and evaluation of the SLAM_Gmapping algorithm under ROS for guiding a differential-wheeled TurtleBot3 robot. This objective was achieved through both simulations in the ROS-Gazebo environment and experimental tests in a real-world environment. The robot's capabilities for mapping and localization were evaluated through separate experiments. In each case, the robot was tested in two modes: manual and automatic guidance. In the automatic mode, the DWA algorithm was used for motion guidance. Static obstacles were present in both the simulated and real-world environments, and the mapping and localization capabilities of SLAM_Gmapping were evaluated and compared in these environments.

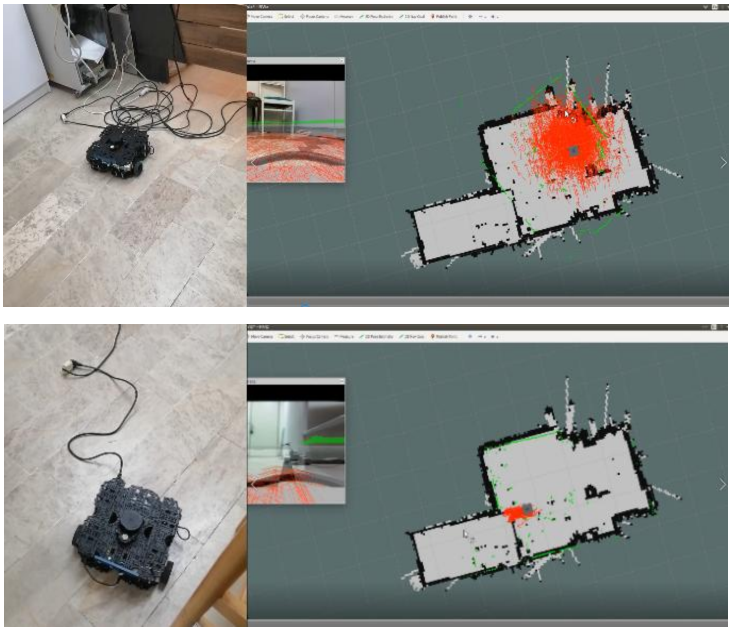


Fig. 8. Localization under manual guidance in the real-world environment.

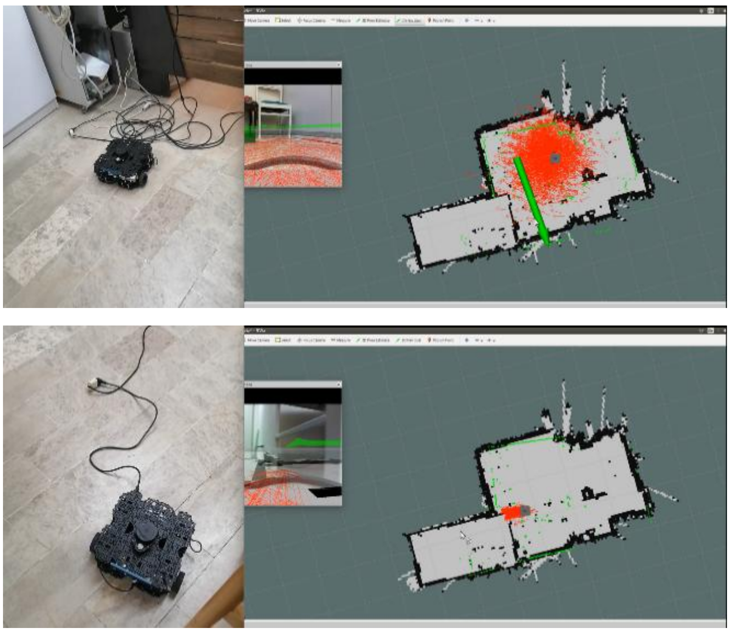


Fig. 9. Localization under automatic guidance in the real-world environment.

The mapping experiment in the simulated environment in the presence of a moving obstacle, controlled using the DWA algorithm, was also conducted, and the results showed that the robot was able to detect dynamic obstacles and avoid collisions with them. However, the time required for this task in the simulation was significantly long, indicating the need for further studies to reduce the duration of the process. The comparison of these experimental outcomes indicates consistent and successful navigation of the robot in both the ROS-Gazebo simulated and real-world environments, with mapping and robot localization evaluated separately.

Overall, it can be concluded that the information obtained from these experiments can be useful in the development of robots with better and more precise navigation capabilities in complex environments. Additionally, these results can aid in the development of improved methods for detecting and avoiding obstacles, particularly moving obstacles.

Declaration

We acknowledge that we used ChatGPT to enhance the academic writing of our manuscript while ensuring the originality and integrity of our work.

Transparency Statement

The data supporting this study are available upon reasonable request to the corresponding author, subject to ethical and confidentiality considerations.

Acknowledgments

We would like to express our gratitude to all individuals who contributed to this project.

Declaration of Interest

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

REFERENCES

[1] Taheri, H., & Xia, Z. C. (2021). SLAM; definition and evolution. *Engineering Applications of Artificial Intelligence, 97*, 104032. https://doi.org/10.1016/j.engappai.2020.104032

[2] Alsadik, B., & Karam, S. (2021). The simultaneous localization and mapping (SLAM): An overview. *Journal of Applied Science and Technology Trends, 2*(2), 147–158. https://doi.org/10.38094/jastt204117

[3] ROBOTIS. (n.d.). *TurtleBot3*. ROBOTIS e-Manual. Retrieved September 10, 2026, from ROBOTIS TurtleBot3 e-Manual

[4] Aydemir, H., Tekerek, M., & Gök, M. (2021). Examining of the effect of geometric objects on SLAM performance using ROS and Gazebo. *El-Cezeri Journal of Science and Engineering, 8*(3), 1441–1454. https://doi.org/10.31202/ecjse.943364

[5] Spina, M. G., Gualtieri, S., & De Rango, F. (2021). Integrating ROS and Gazebo tools with a network security module to support secure autonomous robot coordination. In *Proceedings of the 11th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2021)* (pp. 369–377). SciTePress. https://doi.org/10.5220/0010566303690377

[6] Norzam, W. A. S., Hawari, H. F., & Kamarudin, K. (2019). Analysis of mobile robot indoor mapping using GMapping based SLAM with different parameter. *IOP Conference Series: Materials Science and Engineering, 705*(1), 012037. https://doi.org/10.1088/1757-899X/705/1/012037

[7] Rojas-Fernández, M., Mújica-Vargas, D., Matuz-Cruz, M., & López-Borreguero, D. (2018). Performance comparison of 2D SLAM techniques available in ROS using a differential drive robot. In *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)* (pp. 50–58). IEEE. https://doi.org/10.1109/CONIELECOMP.2018.8327175

[8] Yang, X. (2021). SLAM and navigation of indoor robot based on ROS and lidar. *Journal of Physics: Conference Series, 1748*(2), 022038. https://doi.org/10.1088/1742-6596/1748/2/022038

[9] Chen, S., Zhou, W., Yang, A.-S., Chen, H., Li, B., & Wen, C.-Y. (2022). An end-to-end UAV simulation platform for visual SLAM and navigation. *Aerospace, 9*(2), 48. https://doi.org/10.3390/aerospace9020048

[10] Ali, W., Sheng, L., & Ahmed, W. (2021). Robot Operating System-based SLAM for a Gazebo-simulated Turtlebot2 in 2D indoor environment with Cartographer algorithm. *International Journal of Mechanical and Mechatronics Engineering, 15*(3), 149–157. Publication page

[11] Singandhupe, A., & La, H. M. (2019). A review of SLAM techniques and security in autonomous driving. In *2019 Third IEEE International Conference on Robotic Computing (IRC)* (pp. 602–607). IEEE. https://doi.org/10.1109/IRC.2019.00122

[12] Fan, X., Wang, Y., & Zhang, Z. (2021). An evaluation of Lidar-based 2D SLAM techniques with an exploration mode. *Journal of Physics: Conference Series, 1905*(1), 012021. https://doi.org/10.1088/1742-6596/1905/1/012021

[13] Thrun, S. (2008). Simultaneous localization and mapping. In M. E. Jefferies & W.-K. Yeap (Eds.), *Robotics and cognitive approaches to spatial mapping* (pp. 13–41). Springer. https://doi.org/10.1007/978-3-540-75388-9_3

[14] Stachniss, C., Leonard, J. J., & Thrun, S. (2016). Simultaneous localization and mapping. In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics* (2nd ed., pp. 1153–1176). Springer. https://doi.org/10.1007/978-3-319-32552-1_46