# SLAM AND NAVIGATION APPLICATION WITH CAMERA-INTEGRATED TURTLEBOT3 BURGER

**Dilara GALELİ1, Enes FİDE2, Dr. Öğr. Üy. Kamil ÇETİN1,3,\*, Dr. Öğr. Üy. Fatih Cemal CAN2,#**

**1 İzmir Katip Çelebi University, Faculty of Engineering and Architecture, Department of Electrical and Electronics Engineering, İzmir, Türkiye**

**2 İzmir Katip Çelebi University, Faculty of Engineering and Architecture, Department of Mechatronics**

**Engineering, İzmir, Türkiye**

**ORCID#: 0000-0002-6023-5464**

**Email#: fatihcemal.can@ikcu.edu.tr**

**3 İzmir Katip Çelebi University, Akıllı Fabrika Sistemleri Araştırma ve Uygulama Merkezi, İzmir,** **Türkiye**

**ORCID\*: 0000-0003-1029-5626**

**Email\*: kamil.cetin@ikcu.edu.tr**

**Abstract**

Mapping and navigation are important topics of research and development in the field of robotics, enabling robots to move autonomously in unknown environments. In this study, the Turtlebot3 Burger robot was utilized for Simultaneous Localization and Mapping (SLAM) and navigation application using a LiDAR in both Robot Operating System (ROS)-based Gazebo simulation environment and real world. The Turtlebot3 Burger robot is equipped with a LiDAR sensor, an IMU, four omni-wheels, a ball caster, two Dynamixel motors, an OpenCR controller and a Raspberry Pi 3 running Raspberry Pi OS and ROS Melodic. In addition, a camera was integrated for visualization of the environment. Communication between the robot and computer was estabilished through a Wi-Fi network, with the computer acting as the master node in the ROS network. For the SLAM process, Gmapping algorithm was used. The algorithm utilized the LiDAR sensor data for creating the 2D map of the environment and simultaneously localizing the robot within the map. The generated map at this stage allowed the robot to navigate through the environment afterwards. The system uses ROS packages of the Turtlebot3 for processing the LiDAR data, implementing SLAM Gmapping algorithm and enabling navigation in both Gazebo simulation environment (using the Xacro file of the robot) and real world. The primary goals of this study include navigating through the environment without any collision, following determined paths and successfully reaching the target points along the planned route.

**Keywords:** SLAM, Navigation, Turtlebot3 Burger, ROS

**KAMERA ENTEGRELİ TURTLEBOT3 BURGER İLE SLAM VE NAVİGASYON UYGULAMASI**

**Özet**

Haritalama ve navigasyon, robotik alanında araştırma ve geliştirmenin önemli konularından olup robotların bilinmeyen ortamlarda otonom olarak hareket edebilmelerini sağlar. Bu çalışmada, Turtlebot3 Burger robotu bir LiDAR kullanılarak hem Robot İşletim Sistemi (ROS) tabanlı Gazebo simülasyon ortamında hem de gerçek dünyada Eşzamanlı Yerelleştirme ve Haritalama (SLAM) ve navigasyon uygulaması için kullanılmıştır. Turtlebot3 Burger robotu, bir LiDAR sensörü, bir IMU, dört çok yönlü tekerlek, bir kastor tekerlek, iki Dynamixel motoru, bir OpenCR kontrol kartı ve Raspberry Pi OS ile ROS Melodic çalıştıran bir Raspberry Pi 3 ile donatılmıştır. Ayrıca, robota ortamın görselleştirilmesi için bir kamera entegre edilmiştir. Robot ve bilgisayar arasındaki iletişim, bilgisayarın ROS ağında ana düğüm görevi gördüğü bir Wi-Fi ağı üzerinden kurulmuştur. SLAM süreci için Gmapping algoritması kullanılmıştır. Algoritma, ortamın 2 boyutlu haritasını oluşturmak ve robotu harita içinde eş zamanlı olarak konumlandırmak için LiDAR sensör verilerini kullanmıştır ve bu aşamada oluşturulan harita robotun daha sonra ortamda gezinmesine olanak sağlamıştır. Sistem, LiDAR verilerini işlemek, SLAM Gmapping algoritmasını uygulamak ve hem Gazebo simülasyon ortamında (robotun Xacro dosyasını kullanarak) hem de gerçek dünyada gezinmeyi sağlamak için Turtlebot3'ün ROS paketlerini kullanmaktadır. Bu çalışmanın birincil hedefleri, robotun herhangi bir çarpışma olmadan ortamda gezinmesi, belirlenen yolları takip etmesi ve planlanan rota boyunca hedef noktalara başarıyla ulaşmasıdır.

**Anahtar Kelimeler:** SLAM, Navigasyon, Turtlebot3 Burger, ROS

1. **Introduction**

In recent years, autonomous mobile robots have become crucial in various areas, including industrial automation, smart agriculture and domestic assistance. A critical aspect of robot autonomy is the ability to map and navigate through unknown environments while avoiding obstacles, achieved through Simultaneous Localization and Mapping (SLAM) and navigation algorithms.

SLAM enables a robot to generate a map of the environment while simultaneously determining its position within it (Qu et al., 2021). Features of the environment such as walls, obstacles and open spaces can be accurately detected and represented in the map. This study focuses on implementing Gmapping SLAM technique and enabling autonomous navigation on the Turtlebot3 Burger robot using the Robot Operating System (ROS). ROS offers a modular framework for robotic application development, facilitating efficient communication between nodes and integration of hardware with software components. Its libraries provide robust tools for sensor data processing and navigation tasks, making it an essential platform for this study. By utilizing the computational efficiency of the Gmapping algorithm (Grisetti et al., 2007), the robot generated precise 2D maps in both simulated and real-world environments. The objectives of this study include accurate localization, efficient path planning and collision-free navigation, further enhanced by the integration of an Intel Realsense D455 camera for environmental visualization (Zhang et al., 2014).

1. **Materials and Method**
   1. **Turtlebot3 Burger**

The robot that was used in this study is the Burger model of Turtlebot3. The Turtlebot3 Burger is a compact and versatile mobile robot, which has a 360° LiDAR sensor for precise distance measurements, an IMU for motion and orientation tracking, differential drive wheels for smooth mobility, Dynamixel motors for drive control and a ROS-powered system. Turtlebot3 is a popular choice for educational and research purposes in the field of robotics because of its small size and lightweight design. By utilizing the LiDAR sensor, Turtlebot3 can accurately scan and map its surroundings, enabling it to perform Simultaneous Localization and Mapping (SLAM) and navigate autonomously within its environment. With the help of the differential drive wheels, Turtlebot3 Burger can perform smooth linear and rotational motions, which enable it to execute precise maneuvers during SLAM process and navigation tasks. The robot operates on Robot Operating System (ROS), which provides a flexible framework for programming, monitoring, processing sensor data, controlling the robot’s function and facilitating the communication between robot’s hardware components and software algorithms.

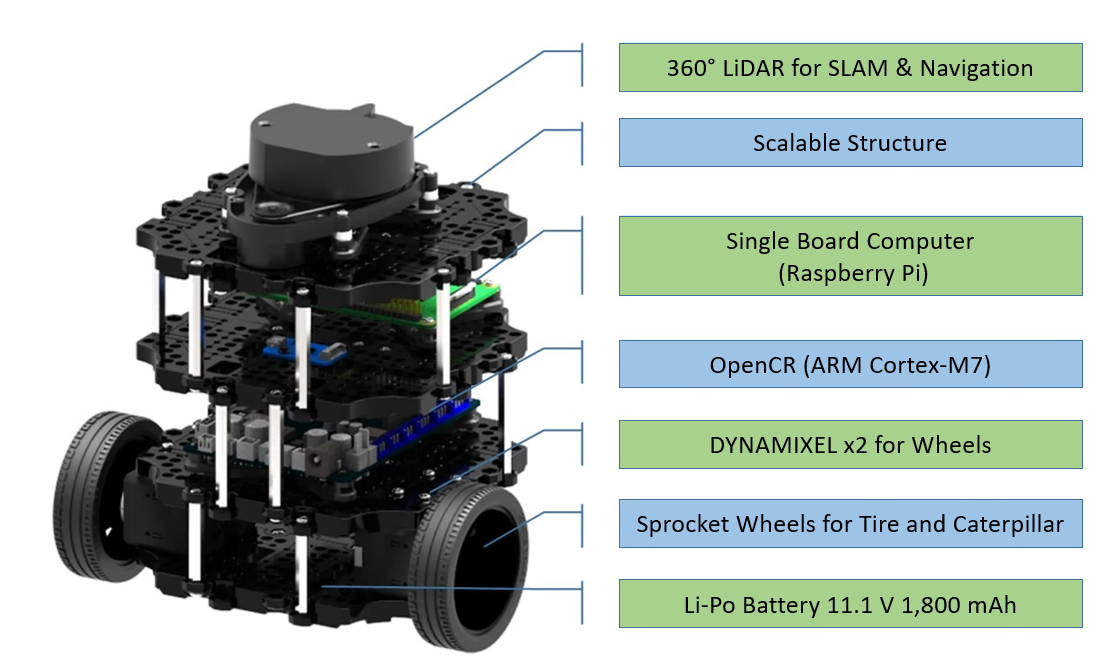
****

Figure 1. Features of Turtlebot3 Burger

* 1. **Raspberry Pi**

Raspberry Pi is a small single-board computer commonly used in robotics applications due to its small size, cost-effectiveness and high performance. It has a series of GPIO pins for interfacing with external devices and USB ports for connections. In addition, it supports various operating systems such as Raspberry Pi OS, which is Linux-based. Since ROS is generally preferred in Linux-based systems, Raspberry Pi is the ideal platform for robotic control and real-time data processing. With its robust features and versatility, Raspberry Pi serves as the main control unit for robotics projects, enabling communication between sensors, actuators and other components. In this study, the Turtlebot3 Burger robot was controlled on computer using Raspberry Pi 3.

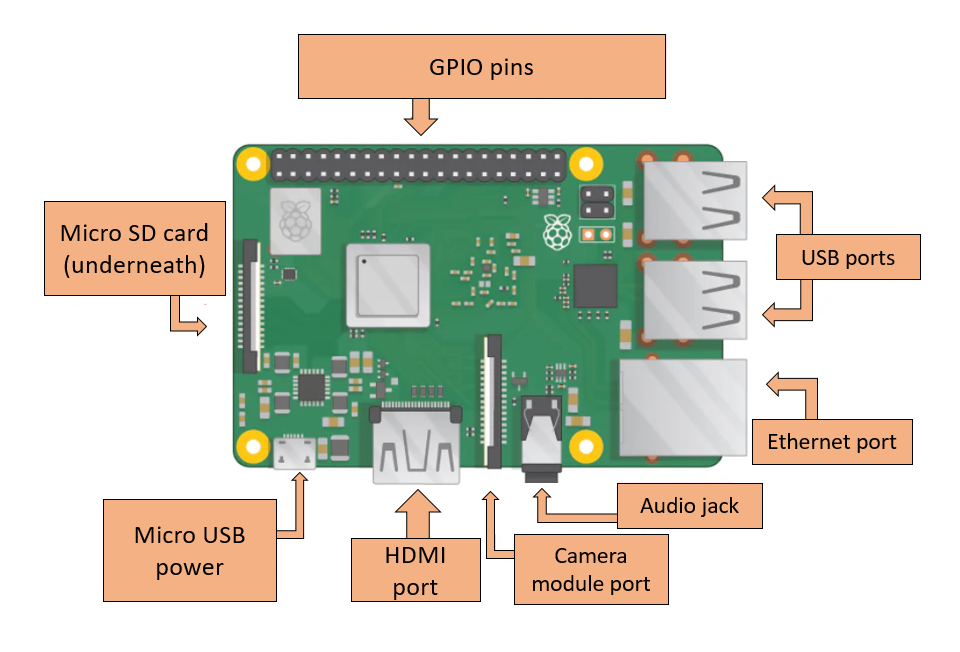


Figure 2. Raspberry Pi 3

* 1. **Intel Realsense D455 Camera**

For viewing the robot’s surroundings in the real world through RViz, an Intel Realsense D455 camera was added to the robot.

* 1. **Simulation**

For the simulation part of the study, the Xacro (XML Macro) file of Turtlebot3 Burger was used for displaying, controlling the robot and enabling SLAM and navigation in Gazebo and RViz. An Intel Realsense D455 camera was added on top of the robot in this Xacro file for visualization. The test simulation was conducted in the Turtlebot3 World environment.



Figure 3. Turtlebot3 Burger with Camera in Gazebo Simulation

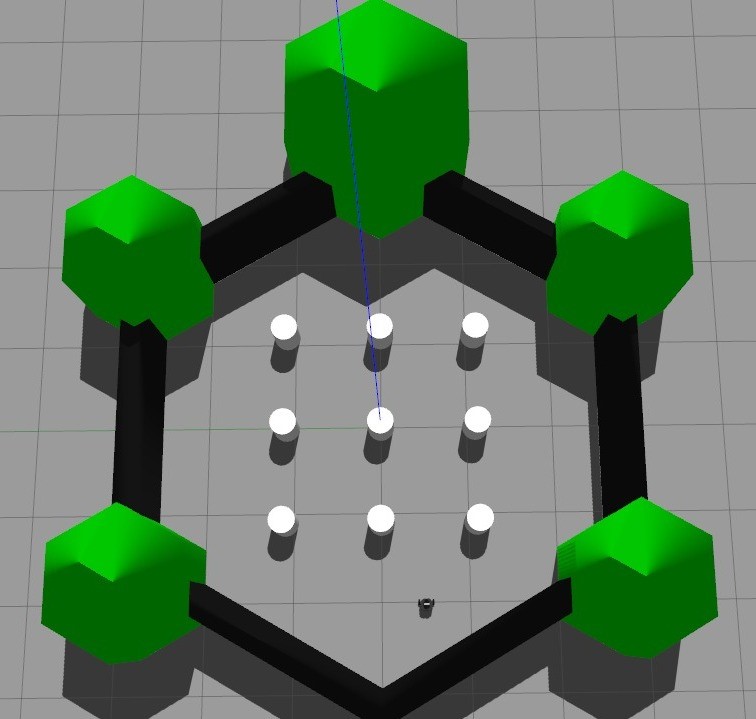


Figure 4. Turtlebot3 World

* 1. **SLAM Gmapping and Navigation**

Simultaneous Localization and Mapping (SLAM) is a technique that allows robots to build a map of an unknown environment while simultaneously determining their position within that map, typically using laser scans obtained from a LiDAR sensor. Gmapping is a SLAM algorithm that utilizes the Rao Blackwellized particle filter (RBPF) to generate occupancy grid maps from laser data (Qu et al., 2021). Gmapping is often preferred in robotic systems with limited processing power because it uses minimal resources to represent the SLAM posterior effectively and minimizes the computations while generating highly accurate maps. Since the Raspberry Pi has limited processing power, the Gmapping SLAM algorithm is preferred in this study to generate an accurate map of the robot’s surroundings.

For the test simulation, the Turtlebot3 Burger model in Turtlebot3 World was launched in Gazebo simulation environment. SLAM Gmapping algorithm was used for mapping process in RViz. The generated world map, given in Figure 5, was saved for the navigation operations afterwards.

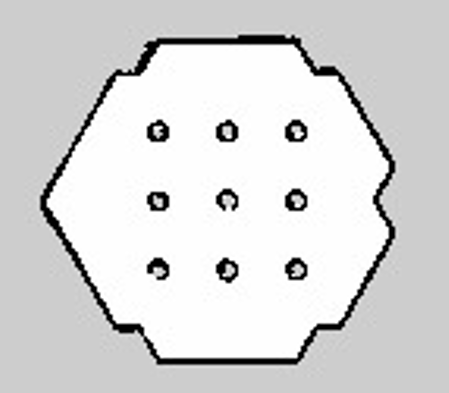


Figure 5. Generated Map of Turtlebot3 World

After the SLAM operation, the navigation process was enabled on the generated world map. When the launch file for the navigation process is run, if the robot does not spawn at position (0,0,0) on the map with the executed command, it is brought to the starting point (0,0,0) using pose estimation via RViz. This helps to prevent potential collisions and errors in the robot's position perception.

Firstly, a 2D navigation goal was given in RViz and the robot is sent to a specific location on the map, which was scanned and detected by Turtlebot3 Burger. A route is generated for the robot to reach this goal in RViz. If an obstacle (scanned with LiDAR sensor) is encountered on the determined route, it will be shown in RViz and a new route will be determined by the robot to reach the desired position without any collision. Another objective is for the robot to navigate through waypoints on the specified route and eventually return to the starting point. A Python script was created for this purpose and this script allows unlimited number of waypoints to be added, enabling the robot to trace a desired trajectory.

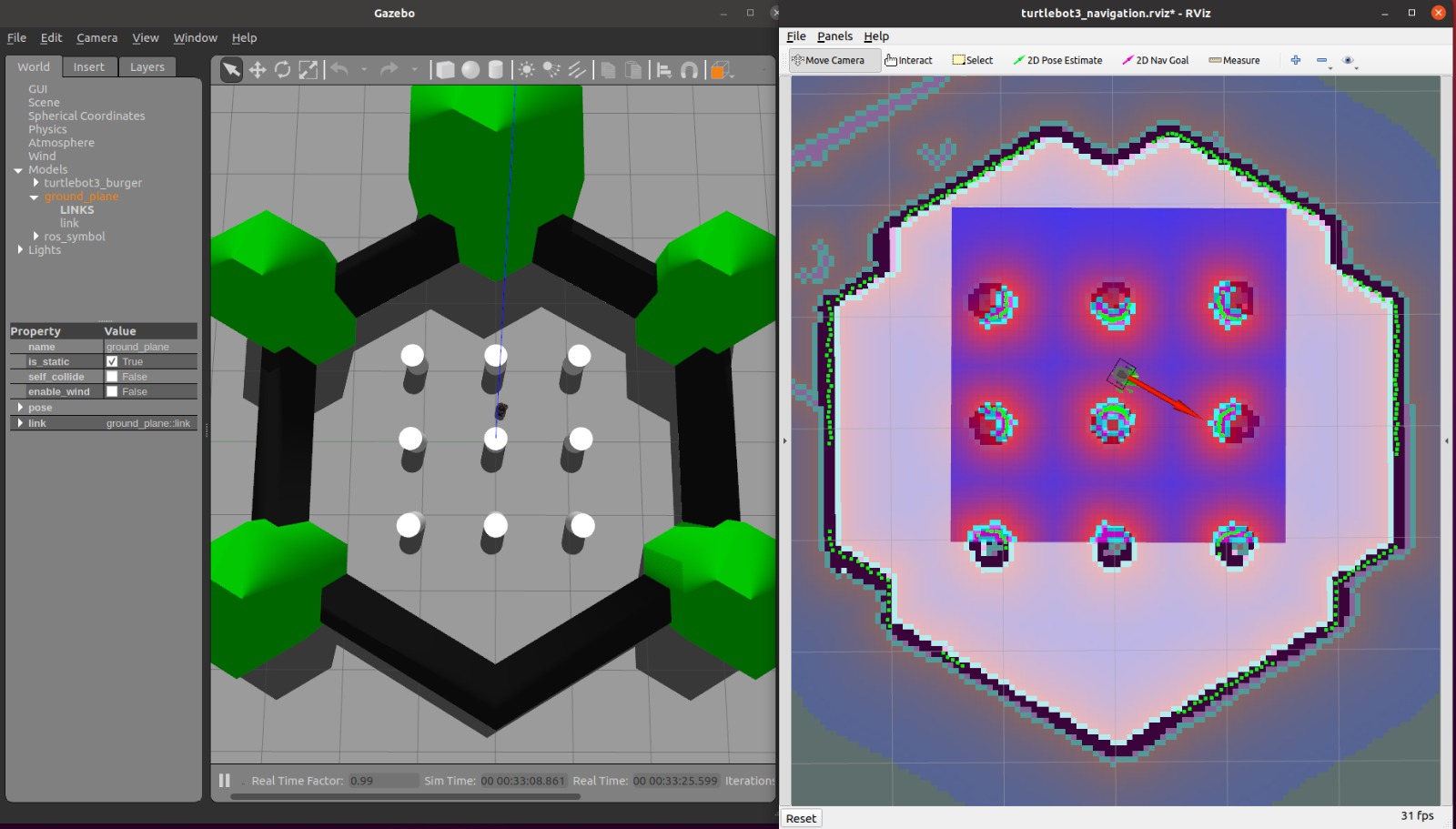


Figure 6. Navigation Process in Gazebo (Left) and RViz (Right)

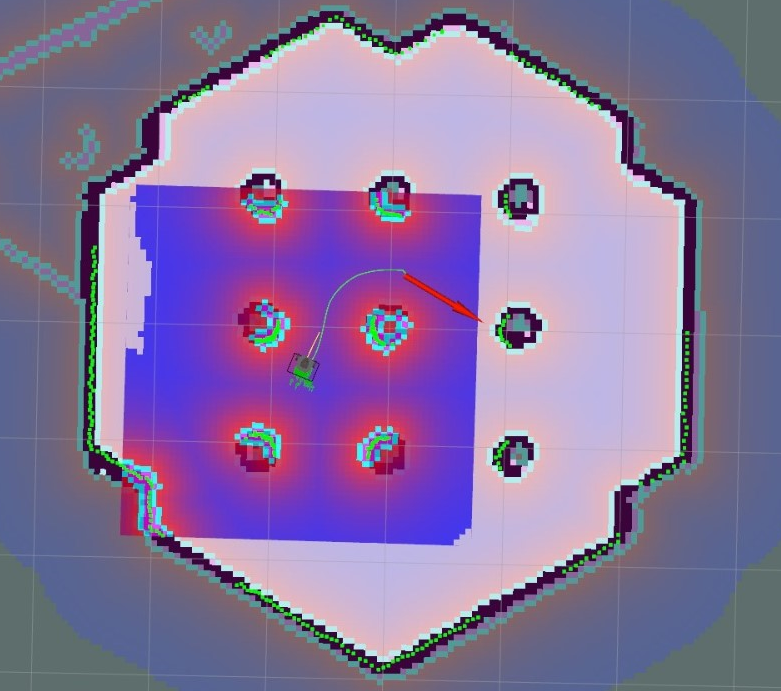


Figure 7. Paths of the Global Planner (Green) and Local Planner (Yellow)

The main topics involved in mapping are ‘/cmd\_vel’, ‘/odom’ and ‘/scan’.

* /cmd\_vel topic: Used for sending linear and angular velocity commands with keyboard for controlling the motion of the robot.
* /odom topic: Provides odometry information, which estimates the robot’s position and orientation.
* /scan topic: Publishes laser\_scan data, which is obtained from LiDAR sensor. Used for obstacle detection and mapping.

For the navigation process, ‘AMCL’ and ‘move\_base’ packages inside the ROS Navigation Stack were utilized. The ROS Navigation Stack is a collection of software packages that allows autonomous robots to perform navigation tasks such as localization, path planning and obstacle avoidance (Walenta el al., 2017).

AMCL (Adaptive Monte Carlo Localization) is a probabilistic localization system used in ROS to estimate a robot's pose (position and orientation) within a known map. It subscribes sensor data topics such as ‘/scan’ and ‘/map’ and publishes topics such as ‘amcl\_pose’. With AMCL, robot was able to understand where it was located within the map that was generated. The move\_base package in ROS integrates the path planning and obstacle avoidance processes, which enables robots to move from the initial point to the goal point without colliding with obstacles in the environment (Chen et al., 2023). In short, AMCL is responsible for estimating the robot’s pose within a map while move\_base uses this pose information to plan the robot’s trajectory to the goal point, ensuring obstacles are avoided along the path.

In real-life tests, Turtlebot3 Burger and the computer were first connected to the same Wi-Fi network to ensure smooth communication between them. After the IP addresses were matched, the real world tests were started by using the bringup package of Turtlebot3. Once the robot is launched, the map of the real environment is created by scanning the surroundings of the robot. In mapping process, the robot is moved using teleoperation commands to scan the area for accurate representation of the environment in the map. After the mapping process is finished, the map is saved and navigation process is initiated. Robot’s position is adjusted in RViz using pose estimation to determine its real location within the mapped area. A 2D navigation goal in RViz is given to observe the robot’s capabilities for safe navigation. After ensuring that the robot can navigate safely in the environment, the Python script for sending the robot to specific waypoints is run. After reaching these designated waypoints, the robot returned to its starting point.



Figure 8. Turtlebot3 Burger on Real Environment

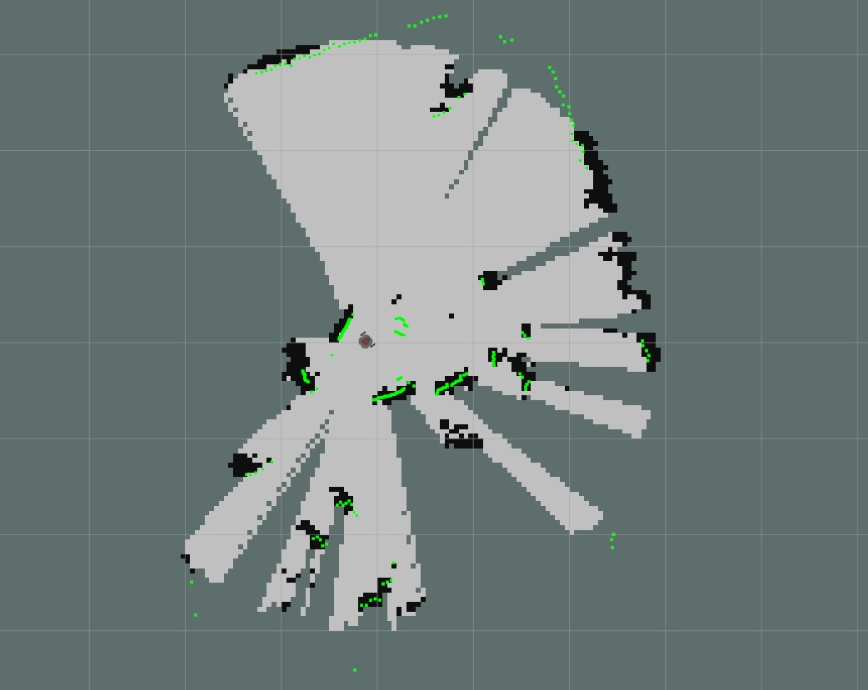


Figure 9. Generated Map of the Environment in RViz

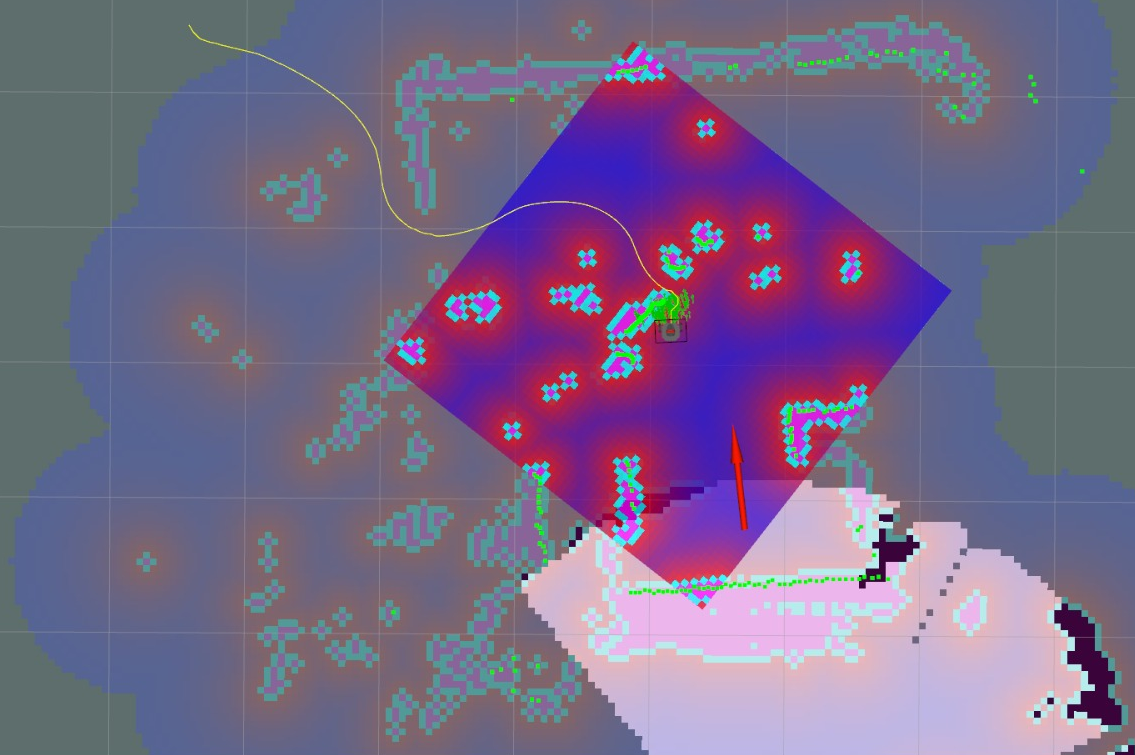


Figure 10. Navigation in Scanned Map

1. **Results**

The study successfully demonstrated SLAM and navigation capabilities using the Turtlebot3 Burger robot. The Gmapping algorithm effectively utilized LiDAR data to create detailed 2D maps, ensuring accurate localization. In simulation tests conducted in the Gazebo environment, the robot navigated efficiently, avoiding obstacles and following planned trajectories to reach predefined waypoints. These results were mirrored in real-world tests, where the robot adhered to its planned routes and safely returned to its starting point. Localization accuracy was achieved using Adaptive Monte Carlo Localization (AMCL), while the move\_base package ensured dynamic path adjustments and obstacle avoidance. The system's integration with a camera enhanced its ability to monitor the environment, demonstrating its robustness for practical applications (Bresson et al., 2017).

1. **Conclusion**

This study highlights the effectiveness of the SLAM Gmapping algorithm and the ROS Navigation Stack in enabling autonomous navigation on the Turtlebot3 Burger robot. By combining LiDAR-based 2D mapping with precise localization and path planning, the robot successfully navigated through environments in both simulation environment and real-world scenarios. The integration of the camera enhanced the visualization of the environment. The findings underline the potential of low-cost, ROS-based robotic platforms for research and real-world applications. Future work could expand the system's capabilities by incorporating 3D mapping techniques and additional sensors to enhance performance in more complex and dynamic environments.

**Acknowledgment**

TÜBİTAK 2209-B, “Akıllı Tarım Teknolojisinde Otonom Mobil Robot Kolu ile Seralarda İnsansız Hasat Uygulaması”, Project No. 1139B412302608, 2023. This project is supported by TÜBİTAK. The authors thank TÜBİTAK for their supports.

**References**

1. Qu, P., Su, C., Wu, H., Xu, X., Gao, S., & Zhao, X. (2021). Mapping performance comparison of 2D SLAM algorithms based on different sensor combinations. *Journal of Physics: Conference Series*. DOI: 10.1088/1742-6596/2072/1/012003.

2. Chen, S. P., Peng, C. Y., Huang, G. S., Lai, C. C., Chen, C. C., & Yen, M. H. (2023). Comparison of 2D and 3D LiDARs Trajectories and AMCL Positioning in ROS-Based move\_base Navigation. In *2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pp. 1-6. IEEE. DOI: 10.1109/COINS57878.2023.10124567.

3. Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1), 34–46. DOI: 10.1109/TRO.2006.889486.

4. Zhang, J., & Singh, S. (2014). LOAM: Lidar Odometry and Mapping in real-time. In *Robotics: Science and Systems Conference (RSS)*. DOI: 10.15607/RSS.2014.X.00157.

5. Walenta, R., Schellekens, T., Ferrein, A., & Schiffer, S. (2017). A decentralized system approach for controlling AGVs with ROS. In *2017 IEEE AFRICON*, pp. 1436-1441. IEEE. DOI: 10.1109/AFRCON.2017.8095694.

6. Bresson, G., Alsayed, Z., Yu, L., & Glaser, S. (2017). Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3), 194–220. DOI: 10.1109/TIV.2017.2749181.