Sigma J Eng Nat Sci, Vol. 43, No. 4, pp. 1233−1247, August, 2025

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2025.00117

Research Article

Precision mapping and navigation: A robotic restaurant management system using SLAM and ROS

Lalit N. PATIL[1,*], Swapnil S. JADHAV[1], Kishor B. WAGHULDE[2], Prathamesh GAIKWAD[1], Shubham DUCHAL[1], Swapnil PATIL[1], Namrata FULSHETE[1], Madhuri CHATUR-DEOKAR[3]

[1]*Department of Automation and Robotics, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, 411033, India*
[2]*Department of Mechanical Engineering, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, 411033, India*
[3]*Department of Mechanical Engineering, JSPM Narhe Technical Campus, Narhe, Pune, 411041, India*

ARTICLE INFO

Article history
Received: 22 April 2024
Revised: 08 July 2024
Accepted: 23 July 2024

Keywords:
Mapping; Navigation; Restaurant Robot; Service Bot; Service Robot; SLAM

ABSTRACT

SLAM system that enables an autonomous robot to efficiently navigate and map a wide range of locations. Numerous industries, such as driverless vehicles, warehouse logistics, search and rescue operations, industrial automation, and more uses the suggested SLAM-based robot. The present research work tackles the vital need for precise mapping and self-localization in dynamic environments by seamlessly merging sensor data with cutting-edge algorithms, thereby contributing to the growth of automation and robotics in a variety of industries. This work demonstrate the seamless integration of Robot Operating System (ROS) with a suite of hardware components, including LiDAR (Light Detection and Ranging), IR (Infrared) sensors, motors, motor drivers, and Raspberry Pi to create a comprehensive autonomous robotics platform. The ROS framework serves as the central nervous system, enabling real-time data processing, sensor fusion, and motor control for diverse applications such as navigation, obstacle avoidance, and environmental mapping. The findings of this study provide a summary of the essential elements and their functions in creating a flexible and competent autonomous robot, showing the potential for developments in areas like robotics, automation, and AI-driven systems.

Cite this article as: Patil LN, Jadhav SS, Waghulde KB, Gaikwad P, Duchal S, Patil S, Fulshete N, Chatur-Deokar M. Precision mapping and navigation: A robotic restaurant management system using SLAM and ROS. Sigma J Eng Nat Sci 2025;43(4):1233−1247.

INTRODUCTION

The field of robotics has grown tremendously in the previous few decades. Autonomous robotic systems promise to completely transform the way humans work by automating risky and repetitive activities, increasing productivity, and addressing the labour crisis [1]. This technology is driven by the increasing availability of hardware as well as the shrinking size and cost of computer resources. Numerous industries, like warehousing, have previously shown how robotic automation may increase both production and safety [2]. Robotic automation has not found on in the

***Corresponding author.**
*E-mail address: lalit.patil@dypvp.edu.in

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

construction industry, though. The dynamic nature of construction sites poses an obstacle to adoption, as it makes infrastructure setup challenging [3]. Therefore, in order to offer more flexibility, transportable systems are required.

Since human-robot interaction is a natural means of communication, it is crucial to the functioning of service robots. There are numerous ways for humans and robots to communicate that been suggested, like voice interaction, facial recognition, and sensor motor interaction. For a highly intelligent service robot, its capacity to locate itself and navigate to a destination within the workplace is important [4]. There are several uses for gesture recognition, such as in social assistive robotics (SAR), human robot interaction (HRI), and human machine interaction (HMI) [5]. The capacity of an autonomous robot to construct a map and locate itself while concurrently navigating in an unfamiliar environment is known as among the most important areas in robotics. The scientific community has focused on simultaneous localization and mapping (SLAM) for more than 20 years, and it is crucial in many applications in real life. The derived terrain maps may prove to be quite beneficial for these kinds of applications and systems, which also involve robot navigation. Stereo or monocular vision-based SLAM techniques are gaining popularity[6]. Modern methods were used, with vision being the average of how the actual surroundings. Human vision typically perceives just two dimensions, with the third dimension commonly referred to as depth being experienced.

ROBOTIC RESTAURANT MANAGEMENT SYSTEM

Robotic restaurant management system consist of following

A. The Robot Operating System: It is the cornerstone of the TurtleBot's software architecture. It offers an adaptable framework for controlling hardware abstraction, inter-component communication, and development tools [7]. The software overview is shown in figure 1.

B. Node-base Architecture: The software components of ROS are arranged as nodes that interact with one another via topics and services. This architecture is built on nodes. This modular design facilitates easy integration of various functionality and scalability [8].

C. Sensor Data Processing:- Data processing modules for sensors installed on the TurtleBot, including LiDAR, cameras, and IMUs, are part of the software. For perception tasks including mapping, localization, object detection, and obstacle avoidance, this processing is essential [9].

D. Navigation Stack: The TurtleBot's paths are autonomously planned and carried out by the navigation stack. It consists of parts for motion control, map construction, localization, and path planning, allowing the robot to move around its surroundings effectively and safely [10]. The navigation system is shown in figure 2.

E. Manipulation Framework: The software design of the TurtleBot includes a manipulation framework, in the event that it is outfitted with grippers or manipulator arms. By providing kinematics, motion planning, grasping, and manipulation planning control over the manipulator, this framework lets the robot engage with its surroundings [11].

F. URDF & XACRO: The XML Macros (XACRO) file and the Unified Robot Description Format (URDF) file are crucial for defining and configuring robots in ROS 2. The URDF standardizes the XML format for describing

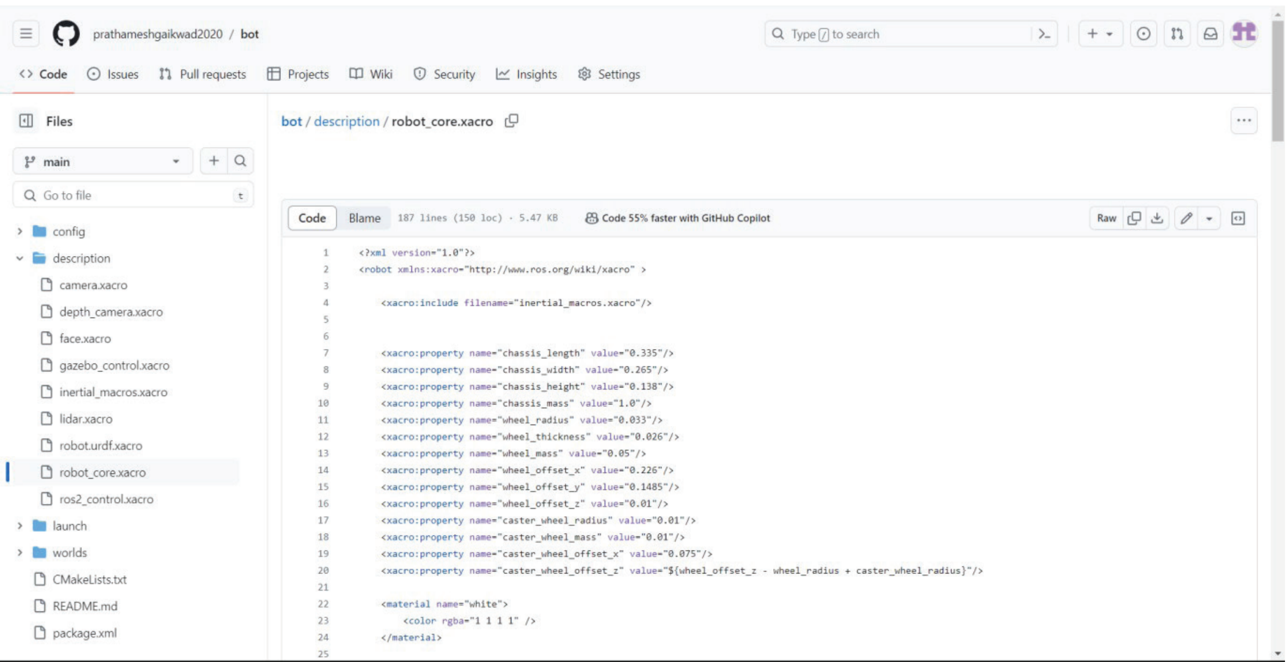


Figure 1. Software overview.

the physical components of a robot, such as its joints, connections, sensors, and other parts as depicted in figure 3. By enabling modular and customizable robot descriptions, XACRO expands upon URDF and streamlines the administration of intricate robot models [12]. The URDF and XACRO files for a SLAM robot in ROS 2 specify the robot's physical characteristics, including its base, sensors (such cameras or LiDAR), and any other parts required for mapping and localization duties as shown in figure 4. These files facilitate the mapping, localization, and navigation functions that are essential for autonomous operation by allowing ROS 2 to precisely simulate and control the robot within a SLAM environment.

G. RViz and Gazebo Simulator: Two crucial tools for robot development and simulation in ROS 2 are RViz and Gazebo Simulator. With the use of a 3D visualization tool called RViz, viewers may see real-time visualizations of sensor data, robot models, and the robot's surroundings. It is a priceless tool for troubleshooting and optimizing robot behavior since it offers a graphical interface for robot monitoring and debugging [13,14].

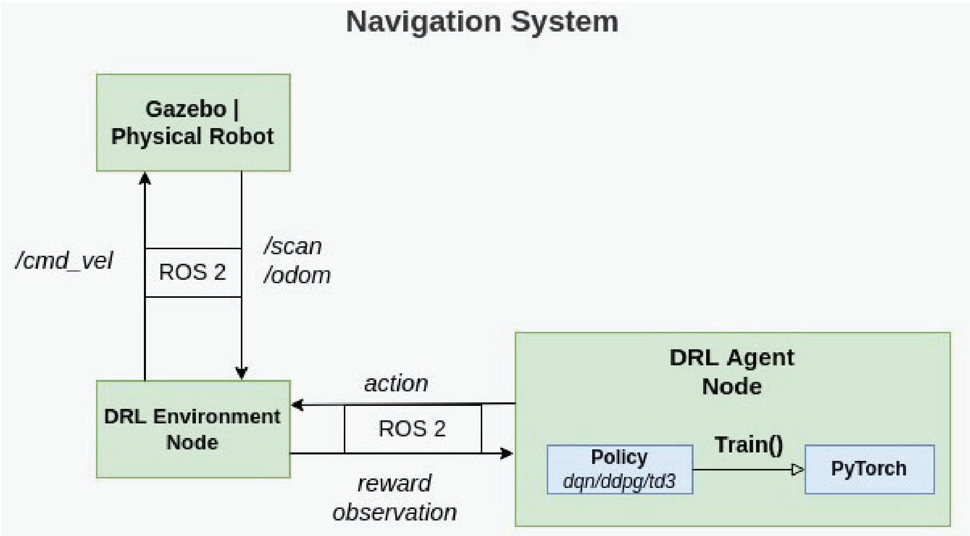
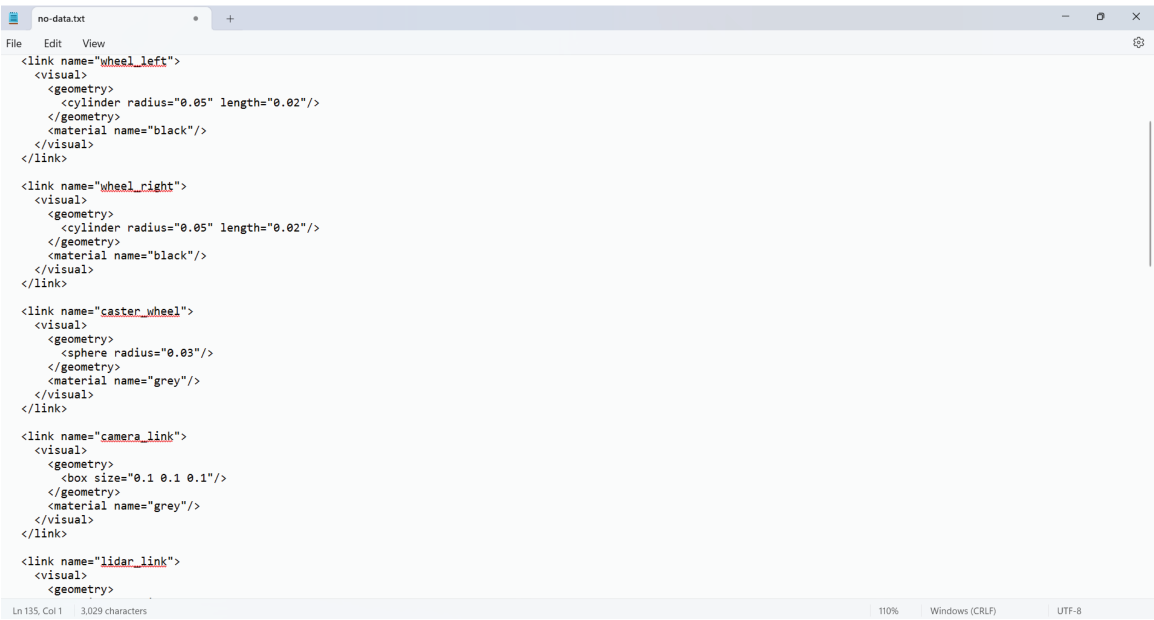


Figure 2. Navigation system.



```
<link name="wheel_left">
  <visual>
    <geometry>
      <cylinder radius="0.05" length="0.02"/>
    </geometry>
    <material name="black"/>
  </visual>
</link>

<link name="wheel_right">
  <visual>
    <geometry>
      <cylinder radius="0.05" length="0.02"/>
    </geometry>
    <material name="black"/>
  </visual>
</link>

<link name="caster_wheel">
  <visual>
    <geometry>
      <sphere radius="0.03"/>
    </geometry>
    <material name="grey"/>
  </visual>
</link>

<link name="camera_link">
  <visual>
    <geometry>
      <box size="0.1 0.1 0.1"/>
    </geometry>
    <material name="grey"/>
  </visual>
</link>

<link name="lidar_link">
  <visual>
    <geometry>
```

Figure 3. URDF describing physical components of a robot, such as its joints, connections, sensors, and other parts.

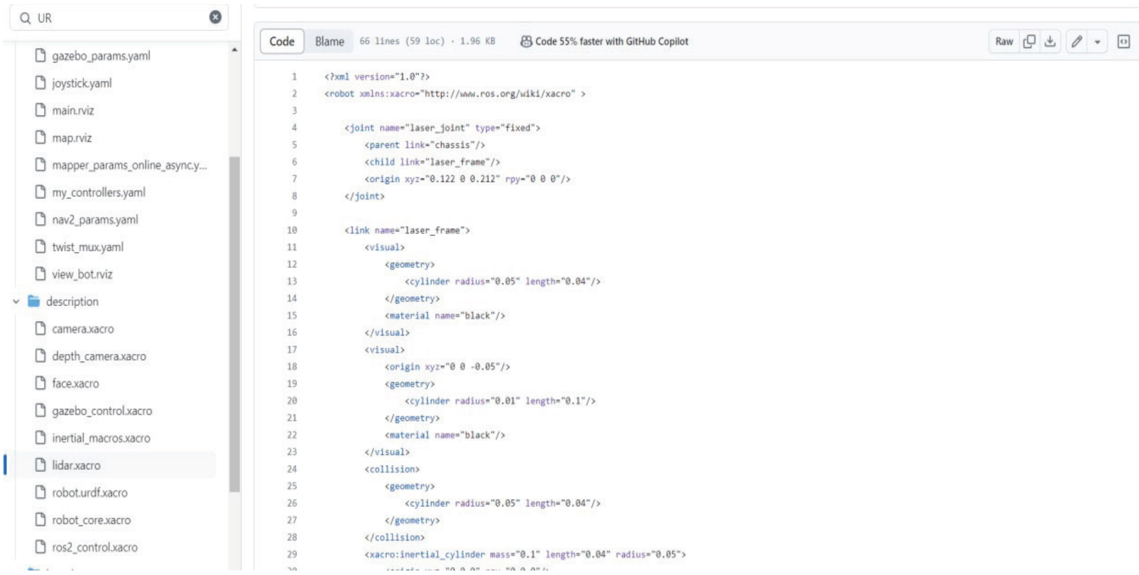


Figure 4. Xacro overview.

Conversely, users can realistically replicate robot models and their interactions with the environment with Gazebo Simulator, a physics-based simulator. Before implementing them on actual robots, it offers a platform for testing control techniques, robot algorithms, and overall system performance in a virtual setting. In ROS 2, RViz and Gazebo Simulator used together to create, test, and validate robotic systems as shown in figure 5 and figure 6.

Product Design for Restaurant Service Robot

For restaurant management robots with SLAM (Simultaneous Localization and Mapping) capabilities, computer-aided design, or CAD, is essential [15,16]. It allows for accurate manufacturing, accurate prototyping, seamless component integration, efficient team collaboration, affordable development, and flexible customization to meet particular operational needs. Engineers may ensure optimal functionality and performance by visualizing and

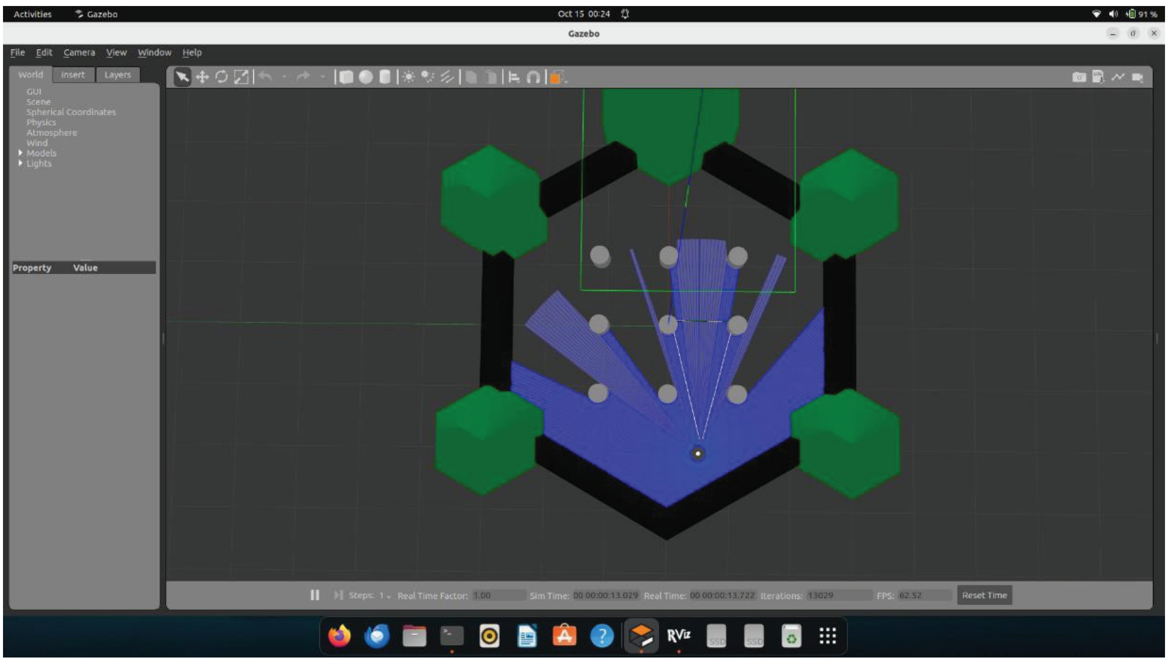


Figure 5. Gazebo simulation for robotic restaurant management model.

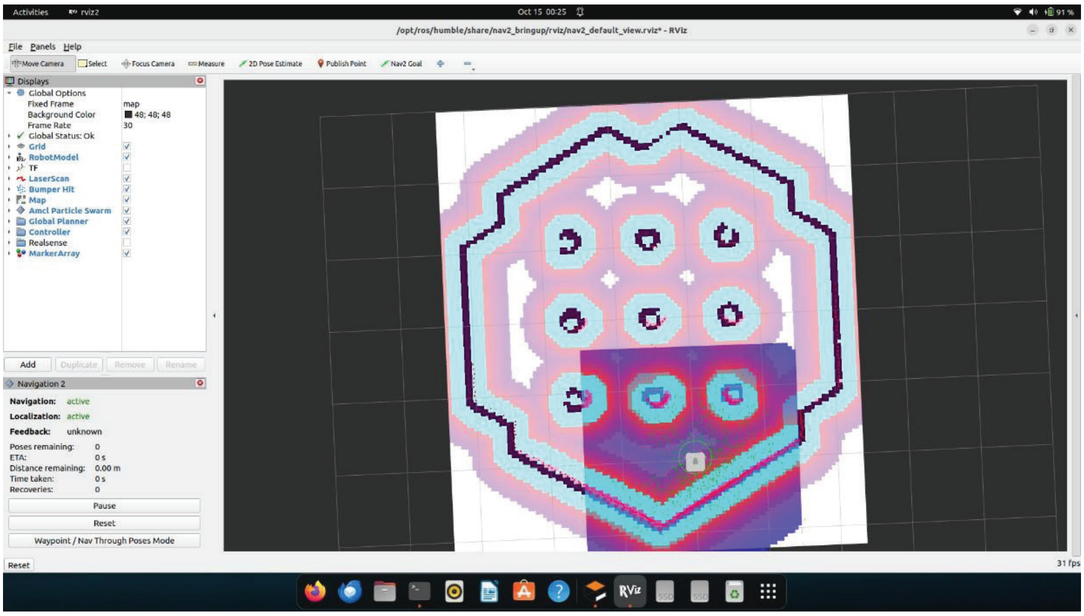


Figure 6. Rviz simulation.

refining designs with the aid of CAD software. Iterative design lowers errors, cuts down on development time, and makes it easier to create creative solutions that are specifically suited to the demands of restaurant situations [17]. Additionally, CAD models provide a common forum for interdisciplinary communication, facilitating the cooperative efforts of mechanical, electrical, and software experts in the effective deployment of robotic systems in restaurant management as shown in figure 7.

System Architecture

In the context of a SLAM-based restaurant service robot [18], ROS topics facilitate communication between various components, allowing for exchange of sensor data, commands, and status updates, enabling seamless coordination and control of the robot's functionalities as shown in figure 8. YAML parameters files store configuration settings and parameters for the robot's behaviour and environment, such as sensor calibration data or navigation parameters, providing a flexible and easily editable way to adjust robot behaviour without modifying code [2]. URDF (Unified Robot Description Format) files describe the robot's physical structure, including links, joints, sensors, and visual properties, enabling accurate visualization, simulation, and kinematic modelling of the robot within the ROS ecosystem [19].

In the context of a SLAM-based restaurant service robot [18], ROS topics facilitate communication between various components, allowing for exchange of sensor data, commands, and status updates, enabling seamless coordination and control of the robot's functionalities as shown in figure 8. YAML parameters files store configuration settings and parameters for the robot's behaviour and environment, such as sensor calibration data or navigation parameters, providing a flexible and easily editable way to adjust robot behaviour without modifying code [2]. URDF (Unified Robot Description Format) files describe the robot's physical structure, including links, joints, sensors, and visual properties, enabling accurate visualization, simulation, and kinematic modelling of the robot within the ROS ecosystem [19].

XACRO (XML Macros) files were used to modularize and simplify URDF files by defining reusable macro elements, reducing redundancy and improving maintainability, making it easier to manage complex robot descriptions as shown in figure 9. The Robot State Publisher node in ROS publishes the robot's kinematic state (e.g., joint angles and transformations) to the TF (Transform) tree, allowing other nodes to access and utilize this information for navigation, manipulation, and visualization tasks [13]. The robot description refers to a comprehensive representation of the robot's physical properties, including its geometry, kinematics, dynamics, and sensor configurations, typically stored in URDF or XACRO files for use within ROS applications as shown in figure 10.

Mechanism for controlling the motion of robots with two wheels, enabling forward, backward, and rotational movements by independently adjusting wheel velocities, commonly used in mobile robots like the restaurant service robot as shown in figure 10. ROS message specifying desired linear and angular velocities for controlling the robot's motion, typically utilized in conjunction with differential drive controllers to generate wheel commands [20]. ROS components are responsible for managing the lifecycle and configuration of robot controllers, facilitating seamless integration and coordination of various control algorithms for different robot

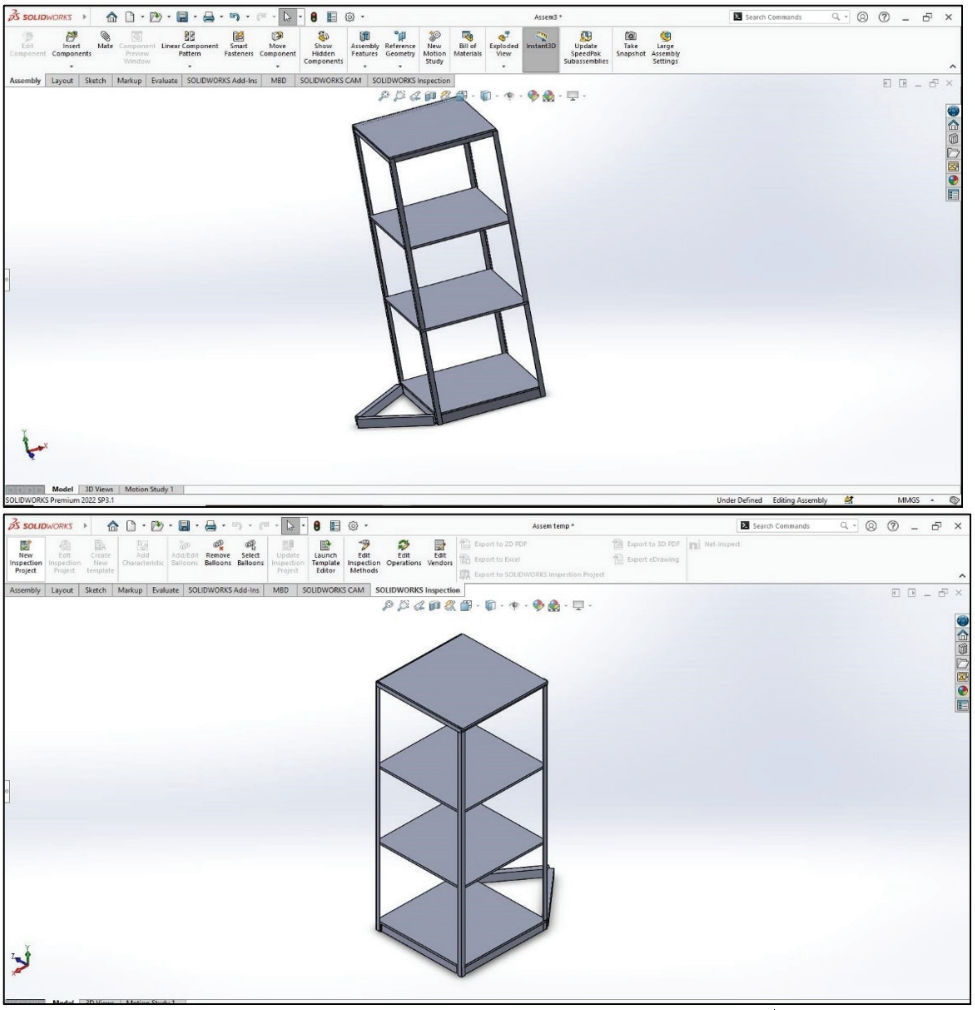


Figure 7. CAD model of restaurant service robot

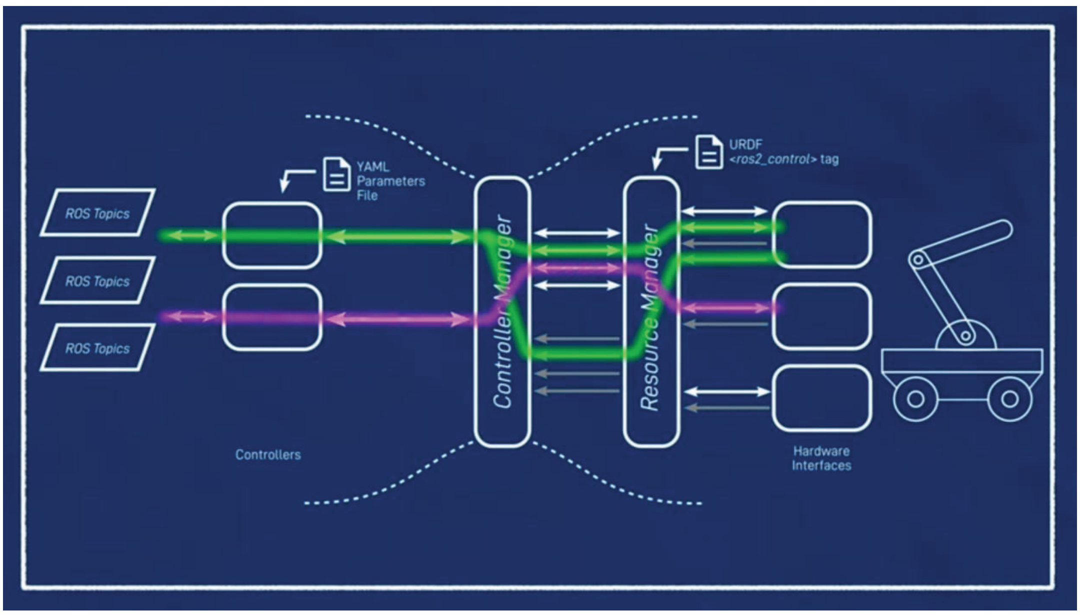


Figure 8. System architecture.

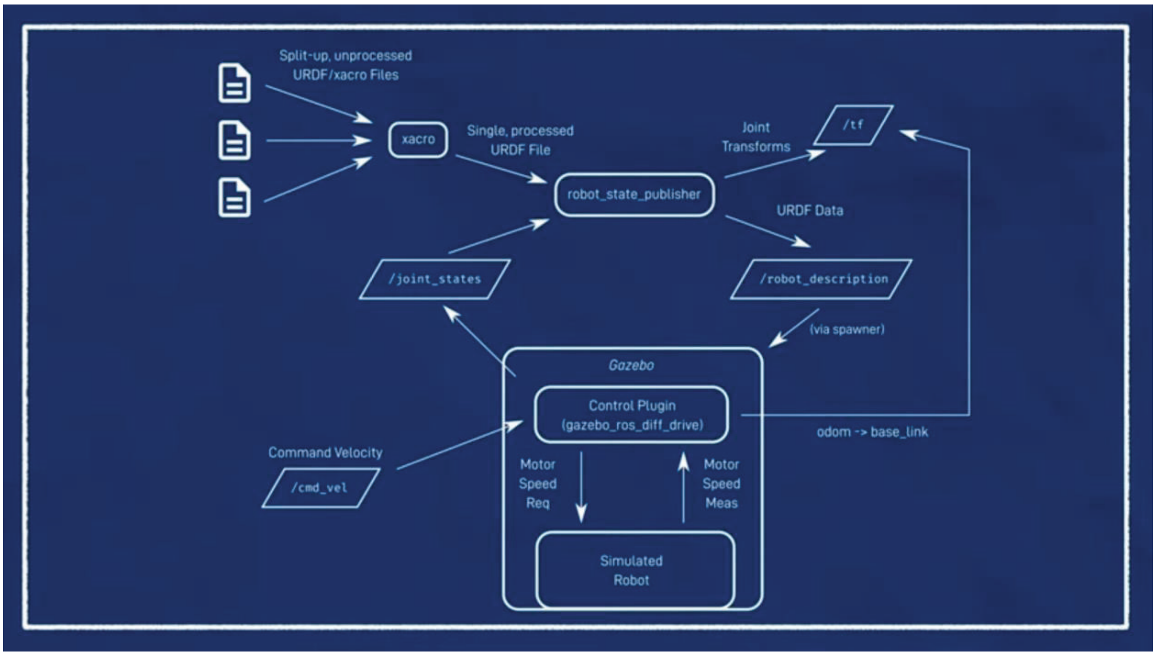


Figure 9. Xacro format.

subsystems. ROS node responsible for publishing the current state of robot joints, such as positions, velocities, and efforts, allowing visualization and monitoring of joint movements and ensuring synchronization with simulation or control algorithms. A control method allowing users to drive the robot using joystick input, enabling intuitive navigation in real-time within the restaurant environment. A ROS topic used to send velocity commands to the robot's motors, facilitating movement control and navigation directives based on SLAM mapping. ROS bridge Server acts as a communication interface between ROS and external systems, enabling bidirectional data exchange, which can be utilized for remote monitoring or control of the restaurant service robot. Software component responsible for interfacing with the Lidar sensor, collecting precise environmental data used for mapping and localization, crucial for the robot's autonomous navigation and obstacle avoidance in the restaurant setting as mentioned in Figure 11.

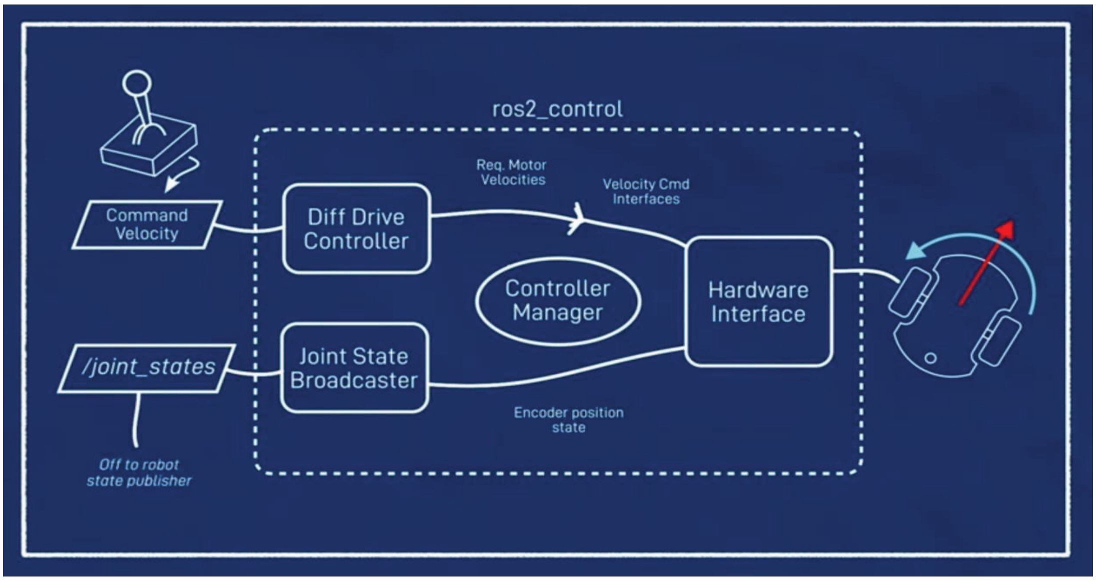


Figure 10. Controller manager.

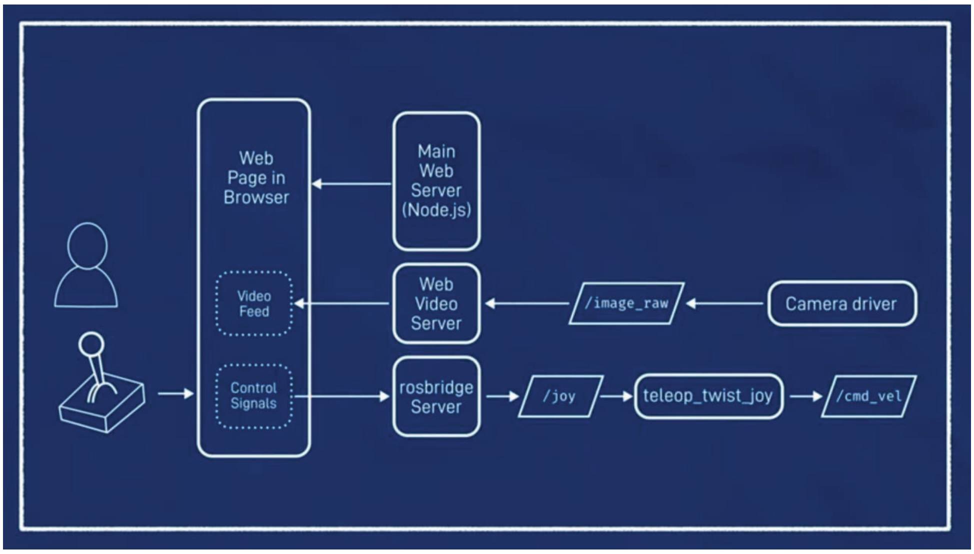


Figure 11. Plug-in controls.

Simultaneous Localization and Mapping

Hardware and software integration is the next step in the development process for a restaurant service robot that uses SLAM (Simultaneous Localization and Mapping), after the CAD design stage. This include building the robot, adding sensors such as cameras and LiDAR, and developing algorithms for object identification, SLAM, navigation, and human-robot communication as shown in Figure 12. Testing guarantees precise mapping, obstacle avoidance, localization, and effective meal delivery. Working together to get input from restaurant employees improves usability. To create a dependable restaurant service robot with SLAM capabilities, mechanical, electrical, and software experts need to work closely together.

Figure 12. Robot service model.

RESULTS AND DISCUSSION

The state-of-the-art Slam-based restaurant service robot is made to navigate restaurant settings on its own and serve patrons effectively [21]. This robot can correctly orient itself and map its surroundings in real-time by utilizing Simultaneous Localization and Mapping (SLAM) technology [22]. This allows it to navigate busy and dynamic restaurant areas with ease. With its sophisticated sensors and clever algorithms, the robot can quickly and safely serve food and beverages to clients' tables by recognizing obstructions, averting collisions, and calculating the best paths. The robot may converse with patrons and employees of the restaurant, adding to the whole dining experience. But like any complicated system, it could occasionally experience problems or malfunctions that require for troubleshooting, which usually entails figuring out where the mistakes are in the sensors, fixing inconsistent navigation, or adjusting the localization algorithms to make sure smooth operation and uninterrupted service delivery as shown in figure 13. Troubleshooting for Localization is shown in figure 14.

SLAM algorithms utilise information from various sensors [23]. For robots to perform activities safely and independently, they need to be able to comprehend where they are in relation to the surroundings. This kind of issue is characterised by the need that the robot be aware of its location within the surroundings. Any robotic application where the robot has to navigate a novel environment and create a map must be based on the vSLAM idea. In contrast to the sensors used in classical SLAM, sensors like LIDAR or GPS (Global Positioning Systems) are less expensive and have a greater capacity to collect environmental data including colour, texture, and appearance. Furthermore, contemporary cameras are inexpensive, small, and low power consumption.

Figure 13. Model Test Track for Localization.

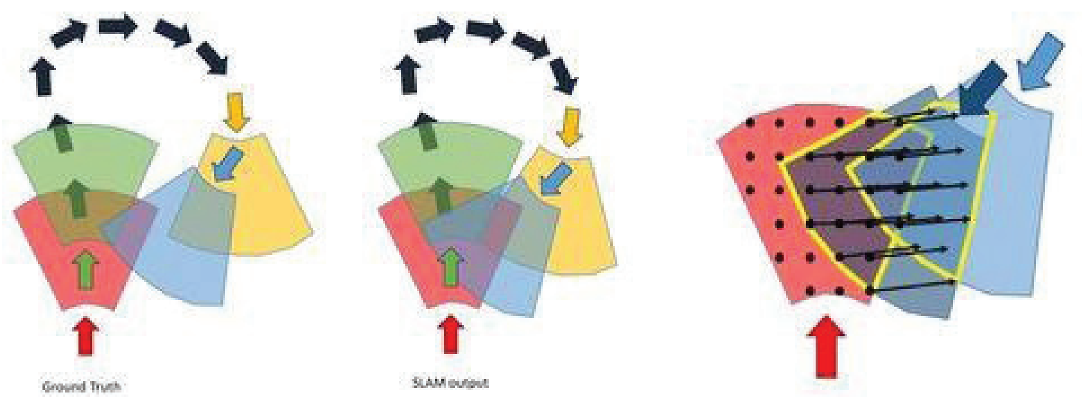


Figure 14. Troubleshooting for Localization.

SLAM

The technique by which a robot or other device may simultaneously map its surroundings and determine its own position within them is known as simultaneous localization and mapping (SLAM). For autonomous robots and vehicles to navigate and function well in uncharted or changing surroundings, this capacity is essential. It explores the basic ideas of statistics and probability theory and explains why they are important for modelling uncertainty in robotic systems. Determining the robot's location and orientation in relation to its surroundings is one of the major robotics issues. It discusses probabilistic methods for localising robots, such as particle filters and Markov localization. This covers methods such as feature-based mapping and grid-based mapping. It introduces the ORB-SLAM system, a simultaneous localization and mapping (SLAM) system. A basic robotics task is known as simultaneous localization and mapping (SLAM), in which a robot or other device maps an uncharted area while also tracking its position on the map. Because the ORB-SLAM system is made to use monocular cameras for SLAM, it can be used in a wide range of robotic and computer vision applications.

From early approaches that relied on geometric and probabilistic methods to more recent developments that integrate sensors like cameras and Lidars for robust perception, the authors explore the development of SLAM techniques. The research highlights the difficulties of SLAM, including how to handle computing limitations, sensor noise, and the requirement for real-time operation in a variety of applications.

Mapping

SLAM methods, which entail mapping the surroundings and concurrently locating the robot on that map. The significance of persistent navigation—which enables the robot to keep functioning and updating its map even in the face of abrupt changes in the surroundings—is emphasised by the authors.

G-Mapping is a probabilistic framework that builds precise and comprehensive maps of the surroundings by fusing odometers and laser range data [24]. One of G-Mapping's primary contributions is its capacity to manage noisy and unclear sensor input, which allows the robot to map partially viewable landscapes. There were issues with accuracy, computational economy, and real-time performance with traditional mapping algorithms. In this work, we assessed the usefulness of SLAM (Simultaneous Localization and Mapping) algorithms in mapping dynamic and crowded surroundings, with a special focus on a restaurant setting. It was intended to measures mapping accuracy, real-time performance, and adaptation to environmental changes. In a simulated restaurant scenario, we used a mobile robot outfitted with a LIDAR sensor and an RGB-D camera. The robot was designed to traverse and map the room automatically, meeting common restaurant components including tables, chairs, and moving customers. Initially, in a controlled setup with minimal movement, the SLAM algorithm (e.g., Hector SLAM) successfully generated detailed and accurate maps. Quantitative analysis showed an average mapping error of less than 5% in terms of object placement and room layout.

Navigation

The importance of coverage path planning in robotics and highlights its uses in agriculture, exploration, surveillance, and environmental monitoring. The authors emphasise how important autonomous robots are for carrying out duties that require covering a certain region completely or partially. They classify the methods according to many parameters, including sensor modalities, robot capabilities, and environment representation. The taxonomy offers an organised approach to comprehending the many methods used in the discipline. Algorithms such as Wave front Expansion and Dynamic Window Approach, which are grid-based, aid in the robot's effective navigation. The main findings include a discussion of the difficulties and potential options for coverage path planning. It highlights the necessity of combining several methodologies and sensor modalities in order to improve robotic systems' performance in coverage jobs.

Learning a policy or a mapping from states to actions that maximises the total reward an agent receives in an environment is the first step in policy search methods. These techniques are especially helpful in scenarios when the system dynamics are intricate and challenging to simulate. The techniques covered in the research most likely rely on algorithms that use trial and error to iteratively enhance the policy, changing course in response to input from the environment. A mobile robot equipped with LIDAR, RGB-D camera, and a robust path planning algorithm was deployed in a simulated restaurant environment. The robot's navigation capabilities were tested under various conditions, including static and dynamic obstacles such as tables, chairs, and simulated human movements. During navigation, the robot displayed real-time response to dynamic changes in its surroundings. It successfully redirected around shifting impediments and changed its course in response to immediate sensor data, allowing for uninterrupted, efficient navigation.

Robot Operating System

ROS can be used by researchers and developers to create and evaluate a wide range of robotic applications, from basic robot prototypes to sophisticated autonomous systems [25]. The value of open-source software in the robotics community, which enables cooperation, information sharing, and a faster pace of robotic technology development. Since its debut, ROS has been widely adopted and has made a substantial contribution to the development of robotics research and applications.

While it gives a broad summary of the study, the technical material contains details on the techniques and algorithms created to deal with the navigational difficulties. It is impossible to give a more thorough outline of the precise methods and findings reported in the study without having access to the complete language [10].

It presents the idea of topics, which are named buses that nodes use to communicate with one another, and nodes, which are programmes that carry out calculations. Device drivers, package management, communication infrastructure, hardware abstraction, and visualisation tools are examples of ROS components. ROS accommodate a large variety of robotic systems and applications because to these features.

The ADAS implementation's technical features are available on ROS. It goes into detail into the functions and interactions of the system's individual modules and parts. An open-source simulator for many robots is called Gazebo. They highlighted about how crucial simulation environments are to the study and advancement of robotics. The ideas used Gazebo's design. It probably covers things like physics modelling, sensor simulation, graphical rendering, and the simulation engine. For robotics experiments, these design principles are essential to building a trustworthy and realistic simulation environment [13].

Robotics and computer vision researchers are paying close attention to the integration of LiDAR data for object detection. Scholars have investigated diverse techniques and algorithms for merging data obtained from these sensors in order to enhance the precision and resilience of object detection [26]. The development of sensor fusion methods, which integrate RGB-D and LiDAR data to produce a more complete picture of the environment, has been the subject of numerous studies. These methods seek to overcome the shortcomings of each sensor by utilising its unique strengths. In sensor fusion, object detection usually entails locating and identifying items in the surrounding environment [27]. To do this, a range of algorithms, including deep learning and conventional computer vision techniques, have been proposed by researchers[28].

Localisation

To provide a thorough comparison, the authors may have assessed various algorithms using simulations and actual experiments. To evaluate their performance, they probably looked at things like the quantity of particles, the effect of sensor noise, and the computing difficulty of the methods.

Graph Simultaneous Localization and Mapping is referred to as GraphSLAM. This method is algorithmic and builds precise maps of an environment while estimating the robot's attitude inside it. It does this by combining motion data and sensor measurements. In order to solve this issue, GraphSLAM models were used to the surrounding world as a graph, with edges standing in for sensor readings and mobility restrictions and nodes representing robot positions and landmarks. It builds a graph with edges denoting the constraints between the robot's poses and the observed landmarks and nodes representing the robot's poses and landmarks. The method determines the most likely combination of poses and landmarks that account for the sensor measurements and motion limitations by optimising this graph. A probabilistic graphical model that shows how variables in a complicated system are related to one another. Factor graphs make it easy to solve the SLAM problem by effectively integrating motion constraints and sensor measurements into the optimisation procedure.

SLAM algorithms make use of data from several sensors [29]. In order for robots to operate in a safe and autonomous manner, they must be able to understand their location in respect to their environment [30]. The requirement that the robot be cognizant of its position within the environment is what distinguishes this type of problem. It examines the fundamental concepts of probability theory and statistics and illustrates their significance for modelling uncertainty in robotic systems. One of the main problems in robotics is figuring out where the robot is and how it is oriented with respect to its environment. The hybrid control system that the authors suggest combines intentional and reactive control strategies. Reactive control enables the robot to respond quickly to immediate sensory inputs and avoid obstacles in its path, whereas deliberative control enables the robot to plan its actions based on an environment map [31,32]. SLAM techniques, which involve mapping the environment and simultaneously locating the robot on it. Uncertain robot motion, noise from sensors, and environmental changes are all challenges in mobile robot localization. As reference points for localization, geometric beacons were used. Geometric beacons are stationary items with recognisable geometric characteristics, such as edges, corners, or certain forms. ROS provides a flexible and distributed platform for developing robot software. Among other things, it offers services for inter process communication, hardware abstraction, package management, and device drivers. Within the robotics community, ROS has gained popularity due to its modular and extensible nature. With edges representing sensor readings and mobility constraints and nodes indicating robot positions and landmarks, GraphSLAM describes the environment as a graph. It constructs a graph where nodes represent the robot's postures and landmarks and edges indicate the limitations between the robot's poses and the observed landmarks. By optimising this graph, the approach finds the most likely set of poses and landmarks that account for the motion limits and sensor measurements. Test Results for SLAM is shown in figure 15 in regards to the numerous trials conducted in workspace. Figure 16 highlights the test results of path traced by the models. Furthermore the movement of models with respect to designed track is shown in figure 17 and finally navigation results are shown in figure 18.

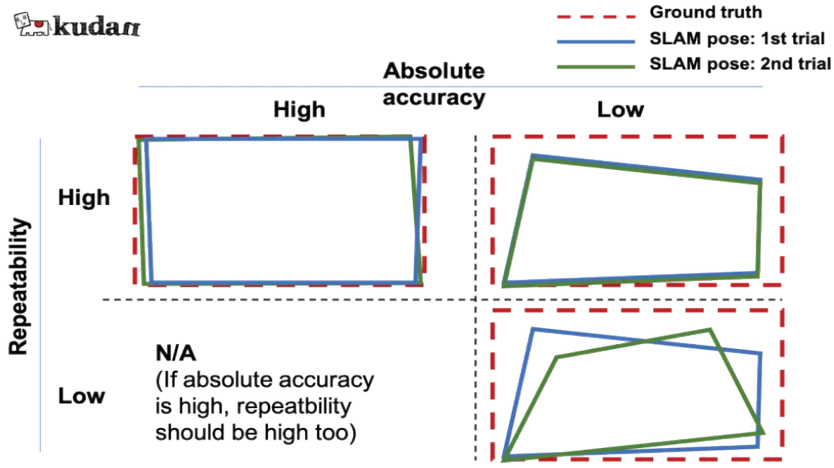


Figure 15. Test Results for SLAM with trials.

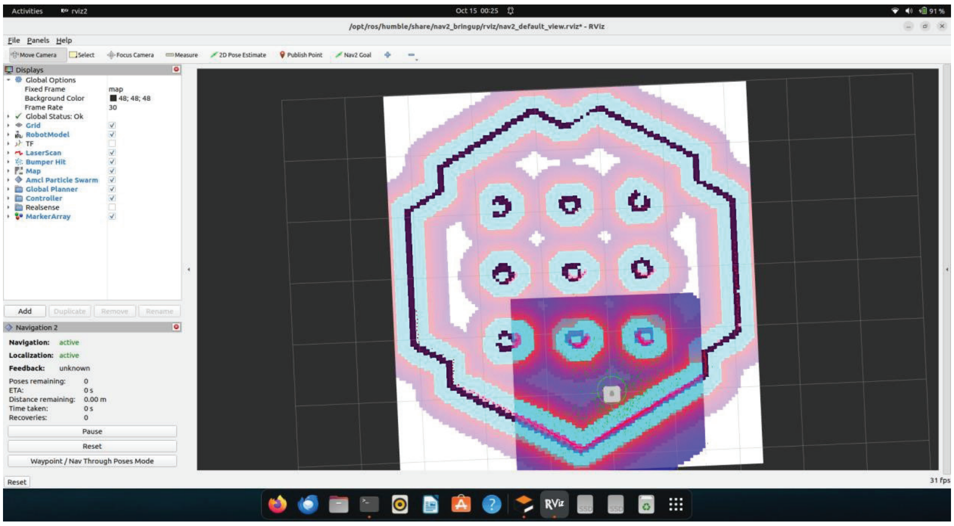


Figure 16. Test results for SLAM - path traced by the models.

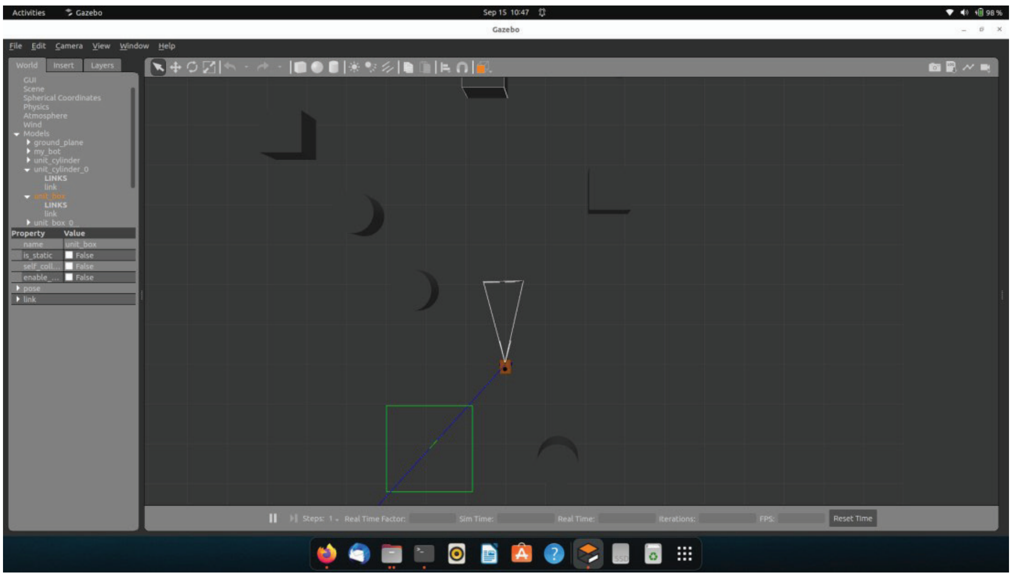


Figure 17. Test results for SLAM.

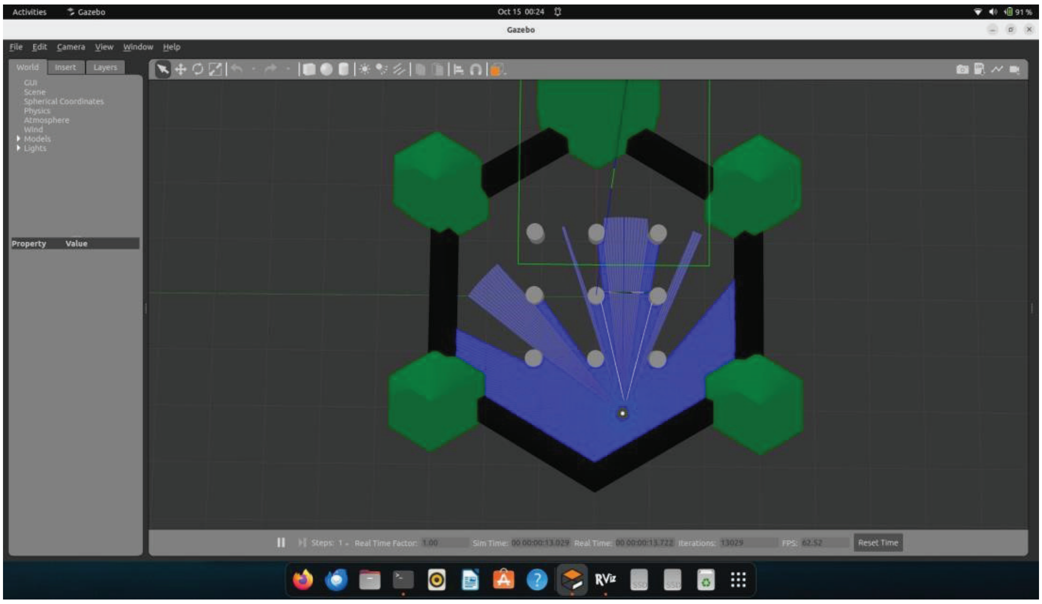


Figure 18. Test results for SLAM - navigation results.

CONCLUSION

The integration of SLAM (Simultaneous Localization and Mapping) with ROS (Robot Operating System) in the context of a restaurant management system has demonstrated a significant improvement in efficiency and automation. The robotic system can navigate complex environments, such as restaurant floors, with high precision, reducing the need for human intervention in tasks like food delivery and table cleaning.

The use of robotics in restaurant management can lead to an enhanced customer experience. Robots can deliver food orders quickly and accurately, reducing wait times and minimizing errors in order delivery. This can result in higher customer satisfaction and loyalty. Despite the initial investment required for implementing a robotic restaurant management system, the long-term cost-effectiveness of such a system is notable. Robots can work continuously without breaks, reducing labour costs over time. Additionally, the system can optimize resource utilization, such as energy and space, leading to overall cost savings.

While the use of SLAM with ROS in robotic restaurant management is promising, there are still challenges to overcome. These include the need for robust algorithms for real-time navigation in dynamic environments and ensuring the safety of human-robot interactions. Future research can focus on addressing these challenges to further enhance the capabilities of robotic restaurant management systems. The successful implementation of a robotic restaurant management system using SLAM with ROS can have a significant impact on the restaurant industry. It can lead to increased adoption of robotics in various restaurant operations, ultimately transforming the way restaurants are managed and operated.

ACKNOWLEDGEMENTS

This work was supported by the Research Fund of the Bursa Uludag University, Project Number OUAP(F)-2019/9. The authors would like to thank for this support.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE

Artificial intelligence was not used in the preparation of the article.

REFERENCES

[1] Chen CS, Lin CJ, Lai CC. Non-contact service robot development in fast-food restaurants. IEEE Access 2022;10:31466–31479. [CrossRef]

[2] Liu M, Chen M, Wu Z, Zhong B, Deng W. Implementation of intelligent indoor service robot based on ROS and deep learning. Machines 2024;12:256. [CrossRef]

[3] Pradhananga P, ElZomor M, Santi Kasabdji G. Identifying the challenges to adopting robotics in the US construction industry. J Constr Eng Manage 2021;147:05021003. [CrossRef]

[4] Delgado JMD, Oyedele L, Ajayi A, Akanbi L, Akinade O, Bilal M, Owolabi H. Robotics and automated systems in construction: Understanding industry-specific challenges for adoption. J Build Eng 2019;26:100868. [CrossRef]

[5] Wang T, Zheng P, Li S, Wang L. Multimodal human–robot interaction for human-centric smart manufacturing: A survey. Adv Intell Syst 2024;6:2300359. [CrossRef]

[6] Montaño-Serrano VM, Jacinto-Villegas JM, Vilchis-González AH, Portillo-Rodríguez O. Artificial vision algorithms for socially assistive robot applications: A review of the literature. Sensors 2021;21:5728. [CrossRef]

[7] Kerr J, Nickels K. Robot operating systems: Bridging the gap between human and robot. In: Proceedings of the 2012 44th Southeastern Symposium on System Theory (SSST); 2012. p. 99–104. [CrossRef]

[8] De D, Mukherjee A, Ray A, Roy DG, Mukherjee S. Architecture of green sensor mobile cloud computing. IET Wirel Sens Syst 2016;6:109–120. [CrossRef]

[9] Krishnamurthi R, Kumar A, Gopinathan D, Nayyar A, Qureshi B. An overview of IoT sensor data processing, fusion, and analysis techniques. Sensors 2020;20:6076. [CrossRef]

[10] Pimentel FDAM, Aquino-Jr PT. Evaluation of ROS navigation stack for social navigation in simulated environments. J Intell Robot Syst 2021;102:87. [CrossRef]

[11] Rozo L, Kupcsik AG, Schillinger P, Guo M, Krug R, van Duijkeren N, Spies M, Kesper P, Hoppe S, Ziesche H. The e-Bike motor assembly: Towards advanced robotic manipulation for flexible manufacturing. Robotics and Comput Integr Manuf 2024;85:102637. [CrossRef]

[12] Tola D, Corke P. Understanding URDF: A dataset and analysis. IEEE Robot Automat Lett 2024;9:4479–4486. [CrossRef]

[13] Furrer F, Burri M, Achtelik M, Siegwart R. RotorS—A modular Gazebo MAV simulator framework. In: Koubaa A, (editor). Robot Operating System (ROS). Springer International Publishing; 2016. p. 595–625. [CrossRef]

[14] Mingo Hoffman E, Traversaro S, Rocchi A, Ferrati M, Settimi A, Romano F, Natale L, Bicchi A, Nori F, Tsagarakis NG. Yarp based plugins for Gazebo simulator. In: Hodicky J, editor. Modelling and Simulation for Autonomous Systems. Springer International Publishing; 2014. p. 333–346. [CrossRef]

[15] Berezina K, Ciftci O, Cobanoglu C. Robots, artificial intelligence, and service automation in restaurants. In: Robots, Artificial Intelligence, and Service Automation in Travel, Tourism and Hospitality. Emerald Publishing Limited; 2019. p. 185–219. [CrossRef]

[16] Ibragimov IZ, Afanasyev IM. Comparison of ROS-based visual SLAM methods in homogeneous indoor environment. In: 2017 14th Workshop on Positioning, Navigation and Communications (WPNC); 2017. p. 1–6. [CrossRef]

[17] Zhao J, Liu S, Li J. Research and implementation of autonomous navigation for mobile robots based on SLAM algorithm under ROS. Sensors 2022;22:4172. [CrossRef]

[18] Aslan MF, Durdu A, Yusefi A, Sabanci K, Sungur C. A tutorial: Mobile robotics, SLAM, bayesian filter, keyframe bundle adjustment and ROS applications. Robot Oper Syst 2021;6:227–269. [CrossRef]

[19] Storiale F, Ferrentino E, Chiacchio P. Robot-agnostic interaction controllers based on ROS. Appl Sci 2022;12:3949. [CrossRef]

[20] Zheng K. ROS navigation tuning guide. In: Koubaa A, (editor). Robot Operating System (ROS). New York: Springer International Publishing; 2021. p. 197–226. [CrossRef]

[21] Leung R, Loo PT. Co-creating interactive dining experiences via interconnected and interoperable smart technology. Asian J Technol Innov 2022;30:45–67. [CrossRef]

[22] Qing-xiao Y, Can Y, Zhuang F, Yan-zheng Z. Research of the localization of restaurant service robot. Int J Adv Robotic Syst 2010;7:18. [CrossRef]

[23] Taheri H, Xia ZC. SLAM; definition and evolution. Eng Appl Artif Intell 2021;97:104032. [CrossRef]

[24] Hu S, Zou L, Yu JX, Wang H, Zhao D. Answering natural language questions by subgraph matching over knowledge graphs. IEEE Trans Knowl Data Eng 2017;30:824–837. [CrossRef]

[25] Abeykoon I, Feng X. A forensic investigation of the robot operating system. In: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData); 2017. p. 851–857. [CrossRef]

[26] Patil LN, Khairnar HP, Hole JA, Mate DM, Dube AV, Panchal RN, Hiwase VB. An experimental investigation of wear particles emission and noise level from smart braking system. Evergreen 2022;9:711–720. [CrossRef]

[27] Patil L, Khairnar H. Python inspired smart braking system to improve active safety for electric vehicles. Int J Automot Mech Eng 2022;19:9447–9459. [CrossRef]

[28] Patil LN, Khairnar HP, Bhirud SG. An intelligent contactless brake blending system with advanced driver assistance technique for electric vehicles. World J Eng 2022;21:44–52. [CrossRef]

[29] Kümmerle R, Steder B, Dornhege C, Ruhnke M, Grisetti G, Stachniss C, Kleiner A. On measuring the accuracy of SLAM algorithms. Auton Robot 2009;27:387–407. [CrossRef]

[30] Patil LN, Khairnar HP. Assessing impact of smart brake blending to improve active safety control by using Simulink. Appl Eng Lett 2021;6:29–38. [CrossRef]

[31] Macario Barros A, Michel M, Moline Y, Corre G, Carrel F. A comprehensive survey of visual SLAM algorithms. Robotics 2022;11:24. [CrossRef]

[32] Taketomi T, Uchiyama H, Ikeda S. Visual SLAM algorithms: A survey from 2010 to 2016. IPSJ Trans Comput Vision Appl 2017;9:16. [CrossRef]