Simulation of autonomous navigation of turtlebot robot system based on robot operating system

Mohammad Talal Ghazal[1], Murtadha Al-Ghadhanfari[2], Najwan Zuhair Waisi[1]
[1]Department of Computer Techniques Engineering, Northern Technical University, Mosul, Iraq
[2]Department of Computer Center, Mosul University, Mosul, Iraq

Article Info

Article history:

Received Apr 7, 2023
Revised May 25, 2023
Accepted Jun 5, 2023

Keywords:

Gazebo
Robot
Robot operating system
Rviz
Simultaneous localization and mapping

ABSTRACT

Complex system science has recently shifted its focus to include modeling, simulation, and behavior control. An effective simulation software built on robot operating system (ROS) is used in robotics development to facilitate the smooth transition between the simulation environment and the hardware testing of control behavior. In this paper, we demonstrate how the simultaneous localization and mapping (SLAM) algorithm can be used to allow a robot to navigate autonomously. The Gazebo is used to simulate the robot, and Rviz is used to visualize the simulated data. The G-mapping package is used to create maps using collected data from a variety of sensors, including laser and odometry. To test and implement autonomous navigation, a Turtlebot was used in a Gazebo-generated simulated environment. In our opinion, additional study on ROS using these important tools might lead to a greater adoption of robotics tests performed, further evaluation automation, and efficient robotic systems.

Corresponding Author:

Mohammed Talal Ghazal
Department of Computer Techniques Engineering, Northern Technical University
Mosul, Iraq
Email: mohammed.ghazal@ntu.edu.iq

1. INTRODUCTION

The field of robotics is an area that has seen a significant increase in the number of research carried out in the last few decades [1], [2]. Numerous software components are shared by most robots, making collaboration crucial. Reusing robot control software developed globally speeds up development [3], [4]. Robot operating system (ROS) has this feature, so it is widely used in research and even in industries [5], [6]. ROS's growing popularity in the robotics scientific community can be attributed to the addition of simulation support with Gazebo and visualization tools such as Rviz and Rqt graph [7]. To understand how the different features of ROS can be used in a basic remote-control application, a holistic approach is needed [8]. Our primary research focus is on the autonomous navigation domain in a closed indoor environment. The majority of the costs related to manual material handling are due to labor costs; however, the use of autonomous robots is becoming increasingly common in industrial applications. By utilizing these autonomous robot vehicles, the plant can save up to 30% on material handling costs [9], [10]. With minimal assistance, these self-driving robots possess the ability to traverse their surroundings autonomously in hazardous areas for workers. Different methods can be used to resolve navigation problems such as mapping, localization, and path planning [11]. The objective of the proposed work focus on describes and show the notion of constructing and designing a mobile robot with the ability to visually identify and evade stationary objects using simultaneous localization and mapping (SLAM) and independently navigating to the intended destination.

Since ROS is free and open-source software, it is widely used in robotics teaching and research [12]. C and Python, two programming languages that are quite popular nowadays, are employed for the development of this software. ROS robot software comprises of several nodes working together and communicating so other nodes may get data. Each node may use multiple languages. The ROS SLAM approach has two distinct phases [13], the first step is to conduct the survey and draw out the map, which is then posted to the Map_Server as a topic. To address the second half of the problem, identifying the survey robot’s position on the map, a method called adaptive monte carlo localization (AMCL) is available for ROS. Different SLAM libraries are available in ROS, including G-mapping, Hector_SLAM, and Karto_SLAM [14]. Many studies [15]–[17] used UGV robots equipped with a Hector SLAM and a LiDAR sensor, both of which utilize laser scans to generate 2D maps and pinpoint locations. Some studies used the SLAM G-mapping method, which can recognize the position and construct a map in a large room using the ROS system, to develop the automated movement robot [18]. Because it creates a two- or three-dimensional model of the robotic work area from the information on the exact geometry of the environment gathered by different sensors, the mapping approach is regarded as an essential duty in mobile robots. When it comes to mobile robotics, this mapping method is required for accurate localization, precise placement relative to both the robot and the target, and collision-free movement [19]. Using both RGB-D and wheel encoder data, researchers in [20] propose an RGB-D SLAM system for interior mapping and navigation. In this paper, we’ll look at how the SLAM technique in ROS and data sent from TurtleBot3’s LiDAR sensor can be used to identify obstacles and build a map that can be saved in the Map_Server. Regarding the ability of ROS to recognize locations, the AMCL algorithm is used by the system in order to locate the position of the robot on the ground. Adjusting the relevant parameters for the robot upon entry into a narrow and enclosed region, such as a pipe, and allowing the robot to travel autonomously according to the map, and determining the proper path when meeting obstacles test the SLAM approach in ROS.

2. SIMULATION ENVIRONMENT

The robot operating system is a Berkeley software distribution (BSD) licensed open-source meta operating system. Using the built-in tools and libraries that are offered by ROS, users are able to write code, construct software, and test it on many computers. ROS makes available a wide variety of services, including the control of low-level devices, the implementation of commonly used functionality, the exchange of messages between multiple processes, and the administration of packages [21]. Because ROS allows us to utilize previously established packages like G-mapping and teleop_key, it shortens the amount of time needed for development [22].

Gazebo serves as a three-dimensional dynamic simulator that can simulate a variety of different models of robots operating in both indoor and outdoor conditions [23]. Gazebo allows for testing robotic models in virtual environments and incorporating sensor data [24]. It enables testing the functionality of a robotic model under particular circumstances without causing any damage to the hardware [25]. It is powered by a physical mechanism that generates light, inertia, and gravity. For the purpose of describing the various components of the robot, an XML file known as the universal robotic description format (URDF) has been utilized. Figure 1 depicts a 3D model of a Turtlebot that has been replicated inside of a Gazebo. The setting shown in Figure 2 is used for the purpose of testing the robotic model.

Figure 1. Turtlebot model in Gazebo

Figure 2. Simulated environment in Gazebo

ROS visualization, often known as Rviz, is an efficient ROS-compatible 3D visualization tool. The name "Rviz" stands for ROS visualization. It gives the user the ability to record Data obtained from the sensors on the robot, see the simulated robot model, and playback the recorded sensor data. By providing a visual representation of the robot's perception, cognition, and actions, the user can effectively debug a robot application, tracing the sensor inputs to the intended or unintended behaviors. Rviz displays 3D sensor data in the form of point clouds. This data may come from a variety of sources, including stereo cameras, lasers, and Kinect. Image data may be seen in Rviz which was captured by 2D sensors such as webcams and RGB cameras.

3. ROBOT IMPLEMENTATION WITHIN ROS

The TurtleBot3 is a mobile robot that may be used in a variety of settings, including teaching, research, hobby, and product development. It is also tiny, inexpensive, and programmable. The primary objectives of the TurtleBot3 project are to significantly lessen the footprint of the platform, bring down its cost, and do all of this without compromising the level of functionality or quality it has, while also making it expandable. Burgers and waffles are the two primary varieties of model constructions that may be created using TurtleBot3; Figure 3 depicts a TurtleBot3 burger. The TurtleBot3 may be modified in a variety of ways, depending on how the mechanical pieces are reassembled and how optional components, such as the computer and the sensor, are used. In addition, the TurtleBot3 has been improved with a single board computer (SBC) that is appropriate for strong embedded systems, 360-degree distance sensors, and 3D printing technologies. This SBC is both cost-effective and tiny in size.

Figure 3. TurtleBot3 burger

The TurtleBot3 is well-suited for domestic service robots due to its key technologies of SLAM, navigation, and manipulation. The TurtleBot can use SLAM to map the environment and drive about. It could be remotely controlled from a laptop, joypad, or Android phone. The TurtleBot can also track a person's legs. The last decade has seen the development of four different versions of the TurtleBot series. TurtleBot3 was built to fill its predecessors' missing functionalities and user needs. Gazebo's simulation platform was used to test two robotics scenarios.

3.1. Scenario 1

A simple teleoperation node is performed on the TurtleBot3 in the Gazebo simulator to verify the essential features of ROS and visualization in Gazebo. With the use of the Roslaunch command, a TurtleBot3-based Gazebo node is launched. Depending on the instruction, the Turtlebot is shown in its simulated world. Then, a teleoperation node is started up and begins sending a stream of Twist messages on the cmd_vel topic, which sends velocity values in response to key presses. Using Twist messages sent on the cmd_vel topic, the location and velocity of the robot's wheels can be adjusted by the node named TurtleBot3-diff-drive. Figure 4 demonstrates teleoperation simulation and Figure 5 is for Rqt graph. The graph is a GUI plug-in for ROS's Rqt tool set. The Rqt graph visualizes nodes, processes, and their communication. It gives a system overview and debugs the system.

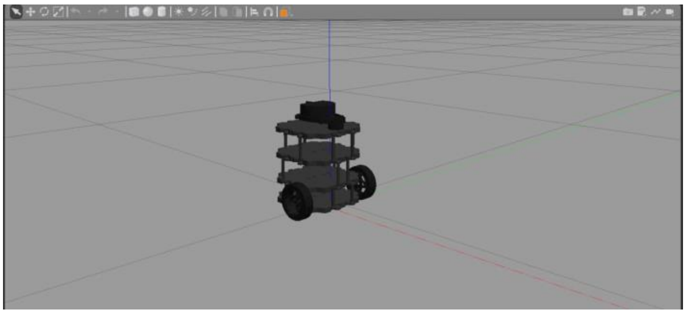

Figure 4. Teleoperation simulation on Gazebo of scenario 1

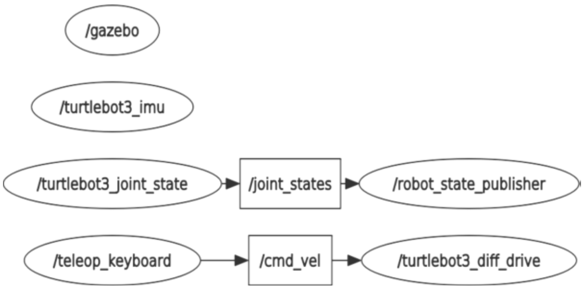

Figure 5. Rqt graph of teleoperation simulation of scenario 1

3.2. Scenario 2

The SLAM algorithm is employed to create and continually update a map for mapping purposes, and estimating the robot location. First, generate the environment, then import Turtlebot in Gazebo. The robotic model consists of a pair of wheels and a Kinect sensor. G-mapping is used to map. To employ G-mapping, the robot model needs to have odometry data and a stationary, horizontal laser range finder. Using infrared sensors, the Kinect records the depth picture of the surroundings. It uses 640×480 color infrared data with 16 bits per pixel. ROS's depth image proc converts this picture into a 3D point cloud. Point cloud to laser scan converts 3D point cloud data into 2D laser scan. By giving G-mapping's needed arguments, Rviz creates a map. Figure 6 illustrates the map. Initially, the robot is driven using ROS's teleop_key package by providing keyboard velocity instructions. Figure 7 shows G-mapping's rqt graph. The map_server package may store the created map using G-mapping's /map topic. Autonomous navigation may now use the map.

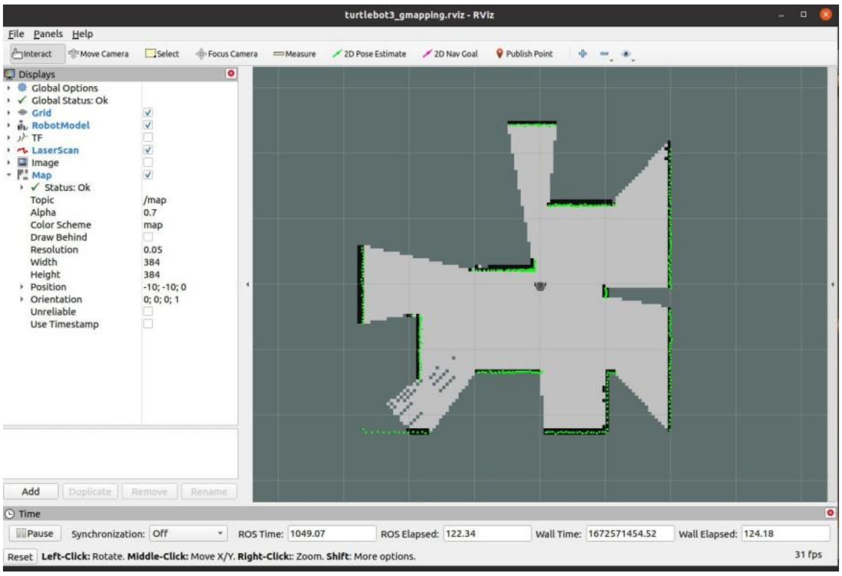

Figure 6. Visualizing map in Rviz of scenario 2

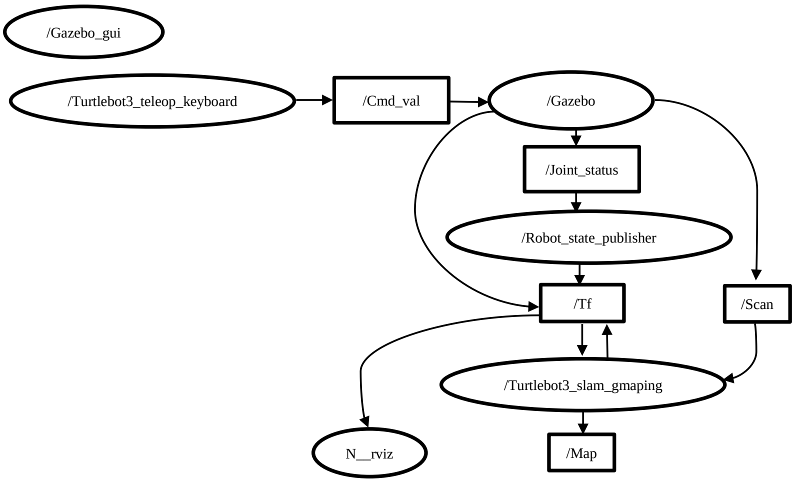

Figure 7. Rqt graph G-mapping of scenario 2

Figure 8 illustrates Rviz's self-localization, through localization, the robot's location and posture are estimated in respect to its surroundings also, it is necessary to have odometry data, laser data, and a map of the surroundings in order to locate the robot. Turtlebot utilizes AMCL to localize, which is use a 2D probabilistic robot localization. In AMCL, the robot's posture is detected using a particle filter.

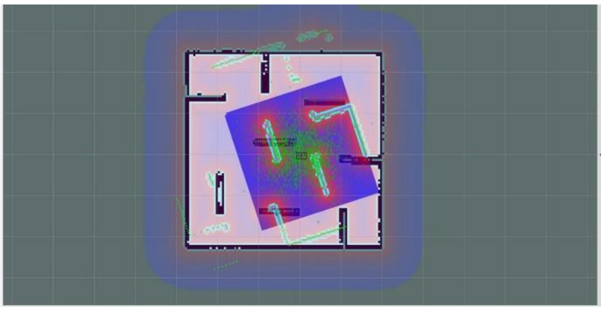

Figure 8. Localized robot in Rviz

After mapping and localization, autonomous navigation will start. ROS navigation stack packages implement AMCL, which includes a node that localizes a static map. This node receives sensors, and static map data, then the AMCL node publishes the robot posture and map position. Two cost-maps store the simulation's obstacles information, the first is utilized for global path planning, and the second for obstacle avoidance and local path planning. Both of them are visualized by Rviz, and finally, the robot can move automatically. The 2D nav objective in Rviz assigns a robot's map destination and direction. The robot plans its path and the controller provides velocity instructions. Figure 9 illustrate the robot's path from its beginning point to the specified target.

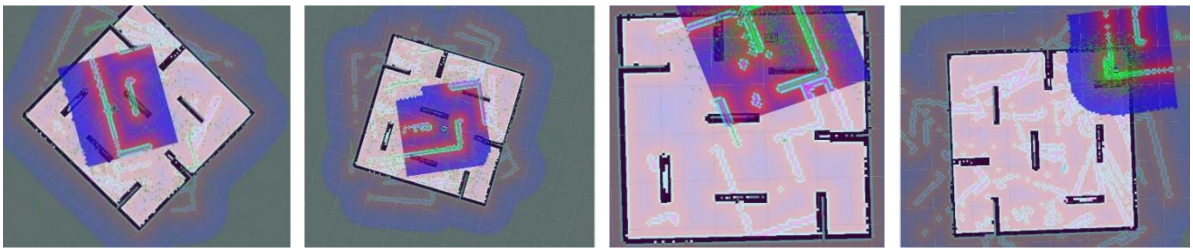

Figure 9. Autonomous navigation

4. CONCLUSION

The capabilities of ROS and Gazebo to be applied to a robot design have been evaluated. As a result, the built-in messaging system of ROS, which has passed laboratory tests, helps save time by handling the complexities of communication between nodes via the anonymous publish and subscribe method. Message passing improves encapsulation and code reuse by forcing the accurate implementation of interfaces across system nodes. For the robot to be understood by the rest of the ROS system, ROS provides a set of features for modeling and describing it. Visualizing the information in one application seems good and helps us to rapidly view what our robot observes and identify sensor difficulties or robot model faults.

ACKNOWLEDGEMENTS

The current research is supported by a collaboration between Northern Technical University NTU and Mosul University in Iraq. The website of NTU is https://www.ntu.edu.iq and Mosul University is https://www.uomosul.edu.iq.

REFERENCES

[1] T. A. Salih, M. T. Ghazal, and Z. G. Mohammed, "Development of a dynamic intelligent recognition system for a real-time tracking robot," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 10, no. 3, p. 161, Sep. 2021, doi: 10.11591/ijra.v10i3.pp161-169.

[2] H. M. Qassim and W. Z. W. Hasan, "A review on upper limb rehabilitation robots," *Applied Sciences (Switzerland)*, vol. 10, no. 19, pp. 1–18, Oct. 2020, doi: 10.3390/app10196976.

[3] N. WehChien, S. Gobee, and V. Durairajah, "Automated Guided Vehicle based with Robot Operating System for Mapping and Navigation Task," in *International Conference on Edge Computing and Applications, ICECAA 2022 - Proceedings*, IEEE, Oct. 2022, pp. 868–872, doi: 10.1109/ICECAA55415.2022.9936362.

[4] V. V. Patel, M. V. Liarokapis, and A. M. Dollar, "Open Robot Hardware: Progress, Benefits, Challenges, and Best Practices," *IEEE Robotics and Automation Magazine*, vol. 30, no. 3, pp. 123–148, Sep. 2023, doi: 10.1109/MRA.2022.3225725.

[5] A. Afzal, C. Le Goues, M. Hilton, and C. S. Timperley, "A Study on Challenges of Testing Robotic Systems," in *Proceedings - 2020 IEEE 13th International Conference on Software Testing, Verification and Validation, ICST 2020*, IEEE, Oct. 2020, pp. 96–107, doi: 10.1109/ICST46399.2020.00020.

[6] M. Quigley *et al.*, "ROS: an open-source Robot Operating System," *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009.

[7] S. Gobhinath, K. Anandapoorani, K. Anitha, D. D. Sri, and R. DivyaDharshini, "Simultaneous Localization and Mapping [SLAM] of Robotic Operating System for Mobile Robots," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, IEEE, Mar. 2021, pp. 577–580, doi: 10.1109/ICACCS51430.2021.9441758.

[8] K. Li and H. Tu, "Design and Implementation of Autonomous Mobility Algorithm for Home Service Robot Based on Turtlebot," in *2021 IEEE 5th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC)*, IEEE, Oct. 2021, pp. 1095–1099, doi: 10.1109/ITNEC52019.2021.9587138.

[9] M. Kulkarni, P. Junare, M. Deshmukh, and P. P. Rege, "Visual SLAM Combined with Object Detection for Autonomous Indoor Navigation Using Kinect V2 and ROS," in *2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA)*, IEEE, Dec. 2021, pp. 478–482, doi: 10.1109/ICCCA52192.2021.9666426.

[10] I. M. Mohammed, M. Z. N. Al-Dabagh, S. J. Rashid, and N. A. M. Isa, "Path discovering in maze area using mobile robot," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 20, no. 2, pp. 416–425, Apr. 2022, doi: 10.12928/TELKOMNIKA.v20i2.19408.

[11] C. Nandkumar, P. Shukla, and V. Varma, "Simulation of Indoor Localization and Navigation of Turtlebot 3 using Real Time Object Detection," in *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, IEEE, Nov. 2021, pp. 222–227, doi: 10.1109/CENTCON52345.2021.9687937.

[12] T. Duckett, S. Marsland, and J. Shapiro, "Learning globally consistent maps by relaxation," in *Proceedings-IEEE International Conference on Robotics and Automation*, IEEE, 2000, pp. 3841–3846, doi: 10.1109/ROBOT.2000.845330.

[13] B. L. E. A. Balasuriya *et al.*, "Outdoor robot navigation using Gmapping based SLAM algorithm," in *2nd International Moratuwa Engineering Research Conference, MERCon 2016*, IEEE, Apr. 2016, pp. 403–408, doi: 10.1109/MERCon.2016.7480175.

[14] Y. Pyo, H. Cho, R. Jung, and T. Lim, *ROS Robot Programming*. ROBOTIS Co., Ltd, 2017.

[15] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *9th IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2011*, IEEE, Nov. 2011, pp. 155–160, doi: 10.1109/SSRR.2011.6106777.

[16] K. O. Arras, S. Grzonka, M. Luber, and W. Burgard, "Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities," in *Proceedings - IEEE International Conference on Robotics and Automation*, IEEE, May 2008, pp. 1710–1715, doi: 10.1109/ROBOT.2008.4543447.

[17] I. Z. Ibragimov and I. M. Afanasyev, "Comparison of ROS-based visual SLAM methods in homogeneous indoor environment," in *2017 14th Workshop on Positioning, Navigation and Communications, WPNC 2017*, IEEE, Oct. 2018, pp. 1–6, doi: 10.1109/WPNC.2017.8250081.

[18] S. P. Thale, M. M. Prabhu, P. V. Thakur, and P. Kadam, "ROS based SLAM implementation for Autonomous navigation using Turtlebot," *ITM Web of Conferences*, vol. 32, Jul. 2020, doi: 10.1051/itmconf/20203201011.

[19] Y. Z. Mu, C. Y. Dong, Q. M. Chen, B. C. Li, and Z. Q. Fan, "Research on Navigation and Path Planning of Mobile Robot Based on Vision Sensor," in *ACM International Conference Proceeding Series*, New York, NY, USA: ACM, Apr. 2020, pp. 519–524, doi: 10.1145/3404555.3404589.

[20] B. Canovas, A. Nègre, and M. Rombaut, "Onboard dynamic RGB-D simultaneous localization and mapping for mobile robot navigation," *ETRI Journal*, vol. 43, no. 4, pp. 617–629, Aug. 2021, doi: 10.4218/etrij.2021-0061.

[21] Q. H. Nguyen, P. Johnson, and D. Latham, "Performance evaluation of ROS-based SLAM algorithms for handheld indoor mapping and tracking systems," *IEEE Sensors Journal*, vol. 23, no. 1, pp. 706–714, Jan. 2022, doi: 10.1109/JSEN.2022.3224224.

[22] A. S. Prakash and A. Mohan, "Design and Simulation of an Autonomous Indoor Robot for Elderly Assistance," in *MysuruCon 2022 - 2022 IEEE 2nd Mysore Sub Section International Conference*, IEEE, Oct. 2022, pp. 1–5, doi: 10.1109/MysuruCon55714.2022.9972508.

[23] R. Tornese *et al.*, "ROS-based simulation environment for obstacle avoidance in autonomous navigation," in *2022 IEEE Conference on Control Technology and Applications, CCTA 2022*, IEEE, Aug. 2022, pp. 17–22, doi: 10.1109/CCTA49430.2022.9965786.

[24] A. Dobrokvashina, S. Sulaiman, T. Gamberov, K. H. Hsia, and E. Magid, "New Features Implementation for Servosila Engineer Model in Gazebo Simulator for ROS Noetic," *Proceedings of International Conference on Artificial Life and Robotics*, vol. 28, pp. 154–157, Feb. 2023, doi: 10.5954/icarob.2023.os6-3.

[25] A. O. Prasad *et al.*, "Design and development of software stack of an autonomous vehicle using robot operating system," *Robotics and Autonomous Systems*, vol. 161, p. 104340, Mar. 2023, doi: 10.1016/j.robot.2022.104340.

BIOGRAPHIES OF AUTHORS

Mohammed Talal Ghazal obtained his M.Sc. degree from Computer Engineering Technology, Northern Technical University, Mosul, Iraq in 2016. His M.Sc. thesis entitled: "wheelchair robot control using EOG signals" and his current research focuses on the development of robot control based on ROS. He can be contacted at email: mohammed.ghazal@ntu.edu.iq.

Murtadha Al-Ghadhanfari obtained his Bachelor's degree from Computer Engineering Technology, Northern Technical University, Mosul, Iraq in 2005 and a master's degree in Computers and Communications from University Kebangsaan Malaysia (UKM), Bangui, Malaysia in 2022. M.Sc project titled: "O3B satellite communication performance evaluation". His current research focuses on ROS. He can be contacted at email: murtadha.gdf@uomosul.edu.iq.

Najwan Zuhair Waisi obtained her M.Sc. degree from Computer Science Department, Mosul University, Mosul, Iraq in 2014. Her M.Sc. thesis entitled: "design and implementation of client honeypot" and her current research interests on ROS. She can be contacted at email: najwan.tuhafi@ntu.edu.iq.