IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 18 September 2023, accepted 17 October 2023, date of publication 26 October 2023, date of current version 13 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327705

RESEARCH ARTICLE

A Novel GUI Design for Comparison of ROS-Based Mobile Robot Local Planners

BILAL GUREVIN [1], FURKAN GULTURK [2], MUHAMMED YILDIZ [2], IHSAN PEHLIVAN [3], TRUNG THANH NGUYEN [4], FATIH CALISKAN [5], BARIS BORU [6], AND MUSTAFA ZAHID YILDIZ [3]

[1]Mechatronics Engineering Department, Graduate Education Institute, Sakarya University of Applied Sciences, 54100 Sakarya, Turkey
[2]Electrical Electronics Engineering Department, Graduate Education Institute, Sakarya University of Applied Sciences, 54100 Sakarya, Turkey
[3]Electrical Electronics Engineering Department, Faculty of Technology, Sakarya University of Applied Sciences, 54100 Sakarya, Turkey
[4]Faculty of Engineering and Technology, Liverpool John Moores University, L3 3AF Liverpool, U.K.
[5]Metallurgy and Materials Department, Faculty of Technology, Sakarya University of Applied Sciences, 54100 Sakarya, Turkey
[6]Mechatronics Engineering Department, Faculty of Technology, Sakarya University of Applied Sciences, 54100 Sakarya, Turkey

Corresponding author: Trung Thanh Nguyen (t.t.nguyen@ljmu.ac.uk)

ABSTRACT The studies such as navigating the AMR between stations, docking to the station, and assigning autonomous tasks to other stations are costly in terms of time and energy consumption. This situation creates the need for an interface where the entire work area can be observed and AMRs can be controlled from a single center in the installation of the system in the field. In this study, an interface that can be used in AMR control and monitoring was designed. With this interface; It is thought to prevent costly situations such as determining the stations, calculating the time spent in the transportation of products between stations, determining the movement route. The interface developed in this context was used in an application where ROS-based path planning algorithms were compared. A total of six different stations was identified. With three different local planners: DWA, TEB and Trajectory planner, AMR was given the task of acting autonomously to each station. Thanks to the developed interface, the distance and time required to reach each station were calculated by performing autonomous movement to the desired points. In this way, a comparison of ROS-based path planning algorithms was made. It was calculated that the DWA was 10.55% more successful than the TEB and 2.33% more successful than the Trajectory in terms of distance covered. Additionally, when examined in terms of arrival time, it was calculated that the DWA was 24.64% more successful than the TEB and 2.39% more successful than the Trajectory.

INDEX TERMS Autonomous mobile robot, path planning, robot operating systems, GUI.

I. INTRODUCTION

Today, the use of autonomous mobile robot (AMR) has been increasing day by day. The usage area of AMRs has been expanding in many fields from civilian life to military life, from the health sector to transportation, and from the livestock sector to agriculture. The use of AMRs, which we frequently encounter outdoors, is also frequently preferred indoors. For example, AMRs are used in cases such as the transportation of products leaving the line in a production facility, and the transmission of products between processes. For the AMR to move between desired points, it must be able to define the environment and follow the route it will go. At this point, path-planning algorithms come into play [1]. Thanks to path-planning algorithms, AMR creates a route from its current location to its destination. And by using sensor equipment, it sends a signal to the motors to follow this route in the appropriate direction and speed [2]. In addition, considering the dynamic obstacles during the movement, the route is instantly renewed. In this way, AMR can reach the desired point safely without hitting obstacles. Criteria such as time and total distance traveled are important in terms of energy consumption and production activities for AMR to reach the desired point [3]. In this context, many studies have been carried out on path planning algorithms in the literature. Some of the studies carried out are mentioned below.

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Liu.

Zhao et al. [4] conducted a path-planning study for unmanned aerial vehicles (UAV) in 3D space. They performed path planning by avoiding dynamic obstacles in the unknown environment in the gazebo simulator on the ROS platform. They also successfully tested this study, in which the planning speed was high, in the physical environment. Liu et al. [5] proposed a new path-planning algorithm to improve the obstacle avoidance and maneuverability of an autonomous vehicle. They tested the proposed algorithm on the tractor vehicle on the robot operating system (ROS) platform. Studies have shown that the tractor works with the desired accuracy in avoiding obstacles.

Gong et al. [6] presented the Long Short-Term Memory (LSTM) algorithm in addition to the traditional Deep Deterministic Policy Gradient (DDPG) algorithm used in path-planning. They determined the robot motion using the current and next states of the mobile robot. Different normalization methods were used to normalize the distance and angle between the mobile robot and the target point. They carried out their experimental studies using the ROS environment and the gazebo platform. As a result of the study, they showed that the mobile robot increased the success of path-planning. Wang et al. [7] presented a new study to minimize collisions in path planning. They created an improvement scheme to find corner points that are important in planning and to remove unnecessary points. As a result of the experiments, they obtained successful results in low-dimensional and high-dimensional space.

Chen et al. [8] proposed an RRT-Connect mobile robot path-planning algorithm developed to solve the inefficient search problem of the Sampling-based Fast Expanding Random Tree path-planning algorithm. For real-world studies, they carried out their experiments on a ROS-based mobile robot. The accuracy of the proposed study was compared between the four algorithms by running them in six different environments. They showed that the developed algorithm works with better performance.

Metaheuristic-based studies have also started to be preferred due to the computational speed in path-planning algorithms [9]. Yang et al. [10] carried out path-planning studies on multiple mobile robots. They suggested the leader follower-ant colony optimization (LF-ACO) algorithm for multi-vehicle studies where it is important to act in harmony as well as to avoid obstacles. As a result of the simulation studies they carried out based on ROS and Matlab, they showed that the proposed method works successfully.

Banjanovic et al. [11] carried out path planning studies on disinfection tools developed due to the COVID-19 pandemic. They used Particle swarm optimization as a global planner in their ROS-based work. They used DWA to avoid dynamic obstacles in the local movement. In this way, they obtained a study that can perform disinfection in contaminated environments without human factors.

Sabiha et al. [12] proposed a teaching–learning based optimization (TLBO) based path-planning algorithm for an autonomous tracked vehicle. They performed the detection processes of the vehicle using IMU, Lidar and odometry data. Using extended kalman filter (EKF), they provided corrections in the localization estimation of the vehicle. They compared the algorithm they developed with the Genetic algorithm (GA), particle swarm optimization (PSO), and hybrid GA-PSO. They carried out their experimental studies in real time in the ROS environment. With the study, they succeeded in obtaining the closest path between the starting and target positions by minimizing the path length, maximizing the path smoothness and preventing possible collisions with nearby obstacles. Choi et al. [13] proposed a Deep reinforcement learning-based algorithm to avoid collisions in path-planning. With this method, they enabled the vehicle to learn to reach the target point in an environment with dynamic obstacles. They successfully carried out the study, in which the robot model with a differential drive system was used, based on ROS, both in virtual and physical environments.

Zhou et al. [14] studied SLAM-based autonomous path-planning in the ROS environment. They addressed the problems of low mapping accuracy, slow path-planning efficiency, and high radar frequency requirements in the mapping and navigating the process in vehicles with all-wheel drive system. They successfully carried out the study in which A* algorithms were used as the global planner and DWA algorithms were used as the local planners by providing ROS and STM32 communication in the real environment.

Li et al. [15] used the Radam algorithm to develop the DDPG algorithm. In their work in the ROS ecosystem and Gazebo simulation environment, they compared DDPG with the DDPG algorithm they developed. They showed that the convergence speed of the DDPG algorithm developed in the path-planning process increased by %21 compared to the original DDPG algorithm and the success rate increased to %90. In their article, Baek and Im [16] discussed the problems such as environment constraints, algorithm computational complexity, and environmental variables in agricultural autonomous vehicle studies. They carried out their work on deep learning and ROS-based autonomous vehicles in a greenhouse environment on a rail and wheeled autonomous vehicle.

Ren et al. [17] carried out an automated vehicle study to ensure personnel safety at substations. In this study, which focused on autonomous navigation, they used A* algorithms globally and DWA algorithms locally. EKF was used in the fusion process of encoder and IMU data. In their study, they developed a novel autonomous vehicle that successfully overcomes obstacles and reaches the desired point. Wang et al. [18] conducted a study on the development of the A* algorithm, which is one of the most frequently used algorithms in path-planning. The expansion distance, which means the distance to the obstacles, has increased the stability of the path by reducing the number of right-angle turns on the paths drawn to the target. Simulation studies were carried out in ROS. In their study, which they compared with the traditional A*, they achieved %278 success in path planning.

In their article study, Chen et al. [19] carried out a study on the avoidance of fast-moving dynamic obstacles in the navigation process of autonomous vehicles. In the study where the cost map and DWA planner were developed, they performed a safe navigation process based on ROS against high-speed obstacles. Wang et al. [20] made a genetic algorithm-based approach to the path-planning of mobile robots. The study, in which vehicle tracking was carried out with a camera placed on the ceiling, was tested both in a ROS-based simulation environment and in a physical environment. In the map environment consisting of cells, it was successfully ensured that the vehicle avoided obstacles and reached the target.

II. SIGNIFICANCE OF STUDY

There may be reasons for preference due to the different features of the path-planning algorithms presented in the literature in different areas. Time to reach the desired point, total distance traveled, hitting obstacles, etc. features such as these can be counted as some of the reasons for preference.

Studies in world-renowned academic platforms such as IEEE Xplore, ScienceDirect, Scopus, and Web of Science in the last 10 years have been taken into account. It is also based on the keywords "ROS, GUI, Path Planning". Generally, studies in the literature were carried out on Matlab. However, there has been no comprehensive study that includes the ROS ecosystem, which is accepted all over the world as one of the standard tools in the control and monitoring of AMR. Although there exist ROS-based interface studies, most of these GUIs perform only very basic functions such as charge indicator, manual control, and some of them just select points and assign tasks [21], [22], [23], [24], or visualization of humidity and temperature of the environment [25]. There is currently no comprehensive GUI in ROS that can allow full control of the robots as well as record and visualize the detailed movements of the robots and the stations they visit. Given that ROS is gradually expanding, and is important academically and industrially [26], [27], the lack of such a GUI on ROS is a clear research gap.

Determining the AMR stations in the field, that is, in the physical environment studies, and evaluating the movements of the AMR between these stations in terms of time and distance is a very costly and time-consuming process. The products coming out of the production lines need to be taken by AMR and transported to the designated points. In addition, the products coming out of these lines have a certain cycle time. It's like a product comes out every 10 minutes. In order to carry out these operations in the field, it is necessary to move the AMR between these points beforehand, and to measure how long it takes and how far it has traveled to the determined points. If AMR cannot reach the products leaving the line in the required time, sufficient number of AMRs must be provided. In this context, studies such as moving the AMR between the stations, docking it to the station, assigning autonomous tasks to other points are costly in terms of time and energy consumption. In addition, flexible working opportunities are not always available according to environmental conditions. Instead of expensive and risky field studies, designing the working environment virtually and visualizing the data in a comprehensive GUI, like what is provided in our proposed GUI, offers very flexible and cost-effective opportunities.

In this study, a study has been made on the performance comparison of the path-planning algorithms by comparing the total distance taken and the time spent during the arrival of the different path-planning algorithms to the desired point. The global planner was kept constant and the ROS compatible global planner algorithm was used. As a local planner, a comparison of ROS-based DWA, TEB, and Trajectory Local Planner algorithms has been made. An interface design has been carried out to ensure the ease of control of the AMR and its tracking during autonomous movement. QtDesigner program and Python software language have been used in the interface design. Operations such as manual control of AMR, determination of speeds, registration and deletion of stations, visualization of stations, autonomous movement to the desired point, display of used path-planning algorithms can be easily achieved thanks to the designed interface. In addition, the time spent and the path traveled by the AMR between two points during autonomous movement can also be followed through the interface. In this way, by using different path-planning algorithms, the comparison process can be easily provided in terms of the time spent and the distance traveled during the movement to the desired points.

In the rest of the paper;

- In Section III: The path-planning algorithms used in the study are mentioned.
- In Section IV: the AMR model and novel interface design are mentioned.
- In Section V: The Gazebo and Rviz environment where the studies were carried out are mentioned.
- In Section VI and Section VII: The experimental results and interpretations of the comparison studies carried out using the interface are given.

III. ALGORITHMS

In the study, three different local planners were used by keeping the global_planner [28] constant to be compatible with the ROS navigation stack. In the following sections, DWA [29], TEB [30] and Trajectory Local planners [31] are mentioned as local planners available in the ROS ecosystem.

A. DWA LOCAL PLANNER

DWA samples linear and angular velocities (v,ω) in velocity space at certain time intervals. It is a predictive local path-planning method that simulates the trajectory at these speeds, taking into account the kinematics of the robot. In trajectory formation, it takes into account the situations of avoiding obstacles, reaching the desired speed, and reaching the target point quickly [32].

B. TEB LOCAL PLANNER

TEB local scheduler, as a derivative of Eband local scheduler, is one of the most common algorithms used in the ROS ecosystem [33]. This method has similar features to the Eband local planner, but instead of contraction and thrust forces, it optimizes every moment of orbital deformation and makes path-planning by minimizing the target cost function. Unlike the Eband local planner, the TEB local planner requires knowledge of the kinematics, dynamics, geometric shape, acceleration and speed limits of the robot [34]. For this reason, CPU consumption is also higher [35].

C. TRAJECTORY LOCAL PLANNER

Dynamic data such as instantaneous position, orientation, velocity and acceleration are used in the dynamic window approach. Considering these data, a limited map area is created in which the tool called the window is located in the center. By creating trajectories in this area, collision is prevented and the closest trajectory to the target is calculated. With this method, a movement trajectory is created on the local map, called a window, by detecting dynamic obstacles that are not found in the static map and sticking to the global plan. These orbits are arc-shaped due to the holonomic structure of the vehicle. The algorithm calculates angular and linear velocity components to follow the calculated trajectory. Calculation costs vary according to the driving technique of the vehicle (diff-drive, steering drive, etc.) [36].

The common parameter values of DWA, TEB and Trajectory local planners used in the study are given in Table 1. The remaining parameters are used as default values provided by the ROS navigation stack.

TABLE 1. Common parameters for local planners.

Parameters	Value
Maximum forward velocity	0.55 m/s
Minimum forward velocity	0.0 m/s
Maximum rotational velocity	0.5 rad/s
Minimum rotational velocity	-0.5 rad/s
Local costmap size	6 x 6 m2
Inflation radius	0.5 m
Obstacle range	2.5 m
Raytrace range	3.0 m

IV. SYSTEM IMPLEMENTATION

In this section, information is given about the characteristics of the computer where the study is performed, the version of the simulation environment (ROS) where autonomous operation and comparisons are made, the autonomous vehicle (AMR) modeled in the simulation, and the interface (GUI) where all these processes are monitored and the system is controlled.

A. ROS AND PC HARDWARE

In Table 2, the computer features and ROS version information on which the studies were carried out are given. In the study, Rviz, Gazebo programs and Lidar sensor used in the ROS environment use the GPU. On the other hand, CPU is used in localization and path planning processes. In the studies carried out, no problems were encountered in the speed of the system. It is seen that the GPU usage is on average 16% and the CPU usage is 30% on average when the whole system is active. From here, it can be seen that the PC hardware is sufficient.

TABLE 2. PC and ROS features.

ROS	Noetic
RAM	32 GB
CPU	İntel Core İ7-9750H @ 12x2.60 GHz
GPU	GeForce RTX 2080 Mobile
OS	Ubuntu 20.04

Thanks to the ROS node graph, the node connections of AMR can be visualized. In this way, the interconnected structures in the ROS environment, which work on the principle of broadcasting and subscribing, can be seen. Figure 1 shows the ROS nodes in the system, and the messages they broadcast and subscribe to. With the ros node named `/gazebo`, the vehicle is run in the simulation environment and the messages `/joint_states`, `/scan`, `/odom`, `/tf` and `/imu` are broadcast and subscribed to the `/cmd_vel` message. `/joint_states` message represents the robot's joint information, `/scan` message represents lidar data, `/odom` message represents encoder based position data, `/imu` message represents 9 axis acceleration sensor data and `/tf` message represents coordinate data which is the result of kinematic calculations. The `/cmd_vel` data is used for speed control of the simulated vehicle in the gazebo environment. The node named `/map_server` publishes the message containing the map data named `/map`. With the `/amr_karakuri_sim_amcl` node, `/tf`, `/map`, `/initialpose` and `/scan` messages are subscribed to. With the Adaptive Monte Carlo method, estimation-based positioning is performed. The kinematic conversion between the "map" coordinate and "odom" is made and the `/tf` message is broadcast. The node `/move_base` subscribes to `/map`, `/scan`, `/odom` and other messages shown in the figure. With these data, the global route and the local route are created. The messages shown in the figure are broadcast using the route planning algorithm specified in the settings uploaded to the parameter server.

B. AMR DESIGN

Ready-made robot models called turtlebot can be used for autonomous work in the ROS ecosystem, as well as originally designed robot models can be integrated into the system. In this study, an originally designed AMR model created in SolidWorks solid model design program was preferred. The designed AMR was made available in the ROS ecosystem thanks to the unified robot description format (URDF) model extraction offered by SolidWorks. The AMR model, in which the differential driving system and the sensors are used, is as shown in Figure 2. AMR has a length of 35 cm, a width of 27 cm and a height of 27 cm. The lidar sensor was placed at the top so that it can see 360 degrees around. The IMU

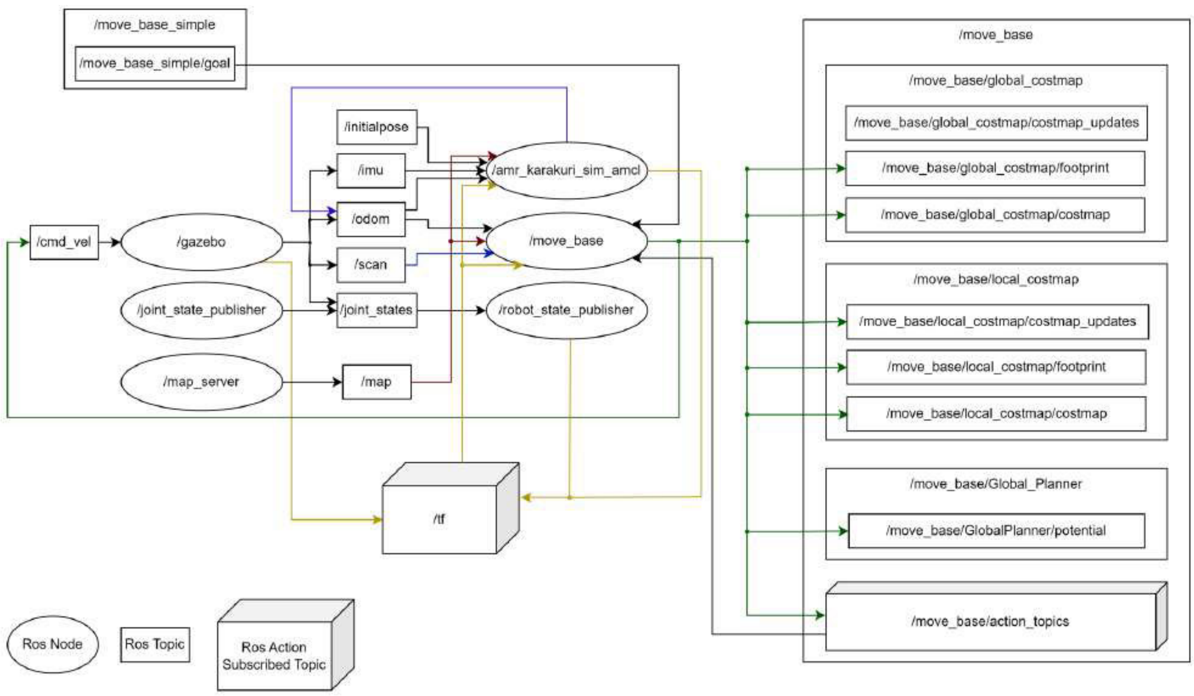

FIGURE 1. AMR ROS node tree.

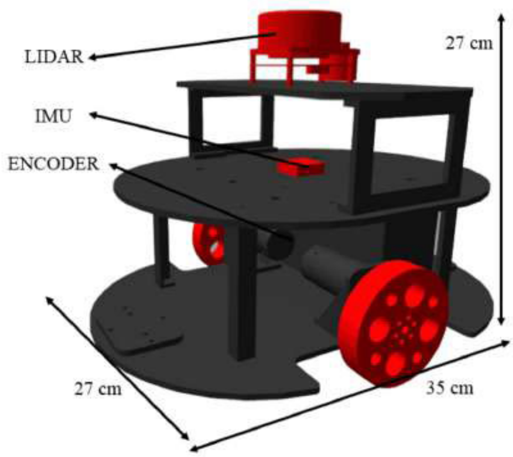

FIGURE 2. Novel AMR design.

sensor was placed in the middle floor. The wheels have been mounted in the middle of the vehicle to enable the AMR to rotate comfortably on its own axis.

Thanks to the transformation tree, the attachment points of the AMR limbs can be visualized. The transformation tree of the AMR designed in the study is given in Figure 3. When the figure is examined, it is seen that there is a main body named `chassis_link`. Thanks to the transformation tree, it can be seen that there are a total of 6 sub-links consisting of drive wheels and sensors under the main body. After the map information is taken as a global reference, AMR determines its position in the gazebo environment by means of odom data. The location and position of the AMR in the environment is created by transferring the position data received by the main body to the lower limbs.

C. PYQT BASED GUI DESIGN

An interface design has been carried out to facilitate autonomous control and analysis of ROS-based AMR. The Python-based interface design was designed using the QtDesigner program. The main features of the interface, which consists of 3 main tabs, are as follows.

In the AMR MANUEL CONTROL tab shown in Figure 4: Angular and linear velocity, orientation angle, x-y position can be monitored instantaneously. Station recording and deletion operations can be performed. AMR manual control can be provided at specified speeds in Figure 5. AMR can be followed instantly on the map. It can be seen that the AMR has the same location and orientation on the map in the Rviz environment in Figure 6 and the interface in Figure 7. Stations are listed with position and orientation information. Stations can be hidden/shown with icons on the map.

In the AMR AUTONOMOUS CONTROL tab shown in Figure 8: Angular and linear velocity, and orientation angle can be monitored instantaneously. Real time x-y-z positions and x-y-z-w orientations can be followed. AMR can be to move autonomously by selecting the desired station. During the autonomous movement between the station, the total elapsed time and the distance traveled can be followed. The planners local and globally used are shown. Real time movement of AMR can be followed on the map.

In the LOG PAGE tab shown in Figure 9: It can be checked in which function of the python codes created with the try-except structure there is an error.

IEEE Access

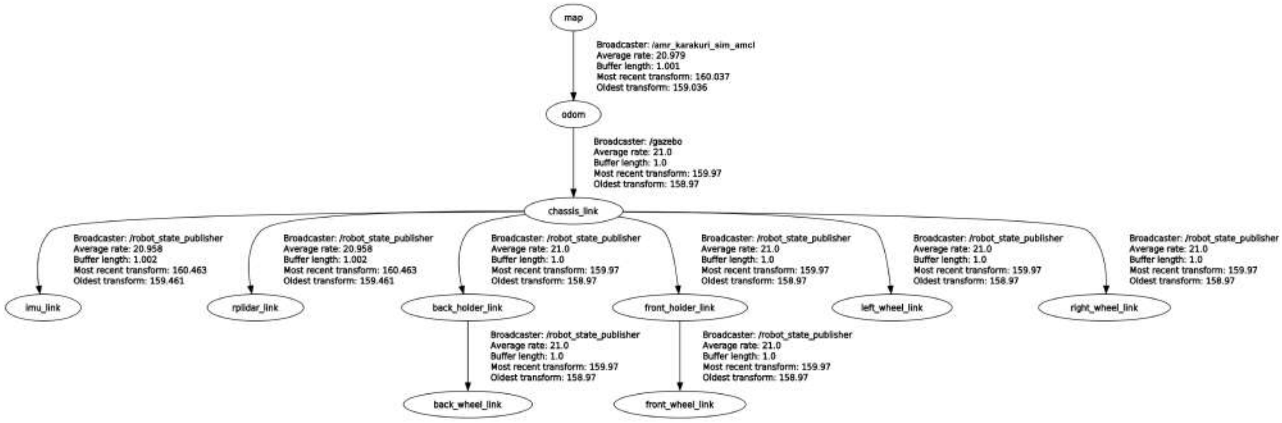

FIGURE 3. AMR URDF tree.

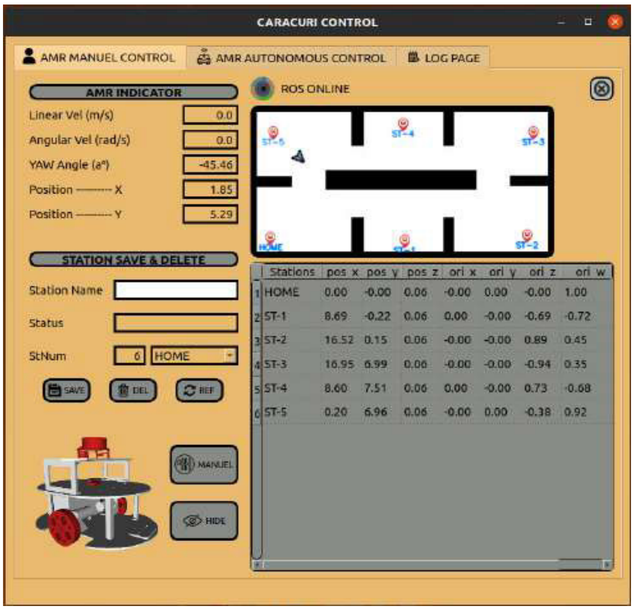

FIGURE 4. AMR MANUAL CONTROL section of the interface.

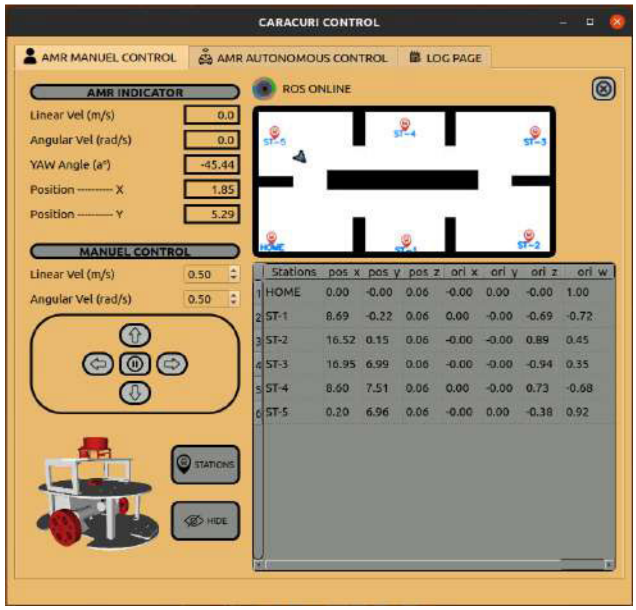

FIGURE 5. Manual control of AMR at the desired velocity.

V. EXPERIMENTAL DESIGN

In order to compare different local planners in ROS-based AMR control via interface, the presented study was carried out in a completely virtual environment. In the realization of the study, the steps in Figure 10 were followed. First, an novel AMR design was made in solidWorks program. Throughout the study, the novel AMR model was used instead of the turtlebot robots offered by ROS by default. In the virtual world of 20 × 10 m created in the gazebo, the AMR was manually navigated and the environment was mapped. The mentioned environment is shown in Figure 11. The novel GUI were used to navigate the AMR. In addition, Rviz program and Gmapping algorithm were used for mapping. The created map was integrated into the GUI, and real time location tracking of AMR in the virtual world was performed. Thanks to the GUI, AMR was brought to the desired point and these points were recorded as stations. After the mapping and determination of stations was completed, the autonomous task was assigned to AMR. AMR was asked to go to these points by selecting the station name on the GUI. A total of 6 stations have been determined, including the home point. By using different local path planning algorithms, autonomous movement of AMR to all stations in the virtual world was provided. Thanks to the GUI, the distance traveled and the time spent between each stop were calculated.

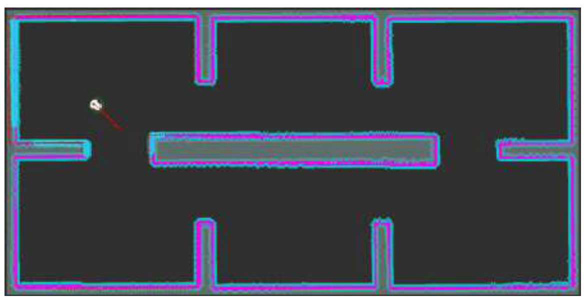

FIGURE 6. Real-time AMR monitoring on Rviz.

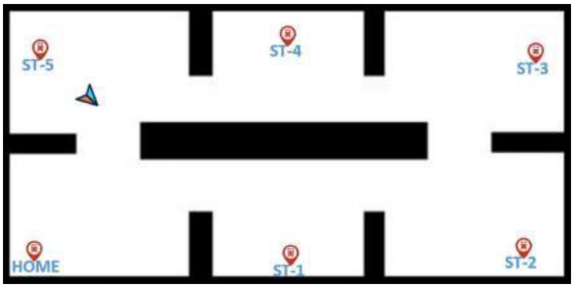

FIGURE 7. Real-time AMR monitoring on the map of the interface.

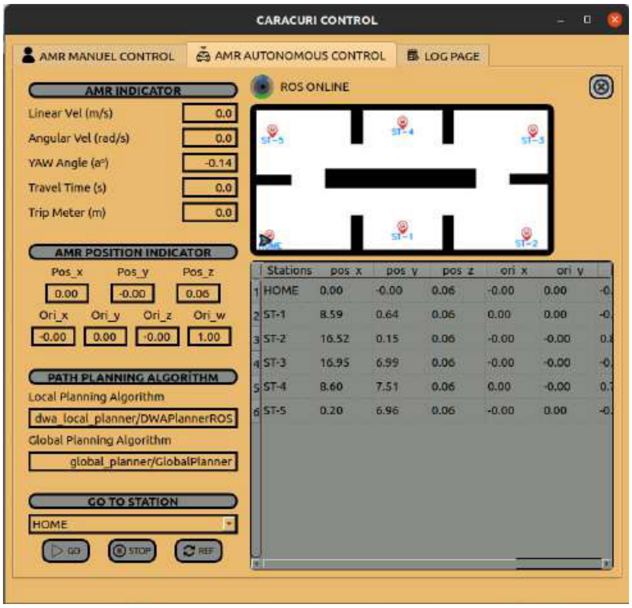

FIGURE 8. AMR AUTONOMOUS CONTROL section of the interface.

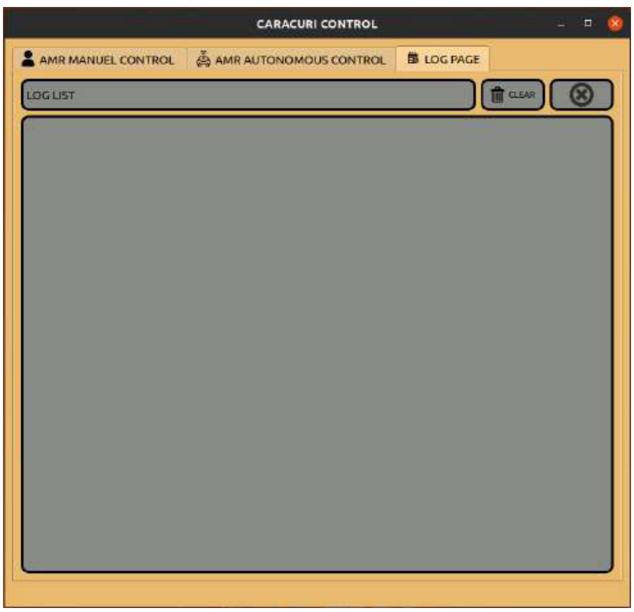

FIGURE 9. LOG PAGE section of the interface.

In addition, the currently used local and global planners were shown.

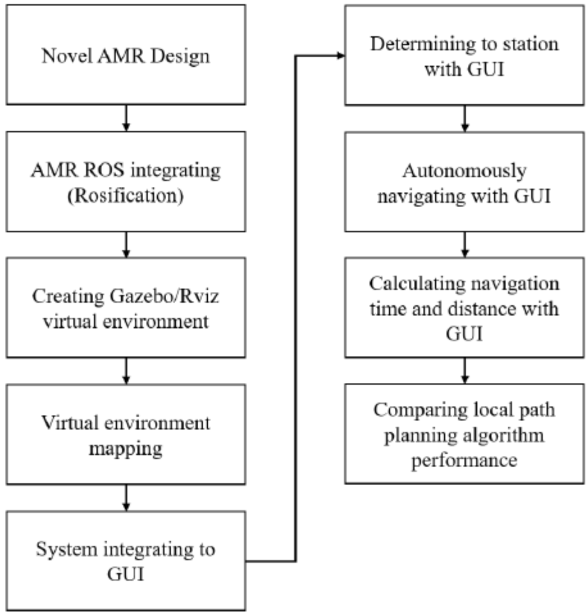

FIGURE 10. Setup steps.

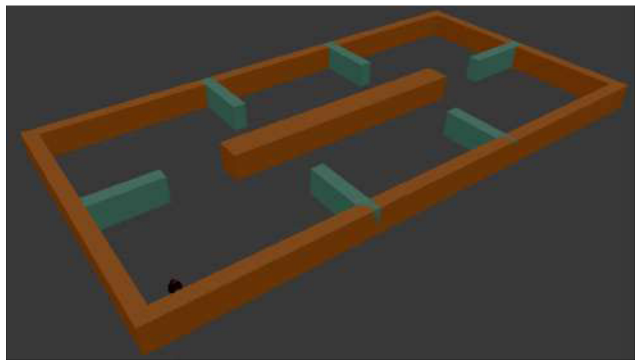

FIGURE 11. Gazebo environment used in the study.

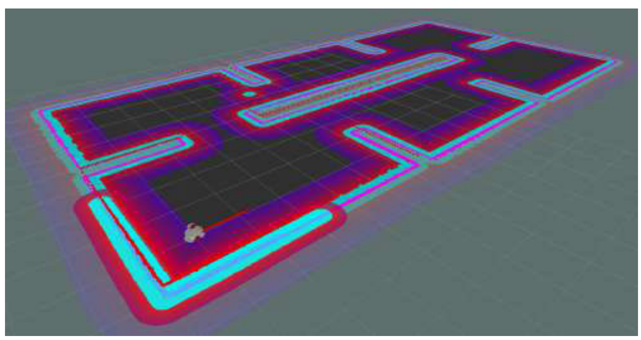

FIGURE 12. Visualization of Gazebo environment in Rviz.

Sensor, location and map information obtained from the AMR in the Gazebo simulator were visualized using the Rviz tool. In many situations such as location and position tracking of AMR, lidar and IMU sensor data, the map information can be observed thanks to the Rviz tool [37]. Figure 12 shows the image of the AMR in the Rviz environment.

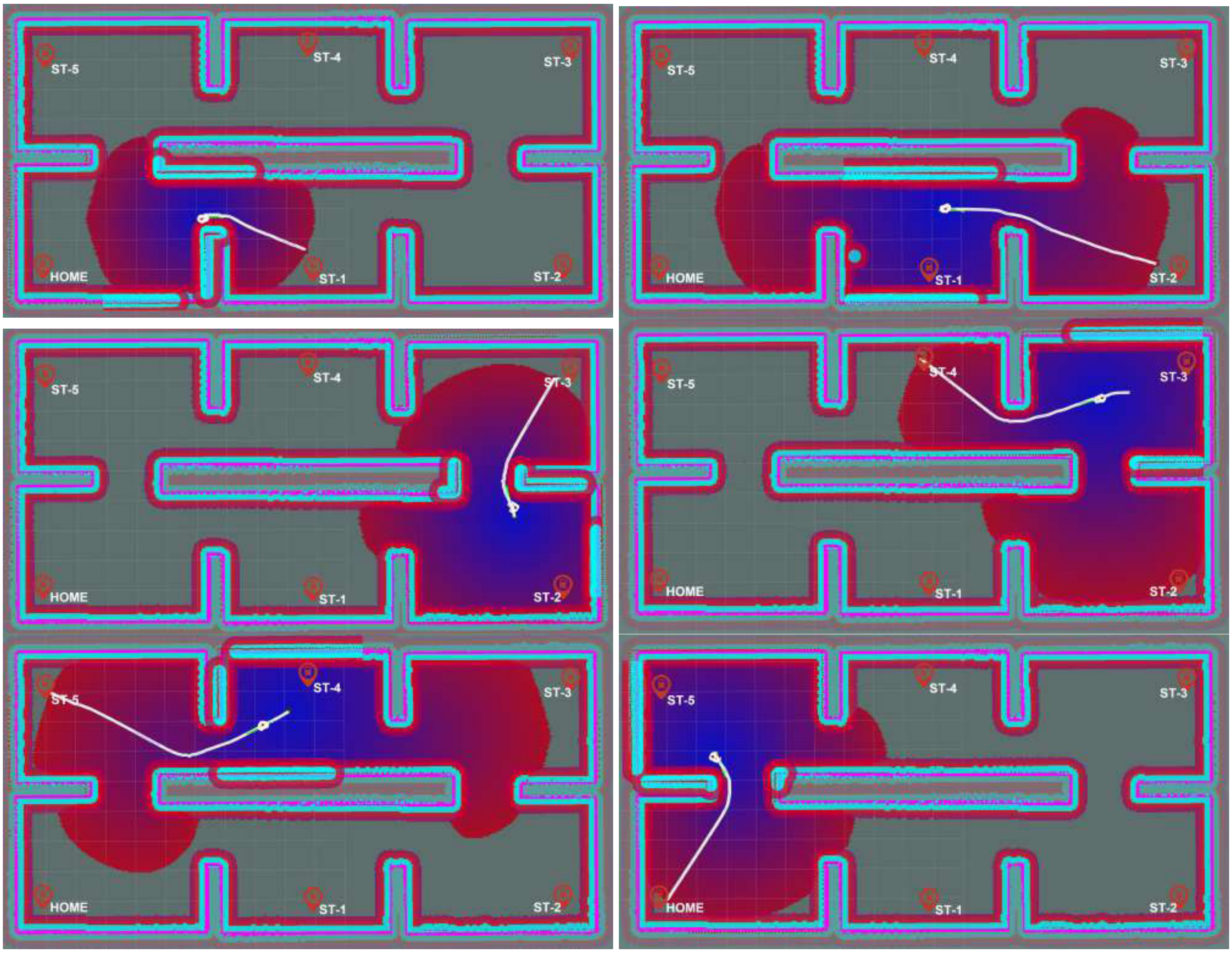

FIGURE 13. Autonomous navigation to each station.

VI. RESULT ANALYSIS

In this section, AMR has been asked to move autonomously to 6 different stations in total, including the pre-recorded starting point and 5 different stations points. DWA, TEB and Trajectory were used as local planners. Global_Planner was chosen as the global planner. The stations on the map in the interface given in Figure 7 are also given in Figure 13. In Figure 13 from home to Station-1, from Station-1 to Station-2, from Station-2 to Station-3, from Station-3 to Station-4, from Station-4 to Station-5, from Station-5 to the home position, images of autonomous movements on Rviz are given. Real time monitoring of AMR, time to reach each point and distance travelled were shown via the new GUI. In addition, the currently used local and global path planners can also be followed with the interface.

The default path planning algorithms (DWA Local Planner - TEB Local Planner - Trajectory Local Planner) offered by ROS are compared with the novel GUI design developed in this article in terms of path and time virtually. The test results are detailed in Table 3. In Table 3, the first row contains the names of the local planners. In the first column, it represents the stops and the starting point, that is, the home point. For example, if we examine the results of the experiment with the DWA local planner;

- While going from Home to St-1, it took 9.13 m and took 27.63 seconds.
- While going from St-1 to St-2, 8.67 m distance was traveled and 27.65 sec. it took.
- On the way from St-2 to St-3, a distance of 7.41 m was traveled and 24.99 seconds. it took
- While going from St-3 to St-4, 9.23 m distance was traveled and 27.65 seconds. it took.
- While going from St-4 to St-5, 9.94 m distance was traveled and 29.41 seconds. it took.
- On the way from St-5 to home, 8.02 m distance was traveled and 25.87 seconds. it took.
- A total of 52.4 m traveled and 163.20 seconds. it took.

Similarly, the measurements of the other two local planners were taken and transferred to Table 3. In this way, it has been shown that different algorithms can be compared in the interface. The comparison of algorithms is described in Section VI.

TABLE 3. Arrival times and distances between stations of local path planners.

	DWA Local Planner		TEB Local Planner		Trajectory Local Planer	
	Distance (m)	Time (sec)	Distance (m)	Time (sec)	Distance (m)	Time (sec)
St-1	9,13	27,63	10,88	45,00	9,39	28,87
St-2	8,67	27,65	9,68	36,58	8,91	27,38
St-3	7,41	24,99	8,48	30,17	7,77	25,32
St-4	9,23	27,65	9,9	37,58	9,55	29,43
St-5	9,94	29,41	10,43	38,56	9,61	30,21
Home	8,02	25,87	9,21	28,67	8,42	25,98
Total	52,40	163,20	58,58	216,56	53,65	167,19

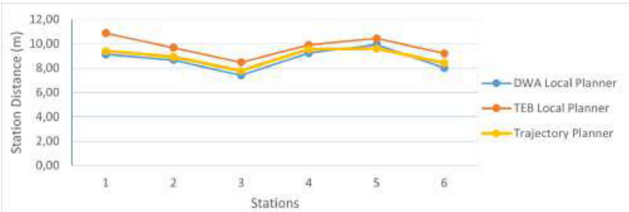

FIGURE 14. Arrival distances between stations.

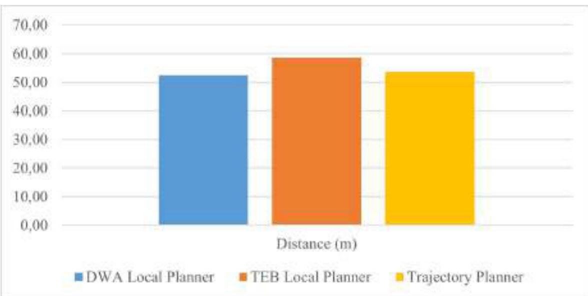

FIGURE 15. Total distance traveled as a result of visiting all stations.

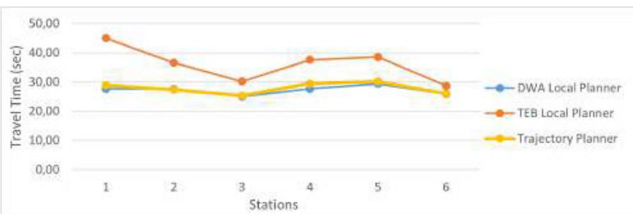

FIGURE 16. Arrival times between stations.

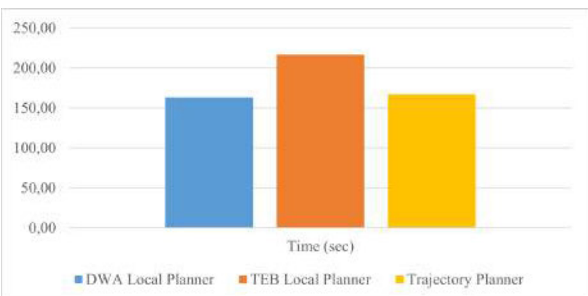

FIGURE 17. Total time elapsed after visiting all stations.

VII. CONCLUSION AND FUTURE WORK

In this section, the results of the autonomous movement of AMR with 3 different local planners to 6 different stations are compared. The comparison is based on time and distance.

Figure 14 shows the distance got over by the different local planners during the autonomous movement of the AMR between the stations. Figure 15 shows the total distance traveled by the AMR during its autonomous movement between all stations. In Figure 16, the arrival time spent by AMR during autonomous movement with different local planners between stations is shown. Figure 17 shows the total time spent by the AMR during its autonomous movement between all stations.

According to Table 3 in Section V, the graphs in Figure 15 and Figure 17 were obtained. When we look at the graph in Figure 15, it is seen that the shortest distance in total is 52.40 m, using the DWA local planner algorithm, while moving to 6 different stations. On the other hand, it is seen that the longest distance is 58.58 m using TEB local planner algorithm. When we look at Figure 17, it is seen that the shortest time in total for the movement to 6 different stations was completed at 163.20 seconds by using the DWA Local planner algorithm. On the other hand, it is seen that the longest time was completed at 216.56 seconds using TEB local planner algorithm. Thus, it was calculated that the DWA local planner was 10.55% more successful than the TEB local planner and 2.33% more successful than the Trajectory local planner in terms of the distance traveled to all stations. In addition, it was calculated that it was 24.64% more successful than the TEB local planner and 2.39% more successful than the Trajectory local planner in terms of the time it took to visit all stations.

In addition, it was observed that the shortest distance between stations was 7.41m, taken in 24.99 seconds when going from Station-2 to Station-3 using the DWA algorithm. On the other hand, it was observed that the longest distance between stations was 10.88m, taken in 45 seconds while going from Home to Station-2 with the TEB algorithm. When this situation, which has the same distances in appearance, was examined, it was seen that the DWA algorithm calculates shorter distances than the TEB algorithm.

In this study, a novel design based on ROS was made. Thanks to this interface, which has ROS connections in the background, AMR tracking was easily performed. The time and distances of the Local path-planning algorithms to reach the target were compared. Thanks to the developed interface such situations can be followed and controlled:

Thanks to the developed interface such situations can be followed and controlled;

- Instantaneous position and orientation of the AMR,
- Autonomous and manual control of AMR,
- Visualization of the locations of the stations as coordinates and image,

- Assigning tasks to determined points autonomously,
- How much time and distance travelled while moving between these stations,
- Which path planning algorithm is used,
- How accurately does it dock when it reaches the desired point?

By transferring the map of the environment to be studied to the GUI, the mentioned studies can be tried quickly and inexpensively. Given the above, we believe that this study is both academically new and will industrially beneficial.

On the other hand, when we look at the studies conducted in world-renowned academic platforms such as IEEE Xplore, ScienceDirect, Scopus and Web of Science in recent years, very few and basic level interface design studies have been carried out on ROS-based mobile robots. In this respect, it is thought that the interface study will be the first in its field and will contribute to the literature.

In future studies, it is planned to compare path planning algorithms in which meta-heuristic-based algorithms are used, unlike traditional algorithms. In addition, it is considered that the studies will be tested in the field in real time.

REFERENCES

[1] Y. Zhang, G. Guan, and X. Pu, "The robot path planning based on improved artificial fish swarm algorithm," *Math. Problems Eng.*, vol. 2016, pp. 1–11, Aug. 2016, doi: 10.1155/2016/3297585.

[2] A. Choudhary, Y. Kobayashi, F. J. Arjonilla, S. Nagasaka, and M. Koike, "Evaluation of mapping and path planning for non-holonomic mobile robot navigation in narrow pathway for agricultural application," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2021, pp. 17–22, doi: 10.1109/IEEECONF49454.2021.9382767.

[3] M. N. Zafar and J. C. Mohanta, "Methodology for path planning and optimization of mobile robots: A review," *Proc. Comput. Sci.*, vol. 133, pp. 141–152, 2018, doi: 10.1016/j.procs.2018.07.018.

[4] J. Zhao, S. Liu, and J. Li, "Research and implementation of autonomous navigation for mobile robots based on SLAM algorithm under ROS," *Sensors*, vol. 22, no. 11, p. 4172, May 2022, doi: 10.3390/s22114172.

[5] Z. Liu, Z. Lu, W. Zheng, W. Zhang and X. Cheng, "Design of obstacle avoidance controller for agricultural tractor based on ROS," *Int. J. Agricult. Biol. Eng.*, vol. 12, no. 6, pp. 58–65, 2019, doi: 10.25165/j.ijabe.20191206.4907.

[6] H. Gong, P. Wang, C. Ni, and N. Cheng, "Efficient path planning for mobile robot based on deep deterministic policy gradient," *Sensors*, vol. 22, no. 9, p. 3579, May 2022, doi: 10.3390/s22093579.

[7] J. Wang, X. Ruan, and J. Huang, "HDPP: High-dimensional dynamic path planning based on multi-scale positioning and waypoint refinement," *Appl. Sci.*, vol. 12, no. 9, p. 4695, May 2022, doi: 10.3390/app12094695.

[8] J. Chen, Y. Zhao, and X. Xu, "Improved RRT-connect based path planning algorithm for mobile robots," *IEEE Access*, vol. 9, pp. 145988–145999, 2021, doi: 10.1109/ACCESS.2021.3123622.

[9] G. Atali, İ. Pehlivan, B. Gürevin, and H. İ. Şeker, "Chaos in metaheuristic based artificial intelligence algorithms: A short review," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 29, no. 3, pp. 1354–1367, May 2021, doi: 10.3906/elk-2102-5.

[10] L. Yang, L. Fu, P. Li, J. Mao, N. Guo, and L. Du, "LF-ACO: An effective formation path planning for multi-mobile robot," *Math. Biosci. Eng.*, vol. 19, no. 1, pp. 225–252, 2022, doi: 10.3934/mbe.2022012.

[11] L. Banjanovic-Mehmedovic, I. Karabegovic, J. Jahic, and M. Omercic, "Optimal path planning of a disinfection mobile robot against COVID-19 in a ROS-based research platform," *Adv. Prod. Eng. Manage.*, vol. 16, no. 4, pp. 405–417, Dec. 2021, doi: 10.14743/apem2021.4.409.

[12] A. D. Sabiha, M. A. Kamel, E. Said, and W. M. Hussein, "Real-time path planning for autonomous vehicle based on teaching–learning-based optimization," *Intell. Service Robot.*, vol. 15, no. 3, pp. 381–398, Jul. 2022, doi: 10.1007/s11370-022-00429-3.

[13] J. Choi, G. Lee, and C. Lee, "Reinforcement learning-based dynamic obstacle avoidance and integration of path planning," *Intell. Service Robot.*, vol. 14, no. 5, pp. 663–677, Nov. 2021, doi: 10.1007/s11370-021-00387-2.

[14] Y. Zhou, Y. Su, A. Xie, and L. Kong, "A newly bio-inspired path planning algorithm for autonomous obstacle avoidance of UAV," *Chin. J. Aeronaut.*, vol. 34, no. 9, pp. 199–209, Sep. 2021, doi: 10.1016/j.cja.2020.12.018.

[15] P. Li, X. Ding, H. Sun, S. Zhao, and R. Cajo, "Research on dynamic path planning of mobile robot based on improved DDPG algorithm," *Mobile Inf. Syst.*, vol. 2021, pp. 1–10, Nov. 2021, doi: 10.1155/2021/5169460.

[16] E.-T. Baek and D.-Y. Im, "ROS-based unmanned mobile robot platform for agriculture," *Appl. Sci.*, vol. 12, no. 9, p. 4335, Apr. 2022, doi: 10.3390/app12094335.

[17] J. Ren, T. Wu, X. Zhou, C. Yang, J. Sun, M. Li, H. Jiang, and A. Zhang, "SLAM, path planning algorithm and application research of an indoor substation wheeled robot navigation system," *Electronics*, vol. 11, no. 12, p. 1838, Jun. 2022, doi: 10.3390/electronics11121838.

[18] H. Wang, S. Lou, J. Jing, Y. Wang, W. Liu, and T. Liu, "The EBS-A* algorithm: An improved A* algorithm for path planning," *PLoS ONE*, vol. 17, no. 2, Feb. 2022, Art. no. e0263841, doi: 10.1371/journal.pone.0263841.

[19] C. S. Chen, C. J. Lin, C. C. Lai, and S. Y. Lin, "Velocity estimation and cost map generation for dynamic obstacle avoidance of ROS based AMR," *Machines*, vol. 10, no. 7, p. 501, Jun. 2022, doi: 10.3390/machines10070501.

[20] W.-C. Wang, C.-Y. Ng, and R. Chen, "Vision-aided path planning using low-cost gene encoding for a mobile robot," *Intell. Autom. Soft Comput.*, vol. 32, no. 2, pp. 991–1006, 2022, doi: 10.32604/iasc.2022.022067.

[21] R. K. Megalingam, B. Tanmayi, G. M. Reddy, I. R. S. Krishna, G. S. Sree, and S. S. Pai, "ROS-based GUI controlled robot for indoor mapping and navigation," in *Computer Networks and Inventive Communication Technologies*, 2021, pp. 283–294, doi: 10.1007/978-981-15-9647-6_22.

[22] D. Bore, A. Rana, N. Kolhare, and U. Shinde, "Automated guided vehicle using robot operating systems," in *Proc. 3rd Int. Conf. Trends Electron. Informat. (ICOEI)*, Apr. 2019, pp. 819–822, doi: 10.1109/ICOEI.2019.8862716.

[23] A. Astudillo, A. Sathya, D. Dirckx, W. Decré, J. Gillis, G. Pipeleers, and J. Swevers, "A web-based graphical user interface for programming optimal control based robot motion tasks," *IFAC-PapersOnLine*, vol. 55, no. 27, pp. 120–125, 2022, doi: 10.1016/j.ifacol.2022.10.498.

[24] H. T. Warku, N. Y. Ko, G. Song, D. B. Jeong, B. Lee, T. E. Derbew, and S.-S. Hwang, "Autonomous navigation of a mobile robot on robot operating system using graphical user interface," in *Proc. Joint 12th Int. Conf. Soft Comput. Intell. Syst., 23rd Int. Symp. Adv. Intell. Syst. (SCIS&ISIS)*, Nov. 2022, pp. 1–6, doi: 10.1109/SCISISIS55246.2022.10001940.

[25] S. Rosa, L. O. Russo, and B. Bona, "Towards a ROS-based autonomous cloud robotics platform for data center monitoring," in *Proc. IEEE Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2014, pp. 1–8, doi: 10.1109/ETFA.2014.7005212.

[26] P. Cheema, S. Luo, and P. Gibbens, "Development of a control and vision interface for an AR.Drone," in *Proc. MATEC Web Conf.*, vol. 56, 2016, p. 7002, doi: 10.1051/matecconf/20165607002.

[27] C. Paolini and G. K. Lee, "A Web-based user interface for a mobile robotic system," in *Proc. IEEE 13th Int. Conf. Inf. Reuse Integr. (IRI)*, Aug. 2012, pp. 45–50, doi: 10.1109/IRI.2012.6302989.

[28] *ROS Wiki, Global_Planner ROS Package*. Accessed: Aug. 21, 2022. [Online]. Available: http://wiki.ros.org/global_planner

[29] *ROS Wiki, Dwalocalplanner ROS Package*. Accessed: Aug. 21, 2022. [Online]. Available: http://wiki.ros.org/dwalocalplanner

[30] *ROS Wiki, Teblocalplanner ROS Package*. Accessed: Aug. 21, 2022. [Online]. Available: http://wiki.ros.org/teblocalplanner

[31] *ROS Wiki, Baselocalplanner ROS Package*. Accessed: Aug. 21, 2022. [Online]. Available: http://wiki.ros.org/baselocalplanner

[32] L. Chang, L. Shan, C. Jiang, and Y. Dai, "Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment," *Auton. Robots*, vol. 45, no. 1, pp. 51–76, Jan. 2021, doi: 10.1007/s10514-020-09947-4.

[33] I. Naotunna and T. Wongratanaphisan, "Comparison of ROS local planners with differential drive heavy robotic system," in *Proc. Int. Conf. Adv. Mech. Syst. (ICAMechS)*, Dec. 2020, pp. 1–6, doi: 10.1109/ICAMechS49982.2020.9310123.

[34] B. Cybulski, A. Wegierska, and G. Granosik, "Accuracy comparison of navigation local planners on ROS-based mobile robot," in *Proc. 12th Int. Workshop Robot Motion Control (RoMoCo)*, Jul. 2019, pp. 104–111, doi: 10.1109/RoMoCo.2019.8787346.

[35] D. To, "Custom global planner for robots in retail environment," *Math. Problems Eng.*, 2016.

[36] S. Karakaya and H. Ocak, "Açısal Duruş Kontrolü Destekli Özgün bir Dinamik Pencere Yaklaşımı," *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi*, vol. 7, pp. 184–200, Jun. 2000, doi: 10.35193/bseufbd.705765.

[37] P. Xu, N. Wang, S.-L. Dai, and L. Zuo, "Motion planning for mobile robot with modified BIT* and MPC," *Appl. Sci.*, vol. 11, no. 1, p. 426, Jan. 2021, doi: 10.3390/app11010426.

BILAL GUREVIN received the B.S. and M.S. degrees in mechatronics engineering from Sakarya University, Sakarya, Turkey, in 2016 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Sakarya University of Applied Sciences. He is a Researcher with Liverpool John Moores University. His current research interests include autonomous mobile robots (AMRs), path-planning algorithms, meta-heuristic algorithms, and interface applications.

FURKAN GULTURK received the B.S. degree in mechatronic engineering from Sakarya University, Sakarya, Turkey, in 2017. He is currently pursuing the Ph.D. degree in electrical and electronics engineering with Sakarya Applied Science University, and the Ph.D. degree in autonomous mobile robot-based intralogistics systems, with a focus on the Tübitak 2244 Project. His current research interests include autonomous mobile robots (AMRs), path planning algorithms, localization algorithms, sensor fusion algorithms, and fleet management algorithms.

MUHAMMED YILDIZ received the bachelor's degree in mechatronics engineering from Firat University and the master's degree from the Sakarya University of Applied Sciences. He is currently pursuing the Ph.D. degree in autonomous logistics storage systems, with a focus on the Tübitak 2244 Project. In the thesis, he worked on upper extremity rehabilitation.

IHSAN PEHLIVAN received the B.S. degree from Istanbul Technical University, in 1997, and the M.S. and Ph.D. degrees in electrical and electronic engineering from Sakarya University, Turkey, in 2001 and 2007, respectively. He is currently a Professor with the Department of Electrical-Electronics Engineering, Sakarya University of Applied Sciences. He is the author of 45 SCI articles and 80 conference papers. He has two international and one national patents. His current research interests include analog electronics, dynamical systems, chaos, chaotic systems, circuit theory, chaotic systems, chaotic circuits, chaotic synchronization, chaotic communication, random number generators, FPGA-based chaotic circuits and random number generators, and chaotic mixers.

TRUNG THANH NGUYEN is currently a Reader of operational research (OR) with Liverpool John Moores University and the Co-Director of the Liverpool Offshore and Marine Research Institute. He has an international standing in operational research for logistics/transport. He has led more than 20 research projects in transport/logistics most with close industry collaborations. He has published about 50 peer-reviewed articles. All of his journal articles are in leading journals (ranked 1st–20th in their fields). He has edited eight books and given speeches at many conferences/events. He co-organized six leading conferences. He was a TPC member of more than 30 international conferences.

FATIH CALISKAN received the master's degree in materials science, in 2006, and the Ph.D. degree, in 2010. He was a Materials Scientist with Sakarya University, from 2000 to 2004. He was a Visiting Lecturer with the University of Limerick, Ireland, in 2008, 2009, and 2012. He has lectured on nitride ceramics to postgraduate students. As a Guest Lecturer within the context of a special intensive program, he attended Governors State University, USA, in 2013. He is currently a Professor of materials science engineering with the Sakarya University of Applied Sciences, Turkey. In recent years, he has focused on industry 4.0 applications. He is conducting many projects on them.

BARIS BORU received the B.S. degree in electronics technical teacher from Sakarya University, in 2005, and the M.S. and Ph.D. degrees from the Institute of Natural Sciences, Department of Electronics and Computer Education, Sakarya University, in 2007 and 2012, respectively. The M.S. thesis is on "Intelligent Video Segmentation Algorithms" and the Ph.D. thesis is on "Performance and Emission Optimization of Diesel Engines Using Blind Measurement and Artificial Intelligence Technics." In 2014, he was a Visiting Researcher with Texas A&M University. He is currently an Associate Professor with the Department of Mechatronics Engineering, Sakarya University of Applied Sciences.

MUSTAFA ZAHID YILDIZ received the Ph.D. degree in biomedical engineering from Boğaziçi University, Istanbul, Turkey, in 2013. He is currently a Faculty Member with the Electrical and Electronics Engineering, Sakarya University of Applied Sciences, Sakarya, Turkey, and the Director of the Biomedical Research Center, Sakarya University of Applied Sciences. His current research interests include biomedical instrumentation, biomedical signal processing, and bio-optics.