Düzce Üniversitesi
Bilim ve Teknoloji Dergisi

Araştırma Makalesi

A Novel Control and Monitoring Interface Design for ROS Based Mobile Robots

Bilal GUREVIN [a, *], Muhammed YILDIZ [b], Furkan GULTURK [b], Ihsan PEHLIVAN [c], Fatih CALISKAN [d], Baris BORU [e], M. Zahid YILDIZ [c]

[a] *Department of Mechatronics Eng., , Graduate School of Education, Sakarya University of Applied Sciences, Sakarya, TURKEY*
[b] *Department of Electrical and Electronics Eng., Graduate School of Education, Sakarya University of Applied Sciences, Sakarya, TURKEY*
[c] *Department of Electrical and Electronics Eng., Faculty of Technology, Sakarya University of Applied Sciences, Sakarya, TURKEY*
[d] *Department of Metallurgy and Materials Eng., , Faculty of Technology, Sakarya University of Applied Sciences, Sakarya, TURKEY*
[e] *Department of Mechatronics Engineering, , Faculty of Technology, Sakarya University of Applied Sciences, Sakarya, TURKEY*
** Sorumlu yazarın e-posta adresi: bilalsau@gmail.com*
DOI: https://doi.org/10.29130/dubited.1214278

ABSTRACT

In this study, an interface design was carried out in order to provide convenience to the user in the control and monitoring of the Robot Operating System (ROS) based autonomous mobile robot (AMR). Qt Designer and Python were used in the interface design. Thanks to the designed interface, autonomous and manual control of AMR was provided. Using the gmapping algorithm, the environment in the virtual world was mapped and transformed into a picture in .png format and visualized in the interface. The location information from the ROS was transferred to the said picture and the instant tracking of the AMR was done via the interface. It was shown which algorithm is used locally and globally at that moment. While in autonomous mode, the vehicle was provided to move to the previously recorded point. The total distance and time spent by the AMR while moving between two points were also calculated by the interface. The location (x, y, z) and orientations (x, y, z, w) of the previously recorded station were monitored from the stop list. On the other hand, the position (x, y, z) and orientation (x, y, z, w) information of the AMR could be followed as real time via the interface. In this way, when the AMR reaches the goal station, the time elapsed between two points, the transportation distance, settlement information such as the location and orientation of the vehicle can be tracked and be compared via the interface.

Anahtar Kelimeler: *AMR, GUI, Path Planning, ROS*

ROS Tabanlı Mobil Robotlar İçin Yeni Bir Kontrol ve Görüntüleme Arayüz Tasarımı

ÖZ

Bu çalışmada, Robot İşletim Sistemi (ROS) tabanlı otonom mobil robotun (AMR) kontrolünde ve izlenmesinde kullanıcıya kolaylık sağlamak amacıyla bir arayüz tasarımı gerçekleştirilmiştir. Arayüz tasarımında Qt Designer ve Python kullanılmıştır. Tasarlanan arayüz sayesinde AMR'nin otonom ve manuel kontrolü sağlanmıştır. Gmapping algoritması kullanılarak sanal dünyadaki ortam haritalanarak .png formatında resme dönüştürülerek arayüzde görselleştirilmiştir. ROS'tan gelen konum bilgisi söz konusu resme aktarılmış ve arayüz üzerinden AMR'nin anlık takibi yapılmıştır. Lokal ve global olarak o an hangi algoritmanın kullanıldığı gösterilmiştir. Otonom modda iken aracın önceden kaydedilen noktaya hareket etmesi sağlanmıştır. AMR'nin iki nokta arasında hareket ederken aldığı toplam mesafe ve süre de arayüz tarafından hesaplanmıştır. Durak listesinden daha önce kaydedilen istasyonun konumu (x, y, z) ve yönleri (x, y, z, w) takip edilebilmiştir. Öte yandan AMR'nin konum (x, y, z) ve yön (x, y, z, w) bilgileri arayüz üzerinden gerçek zamanlı olarak takip edilebilmiştir. Bu sayede AMR hedef istasyona ulaştığında, iki nokta arasında geçen süre, ulaşım mesafesi, aracın konumu, yönü gibi yerleşim bilgileri arayüz üzerinden takip edilip karşılaştırılabilmiştir.

Keywords: *AMR, GUI, Path Planning, ROS*

Geliş: 04/12/2022, Düzeltme: 08/04/2023, Kabul: 09/04/2023

I. INTRODUCTION

Today, the use of AMR is increased day by day. The use of AMRs is expanded in many areas from the food sector to agriculture, from the health sector to animal husbandry [1-4]. The use of AMRs can be carried out indoors as well as outdoors. For example, AMRs are used in various fields such as the transportation of products between processes in a factory environment, and the transmission of goods leaving the line to the warehouse. ROS-based AMR studies are quite common in the academic field as well as in the industry. Some of the studies carried out are as mentioned below.

Liu et al. carried out a study in which A* and one-way graph method algorithms were used in the path planning of multiple AGVs. The studies were tested in a simulation environment with 20 AGVs [5]. Zhang et al. examined the algorithms used in path planning studies of mobile robots. They expressed the most widely used studies in path planning. And they gave advice on future research [6]. Secchi et al. carried out a study to increase the efficiency of the AGV fleet in terms of delivery time. The studies were tested in a small-scale warehouse [7]. Dongyong et al. carried out a multi-purpose algorithm study against problems such as path planning, avoiding obstacles, and time costs in the movements of multiple robots [8].

Goyal and Nagla presented a new path planning algorithm to improve the ability of mobile robots moving in closed areas to pass through narrow places such as doors. By modifying the A* algorithm, they performed the movement of the robot to the target point without hitting obstacles, in the simulation environment, thanks to the new algorithm they presented [9]. In their study, Park et al. made a path planner that determines the appropriate areas to be crossed with a hierarchical road map while the mobile robot navigates in an unknown environment. They tested the study in a home environment [10].

Aslan and Yazıcı carried out a study to prevent the collision problem of multiple autonomous carriers. Thanks to the algorithm they developed based on the A* algorithm, they realized the path planning of multiple mobile robots moving between multiple start and end points. They provided path planning without conflicts with the tests performed [11]. Garip Batik et al. proposed a path planning study to prevent collisions of more than one mobile robot while moving in the same environment. With the images taken from the camera placed on the ceiling, the positions of the objects and robots were determined. And they made path planning with the help of the A* algorithm. They showed the path planning tracking with the interface they designed in Matlab GUI [12].

Metaheuristic-based studies have also started to be preferred in path planning algorithms due to their computational speed [13]. Yıldırım and Akay worked on optimal path planning in mobile robots. They created fixed objects in certain shapes with the interface they designed in the Matlab interface. In the system where the starting and destination points were entered, they carried out the route planning to the target point by using PSO (Particle Swarm Optimization), ABC (Artificial Bee Colony) and GA (Genetic Algorithm). They showed that ABC gave more successful results than other algorithms when compared in terms of the shortest path [14]. Yıldırım and Rüştü compared linear and nonlinear objective functions in their study on path planning of mobile robots. They showed that the linear objective function was superior in terms of both time and distance in the study in which the GA was used in global path planning [15].

In this study, an interface design has been carried out in order to provide ease of manual control and tracking of AMR during autonomous movement. AMR must choose the appropriate route while performing its autonomous movement on the known map. In this way, it is possible to reach the destination in the shortest way and in the shortest time. At the same time, the position and accuracy of its position when it settles in the target station are important according to the task it will perform at that station. For example, it may be necessary to be sensitive to the accuracy of the point placed in the process of being integrated into a line at the arriving station or picking up / dropping a product. With the interface developed in line with these needs, the time spent and the distance covered by the AMR between two points during its autonomous movement can be calculated and followed by the user. In addition, the location and position information of the target station and the location and position information at the time the station is reached can also be compared thanks to the interface. Detailed information about the interface design is given in Chapter 3. Virtual environment studies were carried out in the Gazebo program. Rviz program was used to visualize the data coming from the sensors and to follow the vehicle and route. A model originally developed as AMR was used in the virtual environment studies.

In the continuation of the study, in Chapter 2; The path planning algorithms used are mentioned. In Chapter 3; interface, AMR design and working environment are mentioned. Finally, in Chapters 4 and 5, the experimental results and interpretations of the studies using the interface are given.

II. ALGORITHMS

Local and global path planners, which are frequently used in the ROS ecosystem, were preferred to test the autonomous movement of AMR with the developed interface. Dynamic Window Approach local planner (DWA) [16] was chosen as the local planner, and global_planner [17] was chosen as the global planner.

A. LOCAL PLANNER

AMR needs local planner to detect and avoid surrounding objects while going to the determined target. In this study, DWA local planner was preferred because of its fast computation [18]. In DWA, an objective function is created by considering the distance to the objects and the AMR speed. The mobile robot moves concentrically with the dynamic window drawn around it. Calculations are made in this window. Objective function includes distance to nearest obstacle, robot speed, local target distance and distances to global orbit. According to this function, angular and rotational speed pairs are formed [19, 20].

B. GLOBAL PLANNER

Global planners are used to calculate the optimal path to the destination in the known environment [21]. In this study, the global_planner offered by ROS and using the Dijkstra algorithm was chosen as the global planner. Dijkstra divides the global map into cells to form a grid-cell map. From the starting point to the target, values are assigned to each cell according to the distance from the mobile robot. In each iteration, the path towards the target is established by looking at the occupancy of the next neighboring cells [18].

III. SYSTEM IMPLEMENTATION

In this section, the developed GUI and the original AMR design were included. In addition, the ROS version and computer features used were mentioned.

A. GUI Design

In the user interface design phase of the study, Qt Designer program and Python software language were used. The designed interface consists of three main tabs in total. These are; “Amr Manual Control”, “Amr Autonomous Control”, “Log Page” tabs.

A. 1. AMR Manuel Control

In the Amr Manual Control tab, under the title of Amr Indicator, information about the linear and angular velocities of the vehicle, the angle it makes with the horizontal x-axis, and its position on the x-y axis can be followed. Under the Manual Control title shown in Figure 1, linear and angular speed can be given to the vehicle and manual control can be achieved with direction buttons.

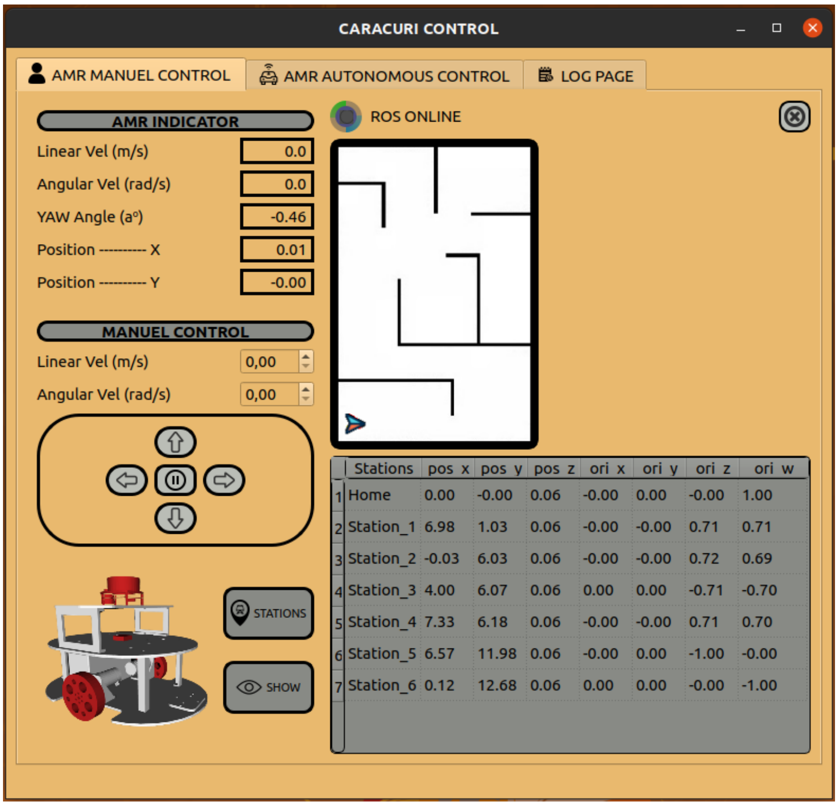

Figure 1. *Manuel control interface section*

It is possible to follow the vehicle real time on the map in the interface. As seen in Figure 2, it can be seen that the position of the vehicle in the Rviz environment (right image) and its positions in the interface (left image) and orientations are the same.

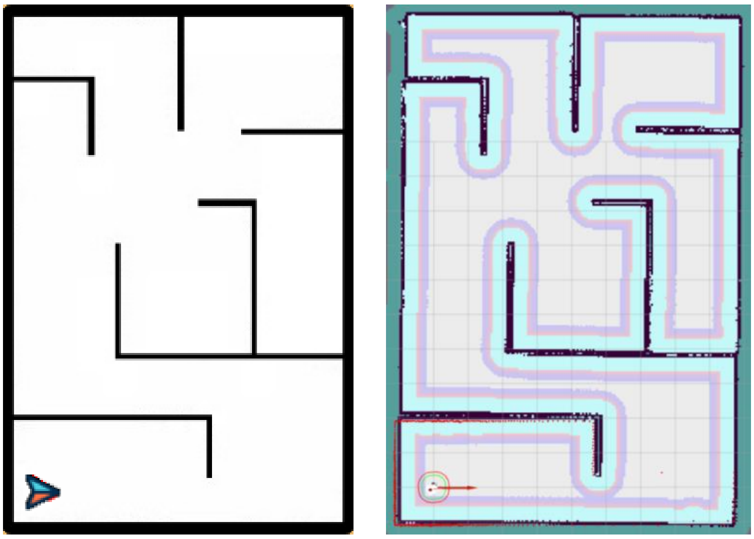

__Figure 2.__ AMR tracking on interface and Rviz

Pressing the Stations / Manual button switches between two different sections. One of these sections is the section under the title Manual Control. The other is the section under the Station Save and Delete heading. Under the Save and Delete Station title, the location information at the point where the vehicle is located can be saved with a name given by the user. During the recording, the position (x, y, z) indicating the position of the vehicle and the orientation (x, y, z, w) indicating the orientation of the vehicle are received. When the Ref button is pressed, the recorded stops are listed in the table on the right with location information in tabular form. Likewise, the desired stop can be selected and deleted. With the Show/Hide button, the locations of the stations on the map can be shown or hidden by station icons. The mentioned features of the interface are shown in Figure 3.

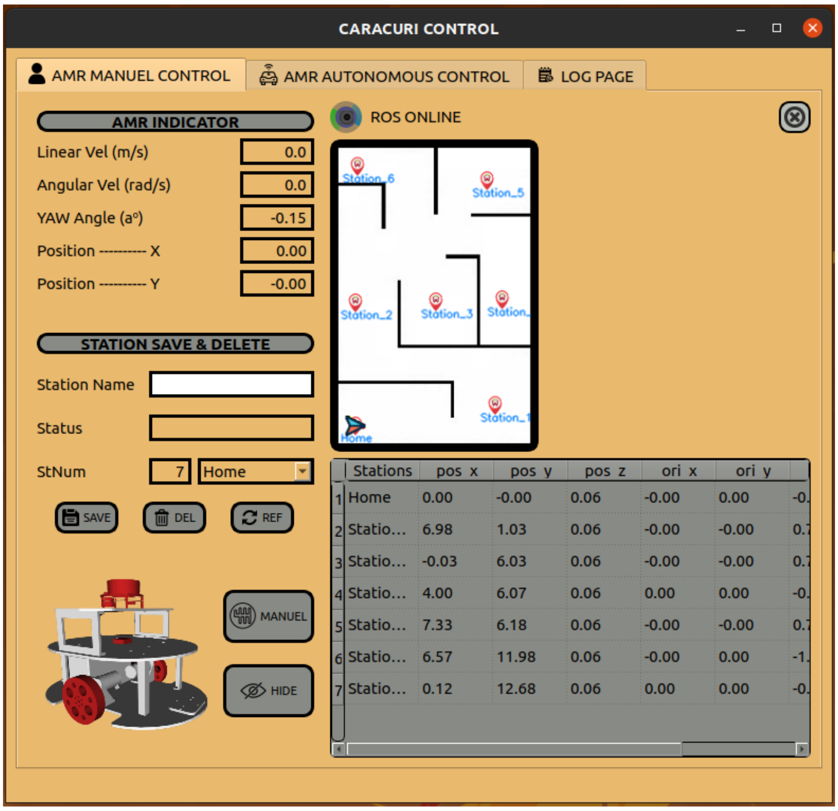

__Figure 3.__ Monitoring of stations in the interface

A. 2. AMR Autonomous Control

AMR can be controlled autonomously in the Amr Autonomous Control tab. The activity of the ROS connection can be seen thanks to the animated icon in the upper middle of the screen. It is possible to exit the program with the cross icon in the upper right corner. In the lower right part of the screen, the previously saved stations are listed with their name, location and position information. After selecting the desired station under the Go To Station title, the autonomous movement of the AMR can be achieved by assigning a task with the Go button. In the same way, the AMR can be stopped by canceling the task thanks to the Stop button during the movement. Under the Amr Indicator title, the real time linear velocity, angular velocity, orientation angle, total distance of the vehicle and how long it took to take that path are displayed. Under the title of Amr Position Indicator, the instant location and position information of the vehicle is displayed. In this way, when the vehicle arrives at any station, the location and position information of the previously recorded station can be compared with the location and position information at the time of arrival. In this way, it is possible to compare which of the different route planning algorithms takes the shortest route and arrives in the least time. Finally, under the title of Path Planning Algorithm, it is shown which path planning algorithms are used locally and globally. The features mentioned in the interface are shown in Figure 4.

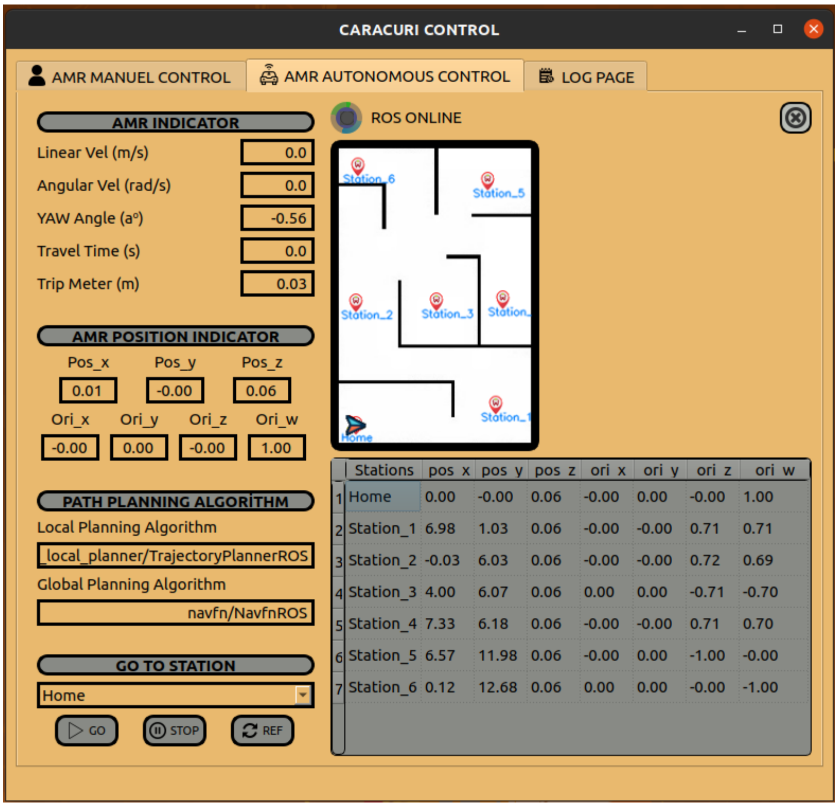

Figure 4. *AMR autonomous control via interface*

A. 2. Log Page

The third and last section used in the interface design is the section called Log Page. In this section, the error outputs of the python-based software run in the ROS and interface program can be viewed by the user. Thanks to the "try-except" structure, the error in which part of the software is directly printed on the log screen. Log Page section of the interface is as shown in Figure 5.

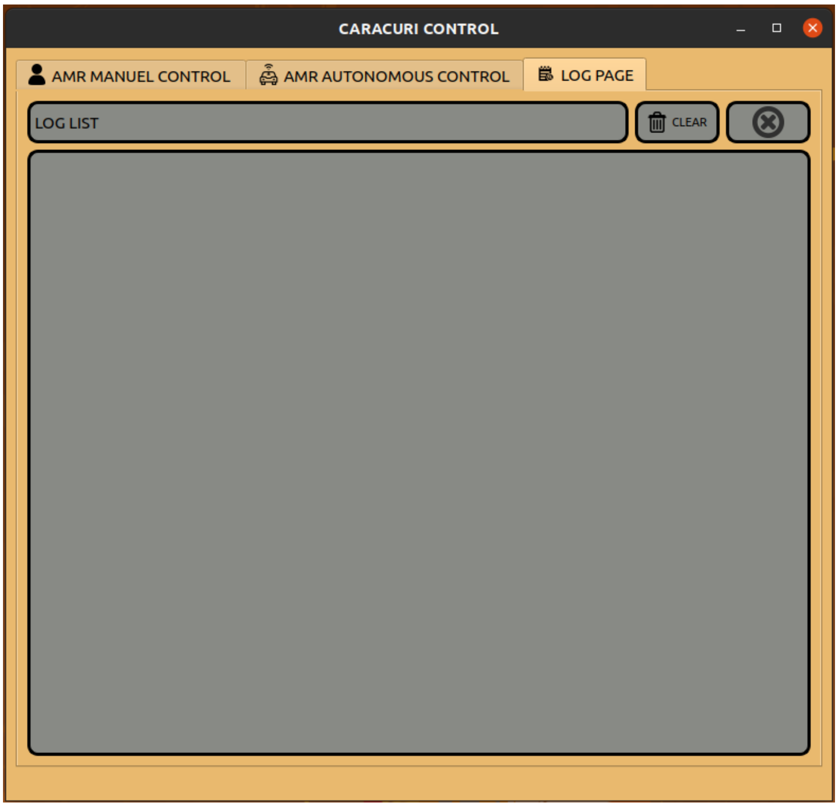

Figure 5. *AMR Log page section*

B. AMR DESIGN

In the virtual environment studies carried out in the ROS environment, the originally designed AMR model was used instead of ready-made models such as Turtlebot. SolidWorks solid model design program was used in AMR design. By using the URDF export extension of the SolidWorks program, AMR was transferred to the ROS environment. There are Lidar, IMU, Encoder sensors on the AMR. Thanks to these sensors, mapping and localization processes were carried out. Differential driving system was preferred as the driving system. The AMR design is as shown in Figure 6. The sensor plugin and technical features of the AMR used in the study carried out in the virtual environment are as shown in Table-1.

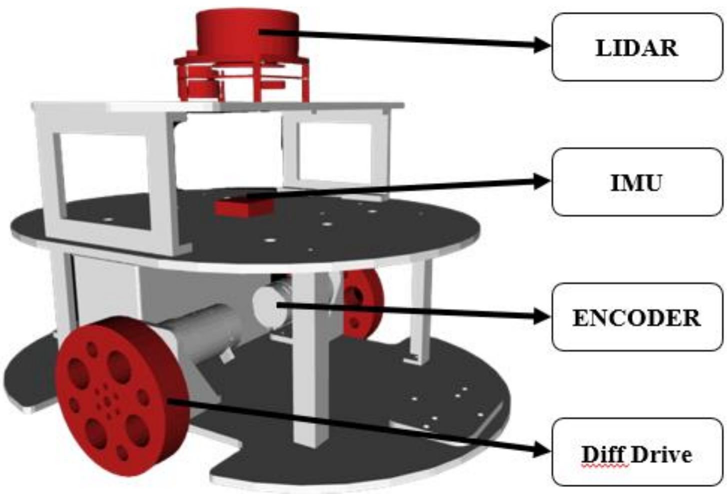

Figure 6. *AMR design*

Table 1. PC and ROS Features

Drive system	Mid-drive differential drive system
IMU sensor	plugin filename="libgazebo_ros_imu_sensor.so"
Lidar sensor	plugin name="gazebo_ros_head_hokuyo_controller"
Motor encoder	plugin name="differential_drive_controller"
Wheel diameter	10 cm
Distance between wheels	21 cm
AMR dimensions	27 x 27 x 35 (cm)

The transform tree obtained from the URDF format of AMR is as shown in Figure 7. Thanks to the transform tree, the structures to which each limb of the robot is connected can be seen. Here chassis_link refers to the main body of the AMR. IMU sensor, lidar sensor, right and left drive wheels as well as front and rear caster wheels are integrated into the main body. AMR takes the map information as a global reference in its environment and determines its location thanks to the odom data. Positions the main body within the detected point. In this way, the location and position information of the AMR is detected and visualized.

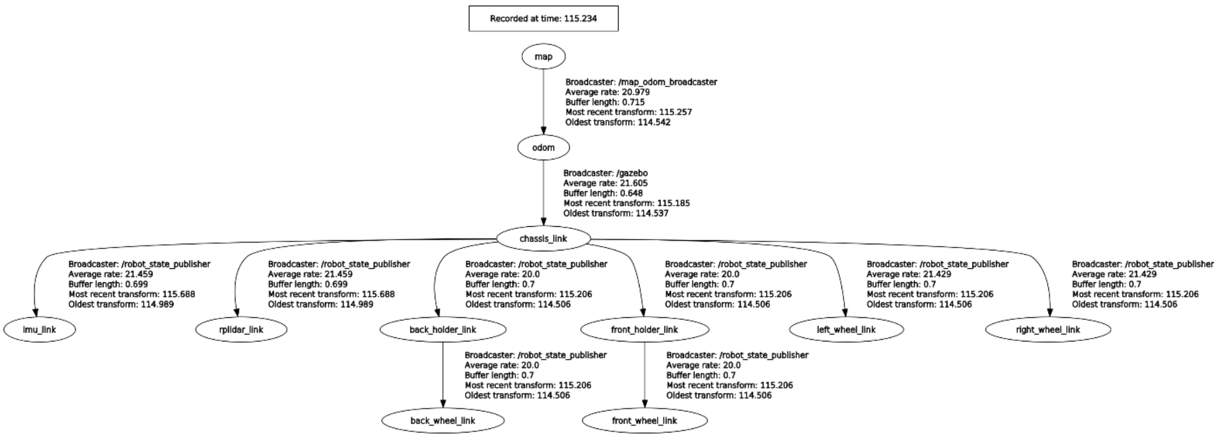

Figure 7. *AMR URDF transform tree*

C. ROS and PC HARDWARE

The computer and ROS information on which the studies were carried out are as shown in the Table 2 below.

Table 2. PC and ROS Features

OS	Ubuntu 20.04
RAM	32 GB
CPU	İntel Core İ7-9750H @ 12x4,5 GHz
GPU	GeForce RTX 2080 Mobile
ROS	Noetic

Structures called nodes in the ROS ecosystem communicate with each other on the principle of publishing and subscribing to the broadcast. In the study carried out, the node communications in the ROS system are as shown in Figure 8. For example, map information published from /map_server and goal information published from /move_base_simple are read by /move_base. This information is then processed and published by /move_base. The topic published by /move_base is read by /cmd_vel and the vehicle in the gazebo is moved. Similarly, the continuing cycle can be seen on the figure.

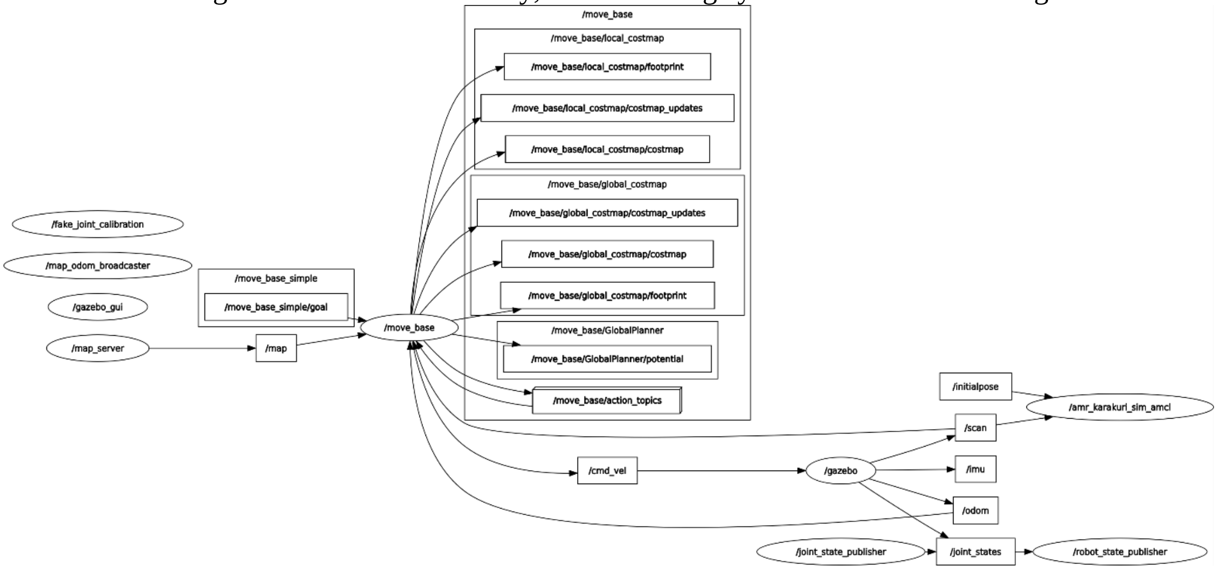

Figure 8. *ROS node graph diagram*

D. EXPERIMENT SETUP

Gazebo was used as a virtual world to compare the performance of local planners used in the ROS ecosystem [22]. Here, a closed environment with chambers of 15x10 m was created. The AMR was placed at the starting point and made ready for assignment. Figure 9 shows the view of the virtual world in Gazebo and the AMR at the starting point.

Figure 9. *Gazebo environment used in the study*

The visualization of sensor information and map information received from AMR was performed in Rviz. Thanks to Rviz, it is possible to visualize many data such as AMR's environment and path planning, real time position, orientation, laser and obstacle [23]. Figure 10 shows the location, orientation and map information of the AMR in the Rviz environment.

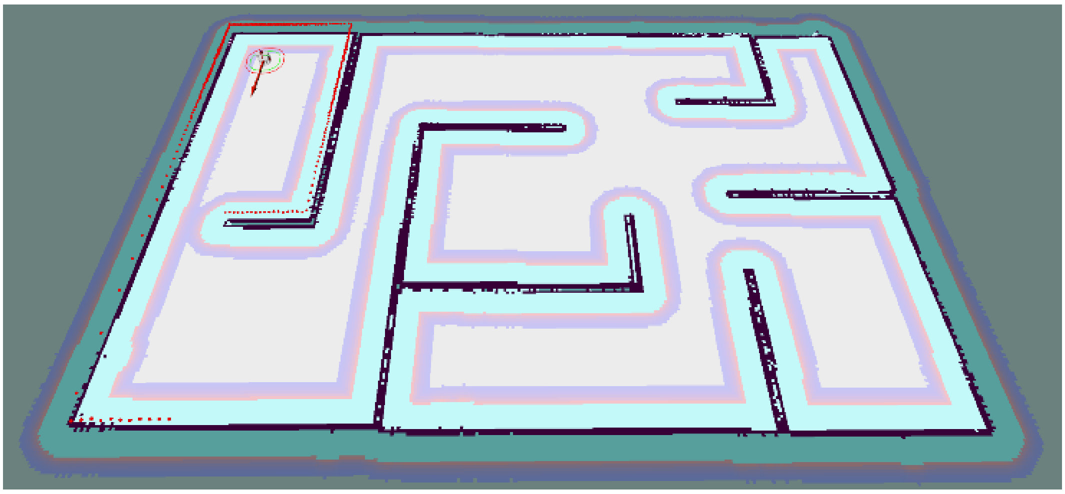

Figure 10. *Visualization of Gazebo environment in Rviz*

IV. RESULTS ANALYSIS

In this chapter; Autonomous control and monitoring of AMR was performed using the developed interface. AMR was requested to go autonomously to 6 different previously registered stations. For this, the desired station was selected from the list and the “GO” button was pressed. Figure 11 shows the Rviz images of the AMR starting from the "Home" point and going to 6 stops in sequence. The time to reach each station and the distance between each station were tracked through the interface. The results obtained were expressed in Table 3. The position and orientation information of the previously recorded stations and the position and location information when the AMR is placed at the station were shown in Table 4 by pulling from the interface. A detailed discussion of the results obtained is given in section 5.

Table 3. *Arrival Times Of Amr And Distance Values Of The Roads*

	Distance (m)	**Time (sec)**
St-1	7,34	17,11
St-2	9,48	21,86
St-3	5,20	16,52
St-4	8,73	25,72
St-5	7,57	21,66
St-6	9,17	24,25
Total	47,49	127,12

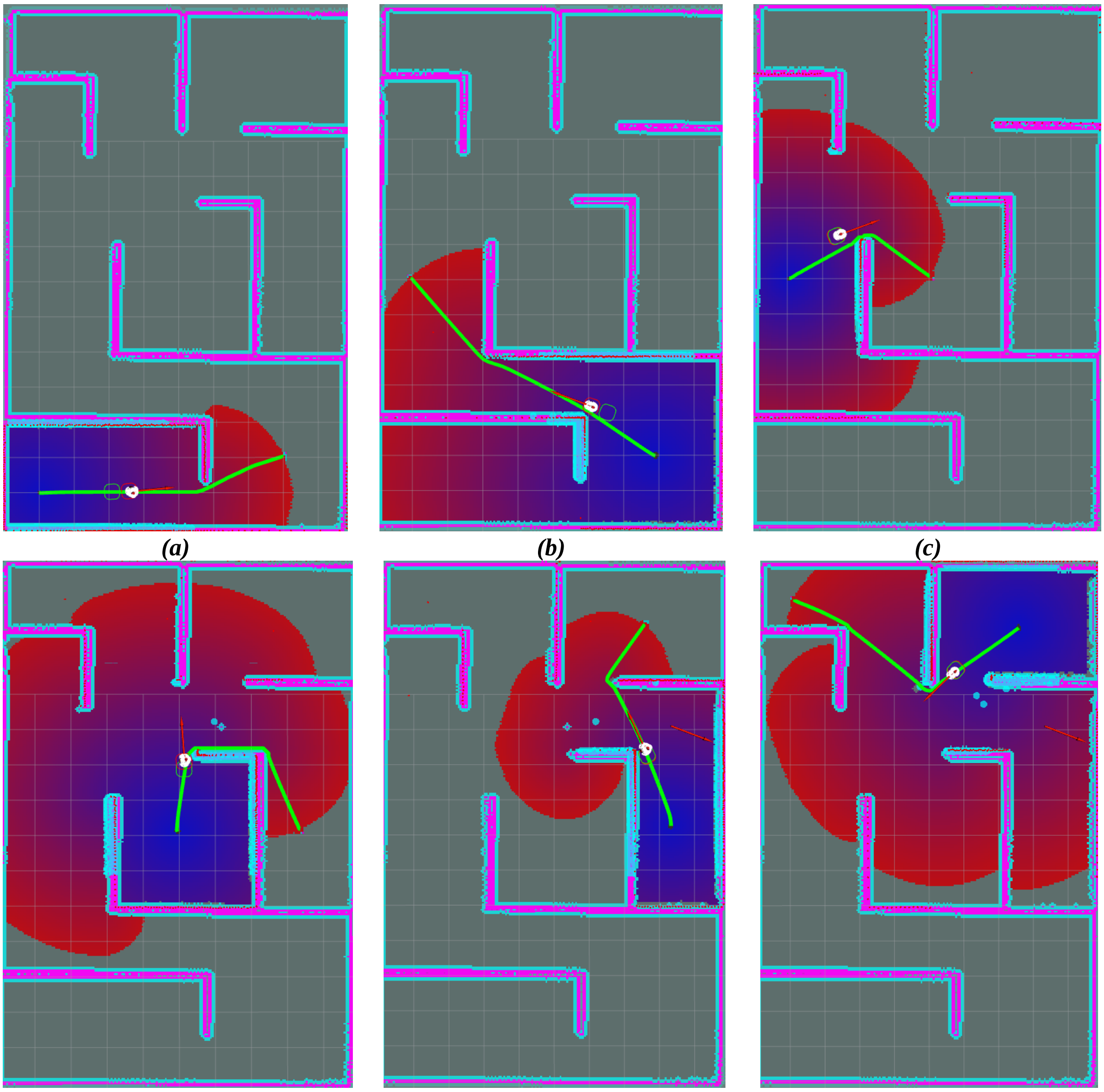

Figure 11. *Assigning tasks to six different stations in order (a) from home to station-1 (b) from station-1 to station-2 (c) from station-2 to station-3 (d) from station-3 to station-4 (e) from station-4 to station-5 (f) from station-5 to station-6*

Table 4. *Comparison Table Of Position And Orientation Values*

		Pos-X	Pos-Y	Pos-Z	Ori-X	Ori-Y	Ori-Z	Ori-W
St-1	Goal	6.98	1.03	0.06	0.00	0.00	0.71	0.71
	Real	**6.97**	**0.97**	**0.06**	**0.00**	**0.00**	**0.65**	**0.76**
St-2	Goal	-0.03	6.03	0.06	0.00	0.00	0.72	0.69
	Real	**-0.02**	**5.91**	**0.06**	**0.00**	**0.00**	**0.73**	**0.68**
St-3	Goal	4.00	6.07	0.06	0.00	0.00	-0.71	-0.70
	Real	**4.05**	**5.99**	**0.06**	**0.00**	**0.00**	**0.64**	**0.77**
St-4	Goal	7.33	6.20	0.06	0.00	0.00	0.71	0.70
	Real	**7.34**	**6.06**	**0.06**	**0.00**	**0.00**	**0.71**	**0.70**
St-5	Goal	6.57	11.98	0.06	0.00	0.00	-1.0	0.00
	Real	**6.60**	**12.02**	**0.06**	**0.00**	**0.00**	**0.99**	**0.14**
St-6	Goal	0.12	12.68	0.06	0.00	0.00	0.00	-1.0

Real	**0.84**	**12.14**	**0.06**	**0.00**	**0.00**	**0.96**	**-0.28**

V. CONCLUSIONS AND FUTURE WORK

In this study, a PyQt-based interface was used for autonomous control and monitoring of ROS-based AMR. DWA_local_planner was used as local planner and global_planner was used as global planner in providing autonomous movement of AMR. As a result of the study, the time spent by the AMR between each station and the distance it took were read from the interface and Table 3 was created.

When we examine Table 3;

- While going from Home to St-1, it took 7.34 m and 17.11 sec.
- While going from St-1 to St-2, it took 9.48 m and 21.86 sec.
- While going from St-2 to St-3, it took 5.20 m and 16.52 sec.
- While going from St-3 to St-4, it took 8.73 m and 25.72 sec.
- While going from St-4 to St-5, it took 7.57 m and 21.66 sec.
- While going from St-5 to St-6, it took 9.17 m and 24.25 sec.
- A total of 47.49 m traveled and 127.12 sec. it took.

as we can see the results.

In addition, when each station is reached, the position and orientation of the AMR was read from the interface again and Table 4 was created.

When we examine Table 4 the recorded goal position values for St-1 are;

- Pos-X=6.98
- Pos-Y=1.03
- Pos-Z=0.06

the orientation values are;

- Ori-X=0.00
- Ori-Y=0.00
- Ori-Z=0.71
- Ori-W=0.71

On the other hand, the position values when the AMR reaches St-1;

- Pos-X=6.97
- Pos-Y=0.97
- Pos-Z=0.06

the orientation values are;

- Ori-X=0.00
- Ori-Y=0.00
- Ori-Z=0.65
- Ori-W=0.76

Similarly, the position and orientation values when the AMR reaches all stations and should be can be seen in Table-4. According to Table-4, the accuracy rates of settling at the target stations were calculated and expressed in Table-5. In this way, it can be measured how accurately the AMR approached when it arrived at the desired station. When the AMR arrived at St-1, it was settled with an accuracy of 99.8% at the x position, 94.1% at the y position, and 100% at the z position. In addition, the x orientation is 100%, the y 100%, the z orientation 91.5%, and the w orientation is 93.4% accurate.

Tablo 5. Table Of Placement Accuracy Of AMRs

	Position			Orientation			
	x (%)	y (%)	z (%)	x (%)	y (%)	z (%)	w (%)
St-1	**99.8**	**94.1**	**100**	**100**	**100**	**91.5**	**93.4**
St-2	**66.6**	**98**	**100**	**100**	**100**	**98.6**	**98.5**
St-3	**98.7**	**98.6**	**100**	**100**	**100**	**90.1**	**90.9**
St-4	**99.8**	**97.7**	**100**	**100**	**100**	**100**	**100**
St-5	**99.5**	**99.6**	**100**	**100**	**100**	**-99**	**7.1**
St-6	**14.2**	**95.7**	**100**	**100**	**100**	**1.04**	**28**

In the autonomous movement of the AMR to a station, the time to reach the station, the distance traveled, the desired position and approach to the station are important work areas. When the literature studies are examined, there are many studies in the field of path planning. However, interface studies that can perform AMR manual control, autonomous control, AMR position and position tracking are negligible.

In this study, a unique ROS-based AMR control interface and AMR design were made and contributed to the literature. Thanks to this interface, which has ROS connections in the background, AMR tracking is easily performed. The path planning study using Local and Global path planning algorithms was carried out and the time and distance covered by AMR in reaching the target were calculated by the interface. In addition, the location and orientation of the target station and the location and orientation information of the AMR when it is placed on the target can be followed by the interface. With the work done, ROS developers have been contributed with interface design in the field of AMR monitoring and control.

ACKNOWLEDGMENT: This study was obtained from the PhD project of SUBU, numbered 075-2022, named "Path Planning with Meta-Heuristic Algorithms in Autonomous Mobile Robots and Designing a New Mobile Robot" supported by BAP. It was also supported within the scope of TUBITAK-2244 project numbered 118C142.

V. REFERENCES

[1] E.T. Baek and D.Y. Im, "ROS-Based Unmanned Mobile Robot Platform for Agriculture," *Appl. Sci.*, vol. 12, no. 9, p. 4335, 2022.

[2] P. Karpyshev, V. Ilin, I. Kalinov, A. Petrovsky, and D. Tsetserukou, "Autonomous Mobile Robot for Apple Plant Disease Detection based on CNN and Multi-Spectral Vision System," 2021.

[3] J. Kriegel, C. Rissbacher, L. Reckwitz, and L. Tuttle-Weidinger, "The requirements and applications of autonomous mobile robotics (AMR) in hospitals from the perspective of nursing officers," *Int. J. Healthc. Manag.*, 2021.

[4] M. Law *et al.*, "Case studies on the usability, acceptability and functionality of autonomous mobile delivery robots in real-world healthcare settings," *Intell. Serv. Robot.*, vol. 14, no. 3, 2021.

[5] C. Liu, J. Tan, H. Zhao, Y. Li, and X. Bai, "Path planning and intelligent scheduling of multi-AGV systems in workshop," 2017.

[6] H. Y. Zhang, W. M. Lin, and A. X. Chen, "Path planning for the mobile robot: A review," *Symmetry (Basel).*, 2018.

[7] C. Secchi, R. Olmi, F. Rocchi, and C. Fantuzzi, "A dynamic routing strategy for the traffic control of AGVs in automatic warehouses," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2015.

[8] Y. Dongyong, C. Jinyin, N. Matsumoto, and Y. Yamane, "Multi-robot path planning based on cooperative co-evolution and adaptive CGA," 2006.

[9] J. K. Goyal and K. S. Nagla, "A new approach of path planning for mobile robots," 2014.

[10] B. Park, J. Choi, and W. K. Chung, "An efficient mobile robot path planning using hierarchical roadmap representation in indoor environment," 2012.

[11] O. Aslan and A. Yazici, "Conflict-Free Route Planning for Autonomous Transport Vehicle," 2020.

[12] Z. B. Garip, D. Karayel, S. S. Ozkan, and G. Atali, "Path planning for multiple mobile robots using A* algorithm," *Acta Phys. Pol. A*, vol. 132, no. 3, pp. 685–688, 2017.

[13] G. Atalı, I. Pehlivan, B. Gürevin, and H. I. Seker, "Chaos in metaheuristic based artificial intelligence algorithms: A short review," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 29, no. 3. 2021.

[14] M. Y. Yıldırım and R. Akay, "Mobil Robotların Yol Planlamasında Doğrusallığın İncelenmesi," *Eur. J. Sci. Technol.*, 2021.

[15] M. Y. Yıldırım and A. Rüştü, "A Comparative Study of Optimization Algorithms for Global Path Planning of Mobile Robots," *Sak. Univ. J. Sci.*, vol. 25, no. 2, pp. 417–428, 2021.

[16] R. Wiki, http://wiki.ros.org/dwa_local_planner; 2022 [accessed 28 August 2022]

[17] R. Wiki, http://wiki.ros.org/global_planner; 2022 [accessed 28 August 2022]

[18] C. Z. Looi and D. W. K. Ng, "A Study on the Effect of Parameters for ROS Motion Planer and Navigation System for Indoor Robot," *Int. J. Electr. Comput. Eng. Res.*, vol. 1, no. 1, 2021.

[19] B. Cybulski, A. Wegierska, and G. Granosik, "Accuracy comparison of navigation local planners on ROS-based mobile robot," 2019.

[20] S. Karakaya and H. Ocak, "Açısal Duruş Kontrolü Destekli Özgün bir Dinamik Pencere Yaklaşımı," *Bilecik Şeyh Edebali Üniversitesi Fen Bilim. Derg.*, vol. 7, no. 1, 2020.

[21] P. Marin-Plaza, A. Hussein, D. Martin, and A. De La Escalera, "Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles," *J. Adv. Transp.*, vol. 2018.

[22] M. M. Moslemi and M. Sadedel, "Behavior Control and Navigation of Two-Wheeled Robot Using ROS-Gazebo," in *2022 8th International Conference on Control, Instrumentation and Automation (ICCIA)*, 2022, pp. 1–6.

[23] 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, 2021.