Research Article

A Multi-Functional Web Control Interface for Industrial Autonomous Mobile Robot Fleets

Sedanur Kirci[1*], Ulas Birgul[2], Gokhan Atali[3]

[1] Karmetal, Orcid ID: https://orcid.org/0000-0001-5089-9243, E-mail: yapayzeka@karmetal.com.tr
[2] Karmetal, Orcid ID: https://orcid.org/0000-0003-0378-312X, E-mail: ulasbirgul@karmetal.com.tr
[3] Sakarya University of Applied Sciences, Orcid ID: https://orcid.org/0000-0003-1215-9249, E-mail: gatali@subu.edu.tr
[*] Correspondence: yapayzeka@karmetal.com.tr

Received 21 September 2024
Received in revised form 20 December 2024
In final form 28 December 2024

Reference: Kirci, S., Birgul, U., & Atali, G. (2024). A multi-functional web control interface for industrial autonomous mobile robot fleets. The European Journal of Research and Development, 4(4), 101-109.

Abstract

This study presents a multifunctional web-based interface developed to facilitate the control and management of autonomous mobile robot fleets in industrial automation. The system enables users to monitor the status of robots in real time, perform mapping, define restricted areas, plan virtual paths, and issue task commands. The developed interface communicates seamlessly with the ROS operating system and is built on a robust backend utilizing a MySQL database, offering a strong technical foundation. With its multi-user management feature, the system allows users to operate efficiently through role-based authorizations.

The interface provides a user-friendly experience, ensuring low error rates, rapid decision-making, and high efficiency. Tests have demonstrated that the system delivers consistent performance in multi-robot scenarios and exhibits resilience against network connectivity issues. This study offers an innovative solution for the management of industrial mobile robots, contributing to digital transformation processes.

Keywords: *Autonomous mobile robot (AMR), Web-Based interface, ROS, Industrial automation, Task management*

1. Introduction

Industrial automation is a field where technology is extensively utilized to optimize manufacturing processes, reduce labor costs, and minimize human-induced errors.

Today, the integration of robotic systems and autonomous devices into production processes is rapidly increasing the level of automation. Autonomous Mobile Robots (AMRs) play a critical role in enhancing the efficiency of logistics processes and are among the foremost systems driving this level of automation. AMR-type mobile robots are designed for dynamic environments and are commonly employed in tasks such as material transport, product picking, and production line optimization. Unlike traditional systems, they operate independently of predefined paths or external aids (e.g., QR codes, lines, or reflectors) thanks to their advanced software infrastructure. Utilizing natural navigation capabilities, these robots first map their surroundings and then calculate their position in real time to follow the most efficient route between stations. Integrated with control interfaces, they can be remotely monitored and managed, significantly reducing human involvement in repetitive tasks, lowering error rates, and accelerating operational processes.

A review of the literature reveals that web-based interfaces are highly advantageous in similar studies. Perier et al. introduced a web-based user interface solution developed for CERN's mechatronics and robotics operations. Their system facilitates real-time monitoring and control of autonomous robotic systems while providing access from multiple devices (Perier et al., 2022). In another study, Panuma focused on creating a web-based user interface for mobile robots, specifically exploring the use of the Three.js library. The study aimed to enable users to plan robot motion paths and manage robots in real time (Panuma, 2023). Gürevin et al. designed and presented a new graphical user interface (GUI) to compare motion planning algorithms for ROS-based autonomous mobile robots (AMRs) (Gürevin et al., 2023).

In the present study, a web-based interface has been developed and introduced.

2. Materials and Methods

In this study, an interface has been developed for the control and management of autonomous mobile robot fleets. The designed interface is compatible with all models of Servant robots developed within the Karmetal company. Figure 1 shows an image of the Servant Z250 model.

Figure 1. Servant Z250

The developed web-based control interface eliminates the need for users to be physically present in the factory. Users can access robots, monitor tasks, and manage operations from any device with an internet connection, providing a significant advantage, particularly in large-scale operations. The interface offers a user-friendly experience through its visually appealing and interactive design. It does not require technical expertise, enabling operators in the factory to make quick and accurate decisions. Real-time monitoring of robot statuses is critical for user companies. Through the interface, any faults or the status of ongoing tasks can be tracked instantly, ensuring seamless operation management.

The developed system also supports multi-user and role management. Different user roles (factory admin, task manager, operator) can be defined, and specific permissions can be assigned to each role. This ensures that each user can perform only the tasks within their designated authority, enhancing security and operational efficiency.

2.1. Technical Infrastructure

The developed web-based control interface is built on robust software technologies. The system's backend is based on Node.js, enabling real-time processing of data received from robots and its seamless transmission to the user interface. The asynchronous architecture of Node.js ensures high efficiency, allowing the system to perform multiple functions simultaneously with optimal performance.

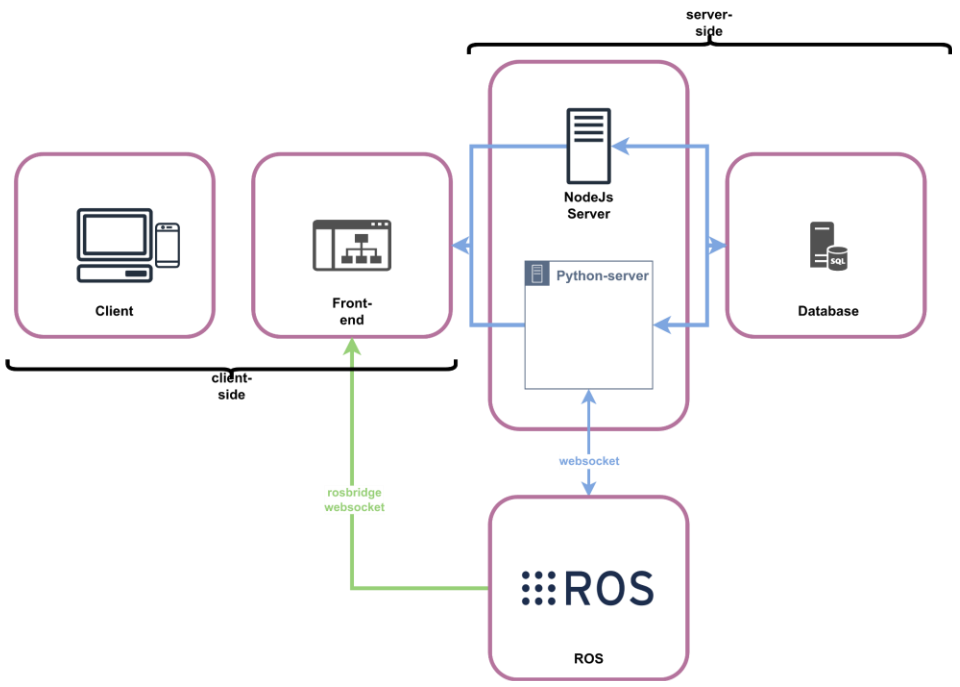

Figure 2. Application architecture

Figure 2 illustrates that the application is composed of server and client sides. The system includes a program named python-server, which listens to and interprets ROS topics. Servant mobile robots utilize the ROS framework for their internal communication. Integration with the ROS infrastructure ensures the collection of data required for functions such as mapping, task management, and status monitoring, and facilitates the seamless transfer of this data to the control interface. When the robots operate as a fleet, they transmit their data to a master computer. This master computer publishes real-time data (e.g., temperature, battery levels, status) via ROS topics. Relatively persistent data (e.g., job orders, station information, virtual path data) is stored in a MySQL database. On the backend side of the interface, real-time data is retrieved by listening to ROS topics and displayed without requiring page refreshes. This system enables bidirectional communication, where data entered through the interface is sent to the database and the robots. This architecture ensures efficient and reliable data flow between components, supporting dynamic operations.

The Three.js library was utilized to display the map in 3D on the web interface. Chosen for its extensive documentation and ease of development, Three.js provides an ideal solution for integrating 3D graphics. The front-end of the system has been developed using a range of open-source libraries managed through npm (Node Package Manager). These libraries ensure that the interface delivers a user-friendly experience, loads quickly, and operates seamlessly across various devices.

2.2. System Functions

The screen for monitoring the status of the robots is presented in Figure 3.

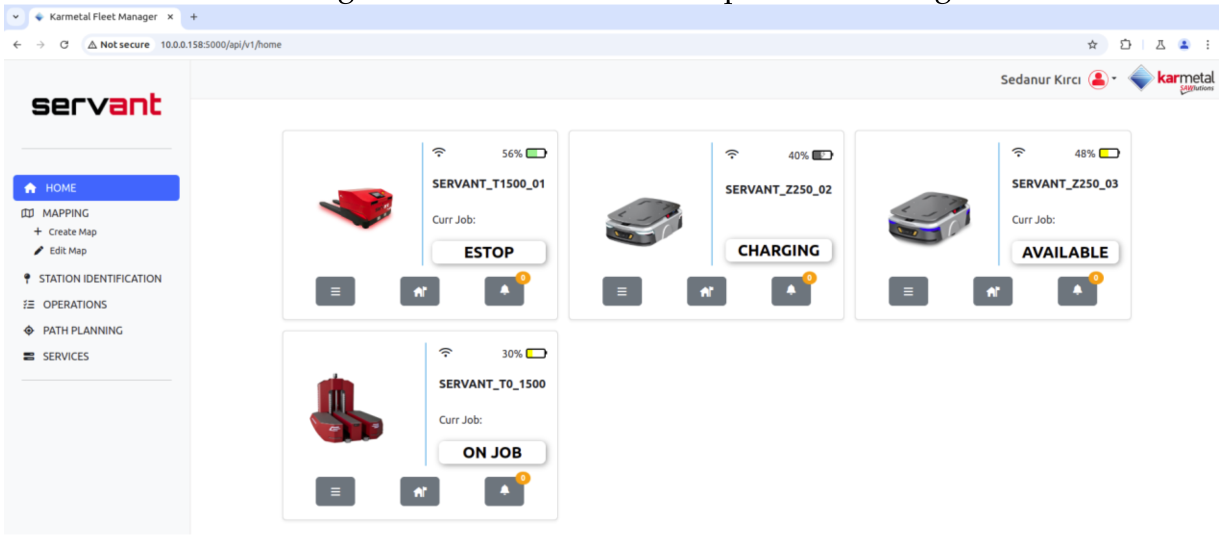

Figure 3. The homepage window

After the user logs in, the homepage of the interface displays all the robots available on the network. The robots' charge level, connection status, and operational status are updated in real time. Additionally, past operation orders and error codes of the robots can be tracked from this page. This information is presented by parsing and interpreting the data shared by the robots through their rostopics.

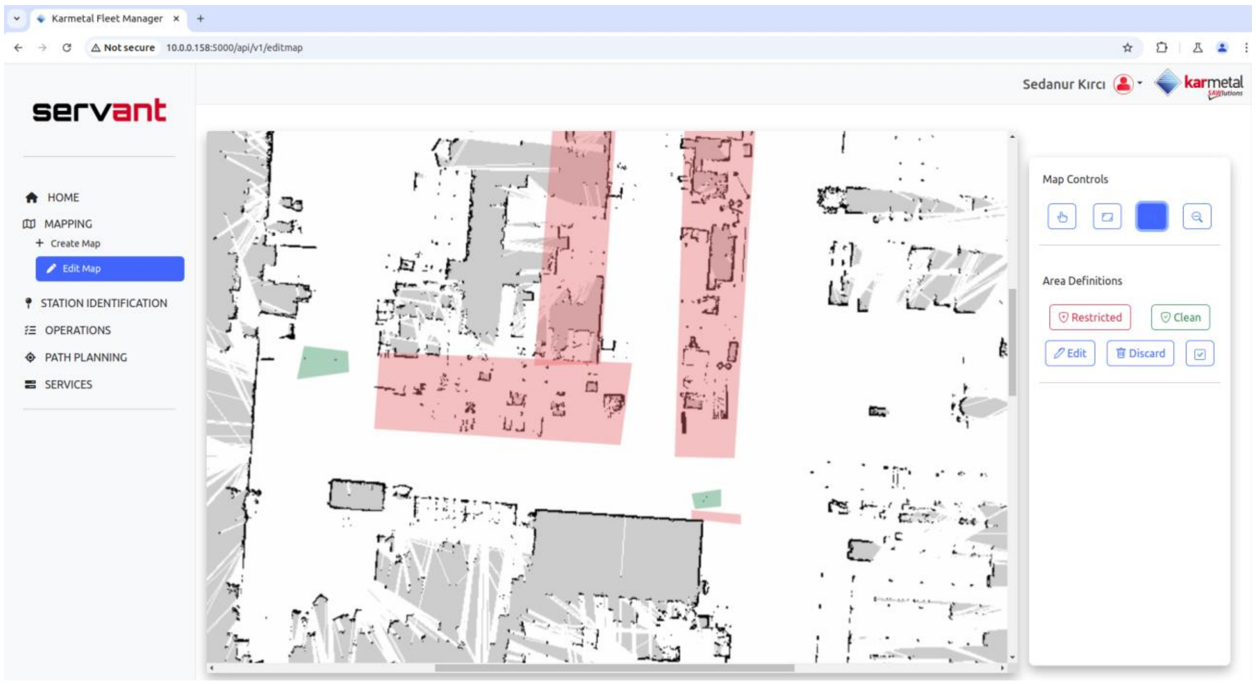

Figure 4. Interface window for defining restricted areas and clearing areas on the map

After one of the robots creates a map, it is saved and distributed to all robots by the master computer. To define restricted areas on the map, the map editing window shown in Figure 4 is used. For this process, the resolution of the actual map and its resolution transformations for the interface must be performed.

The real map used by the robots must be synchronized with the map displayed in the front-end interface. The station locations defined on the actual map and the points of the drawn paths are stored in the database. The station points, which are known based on the map axes on the real map, must be accurately calculated and positioned according to the different resolution of the map displayed in the interface. Similarly, for real-time tracking of the robots on the map, the robot's position (relative to the map) must also be transformed to align with the map shown in the front-end interface. The transformation matrices used in this process are as follows:

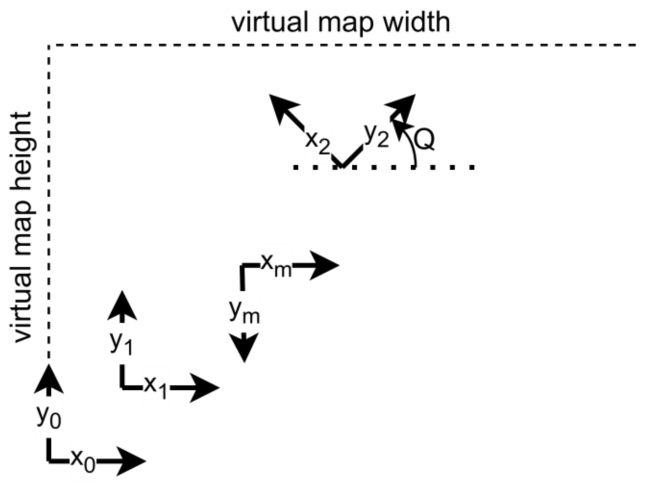

Figure 5. Transformation Axes

In Figure 5, the axes for both the real map and the virtual map are shown. The axes labeled as the virtual map represent the map displayed in the interface. The axis (x_0, y_0) serves as the reference axis of the virtual map in pixel units. The axis (x_1, y_1) represents the center point of the real map relative to the virtual map. The axis (x_2, y_2) denotes the positions of the stations on the real map relative to (x_1, y_1). Finally, the axis (x_m, y_m) is the reference axis of the virtual map. Accordingly, the positions of the stations located on the real map must be transformed to align with the virtual map axes and rendered appropriately.

$$ {}^{m}_{0}T = rot(x,-90) + trans\left(-\frac{virtualmap_{width}}{2{,}0}, \frac{virtualmap_{height}}{2}\right) \tag{1} $$

$$ {}^{1}_{0}T = trans\left(\frac{map_x}{resolution}, \frac{map_y}{resolution}, 0\right) \tag{2} $$

$$ {}^{2}_{1}T = rot(z,\theta) + trans\left(\frac{station_x}{resolution}, \frac{station_y}{resolution}, 0\right) \tag{3} $$

Equations 1, 2, and 3 provide the rotation and translation steps between the axes. Based on these operations, the transformation matrices presented in Equations 4, 5, and 6 have been formulated. Using Equation 7, the transformation of a station point (x_2, y_2) on the real map axes to the virtual map axes (x_m, y_m) has been calculated.

$$ {}^{m}_{0}T = \begin{bmatrix} 1 & 0 & 0 & -virtualmap_{width}/2 \\ 0 & \cos(-90) & -\sin(-90) & 0 \\ 0 & \sin(-90) & \cos(-90) & virtualmap_{height}/2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

$$ {}^{0}_{1}T = \begin{bmatrix} 1 & 0 & 0 & origin\_map_x/\text{resolution} \\ 0 & 1 & 0 & origin\_map_y/\text{resolution} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5} $$

$$ {}^{1}_{2}T = \begin{bmatrix} \cos(Q) & -\sin(Q) & 0 & station_x/\text{resolution} \\ \sin(Q) & \cos(Q) & 0 & station_y/\text{resolution} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6} $$

$$ {}^{m}_{0}T \cdot {}^{0}_{1}T \cdot {}^{1}_{2}T = {}^{m}_{2}T \tag{7} $$

The inverse transformations of these operations are used in the virtual path planning screen. In the virtual path planning screen, the user can input and select path points on the virtual map. The selected path points are transformed according to the real map and stored in the database. These saved points are then sent to the relevant robots through the fleet management software.

To enhance the user experience, customized access levels have been defined among three distinct roles: Factory Admin, Task Manager, and Operator. These roles streamline the interface by eliminating unnecessary complexity. For instance, operators can only access basic functions, while task manager users are granted capabilities for operation planning and task assignment. The Factory Admin, on the other hand, has full access to the system, enabling user management as well as overall control of the robots. This distinction not only improves security levels but also ensures that users interact only with the tools relevant to their specific tasks.

Tests have demonstrated that the system maintains consistent performance even in multi-robot scenarios and exhibits resilience to network connectivity issues. These features highlight the system as an ideal solution for industrial environments.

3. Results

In this study, a control and management interface for a mobile robot fleet was developed, and its technical details were explained. In line with the study scenario, a homepage was created where robots connected to the master computer could be monitored, their status information observed, and error codes and past job orders reviewed. Both the front-end and back-end software for this interface were developed. Mapping and map editing screens were designed, enabling functionalities such as defining restricted areas or clearing areas on the map. The functions of screens for station definition and virtual path planning by the user were also detailed.

This interface additionally provides users with statistical data about the robot fleet. For example, metrics such as the distance covered by each robot and the time taken to complete assigned tasks can be accessed through the service window, offering valuable insights to users.

In future work, innovative technologies such as augmented reality (AR) integration can be added to the system. This would enable users to visualize and interact with the robots' tasks directly in physical environments. Additionally, with AI-powered analytical systems, it would be possible to enhance the efficiency of the robots and proactively detect anomalies.

4. Acknowledge

The authors cordially thank Karmetal for supporting this study.

References

[1] Gurevin, B., Gulturk, F., Yildiz, M., Pehlivan, I., Nguyen, T. T., Caliskan, F., Boru, B., & Yildiz, M. Z. (2023). A Novel GUI Design for Comparison of ROS-Based Mobile Robot Local Planners. *IEEE Access*, 11, 125738–125750.

[2] Panuma, A. (2023). Building a Web-Based User Interface for Mobile Robots. *Bachelor Thesis*, Oulu University of Applied Sciences.

[3] Perier, H., Matheson, E., & Di Castro, M. (2022). Web-Based User Interface Solution for Remote Robotic Control and Monitoring Autonomous System. In *Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2022)* (pp. 680–687). SCITEPRESS.