International Journal of Electrical and Computer Engineering (IJECE)
Vol. 16, No. 1, February 2026, pp. 158~173
ISSN: 2088-8708, DOI: 10.11591/ijece.v16i1.pp158-173

Autonomous mobile robot implementation for final assembly material delivery system

Ahmad Riyad Firdaus[1], Imam Sholihuddin[2], Fania Putri Hutasoit[1], Agus Naba[2], Ika Karlina Laila Nur Suciningtyas[1]
[1]Department of Electrical Engineering, Politeknik Negeri Batam, Batam, Indonesia
[2]Department of Physics, Brawijaya University, Malang, Indonesia

Article Info

Article history:

Received Nov 7, 2024
Revised Sep 19, 2025
Accepted Nov 23, 2025

Keywords:

Autonomous mobile robot
Final assembly
Industry 4.0
Material delivery system
Simultaneous localization and mapping

ABSTRACT

This study presents the development and implementation of an autonomous mobile robot (AMR) system for material delivery in a final assembly environment. The AMR replaces conventional transport methods by autonomously moving trolleys between the warehouse, production stations, and recycling areas, thereby reducing human intervention in repetitive logistics tasks. The proposed system integrates a laser-SLAM navigation approach, customized trolley design, RoboShop programming, and robot dispatch system coordination, enabling real-time route planning, obstacle detection, and material scheduling. Experimental validation demonstrated high accuracy in path following, with root mean square error values ranging between 0.001 to 0.020 meters. The AMR achieved an average travel distance of 118.81 meters and a cycle time of 566.90 seconds across three final assembly stations. The overall efficiency reached 57%, primarily due to reduced idle time and optimized material replenishment. These results confirm the feasibility of AMR deployment as a scalable and flexible intralogistics solution, supporting the transition toward Industry 4.0 smart manufacturing systems.

Corresponding Author:

Ahmad Riyad Firdaus
Department of Electrical Engineering, Batam State Polythecnic
Jl. Ahmad Yani, Teluk Tering, Batam, Batam 29461, Kepulauan Riau, Indonesia
Email: rifi@polibatam.ac.id

1. INTRODUCTION

The implementation of autonomous mobile robots (AMRs) in final assembly material delivery systems represents a pivotal advancement toward achieving flexible, efficient, and resilient intralogistics within the Industry 4.0 paradigm. Unlike traditional conveyor-based transport or automated guided vehicles (AGVs), AMRs employ advanced perception, mapping, and decision-making techniques, particularly simultaneous localization and mapping (SLAM) and dynamic path planning, enabling them to autonomously adapt to frequently changing assembly layouts and unstructured environments. This adaptability is especially valuable for high-mix, low-volume (HMLV) manufacturing systems, where responsiveness and scalability are paramount [1]–[3].

The kitting and delivery process, integral to final assembly, has traditionally relied on manual or semi-automated handling. These conventional approaches are often labor-intensive, error-prone, and vulnerable to delays when faced with rapidly evolving production requirements. AMRs address these shortcomings by autonomously transporting materials, dynamically avoiding obstacles, and ensuring precise, timely deliveries. Such capabilities not only increase accuracy and throughput but also reduce repetitive and physically demanding tasks for human workers, thereby improving ergonomics, safety, and operational cost efficiency [4]–[6].

Recent empirical studies underscore these benefits. For instance, Hercík *et al.* [7] demonstrated that AMRs in SmartFactory contexts can achieve repeatable positioning accuracy of ±3 mm, while AI-enhanced navigation has been shown to significantly improve perception and path planning in dynamic assembly settings [8], [9]. Moreover, case studies in the automotive industry report that AMR integration reduces shift times by 1.14 hours and lowers capital expenditure by up to 30% compared with conventional material-handling solutions [10]. These findings illustrate the economic and operational viability of AMR deployment in final assembly intralogistics.

Despite such progress, critical research gaps remain in areas such as high-accuracy localization under dynamic conditions, seamless integration into digital manufacturing systems, and real-world validation of SLAM performance in complex environments. Existing systems are often limited by sensor drift, calibration demands, and environment clutter, restricting their reliability at industrial scale [11]–[16]. Addressing these challenges is essential for realizing the full potential of AMRs in assembly line delivery.

To this end, the present study contributes a systematic approach to AMR implementation in final assembly material delivery, comprising: i) Designing the delivery system architecture, with emphasis on obstacle avoidance and path adaptability; ii) Evaluating mapping and localization accuracy using SLAM under dynamic factory-like conditions; and iii) Measuring delivery performance metrics—accuracy, responsiveness, and scalability—through experimental trials.

The novel contribution of this work lies in bridging the gap between algorithmic performance evaluation and practical assembly integration, providing quantifiable evidence of AMR effectiveness in material delivery. Unlike prior studies that focus narrowly on either navigation algorithms or logistics feasibility, this research offers an end-to-end validation framework, combining experimental SLAM assessment with intralogistics performance benchmarking [13], [17], [18].

The implications of this study are far-reaching. Beyond improving final assembly operations, the outcomes provide a roadmap for industries to adopt AMRs as scalable, modular logistics solutions without costly infrastructure modifications. Furthermore, the demonstrated improvements in localization accuracy and adaptive navigation lay the groundwork for integrating AMRs with digital twins, predictive maintenance systems, and AI-driven scheduling platforms. Ultimately, this research advances the development of resilient and intelligent intralogistics systems, positioning AMRs as a cornerstone technology in next-generation manufacturing, particularly in high-value industries such as automotive, aerospace, and electronics assembly [5], [6], [10], [16].

This paper is structured as follows: Section 2 outlines the methodology, and the steps involved in implementing the AMR. Section 3 evaluates the AMR's accuracy in following virtual paths based on experimental results and data gathered from the methodology. The study’s conclusions and potential implications for AMR technology in assembly processes are discussed in the final section.

2. METHOD

The implementation of an AMR in the final assembly material delivery system was conducted through a structured series of steps. These included designing the AMR system architecture, defining the material delivery process, developing a customized trolley, configuring and programming the AMR, validating navigation accuracy, and integrating the robot dispatch system (RDS) for coordinated operation. Each stage is described in detail below.

2.1. AMR system configuration

The AMR system consisted of five main components: the AMR unit, RoboShop application, RDS, warehouse control system (WCS) server, and WCS client. Figure 1 illustrates the overall system configuration, while Table 1 summarizes the specifications, functions, and communication protocols for each component. Figure 1 and Table 1 outline the architecture, communication flow, and functional roles within an AMR system used for material handling in a warehouse or final assembly environment. The figure demonstrates the interactions between five main components: WCS web client, WCS web server, RDS, RoboShop, and the AMR itself. Communication flows through various protocols, including HTTP and REST API over Ethernet, and TCP/IP over Wi-Fi or Ethernet, ensuring seamless data transfer and coordination across components. The table provides a detailed breakdown of each component's specifications, functions, and connection methods:

a. AMR: The robotic unit responsible for moving materials, with a load capacity of 500 kg and laser SLAM navigation. It receives instructions from the RDS and can communicate with RoboShop for setup over TCP/IP.
b. RoboShop: A desktop application provided by the AMR supplier for configuring AMR parameters, including network setup, route planning, and integration with the RDS via TCP/IP.

c. RDS: A dispatching web application that orchestrates AMR movements, monitors routes and task statuses, and communicates bi-directionally with both the AMR and the WCS web server using Wi-Fi and Ethernet with REST API.
d. WCS web server: A central application with a GUI for material request handling, sending commands to the RDS for AMR task management, and receiving real-time updates. It communicates with the RDS using Ethernet via REST API.
e. WCS web client: An interface through which users can control AMR operations, connected to the WCS web server over Wi-Fi using HTTP.

Together, the figure and table illustrate a cohesive system that integrates software and hardware for automated, real-time material transport within industrial settings, facilitating efficient coordination of logistics and assembly tasks. This structured approach enhances the flexibility and responsiveness of the material delivery process, aligning with modern automated warehousing and Industry 4.0 standards.

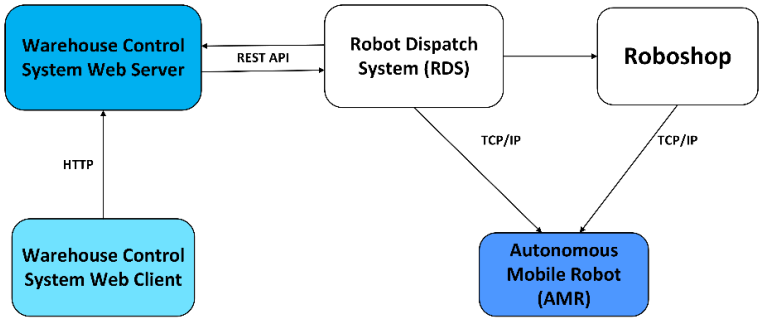


Figure 1. AMR system

Table 1. Specifications and functions of AMR systems

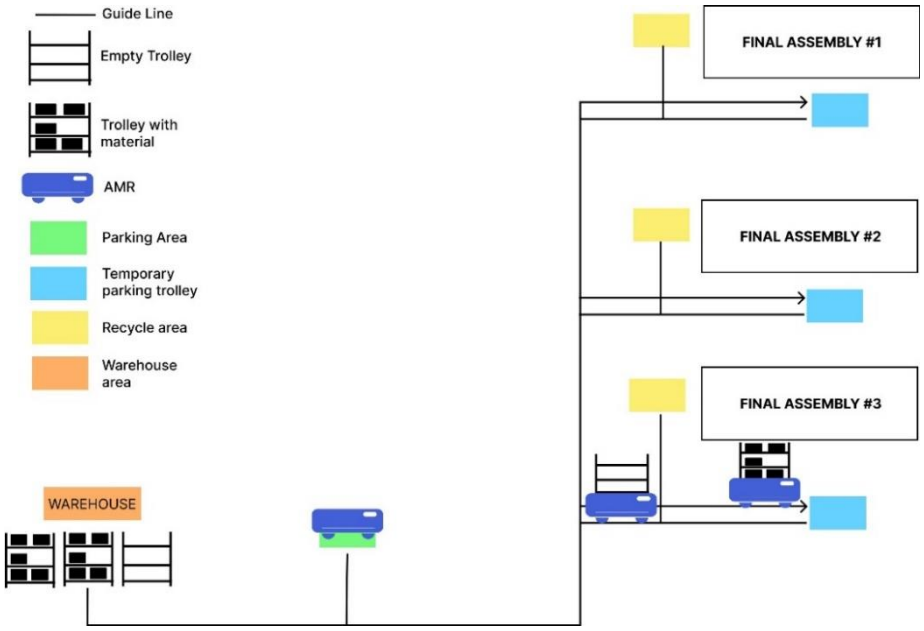
No.	Component	Description	Function in the project	Connection methods
1.	Autonomous mobile robot (AMR)	Model: Seer SJV-SW500 Dimension: 924*762*300 (mm) Navigation: laser SLAM Load capacity: 500 kg Controller: SRC-2000-I(S)	– Deliver material from one point to another point – Receive task from RDS	– With RoboShop either using Ethernet or Wi-Fi with protocol TCP/IP – With RDS using Wi-Fi with protocol TCP/IP
2.	RoboShop	Desktop application provided by AMR supplier	Set-up AMR: – Ethernet – Wi-fi – Localization – Create Route – Set-up connection AMR with RDS	– With AMR using either Ethernet or Wi-Fi with protocol TCP/IP
3.	Robot dispatch system (RDS)	Web server application provided by AMR supplier	– Coordinates movement of AMR – Send tasks to AMR. – Monitoring of AMR routes, stations, and zones – Real-time display of AMR work status and operation – Receives tasks from WCS – Send status updates back to WCS	– With AMR using Wi-Fi with protocol TCP/IP – With WCS using ethernet with protocol Rest API
4.	Warehouse control system web server	Web server application	– Provides a graphical user interface (GUI) for request material to warehouse – Provide GUI to receive – Send commands to the RDS	– With RDS using ethernet with protocol Rest API
5.	Warehouse control system web client	Web server application	– Send commands to the server to control the movement of AMR	– With WCS using Wi-Fi with protocol HTTP

2.2. Project scope and delivery process

In this project, a single AMR is deployed to transport materials to three designated areas for final assembly. Figure 2(a) presents the delivery route encompassing three final assembly locations: one trolley positioned within the final assembly area, a second trolley serving as a recycling trolley in the material staging area, and a third trolley located at the warehouse. The AMR initiates movement upon receiving

material request data from the WCS web client. A Kanban card, depicted in Figure 2(b), provides essential information regarding the material requirements for final assembly, including the material's name, quantity, and intended delivery location. The steps for material delivery are outlined as follows.

a. The initial step is to prepare a card that specifies the delivery location and the type of material to be transported. This card is then placed in the recycling trolley, allowing the feeder to identify it and obtain details about the delivery route and final destination. Table 2 presents data on replenishment time over one hour, production output per hour, and the material capacity of the trolley. This material requirement card is designed with a one-hour replenishment cycle, ensuring that the quantity of material sent aligns with the planned production output.
b. When materials are depleted in the final assembly area, the operator must activate the trigger on the WCS web client to alert the AMR that material replenishment is needed. The operator then removes the empty trolley and loads it with the required components for the next delivery.
c. Upon receiving the signal, the AMR moves to the designated material pickup area as specified by the RDS task. Once it arrives, the AMR unloads the materials at a temporary staging area, then proceeds to the recycling area to collect an empty trolley and the Kanban card. The operator transports the materials to the final assembly stations.
d. The warehouse receives empty carts and Kanban cards from the AMR. This process is repeated whenever materials need to be delivered to various locations. With the AMR using Kanban cards, material deliveries can be carried out automatically and efficiently without constant human supervision.



(a)

Location: Final Assembly PO: 001

No	Part	Qty
1	Part 1	180
2	Part 2	100
3	Part 3	200
4	Part 4	180
5	Part 5	150

(b)

Figure 2. Material delivery system: (a) the route of travel that AMR takes to deliver materials and (b) Kanban cards

Table 2. Replenishment table

Replenishment delivery material = 1 hour	
Output/hour/Final Assembly	550 Pcs
Material on trolley	550 Pcs
Total Material	1100 Pcs

2.3. Trolley design

The AMR requires a trolley capable of transporting multiple items in a single trip to efficiently move kitting materials from the warehouse to the final assembly area. Materials prepared in the warehouse will be organized in bins, which are arranged in rows on the trolley during delivery. In this project, the trolley base features a specially designed space allowing the AMR to maneuver beneath it and lift the trolley. Consequently, the trolley frame must be constructed from durable iron to support the transported load and must be sized to match the AMR's capacity, preventing any tilt that could dislodge the bins. The frame consists of three flat pallets, each accommodating up to 10 to 12 bins. The trolley wheels are built for smooth movement over a stable, flat surface, with locking mechanisms on both the front and rear wheels to secure them once positioned. Figure 3 displays the SolidWorks 3D design specifications, including trolley dimensions. This trolley, operated with AMR assistance, can be transported to designated locations. Once in position, the AMR's control system lifts the materials; after unloading, the trolley is lowered, and the AMR returns it to the warehouse.

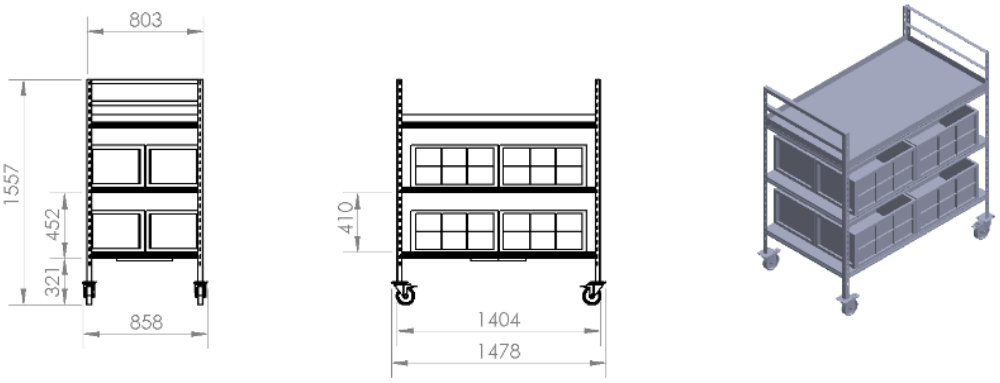


Figure 3. Trolley design

2.4. AMR set-up and programming

This section outlines the steps required to implement an AMR for material distribution, detailing the integration between the AMR and the software used for its configuration and control. It explains how to connect the AMR to RoboShop and covers the processes for localizing and mapping the AMR, as well as setting its waypoints and travel routes. These routes define the path the AMR will follow to deliver materials to designated locations. The following provides a comprehensive guide to the AMR setup procedures.

2.4.1. System connection

To connect the AMR to RoboShop, several steps are required to establish communication between the AMR and the RoboShop platform. First, connect the AMR to a computer using a cable to adjust its local IP address, then add this IP address to RoboShop. Next, input the AMR's wireless IP address assigned by RoboShop to enable Wi-Fi connectivity. Once the AMR recognizes the RoboShop platform, simulation tests and data collection can be performed to validate performance and ensure system functionality.

2.4.2. Mapping and localization

Determining the exact location of AMR movements is one of the most challenging aspects of map building. In selecting an appropriate map representation, three primary factors should be considered: i) the map's accuracy in representing the AMR's target destination, ii) the alignment of the AMR sensor's data type with the map's requirements, and iii) the map's complexity, which can impact the AMR's navigation and mapping performance [19]. In autonomous mobile robots, mapping and localization are foundational concepts. Mapping involves creating a representation of the AMR's physical environment, while localization

determines the AMR's position relative to this map. The AMR relies on sensors such as light detection and ranging (LiDAR), cameras, and proximity sensors to gather environmental data, which is then processed to produce detailed maps [20]. These maps guide the AMR's autonomous navigation [9], [21].

Localization relies on technologies such as odometry, inertial sensors, and GPS to establish the AMR's precise location on the environmental map. Integrating these sensors enables high-precision positioning essential for AMR decision-making and task execution accuracy [22], [23]. Mapping and localization systems work in conjunction, supporting AMR operations by minimizing errors and enhancing productivity [24], [25]. In RoboShop-based mapping and localization, the initial step involves preparing the environment by clearing clutter and configuring a new project with the AMR's IP address and parameters. Using the SLAM tool, the AMR scans its route, identifies obstacles, and generates a map. Data from onboard sensors, including LiDAR and inertial measurement unit (IMUs), are continuously analyzed and refined as the AMR navigates, resulting in a highly accurate and dynamic map, as shown in Figure 4 [26], [27].

Once the map is complete, the AMR is positioned within the mapped environment, allowing the localization process to begin. Localization ensures the AMR can determine its position in the pre-mapped environment, performing tasks with accuracy and efficiency. Initially, the AMR's starting position is defined using sensor data, and as it moves, its sensors continuously update the AMR's position on the real-time map. This iterative process of position adjustment and map refinement enables the AMR to maintain high precision in dynamic, real-world environments. The localization process will continue in real time, updating the AMR's position as it navigates the environment [21], [28], [29].

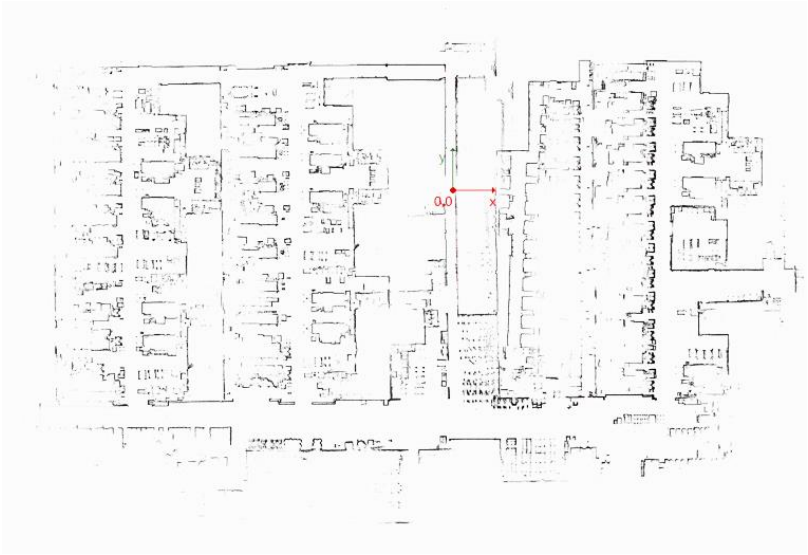


Figure 4. Location map generated by SLAM

2.4.3. Route and set-point creation

Once the AMR completes the map creation, the next step is to designate the path it will follow to reach the final assembly area. Using the RoboShop application, users can plot a route and add checkpoints at various intervals along the way. The process for establishing an AMR travel route is as follows.

Step 1: Select the "Map Editor" option. Using the "Mark Shortcut" tool, mark specific points on the map, such as parking points, charging points, and action points, where the AMR will perform predefined actions. Each point type corresponds to specific actions required along the route.

Step 2: After defining the points, use the "Add Path" tool to connect them, adjusting the path's shape and length based on the AMR's route requirements. Then, in "Map Editor," open the "Route Editor" to arrange the points and paths in sequence. The "Route Editor" establishes the order in which the AMR will follow the path and interact with each point during tasks. After configuring the routes and points, save and apply them to the AMR. Once complete, the system will generate a route, as shown in Figure 5, illustrating the configured travel path.

The delivery route for the AMR to reach final assembly 3 is illustrated in Figure 6, showing various predefined set-points and AMR positions where actions will be performed, along with the route the AMR will follow for material delivery. Beyond creating set-points and routes, the editor menu includes additional options that enhance the efficiency of routes and checkpoints. These options include an eraser to remove

moving obstacles, an advanced line tool to restrict AMR access to specific areas by drawing virtual lines, and an advanced area tool for configuring customized operations in defined zones. Additionally, QR codes can be added for navigation details or to integrate external devices. When designing AMR routes and waypoints in the RoboShop application, it is crucial to account for factors such as distance, speed, and AMR position to perform actions like stopping, turning, and raising or lowering the trolley effectively.

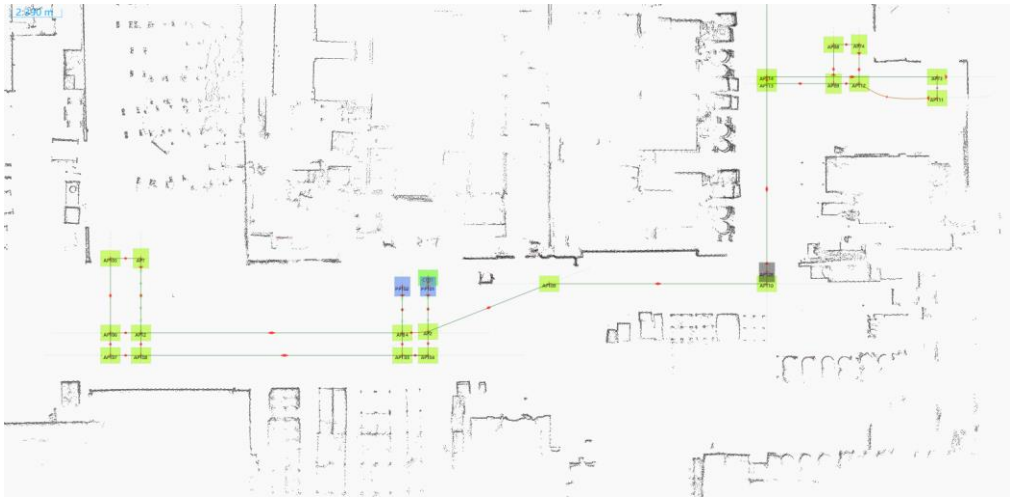


Figure 5. Route of AMR in final assembly 3

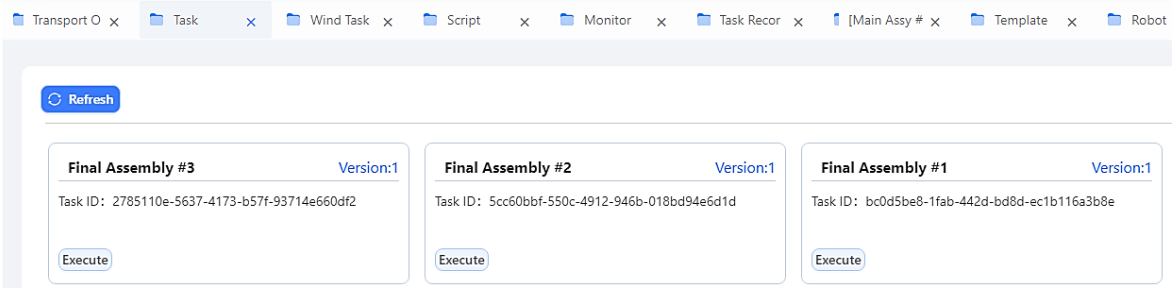


Figure 6. Wind task

2.4.4. Task chain programming

The next step is to create a task chain using the task chain feature in the RoboShop application, following the establishment of the AMR's route and waypoints. Tasks in RoboShop represent the types of jobs that can be assigned to an AMR. To begin, open the "Task Chain" menu in RoboShop and select "Create New Task." Assign a name to the task, specify the work zone the AMR will operate in, and define the actions required, such as pick-up, drop-off, and navigation to designated locations. Additional commands can be added, like picking up or placing objects at specific points. Then, select the type of task, whether for material pick-up or delivery, and set the start and end points as well as the path the AMR will follow. These steps can be customized to meet the task's specific requirements. Once complete, save and execute the task, and the AMR will follow the predefined steps. With the task chain feature, users can easily organize and manage AMR tasks for various scenarios. As the AMR executes the task, users can monitor its progress in real time and make any necessary adjustments through the RoboShop application.

The AMR is equipped with an obstacle-detection system, including an ultrasonic sensor that identifies objects in its path. This sensor emits sound waves and calculates the distance to obstacles based on the time it takes for the sound to reflect back. With these ultrasonic sensors, the AMR can detect obstacles nearby or approaching. If an obstacle is detected while the AMR is in motion, it will slow down and stop, ensuring it reaches its destination safely. Should the obstacle remain for an extended period, the AMR will notify the user that the route is blocked and request further instructions. At this point, the operator can assess the situation to decide whether to remove the obstruction or reroute the AMR to a safer path.

2.4.5. Execution and monitoring

After designing a task chain, the next step in AMR deployment involves programming and defining specific tasks, known as missions, which are essential for the AMR to fulfill operational objectives like material transport, stock picking, or order fulfillment. RoboShop software facilitates this by allowing users to define the task area, material locations, and travel distances, generating optimal movement plans for mission success. It continuously monitors task progress, enabling real-time performance tracking and dynamic adjustments to overcome obstacles, which enhances operational efficiency and task completion accuracy.

2.4.6. Accuracy testing

To validate the accuracy of an AMR, systematic tests were conducted focusing on navigation and object recognition within a controlled environment. Typically, ten tests are performed in identical settings to obtain reliable data, minimizing random variations and reflecting AMR's consistent performance. For precise path-following accuracy, the AMR's line-tracking ability can be quantified by calculating the slope mm between two points, expressed as:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \tag{1}$$

where x_1, y_1 and x_2, y_2 are coordinates of two points along the desired path.

Further precision evaluation utilizes the root mean square error (RMSE) to measure the average deviation between actual and predicted AMR positions.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

Here, y represents the actual position, $\hat{y}$ the predicted position, and n the sample count. A lower RMSE indicates a more accurate path-following performance [30]–[32].

Additionally, the Euclidean distance between points is calculated as (1).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3}$$

This provides the distance between AMR's consecutive coordinates, supporting assessments of adherence to virtual paths. Collectively, these tests ensure the AMR's accurate movement along predefined paths [30], [33], [34].

2.5. Robot dispatch system integration

This section outlines the configuration of the RDS, which provides a centralized platform for managing, supervising, and controlling multiple robots. It details the process required to integrate AMRs with the RDS, including essential connectivity and setup steps to ensure seamless communication and coordinated functionality. The steps needed for establishing and maintaining this connection are presented systematically to guide users through each stage of configuration and deployment.

2.5.1. Connection setup

To connect an AMR to the RDS, a server IP address must first be configured within RoboShop to enable RDS-AMR communication. RoboShop and RDS must operate on the same network. After entering the server IP, select the upgrade/backup option, then drag the RDS installation package to its designated area. The RDS will automatically restart, confirming successful integration. Once the login page appears, RDS can be accessed via a web browser, indicating a completed setup.

2.5.2. Task management (Wind Task)

In the RDS web application, the "Wind task" feature, illustrated in Figure 6, assigns specific missions to AMRs based on pre-set parameters to optimize material delivery. By designing tasks around speed-optimized routes, RDS can enhance delivery efficiency and minimize AMR energy usage, aligning with best practices in autonomous logistics systems [35]. To initiate a Wind Task, users navigate to the RDS dashboard, select "Create Task," then configure key parameters such as task type, route, location, and simulation settings to optimize AMR performance. Once configured, the task appears on the dashboard with detailed route information, including travel distance and time, to facilitate easy tracking of ongoing and upcoming AMR activities. To start the AMR's delivery, the operator can activate the task with the "Running" button, allowing the robot to proceed to its designated final destination efficiently.

2.5.3. API integration

The RDS streamlines material delivery through "Wind Tasks" integrated with JavaScript, as illustrated in Figure 7. This flowchart outlines the process of setting up RDS to respond to API calls and handle tasks. The *registerHandler* function enables RDS to manage HTTP requests, specifically with:

a. *POST*: the HTTP request method.
b. */script-api/runTask*: the designated URL endpoint.
c. *runTask*: the function triggered by this endpoint.

In Figure 7, the RDS workflow begins with registering the handle and creating a boot function. Once an API call is received, the system processes the data and verifies the task name in the cache. If no cache entry is found, the task is cancelled; otherwise, it proceeds to issue the task to the "Wind Task" component, enhancing automated material handling in industrial settings.

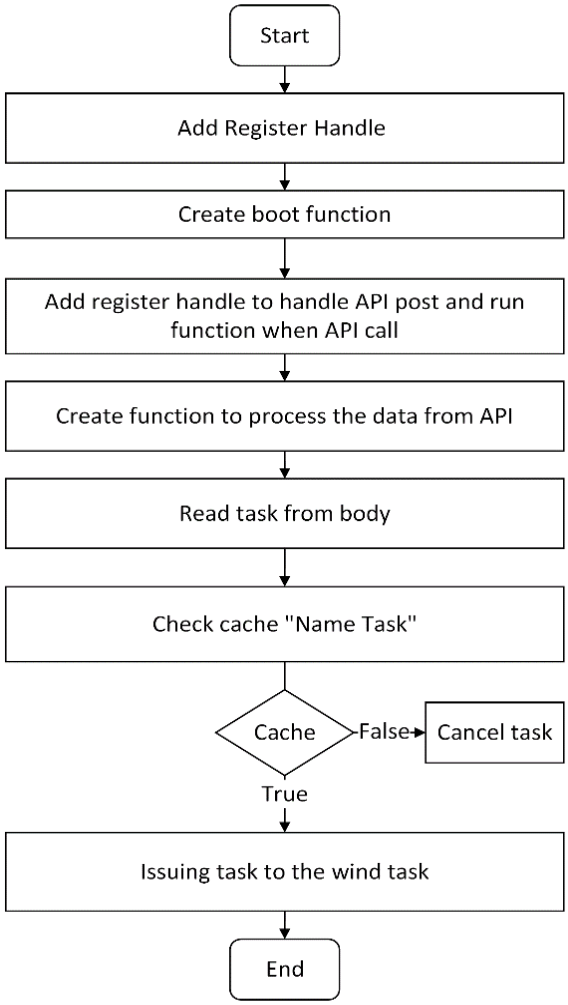


Figure 7. RDS integration flowchart with calling system

The "*POST*" parameter specifies the HTTP method that the *runTask* function will handle. The second parameter, "*/script-api/runTask*," designates the API's endpoint. The function *runTask* is invoked when the specified HTTP method and endpoint are called. The fourth parameter, set to false, ensures that the *runTask* function executes whenever the endpoint */script-api/runTask* receives a POST request, enabling the function to process the API data. After initiating the function, the ta*skName* cache is examined. Figure 8 illustrates the design of a calling system with a trigger button and trolley status display. This system activates the trolley when needed to transport materials: pressing the "Deliver trolley" button prompts the AMR to transport the pre-prepared materials. Additionally, the order monitoring system provides real-time status updates on the trolley, indicating whether it is ready or currently in use.

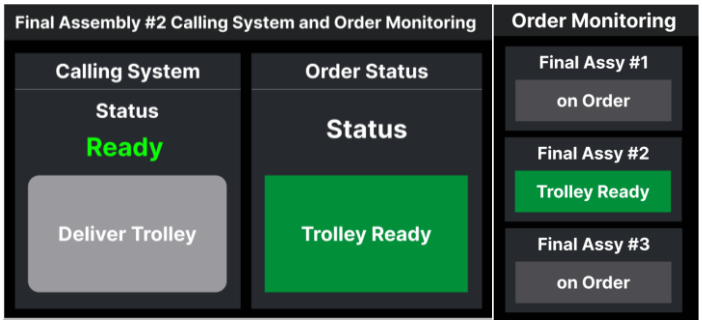


Figure 8. Calling system design

3. RESULTS AND DISCUSSION

This chapter presents an analysis and interpretation of the test results for the implementation of an AMR in material delivery within the three final assembly stages, providing insights into the AMR's performance in executing assigned tasks. Key performance metrics, including the AMR's travel precision and efficiency, were measured. To ensure reliability and robustness, each test was conducted ten times. The results of these accuracy tests serve as benchmarks for evaluating the success of AMR development. Furthermore, this section examines the AMR's navigational accuracy, comparing its actual travel paths to pre-established virtual routes on the map, along with a detailed review of the testing procedures.

3.1. AMR trial in following virtual line on map

Testing the AMR's ability to accurately follow a defined path is essential for its effective implementation. This test aims to validate the AMR's precision in adhering to a predetermined route autonomously. During the test, the AMR is directed to navigate along a specific path, demonstrating its capacity to follow the route accurately. Figure 9 illustrates two reference points with known coordinates (x_1, y_1; x_2, y_2) mapped on the route, which serve as checkpoints for assessing the AMR's path-following precision.

In two-dimensional space, coordinate points, represented as pairs of x and y values, specify the location of an object. When testing AMR accuracy, these coordinate points establish a virtual line that the robot must follow. The gradient (slope) of this line is calculated by selecting two coordinate points and dividing the difference in y-values by the difference in x-values, as shown in (1). This gradient defines the inclination of the line connecting the two points, as illustrated in Figure 9, and serves as a virtual guide that the AMR should follow throughout its route. The robot's accuracy is then assessed by comparing its actual position to the expected position along the virtual line. Consequently, the use of coordinate points and line gradient calculations offers a method to test the AMR's travel precision. Table 3 presents the target coordinate points alongside the actual points achieved, as well as the RMSE values calculated using (2).

Table 3 provides the coordinate points from the previous map. The target coordinates denote the points through which the AMR is intended to pass as it follows the virtual line, while the actual target coordinates represent the locations recorded by the AMR during its journey. To evaluate the precision of the AMR's travel, the RMSE was calculated for each data point. The RMSE values in the table are minimal, ranging from 0.001 to 0.020, indicating that the AMR closely followed the virtual line defined by the coordinate points with high accuracy.

Figure 10 illustrates the actual route taken by the AMR, as captured by RoboShop software at the final stop coordinates. The x- and y-axes in Figure 10 represent the AMR's intended and actual final positions. The sloped lines in the graph depict the AMR's path on the map in relation to its recorded locations. The close alignment between the two lines in Figure 10 indicates that the AMR accurately followed the designated points in the RoboShop software. Notably, point LM137, highlighted in Figure 10, shows the highest discrepancy, with an error of 0.016, attributed to the uneven floor texture and slight misalignment of the AMR's final position. This error could be reduced in a stable environment with a level surface, where the AMR's final position aligns closely with the mapped coordinates.

Figure 10 presents an evaluation of the slopes created by the target coordinates on the map and the actual coordinates reached by the AMR. The results indicate that the slopes of the lines formed by the target and actual coordinates are nearly identical across multiple points, demonstrating the AMR's high travel precision. This alignment confirms that the AMR technology effectively follows the predefined virtual path with accuracy, as evidenced by the relatively low RMSE values.

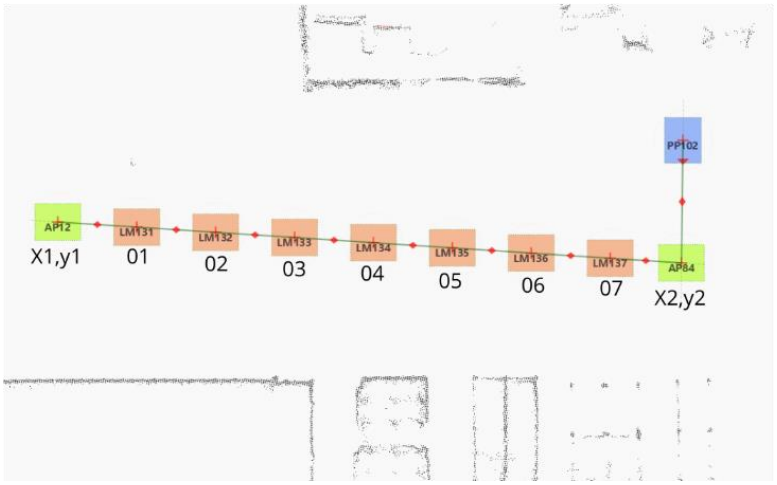


Figure 9. Gradient line coordinate points

Table 3. Coordinate points of the gradient line based on the test

Target	Target Coordinate (m)		Actual Coordinate (m)	Accuracy (m)	
	X	Y		Min	Max
AP12	1.891	-33.044	-33.037 ± 0.001	0.001	0.001
LM131	3.455	-33.147	-33.144 ± 0.006	0.006	0.009
LM132	5.019	-33.250	-33.243 ± 0.005	0.005	0.007
LM133	6.583	-33.353	-33.343 ± 0.006	0.006	0.009
LM134	8.147	-33.456	-33.450 ± 0.002	0.002	0.003
LM135	9.711	-33.560	-33.557 ± 0.007	0.007	0.010
LM136	11.275	-33.663	-33.645 ± 0.003	0.003	0.007
LM137	12.839	-33.766	-33.760 ± 0.016	0.016	0.020
AP84	14.252	-33.859	-33.865 ± 0.004	0.003	0.011

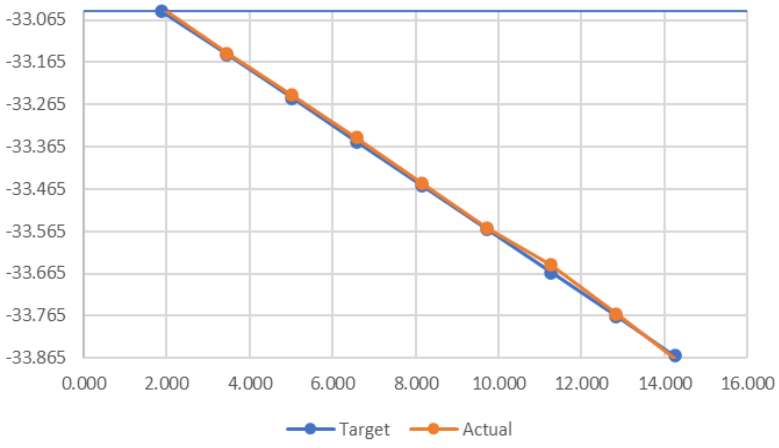


Figure 10. AMR accuracy testing of virtual line

3.2. Material delivery process

This study evaluates the performance of an AMR in transporting material trolleys between charging points, warehouses, and final assembly areas. During recycling transport, the AMR is programmed to follow a mapped route with designated coordinates, loading and unloading the trolley at specified locations. Upon completing the test, the RMSE is calculated using (2) to assess the accuracy of the AMR's movement in reaching target coordinates. A lower RMSE value indicates higher precision in reaching the specified map coordinates. The final coordinates for the AMR's journeys from the charging point to final assembly and back to the warehouse are detailed in Tables 4 to 6.

Table 4. AMR's travel coordinates to final assembly 1

Step	Description	Type Task	Starting Point		Destination on Map		Actual Destination	
			X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)
1	Go from parking to warehouse	movement	14.282	-31.421	1.881	-30.151	1.879 ± 0.005	-30.165 ± 0.007
2	Loading trolley	action	-	-	-	-	-	-
3	Deliver trolley from warehouse to production	movement	1.881	-30.151	38.114	13.469	38.113 ± 0.005	13.464 ± 0.020
4	Unload trolley	action	-	-	-	-	-	-
5	Go to deliver to recycle area	movement	38.114	13.519	35.268	15.006	35.257 ± 0.001	14.945 ± 0.036
6	Loading trolley recycle	action	-	-	-	-	-	-
7	Deliver trolley from production to warehouse	movement	35.268	15.006	1.881	-30.151	1.857 ± 0.008	-30.153 ± 0.001
8	Unload trolley	action	-	-	-	-	-	-
9	Back to parking	movement	1.881	-30.151	14.282	-31.421	14.276 ± 0.006	-31.424 ± 0.001

Table 5. AMR's travel coordinates to final assembly 2

Step	Description	Type Task	Starting Point		Destination on Map		Actual Destination	
			X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)
1	Go from parking to warehouse	movement	14.282	-31.421	1.881	-30.151	1.874 ± 0.004	-30.169 ± 0.005
2	Loading trolley	action	-	-	-	-	-	-
3	Deliver trolley from warehouse to production	movement	1.881	-30.151	38.700	-5.152	38.698 ± 0.002	-5.154 ± 0.003
4	Unload trolley	action	-	-	-	-	-	-
5	Go to deliver to recycle area	movement	38.700	-5.152	36.571	-7.817	36.549 ± 0.004	-7.779 ± 0.022
6	Loading trolley recycle	action	-	-	-	-	-	-
7	Deliver trolley from production to warehouse	movement	36.571	-7.817	1.881	-30.151	1.861 ± 0.007	-30.155 ± 0.001
8	Unload trolley	action	-	-	-	-	-	-
9	Back to parking	movement	1.881	-30.151	14.282	-31.421	14.269 ± 0.008	-31.425 ± 0.005

Table 6. AMR's travel coordinates to final assembly 3

Step	Description	Type Task	Starting Point		Destination on Map		Actual Destination	
			X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)
1	Go from parking to warehouse	movement	14.282	-31.421	1.881	-30.151	1.881 ± 0.002	-30.166 ± 0.002
2	Loading trolley	action	-	-	-	-	-	-
3	Deliver trolley from warehouse to production	movement	1.916	-30.151	38.754	-21.729	38.769 ± 0.020	-21.741 ± 0.031
4	Unload trolley	action	-	-	-	-	-	-
5	Go to deliver to recycle area	movement	38.754	-21.729	35.098	-20.225	35.128 ± 0.004	-20.198 ± 0.011
6	Loading trolley recycle	action	-	-	-	-	-	-
7	Deliver trolley from production to warehouse	movement	35.098	-20.225	1.916	-30.151	1.873 ± 0.009	-30.156 ± 0.005
8	Unload trolley	action	-	-	-	-	-	-
9	Back to parking	movement	1.881	-30.151	14.282	-31.421	14.267 ± 0.007	-31.422 ± 0.002

Based on the RMSE values shown in Tables 4 to 6, the AMR demonstrated high accuracy in reaching specified map coordinates, with delivery errors ranging from 0.002 to 0.031. This indicates that the AMR can reliably transport materials from origin to destination. The distance traveled by the AMR can be calculated using (3), where d represents the distance between two coordinate points, with (x_1, y_1) as the starting point and (x_2, y_2) as the destination. The AMR's speed is then calculated by dividing the traveled distance by the travel time. The AMR's journey begins at the charging station, proceeds to the warehouse, and then continues to each final assembly area, following the designated layout. The AMR's starting coordinate is based on its last stopping point; from there, it moves toward the target coordinates, recording its final position upon stopping.

Table 7 provides details on the distance, time, and average speed of the AMR during material deliveries from the warehouse to final assembly stations 1, 2, and 3. Distance is calculated using (1), while time is measured with a stopwatch, starting from the AMR's departure from the charging point to the warehouse and final assembly areas, and ending upon its return to the charging point. The AMR's average speed is then calculated by dividing the distance by the total time. Results indicate that the AMR can deliver materials over varying distances to final assembly stations 1, 2, and 3, achieving different average speeds. Factors such as road conditions, material type, and the AMR's technical capabilities may influence these variations in average speed.

Table 8 summarizes the total distance and time taken by the AMR to deliver materials to the three final assembly locations. The data indicate that the AMR's average distance traveled for these deliveries is

approximately 118.81 meters, with an average time of around 566.90 seconds. The longest distance and time were recorded for deliveries to final assembly 1, at approximately 141 meters and 638.32 seconds. Conversely, the shortest distance and time were for deliveries to final assembly 3, measuring around 101.31 meters and 512.76 seconds.

Figure 11 illustrates the AMR system's efficiency over a one-hour period prior to replenishment. Efficiency is calculated based on the AMR's operational movements between final assembly stations 1, 2, and 3, alongside the required charging time. The overall efficiency of the AMR system is determined by the total time required to complete the material delivery within this hourly replenishment cycle. According to the graph, the AMR achieves an efficiency rate of 14% for deliveries to final assembly 1, and 12% for final assembly 2 and 3, while charging occupies 10 minutes, or 19% of the total hour. Additionally, there is an idle time of 43%, indicating that nearly half of the AMR's operational cycle remains unused, presenting an opportunity to further optimize the system's efficiency. Overall, the implementation of the AMR system has led to a marked improvement, with a 57% efficiency rate post-implementation, primarily due to reduced idle time and accelerated material delivery, resulting in a more efficient manufacturing process.

Table 7. Distance, time, and average speed of AMR when delivering materials

Description	Final Assembly 1			Final Assembly 2			Final Assembly 3		
	Distance (m)	Time (s)	Avr Speed (m/s)	Distance (m)	Time (s)	Avr Speed (m/s)	Distance (m)	Time (s)	Avr Speed (m/s)
Go from parking to warehouse	12.47	73.23	0.17	12.47	71.98	0.17	12.47	74.88	0.17
Loading trolley	-	7.32	-	-	7.36	-	-	7.96	-
Deliver trolley from warehouse to production	56.71	187.57	0.30	44.50	153.02	0.29	37.79	123.59	0.31
Unload trolley	-	7.44	-	-	6.91	-	-	7.34	-
Go to deliver to recycle area	3.21	80.72	0.04	3.41	64.28	0.40	3.95	84.60	0.05
Loading trolley recycle	-	7.26	-	-	8.56	-	-	7.33	-
Deliver trolley from production to warehouse	56.16	180.17	0.31	41.26	147.46	0.28	34.63	120.51	0.29
Unload trolley	-	7.38	-	-	7.12	-	-	6.74	-
Back to parking	12.47	87.24	0.14	12.47	82.88	0.15	12.47	79.83	0.16

Table 8. Total distance and time delivering materials

Destination	Total Distance (m)	Total Time (s)
Final Assembly 1	141.01	638.32
Final Assembly 2	114.10	549.61
Final Assembly 3	101.31	512.76
Average	118.81	566.90

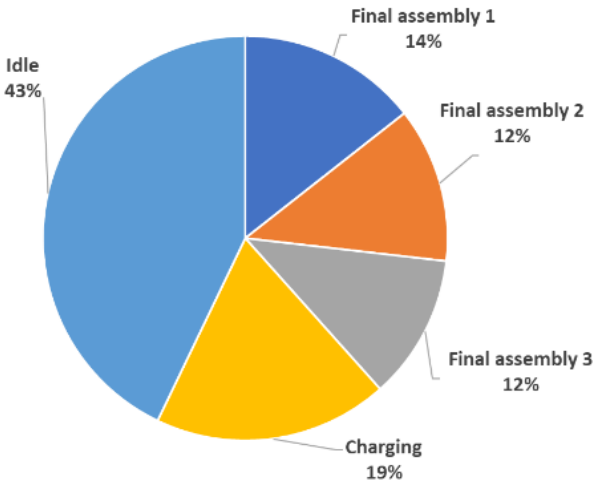


Figure 11. AMR efficiency graph for delivery of 1 AMR material at 3 final assemblies

4. CONCLUSION

This research successfully implemented an AMR-based material delivery system to support final assembly operations. By integrating SLAM-based navigation, customized trolley design, and RDS task coordination, the AMR effectively handled material movement across three final assembly stations. Experimental trials confirmed the robot's capability to follow predefined routes with high accuracy, achieving minimal RMSE values (0.001–0.020 m) and consistent adherence to virtual paths. The AMR system recorded an average travel distance of 118.81 meters and a cycle time of 566.90 seconds, leading to a measured operational efficiency of 57%.

The findings highlight that AMRs can reduce idle time, optimize material replenishment, and increase productivity, thereby offering significant improvements over manual or semi-automated methods. Beyond immediate gains in efficiency, this work demonstrates the broader potential of AMRs to enable scalable, modular, and flexible logistics systems. These results provide a pathway for industries to integrate AMRs with digital manufacturing platforms, such as digital twins, predictive maintenance, and AI-driven scheduling systems, advancing the vision of resilient and intelligent Industry 4.0 intralogistics ecosystems.

REFERENCES

[1] X. Zhao and T. Chidambareswaran, "Autonomous mobile robots in manufacturing operations," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE),* Auckland, New Zealand, 2023, pp. 1-7, doi: 10.1109/CASE56687.2023.10260631.

[2] J. Zhang, X. Yang, W. Wang, J. Guan, L. Ding, and V. C. S. Lee, "Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering," *Automation in Construction*, vol. 146, p. 104699, 2023, doi: 10.1016/j.autcon.2022.104699.

[3] S. Wang and W. Zhang, "Slam algorithms for autonomous mobile robots," in *Emerging Methodologies and Applications in Modelling: Modeling, Identification, and Control for Cyber-Physical Systems Towards Industry 4.0*, P. Mercorelli, W. Zhang, H. Nemati, and Y. Zhang, Eds. Academic Press, 2024, pp. 115–134, doi: 10.1016/B978-0-32-395207-1.00016-0.

[4] P. Barosz, G. Gołda, and A. Kampa, "Efficiency analysis of manufacturing line with industrial robots and human operators," *Applied Sciences*, vol. 10, no. 8, p. 2862, 2020, doi: 10.3390/app10082862.

[5] Z. L. Gan, S. N. Musa, and H. J. Yap, "A review of the high-mix, low-volume manufacturing industry," *Applied Sciences*, vol. 13, no. 3, p. 1687, 2023, doi: 10.3390/app13031687.

[6] M. Čech *et al.*, "Autonomous mobile robot technology for supplying assembly lines in the automotive industry," *Acta Logistica*, vol. 7, no. 2, pp. 103–109, 2020.

[7] R. Hercík, R. Byrtus, R. Jaros, and J. Koziorek, "Implementation of autonomous mobile robot in SmartFactory," *Applied Sciences*, vol. 12, no. 17, p. 8912, 2022, doi: 10.3390/app12178912.

[8] J. Sánchez-Cubillo, J. Del Ser, and J. L. Martín, "Toward fully automated inspection of critical assets supported by autonomous mobile robots, vision sensors, and artificial intelligence," *Sensors*, vol. 24, no. 12, p. 3721, 2024, doi: 10.3390/s24123721.

[9] J. Galarza-Falfán *et al.*, "Path planning for autonomous mobile robot using intelligent algorithms," *Technologies*, vol. 12, no. 6, p. 82, 2024, doi: 10.3390/technologies12060082.

[10] A. D. Hulwan, U. Rajurkar, and A. Gaikwad, "Implementation of Industry 4.0 in manufacturing industry: An autonomous mobile robots case study," *International Journal of Performability Engineering*, vol. 21, no. 4, pp. 188–198, 2025, doi: 10.23940/ijpe.25.04.p2.188198.

[11] U. Othman and E. Yang, "Human–robot collaborations in smart manufacturing environments: Review and outlook," *Sensors*, vol. 23, no. 12, p. 5663, 2023, doi: 10.3390/s23125663.

[12] C. D. de Sousa Bezerra, F. H. T. Vieira, and D. P. Q. Carneiro, "Autonomous robotic navigation approach using deep Q-network late fusion and people detection-based collision avoidance," *Applied Sciences*, vol. 13, no. 22, p. 12350, 2023, doi: 10.3390/app132212350.

[13] J. Pizoń, Ł. Wójcik, A. Gola, Ł. Kański, and I. Nielsen, "Autonomous mobile robots in automotive remanufacturing: A case study for intra-logistics support," *Advances in Science and Technology Research Journal*, vol. 18, no. 1, pp. 213–230, 2024, doi: 10.12913/22998624/177398.

[14] J. Escobar-Naranjo, G. Caiza, P. Ayala, E. Jordan, C. A. Garcia, and M. V Garcia, "Autonomous navigation of robots: Optimization with DQN," *Applied Sciences*, vol. 13, no. 12, p. 7202, 2023, doi: 10.3390/app13127202.

[15] G. Fragapane, R. B. M. de Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journal of Operational Research*, vol. 294, no. 2, pp. 405–426, 2021, doi: 10.1016/j.ejor.2021.01.019.

[16] T. Lackner, J. Hermann, C. Kuhn, and D. Palm, "Review of autonomous mobile robots in intralogistics: State-of-the-art, limitations and research gaps," in *Procedia CIRP*, 2024, vol. 130, pp. 930–935, doi: 10.1016/j.procir.2024.10.187.

[17] A. F. Costa, M. S. Carvalho, M. Henriques, and P. V Ferreira, "Strategy for the introduction of autonomous driving technologies: A case study in the logistics area of an automotive company," in *Procedia Computer Science*, 2022, vol. 204, pp. 337–345, doi: 10.1016/j.procs.2022.08.041.

[18] A. K. Grover and M. H. Ashraf, "Leveraging autonomous mobile robots for Industry 4.0 warehouses: A multiple case study analysis," *International Journal of Logistics Management*, vol. 35, no. 4, pp. 1168–1199, 2024, doi: 10.1108/IJLM-09-2022-0362.

[19] H. Xie, D. Zhang, X. Hu, M. Zhou, and Z. Cao, "Autonomous multirobot navigation and cooperative mapping in partially unknown environments," in *IEEE Transactions on Instrumentation and Measurement,* vol. 72, pp. 1-12, 2023, Art no. 4508712, doi: 10.1109/TIM.2023.3327469.

[20] H. Zhu, Y. Qiu, Y. Li, L. Mihaylova, and H. Leung, "An adaptive multisensor fusion for intelligent vehicle localization," in *IEEE Sensors Journal,* vol. 24, no. 6, pp. 8798-8806, 15 March15, 2024, doi: 10.1109/JSEN.2024.3360083.

[21] L. Wijayathunga, A. Rassau, and D. Chai, "Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: A review," *Applied Sciences*, vol. 13, no. 17, p. 9877, 2023, doi: 10.3390/app13179877.

[22] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet, "A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot," *Autonomous Robots*, vol. 27, no. 3, pp. 93–103, 2009,

doi: 10.1007/s10514-009-9130-2.
[23] P. Torres, H. Marques, and P. Marques, "Pedestrian detection with LiDAR technology in smart-city deployments: Challenges and recommendations," *Computers*, vol. 12, no. 3, p. 65, 2023, doi: 10.3390/computers12030065.
[24] J. Zhu, H. Li, and T. Zhang, "Camera, LiDAR, and IMU based multi-sensor fusion SLAM: A survey," in *Tsinghua Science and Technology,* vol. 29, no. 2, pp. 415-429, April 2024, doi: 10.26599/TST.2023.9010010.
[25] T. Su, H. Zhu, P. Zhao, Z. Li, S. Zhang, and H. Liang, "A robust LiDAR-based SLAM for autonomous vehicles aided by GPS/INS integrated navigation system," in *2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE)*, Dalian, China, 2021, pp. 351-358, doi: 10.1109/CACRE52464.2021.9501326.
[26] S. Yousuf and M. B. Kadri, "Information fusion of GPS, INS and odometer sensors for improving localization accuracy of mobile robots in indoor and outdoor applications," *Robotica*, vol. 39, no. 2, pp. 250–276, 2021, doi: 10.1017/S0263574720000351.
[27] C. Chen *et al.*, "OL-SLAM: A robust and versatile system of object localization and SLAM," *Sensors*, vol. 23, no. 2, p. 801, 2023, doi: 10.3390/s23020801.
[28] W. Wang, X. Liu, M. Zhao, and X. Xu, "VIS-SLAM: A real-time dynamic SLAM algorithm based on the fusion of visual, inertial, and semantic information," *ISPRS International Journal of Geo-Information*, vol. 13, no. 5, p. 163, 2024, doi: 10.3390/ijgi13050163.
[29] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 6019–6039, 2022, doi: 10.1016/j.jksuci.2021.02.015.
[30] K. K. Borkar *et al.*, "Stability analysis and navigational techniques of wheeled mobile robot: A review," *Processes*, vol. 11, no. 11, p. 3302, 2023, doi: 10.3390/pr11123302.
[31] Y. Cao, K. Ni, T. Kawaguchi, and S. Hashimoto, "Path following for autonomous mobile robots with deep reinforcement learning," *Sensors*, vol. 24, no. 2, p. 561, 2024, doi: 10.3390/s24020561.
[32] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao, "Review of autonomous path planning algorithms for mobile robots," *Drones*, vol. 7, no. 3, p. 211, 2023, doi: 10.3390/drones7030211.
[33] S. Noh, J. Park, and J. Park, "Autonomous mobile robot navigation in indoor environments: Mapping, localization, and planning," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea (South), 2020, pp. 908-913, doi: 10.1109/ICTC49870.2020.9289333.
[34] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar, and A. García-Cerezo, "Path planning for autonomous mobile robots: A review," *Sensors*, vol. 21, no. 23, p. 7898, 2021, doi: 10.3390/s21237898.
[35] Á. Bányai *et al.*, "Smart cyber-physical manufacturing: Extended and real-time optimization of logistics resources in matrix production," *Applied Sciences*, vol. 9, no. 7, p. 1287, 2019, doi: 10.3390/app9071287.

BIOGRAPHIES OF AUTHORS

Ahmad Riyad Firdaus completed bachelor and master programs at the Institut Teknologi Bandung (ITB) in 2000 and 2008 respectively, and received his PhD in automatic control and systems engineering at The University of Sheffield, United Kingdom in 2018. He is a Lecturer in the Department of Electrical Engineering, Politeknik Negeri Batam, Indonesia. His research interests cover issues related to nonlinear control systems, soft-computing, unmanned aerial vehicles (UAVs), and robotics. He is the author of various research papers published in national and international journals, and international conference proceedings. He can be contacted at rifi@polibatam.ac.id.

Imam Sholihuddin received a B.Sc. degree in physics from Brawijaya University, Indonesia in 2012. He is currently studying for a master's degree in physics at Brawijaya University, Indonesia. His research interests include machine learning, autonomous mobile robot, vision system, automation, manufacturing execution system and Industry 4.0. He can be contacted at imamsholihuddin@gmail.com.

Fania Putri Hutasoit currently pursuing a bachelor's degree in applied robotics engineering at Batam State Polytechnic. She can be contacted at hutasoitvania@gmail.com.

Agus Naba is a researcher and professor at Department of Physics, Brawijaya University (UB). He completed bachelor at Brawijaya University in 1995, master programs at the Institut Teknologi Bandung (ITB) in 2000, and received his Dr. Eng. in computer science at Univ. of Tsukuba Japan in 2007. His research interests cover issues related to smart systems and signal processing. He can be contacted at anaba@ub.ac.id.

Ika Karlina Laila Nur Suciningtyas is a researcher and lecturer at Department of Electrical Engineering, State Polytechnic of Batam. She completed her Physics B.S. degrees and M.S. degrees from Brawijaya University. She can be contacted at ikakarlina@polibatam.ac.id.