Robotica (2026), **44**, pp. 408–470
doi:10.1017/S0263574725102944

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

A hybrid optimization algorithm combining A* and the dynamic window approach for automated guided vehicle control in static and dynamic environments

Ankur Bhargava[1], Mohammad Suhaib[1] and Ajay K. S. Singholi[2]

[1]Department of Mechanical Engineering, Faculty of Engineering and Technology, Jamia Millia Islamia, New Delhi, India
[2]University School of Automation & Robotics, Guru Gobind Singh Indraprastha University, Delhi, India
Corresponding author: Ankur Bhargava; Email: ankurgsb21@gmail.com

Received: 24 October 2024; **Revised:** 25 September 2025; **Accepted:** 7 November 2025;
First published online: 12 December 2025

Keywords: automated guided vehicles (AGVs); modified A* algorithm; dynamic window approach (DWA); path planning; obstacle avoidance

Abstract

This study presents the control of an omnidirectional automated guided vehicle (AGV) with mecanum wheels using a hybrid optimization algorithm that combines a modified A* algorithm and the dynamic window approach (ADWA-HO). The method ensures efficient and precise navigation in both static and dynamic environments. The modified A* algorithm generates global paths, removes redundant nodes, and refines trajectories to improve efficiency and smoothness. At the same time, the dynamic window approach (DWA) enables real-time local path planning and obstacle avoidance. By evaluating the AGV's motion commands in real time, ADWA-HO selects optimal velocity commands within a dynamically updated window, thereby reducing route conflicts and ensuring stable movement. Compared with benchmark methods including dynamic A* (D*), artificial potential field (APF), DWA, probabilistic roadmap (PRM) & rapidly exploring random tree (RRT) fusion, and PRM & DWA fusion, the proposed ADWA-HO achieves improvements in average path length of 28.10%, 22.95%, 21.16%, 17.35%, and 10.71% and in average motion time of 23.48%, 17.85%, 15.47%, 11.86%, and 7.53% on both Map 1 and Map 2, respectively. The difference between simulation and real-world experiments is limited to 5.35% in path length and 4.38% in motion time, confirming the method's practical reliability. Furthermore, the algorithm achieves lower standard deviation in both metrics, indicating higher consistency of performance. This work also introduces a novel map-building strategy based on geometric and semantic data modules, which enhances the adaptability of real-world AGV deployment.

1. Introduction

Recent advances in automation and digitization have significantly transformed industrial mass production. However, the technological landscape presents significant challenges, including increased costs and operational complexity. Future organizations will need technologies to meet their needs to thrive in this ever-changing environment. Medium-sized self-driving vehicles operating in factories, assembly lines, and warehouses require an Internet of Things (IoT)-enabled hybrid navigation system. The ability of these automated vehicles to accelerate the transport of supplies from their source to designated workstations can enhance overall operational efficiency. A self-driving automated guided vehicle (AGV) needs to be furnished with a wide range of sensors, including light detection and ranging (LiDAR) [1], GPS [2], vision [3], ultrasonic [4], infrared [5], and radar [6], for it to be able to guide itself through new and tricky situations in real time. A complex control system is essential for processing sensor data to estimate navigation paths and detect obstacles. This enables the design of a safe and effective route to the destination or station. Integrating software with mechanical and electrical hardware, supported by robust local and remote communication networks, can reduce congestion and minimize collisions [7]. Furthermore, the system must be robust against external disturbances and capable of recovering from

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

faults. This will ensure that there is not much need for outside interference during the system's long operational life. An AGV requires information from various sensors to accurately sense its environment. These sensors differ in terms of expense, capability, and data collection capacity [8]. A medium-sized AGV employs visual recognition [9], deep learning [10], artificial neural networks [11], and perceptual intelligence [12] for autonomous driving. It is, therefore, a potent tool in the automation landscape. While existing navigational autonomy techniques are helpful in specific situations, they have numerous limitations in complex and dynamic industrial settings [13]. Although widely used for global path planning, the conventional A* method often produces inefficient paths with excessive turns, resulting in longer routes and higher runtimes [14]. Due to local minima problems, the vehicle may become stuck in confined spaces or crowded areas using the artificial potential field (APF) technique [15]. The dynamic window approach (DWA) leads to many diversions since it performs poorly in long-range path planning but well in short-range dynamic obstacle avoidance [16]. There are further drawbacks to the rapidly exploring random tree (RRT) [18], Boolean, Bayesian [17], and probabilistic roadmap (PRM) fusion approaches. In scenarios with numerous obstacles, PRM is ineffective, whereas RRT's randomization may result in suboptimal and unpredictable paths [19]. Utilizing such approaches can escalate difficulty without strengthening the ease of navigation. Given these weaknesses and increasing operational complexity, alternative strategies such as the modified A* and DWA hybrid optimization (ADWA-HO) system should be adopted to address evolving industrial processes and real-time control requirements.

This article is formatted as follows: Section 2 provides an in-depth outline of the methods employed in this study, including a literature review, and highlights areas that require further research. Section 3 delves into the kinematic and dynamic modeling. Section 4 details the ADWA-HO algorithm employed in this research, while Section 5 discusses the hardware and software implementation. Finally, Section 6 presents the experimental validation, and the study's outcomes and conclusions are presented in Section 7.

2. Literature review

In the 1950s, Barrett Electronics Corporation pioneered the development of AGVs for industrial applications. AGVs have become a widely adopted transportation technology to perform diverse tasks with efficiency, safety, and precision. Extensive research has been conducted in transportation, AGVs, and mobile robotics, covering static and dynamic environments. Connectivity advancements have further solidified the trend toward the now-necessary integration of intelligence into these systems. The emergence of IoT-enabled smart devices and the creation of smart vehicles have made major contributions to advancements in this field. Intelligent uncrewed vehicles are increasingly essential for modern companies to perform tasks such as loading and unloading. These vehicles enhance productivity in complex environments by continuously adapting to their surroundings. They update maps, recalculate paths, and maintain route accuracy to optimize material handling and shipment. These methods also show strong potential for adoption across other industries. According to research by Zheyi et al. [20], AGVs can function well in directed and unguided situations. The AGV uses a "potential function" to navigate in unguided environments. In guided environments, it utilizes radio frequency identification (RFID) tags as nodes to guide its travel from origin to destination, selecting the most optimal route among a range of feasible options. Achieving high precision and accuracy in robotics and autonomous vehicles is essential, as highlighted by Jans et al. [21]. This can be achieved using three primary approaches: mechanical, programmable, and sensor-based techniques. Sensor-based techniques leverage a variety of sensors, including RGB vision cameras, 9-axis IMU sensors, 3-axis infrared and sonar sensors, gyroscopes, magnetometers or compasses, accelerometers, and more. Optical shaft encoders measure the motor's angular position and provide digital output. The system achieves significantly improved motion precision by integrating this data with gyro sensor measurements in a PID controller. Caitité et al. [22] developed a line-follower AGV capable of identifying colored lines to restrict operator authorization and distinguish between courses. The system's microcontroller utilizes fuzzy logic with the Mamdani inference method to accurately detect lines and control AGV movement smoothly. The AGV utilizes RFID-based identity and authorization, as well as LDR (light-dependent resistor)-based color sensors,

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

to detect lines. To ensure smooth and precise stopping, some systems also incorporate a fuzzy AGV control mathematical model, which adjusts the electric motor's power based on speed and the distance to obstacles [23]. Gu et al. [24] propose a hybrid path planning technique for inspection robots on coal mine roads that combines the DWA with an enhanced A* algorithm and the Floyd method. B-spline slopes are utilized to optimize the route, eliminate duplicated nodes, and incorporate a target-weighted heuristic function into the A* algorithm. Findings show that this hybrid technique enhances safety, efficiency, and path smoothness, making it suitable for inspection robots in complex coal mine environments. Lazreg et al. [25] proposed a fusion-based path planning technique that combines particle swarm optimization (PSO) with an adaptive neuro-fuzzy inference system (ANFIS). While ANFIS has limited speed optimization capabilities, it enables learning-based control for accurate target reaching. PSO enhances the hybrid controller by optimizing speed and position adjustments, improving overall navigation performance. Simulation results indicate that the hybrid ANFIS-PSO approach outperforms conventional methods in movement efficiency and reduces travel distance. This method is particularly effective for mobile robotics, improving collision avoidance and enabling smoother, faster navigation. Jiang et al. [26] described AGVs that follow guided lines, select optimal pathways, and transmit their positions via a wireless sensor network. However, their approach may face limitations when sensors encounter black-and-white line patterns. Logistics automation and real-time data for informed decision-making are greatly enhanced by RFID technology, an integral part of Industry 4.0. Using a camera and a microcontroller, A.M. et al. developed a system for tracking lines and objects [27]. Meanwhile, Bhargava et al. developed a real-time mobile AGV control method that utilizes a Microsoft Kinect sensor connected to a laptop or PC, along with a mobile AGV equipped with LiDAR [28]. Their system uses microcontroller programming to control a wireless AGV via hand gestures detected by the Kinect sensor. Gestures such as start, progress, right, left, and backward are recognized, and MATLAB is used to generate movement commands, which are then transmitted to the AGV. Kokot et al. [29] implemented AGVs powered by the robot operating system (ROS) on assembly lines, employing two sets of LiDAR for autonomous navigation, mapping, and localization. An RGB-D camera was used to understand the workspace and detect workers, with YOLO V3's AI deep learning employed for this purpose. The AGV robot manipulator was tasked with pick-and-place operations. The proposed AGV demonstrated effective performance in SLAM, autonomous navigation, and collision avoidance in typical indoor settings. Ajeil et al. [30] addressed the route planning problem for autonomous robots in static and dynamic environments, aiming to generate short, smooth, and collision-free paths. They proposed a hybrid PSO-modified Firefly algorithm (MFB) to maximize path efficiency and smoothness. A local search (LS) algorithm identifies and corrects impractical path segments. An obstacle detection and avoidance module enable the AGV to respond dynamically to its environment. Simulation results demonstrate that this hybrid approach produces more feasible and optimized routes than conventional path planning methods. J. Zheng et al. highlighted that AGVs often operate in confined layouts, where dynamic interactions can lead to collisions and deadlocks. Although numerous solutions have been proposed, many fail to prevent dynamic collisions and deadlocks effectively. To address this, their research recommends scheduling multiple AGVs using dynamic resource reservations to minimize collisions. The layout is divided into uniform square blocks, representing points in an undirected graph. Vehicle trajectories are dynamically managed to resolve collision and deadlock issues by identifying shared resource locations [31–32]. AGV guidance technologies are rapidly evolving, with applications in smart delivery, automated pickup, and intelligent storage systems. The visual perception capabilities of AGVs have been significantly improved through advanced image processing techniques, enabling accurate and reliable interpretation of complex environments. Before these advancements, pixel interference was common due to variable lighting, inconsistent road surfaces, poor marking reflectivity, uneven marker widths, and other environmental inconsistencies. Qin et al. [33] presented a gray barycentric approach to reliably determine path marker centerlines and address these issues. This concept uses traditional methods to produce a normal vector at each array location. An integrated image-capturing card installed on a specially designed cart prototype was used to test the system, allowing real-time visual data capture for performance assessment. The improved approach decreases pixel interference, improving image durability and extraction accuracy.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

A storage AGV with optical navigation uses fast and flexible image processing. Visual feedback controls motors and servos in hybrid, position-based, or image-based systems. Visual serving systems typically use either an "eye-in-hand" or "hand-in-eye" camera configuration. In the eye-in-hand setup, the AGV's camera faces the target directly, while the hand-in-eye setup uses an external camera to monitor the AGV and the target object. These systems guide AGVs in industrial settings, frequently following norms. Despite their robustness, these systems can be disrupted by AGV paths or surrounding obstacles. Global vehicle monitoring enables dynamic route adaptation, thereby reducing the risk of collisions and facilitating more flexible navigation. Although these systems can be expensive for simple obstacle avoidance tasks, the AGV platform remains essential for industrial automation, smart warehouses, and intelligent logistics. This platform facilitates controlled loading and unloading, making it essential for research on AGV automated positioning and navigation [34]. The growing demand for AGVs in smart manufacturing emphasizes their increasing role in transporting machinery and supplies. In this regard, Wang et al. [35] investigated AGV path planning for a single manufacturing line in a workshop. They developed a computational model designed to minimize transit time and proposed an enhanced PSO method for path planning and optimization.

Science and technology influence all aspects of human activity, and modern industrial practices are increasingly outsourcing human tasks to machines and automation. Automation frees people from repetitive activities, which are typically governed by rigid formulas. Kim et al. [36] describe a path planning and control module for an omni-wheeled mobile robot (OWMR). To improve robustness and adaptation speed in OWMR motion control, a brain limbic system (BLS)-based regulation technique was developed. The A* algorithm and the fuzzy analytic hierarchy process (FAHP) create the most effective route planning module for effective navigation. Simulations validate the proposed motion control and path planning framework for dynamic target tracking, circular path tracking, and point-to-point motion. The A*-FAHP algorithm guarantees collision-free, optimal path planning in various settings, including warehouses and automated valet parking, and the findings show that the BLS-based controller outperforms conventional PID controllers. The Halton-biased RRT algorithm for efficient robotic route planning is described in this study by Zhong et al. [37]. Using uniform sampling based on the Halton sequence guarantees consistent search space coverage and successfully addresses the uneven node distribution issue in conventional RRT. Both candidate sampling pool techniques and a mouse-inspired goal-oriented strategy optimize sampling quality and memory economy. Path redundancy is removed using a multilevel planning technique, and the final path is smoothed using a B-spline technique. Regarding planning time, path length, and path quality, HB-RRT outperforms RRT, Bionic Target Bias-RRT, and Informed-RRT* in simulations and real-world tests. To overcome the constraints in the spatial localization of AGV LiDAR data in the industrial sector, Yan et al. [38] examined the application of machine learning methods. Their research explores the application of bagging and boosting techniques to decision trees and the k-nearest neighbor method. They evaluated various algorithms using different criteria and hyperparameter adjustments, with multiple regression used for comparative analysis. For automated vehicles, 3D point cloud data is essential for environmental assessment. Miljkovic et al. [39] presented a hybrid AGV control system that utilizes image-based visual serving (IBVS) for precise movements near loading/unloading points and position-based control (PBC) for global navigation. Unlike conventional AGV navigation, monocular SLAM eliminates the requirement for precise environmental maps or markers. The neural extended Kalman filter (NEKF) model uses a feedforward neural network to improve state estimations. The hybrid control system and NEKF-based position estimate were tested in a laboratory setting with a mobile robot equipped with a simple camera. Experimental results demonstrate improved motion precision and navigation accuracy, confirming the viability of the proposed AGV hybrid control system. Gao et al. [40] combine deep reinforcement learning and actor-critic control to create an integrative control system. The Lyapunov method creates a kinematics control law that minimizes pose errors as an initial control input. In a finite Markov decision process tracking control problem, deep deterministic policy gradient (DDPG) maximizes control actions. Table I presents a chronological and comparative overview of the primary navigation techniques developed for AGVs.

Table II presents a detailed comparative analysis of the advanced algorithms, including the author, method, objective, contribution, and advantages of the proposed ADWA-HO algorithm.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

Table I. *Evolution of AGV navigation techniques: descriptions and key features.*

Navigation technique	Description	Key features	Sensors used
Potential function *Early 1990s*	Used in unguided environments for navigation.	Adapts to dynamic surroundings and updates paths in response to environmental changes.	Ultrasonic sensors and infrared sensors.
Radio-frequency identification tags *Mid 2000s*	Utilizes radio frequency identification tags in guided environments.	Follows predefined routes and selects the optimal path from the origin to the destination.	RFID readers and RFID tags.
Mamdani fuzzy inference system *Early 2005s*	Uses fuzzy logic for decision-making in line-follower AGVs.	Enables smooth stopping, color detection, and operator authorization.	Passive infrared (PIR) sensors and ultrasonic sensors.
Swarm-optimized neuro-fuzzy inference system *Early 2010s*	Combines PSO and adaptive neuro-fuzzy intelligence.	Improves navigation and target reach by optimizing speed and location.	Ultrasonic sensors, color sensors, and encoders.
Barycentric gray system methodology *Mid 2014s*	Determines path marker center lines using image processing.	Reduces pixel interference, improves image durability, and increases extraction accuracy.	RGBD cameras and image processing sensors.
Quasi-random biased RRT algorithm *Late 2017s*	Enhances rapidly exploring random tree (RRT) with Halton sequence sampling.	Reduces path redundancy, improves planning time, and ensures smooth paths.	Ultrasonic sensors, RGB, and depth cameras.
Hybrid neural-extended Kalman filter *Early 2021s*	Improves state estimation using neural networks and Kalman filtering.	Enhances motion precision and navigation accuracy in dynamic environments.	LiDAR and encoders.
Sensor-based techniques *Mid 2022s*	Employs multiple sensor fusion.	Enhances precision with PID control algorithms and real-time sensor feedback.	RGB and depth cameras, IMU sensors, infrared sensors, and gyroscopes.
Hybrid path planning *Late 2023s*	Combines DWA, Floyd method, and B-spline curves for path optimization.	Improves safety, efficiency, and path smoothness in complex environments.	LiDAR, ultrasonic sensors, RGBD cameras, and IMU sensors.
Deep reinforcement learning *Mid 2024s*	Combines deep learning and actor-critic control for integrative AGV control.	Minimizes pose errors and maximizes control actions in complex tasks.	RGB cameras, LiDAR, IMU sensors, and encoders.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

Table II. *Comparative analysis of advanced algorithms.*

Author	Method	Objective	Contribution	Proposed method performance
A.K.S. Singholi et al., Alkhawaja et al. , P.S. Bharti et al. [1], [4], [12]	IoT-based methodology.	To identify the hybrid navigation method for AGVs and EVs.	The model for the hybrid navigation technique has been developed for AGVs and EVs.	The modified A*-DWA hybrid optimization algorithm offers better adaptability in dynamic environments than IoT-based methods.
Zheyi et al., Reis et al., A.M. et al. [20], [22], [27]	An approach based on artificial potential and line following.	Line following in a typical setting to pursue an objective.	Routing using nodes, infrared, and RFID identifiers.	Provides a more flexible and efficient approach than fixed-line following.
R. M. Jans et al. [21]	Methods of programming and sensor-based techniques.	Mechanical, PID, and sensor-based technique integration.	PID controller-based motor angular position control utilizing sensor data.	Ensures real-time adaptability and higher precision in navigation.
A. E. Setiawan et al. [23]	Fuzzy logic regulation.	Control utilizing fuzzy logic for the high-level AGV.	Achieve precise pausing and flexible movement.	Offers better decision-making in dynamic environments compared to fuzzy logic.
Gu et al. [24]	DWA-based hybrid path planning method.	Autonomous navigation in static–dynamic scenes under challenging geological conditions.	Implement a grid-based map for simulations and validation through real-world experiments in a simulated coal mine roadway.	Provides enhanced path planning efficiency and robustness in dynamic terrains.
Lazreg et al. [25]	Combining ANFIS and PSO.	Overcome the limitations of ANFIS by integrating PSO for improved speed optimization and positioning.	Developing a hybrid controller that combines the advantages of both methods leads to improved movement performance.	Surpasses ANFIS-PSO by ensuring smooth transitions and improved obstacle avoidance.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

Table II. *Continued.*

Author	Method	Objective	Contribution	Proposed method performance
M. Kokot et al. [29]	Making use of ROS with a SLAM-based hybrid algorithm.	SLAM and LiDAR-based system.	Deep learning in AI utilizing YOLO/RGBD/LiDAR.	Integrates visual and LiDAR SLAM, offering enhanced mapping accuracy and real-time navigation.
Ajeil et al. [30]	A hybrid PSO-MFB algorithm with local search.	To develop short, smooth, collision-free autonomous robot pathways in both static and dynamic situations.	Impractical pathways can be refined using a PSO-MFB-based path planning approach with an integrated local search (LS) algorithm. Enables dynamic obstacle detection and avoidance.	The modified A*-DWA hybrid optimization algorithm enhances collision avoidance through multi-sensor data fusion.
J. Zheng et al. [32]	Dynamic resource reservation for AGV scheduling in a grid-based environment.	To minimize collisions and deadlocks in confined layouts with multiple AGVs.	Introduced a dynamic scheduling method that divides the layout into uniform square blocks.	Offers superior real-time flexibility and collision avoidance in dynamic situations compared to grid-based techniques.
S. Qin et al. [33]	Gray barycentric method for visual perception-based AGV navigation.	To improve AGV path tracking accuracy by reducing pixel interference and enhancing image durability.	Developed a gray barycentric approach for exact centerline extraction of path markers to overcome uneven lighting and poor reflectivity.	Enhances real-time adaptation, obstacle avoidance, and path planning compared to static visual methods.
C. Wang et al. [34]	Improved particle swarm optimization algorithm.	AGV route planning difficulty.	Define a computational formula that requires the least amount of time to transport.	Improves route efficiency with dynamic path adjustments.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

Table II. *Continued.*

Author	Method	Objective	Contribution	Proposed method performance
Kim et al. [36]	A* algorithm is coupled with the FAHP to develop the best course for the path planning module.	Enhance real-time navigation performance by combining a BLS-based motion controller with a hybrid A*-FAHP algorithm for path planning.	Simulations in warehouse logistics and automated valet parking validate the proposed control and planning method.	Refines A*-FAHP with enhanced obstacle avoidance and adaptability.
Zhang et al [37]	The Halton sequence, known for its uniform distribution and low disagreement, is used for sampling to avoid the problems with unequal distribution in standard RRT.	To overcome the disadvantages of the standard RRT algorithm and increase the efficacy, quality, and smoothness of robot path planning.	Developing a candidate sample pool and goal-oriented procedures, modeled after mice, improves node expansion and reduces memory constraints.	Enhances memory efficiency and computational performance in complex environments.
Hata et al. [38]	Utilizing machine learning methods to enhance AGV LiDAR data spatial localization in industrial settings.	It aims to determine the most effective data-driven approach for processing 3D point cloud data, ensuring precise environmental assessment.	Optimizing LiDAR-based localization utilizing sophisticated machine learning algorithms (bagging, boosting, decision trees, and K-nearest neighbors).	Integrates SLAM-based localization for improved AGV accuracy.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

Table II. *Continued.*

Author	Method	Objective	Contribution	Proposed method performance
Miljkovic et al. [39]	Presents a hybrid AGV control system combining image-based visual serving (IBVS) for precise motion control with position-based control (PBC) for global navigation.	Develop a robust and flexible AGV navigation system that ensures precise movement without relying on predefined environmental maps or external markers.	To increase the precision of AGV positioning, a novel dual-loop control system was developed that combines IBVS for fine motion control and PBC for global navigation.	Ensures superior precision and adaptability without dependency on markers.
Gao et al. [40]	This hybrid control strategy for nonholonomic wheel robots combines mode-based management with an actor-critic deep learning reinforcement learning approach.	Combine deep reinforcement learning with traditional kinematic control to enhance the tracking and control capabilities of nonholonomic wheeled mobile robots.	Consider the limits of acceleration, velocity, and uncertainty to guarantee stable and smooth motion control.	Integrates deep learning-based path planning for higher efficiency and smoother trajectory execution.

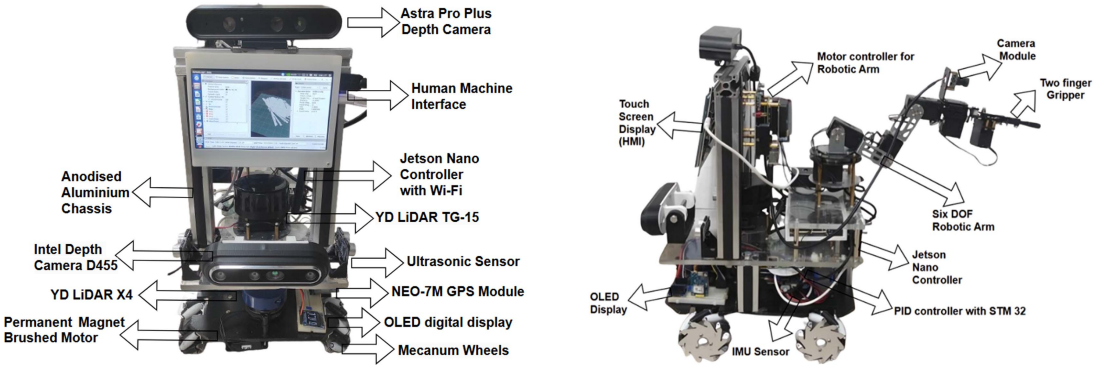


Figure 1. *The automated guided vehicle labeled diagrams.*

- Many studies focus on static or regulated industrial settings, whereas dynamic contexts with unexpected problems are understudied. Most AGV systems are designed for controlled, predictable environments, and little is known about how they can adapt to dynamic situations, such as crowded rooms or rapidly changing industrial floors. To address these dynamic challenges, the ADWA-HO algorithm optimizes real-time obstacle avoidance and routing in dynamic indoor and outdoor environments.
- Studies have employed LiDAR, vision systems, and global navigation satellite system (GNSS) for navigation, but few have explored combining sensing modalities to enhance AGV navigation and control in complex scenarios. These sensors have not yet been incorporated into an indoor and outdoor system. The AGV can better perceive and react to its environment with the ADWA-HO and multi-sensor data fusion, including LiDAR, vision sensors, IMUs, and ultrasonic sensors.
- Potential functions, improved particle swarm optimization (IPSO), and visual-SLAM are path planning approaches. However, real-time algorithms that quickly compute the best pathways while considering static and dynamic barriers are needed. Complex conditions can make A^* or DWA rigid or inefficient. However, the A^* and DWA combination enables the real-time scheduling of routes, facilitating quick decision-making in changing circumstances.
- AGVs often use traditional systems for line-following, obstacle avoidance, and other warehouse operations. The flexible, high-performance AGVs for numerous applications and settings are understudied. The customized omnidirectional AGV, as shown in Figure 1, equipped with an ADWA-HO algorithm, may be suitable for various applications in busy industrial and public locations.
- Many methods fail to prevent dynamic collisions or are oversimplified despite deadlock and collision avoidance challenges. Few dynamic solutions to these problems have been studied, particularly in the context of AGVs operating in external or outdoor environments. By establishing a stable foundation, ADWA-HO's dynamic performance can minimize collisions and deadlocks in real-time environments.
- Prior research has stressed the need for quick processing to control the AGV, but it has rarely examined the most efficient methods under computing constraints. Dynamic real-time decision-making is complex for traditional algorithms. The modified A^*-DWA fusion may enhance AGV decision-making by integrating real-time sensor data processing with efficient path planning and obstacle avoidance. The ADWA-HO, with an omnidirectional AGV control system, may fill these gaps. To advance industrial and AGV technology, this research focuses on scalability, real-time path planning, multi-sensor integration, and dynamic settings.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

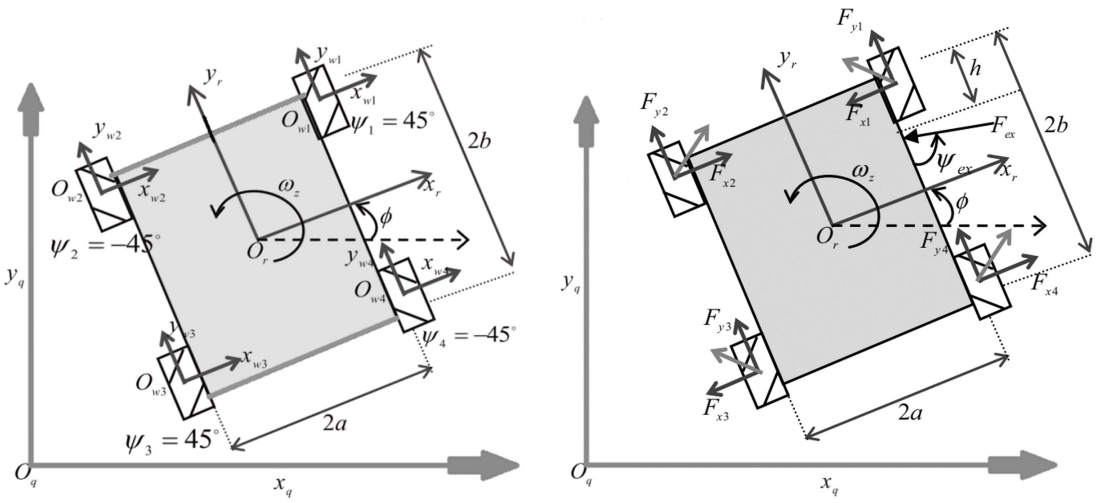


Figure 2. *The automated guided vehicle free body diagrams.*

Advanced methods for thorough comparison are presented in this research, such as dynamic A* (D*), APF, DWA, PRM, PRM & RRT fusion, and PRM & DWA fusion. This comprehensive analysis should help us understand the merits and demerits of each algorithm in various situations. This comparison assesses the performance of each technique and suggests future research directions, including the development of robust collision avoidance algorithms for highly dynamic environments and the creation of hybrid algorithms that can operate effectively in both structured and unstructured settings. Ultimately, path planning and obstacle avoidance enhance the intelligence and success of industrial AGVs.

3. Locomotion of the AGV

Four-wheeled AGVs with mecanum wheels provide remarkable maneuverability and agility. The omnidirectional mobility made possible by the 45-degree rollers on these wheels enables the AGV to move forward, backward, sideways, and diagonally without changing its orientation. Mecanum wheel AGVs are well suited for maneuvering in confined spaces and complex environments due to their ability to maintain accuracy and a variety of movements in industrial applications [41]. The AGV can perform precise movements, such as spinning on the spot, by controlling the speed and direction of each wheel, increasing its operational efficiency in various scenarios.

Figures 1 and 2 depict the four-wheel omnidirectional mobile AGV, which consists of four mecanum wheels. The rollers positioned around the edges of the wheel are consistently oriented at an angle ψ. In this specific case, with ψ set to 45 degrees, the wheel can move smoothly at a 45-degree angle relative to the direction of the applied motion. Each wheel has a permanent magnet-brushed motor, as described in ref. [42], which delivers the necessary torque to drive the AGV's movement. The center of gravity of the AGV, denoted as O_r in Figure 2, is considered the reference point for motion analysis. The AGV is thought to travel on a flat, evenly spaced surface, with all its components, including the wheels, treated as rigid bodies to simplify the equation of motion. The kinematic and dynamic modeling is as follows:

3.1. The kinematics and dynamics of AGV

The fixed coordinate frame is denoted as O_q, while O_r represents the mobile coordinate frame attached to the AGV. Each wheel has its wheel coordinate frame, represented as O_{wi} (where i = 1 to 4). As described

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

in ref. [43], we express the wheel position vector in its respective coordinate frame, Owi, as P_{wi}. The corresponding equation is given by:

$$P_{wi}(\text{where i} = 1 \text{ to } 4) = [x_{wi}\ y_{wi}\ \varphi_{wi}]^T \tag{3.1a}$$

The rotational speed of each wheel around its hub is represented by $\dot{\theta}_{ix}$, while the roller's angular velocity is denoted by $\dot{\theta}_{ir}$, and the angular velocity of the wheel at the point of contact with the ground is given by $\dot{\theta}_{iz}$. Additionally, the wheel radius is represented by R_i, and the roller slope angle for each wheel is denoted by ψ_i (i = 1 to 4). The radius of the roller is symbolized by ref. [44]. Based on these parameters, the motion vector of the AGV can be formulated as described in ref. [45].

$$\dot{\mathbf{P}}_{wi} = \begin{bmatrix} \dot{x}_{wi} \\ \dot{y}_{wi} \\ \dot{\phi}_{wi} \end{bmatrix} = \begin{bmatrix} 0 & r\sin(\psi_i) & 0 \\ R_i & -r\cos(\psi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{ix} \\ \dot{\theta}_{ir} \\ \dot{\theta}_{iz} \end{bmatrix} \tag{3.1b}$$

$$\mathbf{T}^{\mathbf{r}}_{\mathbf{wi}} = \begin{bmatrix} \cos(\phi^r_{wi}) & -\sin(\phi^r_{wi}) & d^r_{wiy} \\ \sin(\phi^r_{wi}) & \cos(\phi^r_{wi}) & d^r_{wix} \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1c}$$

Equations show the rotational angle of O_{wi} w.r.t. O_r and is represented by ϕ^r_{wi} (i = 1 to 4). Additionally, the translational distance between the two coordinate frames is denoted by d^r_{wix} and d^r_{wiy}, which define the displacement along the x and y axes, respectively. Suppose the position vector of the AGV in the O_r coordinate frame is given by $P_r = [x_r\ y_r\ \phi_r]^T$. In this case, the equation connecting P_r and P_{wi} can be given as $P_r = T^r_{wi}.P_{wi}$. Additionally, as shown in Figure 2, it is evident that $\phi^r_{w1} = \phi^r_{w2} = \phi^r_{w3} = \phi^r_{w4} = 0$. Based on these conditions, the AGV's vector of velocity is presented as follows:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\phi}_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & d^r_{wiy} \\ 0 & 1 & d^r_{wix} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_{wi} \\ \dot{y}_{wi} \\ \dot{\phi}_{wi} \end{bmatrix} \tag{3.1d}$$

From (3.1b) and (3.1d), we get,

$$\dot{\mathbf{P}}_{\mathbf{r}} = \mathbf{J}_{\mathbf{i}}\dot{\mathbf{q}}_{\mathbf{i}} \tag{3.1e}$$

where

$$\mathbf{J}_{\mathbf{i}} \in R^{3\times 3} = \begin{bmatrix} 0 & r\sin(\psi_i) & d^r_{\text{wiy}} \\ R_i & -r\cos(\psi_i) & d^r_{\text{wix}} \\ 0 & 0 & 1 \end{bmatrix}$$

The ith wheel's Jacobian matrix is represented as $\mathbf{J_i}$ and, $\dot{q}_i = [\,\theta_{ix} \quad \theta_{ir} \quad \theta_{iz}\,]$.
AGV dynamics on a flat surface can be calculated using the Newton–Euler method. To simplify the computation of the equation of motion, the coordinate frame Or is assumed to lie at the AGV's center of gravity, as this optimizes computational efficiency. The AGV's mass matrix is denoted by M_m.

$$S_q = \begin{bmatrix} x_q & y_q \end{bmatrix}^T \tag{3.1f}$$

S_q is the vector showing the position in a frame with fixed coordinate "O_q" as in ref. [46].

$$F_q = \begin{bmatrix} F_{qx} & F_{qy} \end{bmatrix}^T \tag{3.1g}$$

F_q is the vector showing the force in a frame with fixed coordinate "O_q" as in ref. [46].
When the second law of Newton is applied, the equation of motion of the AGV (shown in Figure 2) in the coordinate frame O_q can be represented as in ref. [47]:

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

$$F_q = M_m \ddot{S}_q \tag{3.1h}$$

$$\begin{bmatrix} F_{qx} \\ F_{qy} \end{bmatrix} = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \ddot{x}_q \\ \ddot{y}_q \end{bmatrix} \tag{3.1i}$$

$$\text{if, } {}^{q}R_r(\varphi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{3.1j}$$

where ${}^{q}R_r(\varphi)$ is the matrix that represents the transformation of O_r w.r.t O_q; therefore,

$$\dot{S}_q = {}^{q}R_r(\varphi)\,\dot{S}_r \text{ and } F_q = {}^{q}R_r(\varphi)\,F_r \tag{3.1k}$$

Eq. (3.1h) can be simplified by using the transformation matrix as follows:

$$M_m\left({}^{q}R_r(\varphi)\,\ddot{S}_r + {}^{q}R_r(\dot{\varphi})\,\dot{S}_r\right) = {}^{q}R_r(\varphi)\,F_r \tag{3.1l}$$

Multiply both sides by ${}^{q}R_r(\varphi)^{-1}$ gives,

$$M_m({}^{q}R_r(\varphi)^{-1}\,{}^{q}R_r(\varphi)\ddot{S}_r + {}^{q}R_r(\varphi)^{-1}{}^{q}R_r(\dot{\varphi})\dot{S}_r) = {}^{q}R_r(\varphi)^{-1}{}^{q}R_r(\varphi)F_r \tag{3.1m}$$

Since ${}^{q}R_r(\varphi) = {}^{q}R_r(\varphi)^{T}$, ${}^{q}R_r(\varphi)^{-1}\,{}^{q}R_r(\dot{\varphi}) = \dot{\varphi}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. Therefore,

$$\begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \ddot{x}_r - \dot{\phi}\dot{y}_r \\ \ddot{y}_r + \dot{\phi}\dot{x}_r \end{bmatrix} = \begin{bmatrix} F_{xr} \\ F_{yr} \end{bmatrix} - \begin{bmatrix} \beta_x \dot{x}_r \\ \beta_y \dot{y}_r \end{bmatrix} + \begin{bmatrix} -F_{ex}\cos(\psi_{ex}) \\ F_{ex}\sin(\psi_{ex}) \end{bmatrix} \tag{3.1n}$$

In the context of this equation, the symbol F_{xr} is used to represent the total force that is doing its work in the x direction, and the symbol F_{yr} is used to represent the total force that is doing its work in the y direction. The β_x is the coefficient of linear friction along the *x*-axis, and β_y is the coefficient of linear friction along the *y*-axis. The body of the AGV is being subjected to the influence of an external force at a distance (h) from the upper edge of the AGV, as in Figure 2. In this context, the symbol F_{ex} is used to denote this force. The force F_{ex} is situated at an angle of ψ_{ex} w.r.t. y_r. Consider that τ, I_q, and β_z represent the AGV's center of gravity moment, inertia moment, and linear friction coefficient in the z-direction, as described in ref. [48]. The Euler equation can be represented in the following fashion by making use of the free body model, which is depicted in Figure 2.

$$I_q\ddot{\phi} = \tau - \beta_z\dot{\phi} + F_{ex}\cos(\psi_{ex})\,a - F_{ex}\sin(\psi_{ex})\,(b-h) \tag{3.1o}$$

If the dynamics of the motor that is coupled to each wheel are taken into consideration, the driving force F_{di} (where i = 1 to 4) that is produced by the DC motor can be stated in the following manner, as in ref. [49]:

$$F_{di}\,(\text{where } i = 1||4) = \alpha u_i - \beta R\dot{\theta}_i \tag{3.1p}$$

The input voltage that is applied to each motor is denoted by u_i (where i = 1 to 4). The motor coefficients, denoted by α and β, are computed by employing the following formulas as in ref. [49]:

$$\alpha = \frac{k_\tau}{RR_a} \tag{3.1q}$$

$$\beta = \frac{k_\tau k_e n}{R^2 R_a} \tag{3.1r}$$

The torque coefficient of the motor is shown by the symbol k_τ, the gear ratio is indicated by n, the armature resistance is indicated by R_a, and the back emf coefficient is indicated by k_e in Eqs. (3.1q) and (3.1r).

Based on the free body diagram (FBD) in Figure 2, the elements F_{xr}, F_{yr}, and τ are represented in terms of driving force, respectively, as in ref. [50].

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

$$F_{xr} = \frac{1}{2}\left(-F_{d1} + F_{d2} - F_{d3} + F_{d4}\right) \tag{3.1s}$$

$$F_{yr} = \frac{1}{2}\left(F_{d1} + F_{d2} + F_{d3} + F_{d4}\right) \tag{3.1t}$$

$$\tau = \frac{a}{2}\left(F_{d1} - F_{d2} - F_{d3} + F_{d4}\right) + \frac{b}{2}\left(F_{d1} - F_{d2} - F_{d3} + F_{d4}\right) \tag{3.1u}$$

For efficient route planning and real-time obstacle avoidance, a mecanum wheel AGV must precisely manage its revolutions per minute (rpm) and wheel velocities to perform linear and angular motions in varied surroundings, as determined by the above equations. These movements enable the AGV to avoid obstacles and navigate complex terrains with precision. Omnidirectional mobility requires wheel speed control to allow the AGV to travel forward, backward, sideways, and diagonally without changing orientation. Static and dynamic control equations optimize AGV performance by ensuring efficient motion planning, stability, and responsiveness in both organized and unstructured situations. To validate the kinematic and dynamic equations governing the motion of the mecanum wheel AGV, we conducted MATLAB-based simulations to generate and analyze various trajectories, including circular, elliptical, triangular, tetragonal, lemniscate, and five-petal rose curves. These trajectories serve as benchmarks to assess the AGV's ability to execute precise path-following while maintaining stable motion control. Figure 3 shows the AGV testing simulations based on the equations.

4. Modified A* and dynamic window approach evolutionary algorithms

4.1. Modified A algorithm*

The modified A* (MA*) method enhances the traditional A* algorithm by optimizing path searching in dynamic environments. MA* is ideal for real-world settings with constant changes because it incorporates dynamic obstacles, real-time updates, and grid structures, whereas A* utilizes heuristics to determine the shortest path. For reliable and effective AGV path planning, MA* combines BFS with Dijkstra's global search [51]. It evaluates nodes one after the other, choosing the next node based on the lowest $f(n)$ as $f(n) = g(n) + h(n)$, where n is the number of nodes, $g(n)$ is the starting cost, and $h(n)$ is the goal heuristic cost [52–54]:

$$\left.\begin{aligned} g\,(n) &= \sqrt{(y_s - y_n)^2 + (z_s - z_n)^2} \\ h\,(n) &= \sqrt{(y_n - y_t)^2 + (z_n - z_t)^2} \end{aligned}\right\} \tag{4.1a}$$

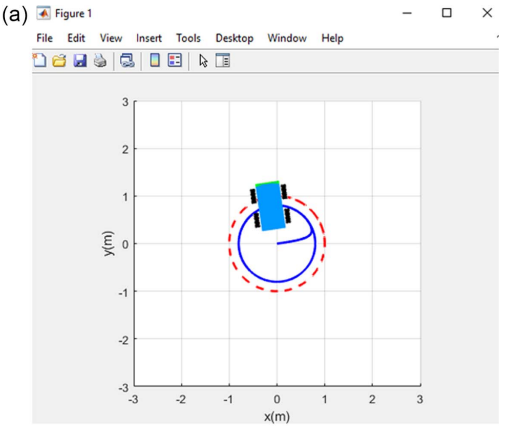
The start, current, and target node positions are shown by (y_s, z_s), (y_n, z_n), and (y_t, z_t), respectively. In complex scenarios, the MA* algorithm employs an eight-directional analysis on a raster map to ensure accurate and collision-free route planning. Figure 4 shows the illustration of the MA* algorithm. Figure 5 shows the Simulation results of the MA* algorithm.

4.2. Dynamic window approach

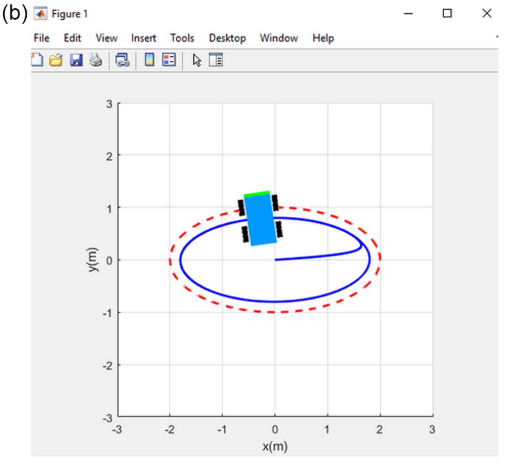
The DWA is a well-liked method for AGVs to use when navigating locally and overcoming obstacles. Analyzing and selecting the appropriate motion commands allows real-time navigation in changing environments. The DWA calculates a dynamic window to estimate linear and angular velocities based on the AGV's physical limits and current state. The AGV trajectory is simulated for each conceivable velocity generated inside this window for a short time [55–57]. A scoring system that balances safety and efficiency evaluates these trajectories, considering target proximity and obstacle avoidance variables. The trajectory with the highest rating is chosen by the velocity that best achieves the goal without colliding. The operation is repeated once the AGV moves at the set velocity [58]. Rapid velocity optimization ensures safe and agile navigation in complex and dynamic locations in the DWA.

The initial nodes on the DWA model result in "Calculate Dynamic Window." A time step, the vehicle's present condition, and limitations are used to calculate a range of possible velocities. "Generate

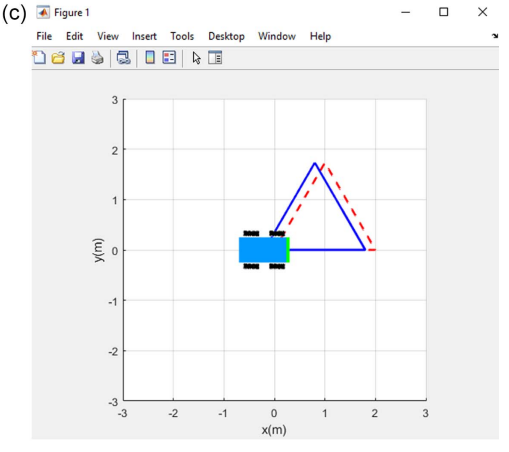
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Circular Trajectory.



Elliptical Trajectory.

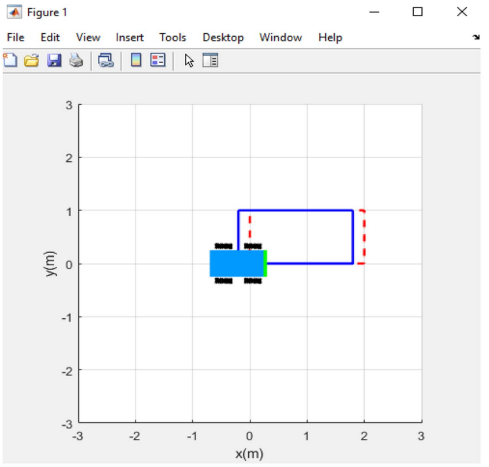


Triangular Trajectory.

Figure 3. *AGV testing simulations based on the equations: a) circular trajectory, b) elliptical trajectory, c) triangular trajectory, d) tetragonal trajectory, e) lemniscate trajectory, and f) five-petal rose curve.*

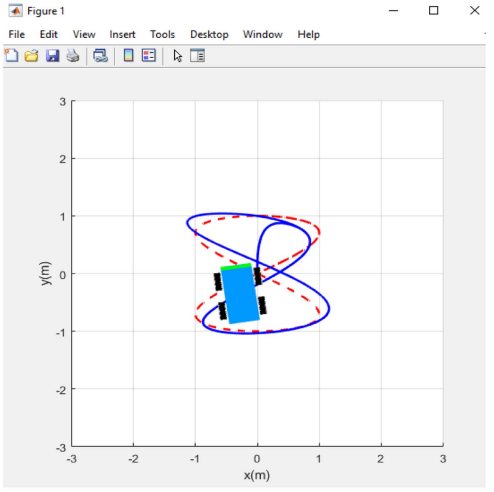
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

(d)



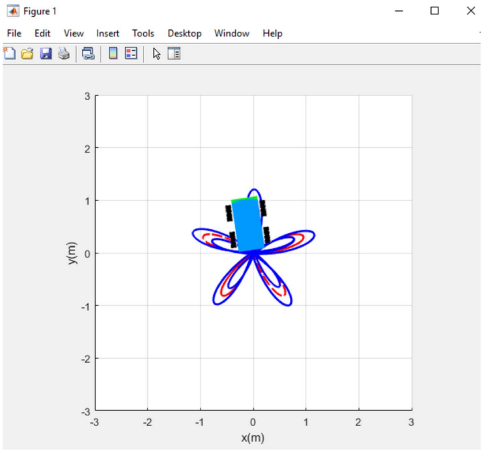
Tetragonal Trajectory.

(e)



Lemniscate Trajectory.

(f)



Five-Petal Rose Curve.

Figure 3. *Continued.*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

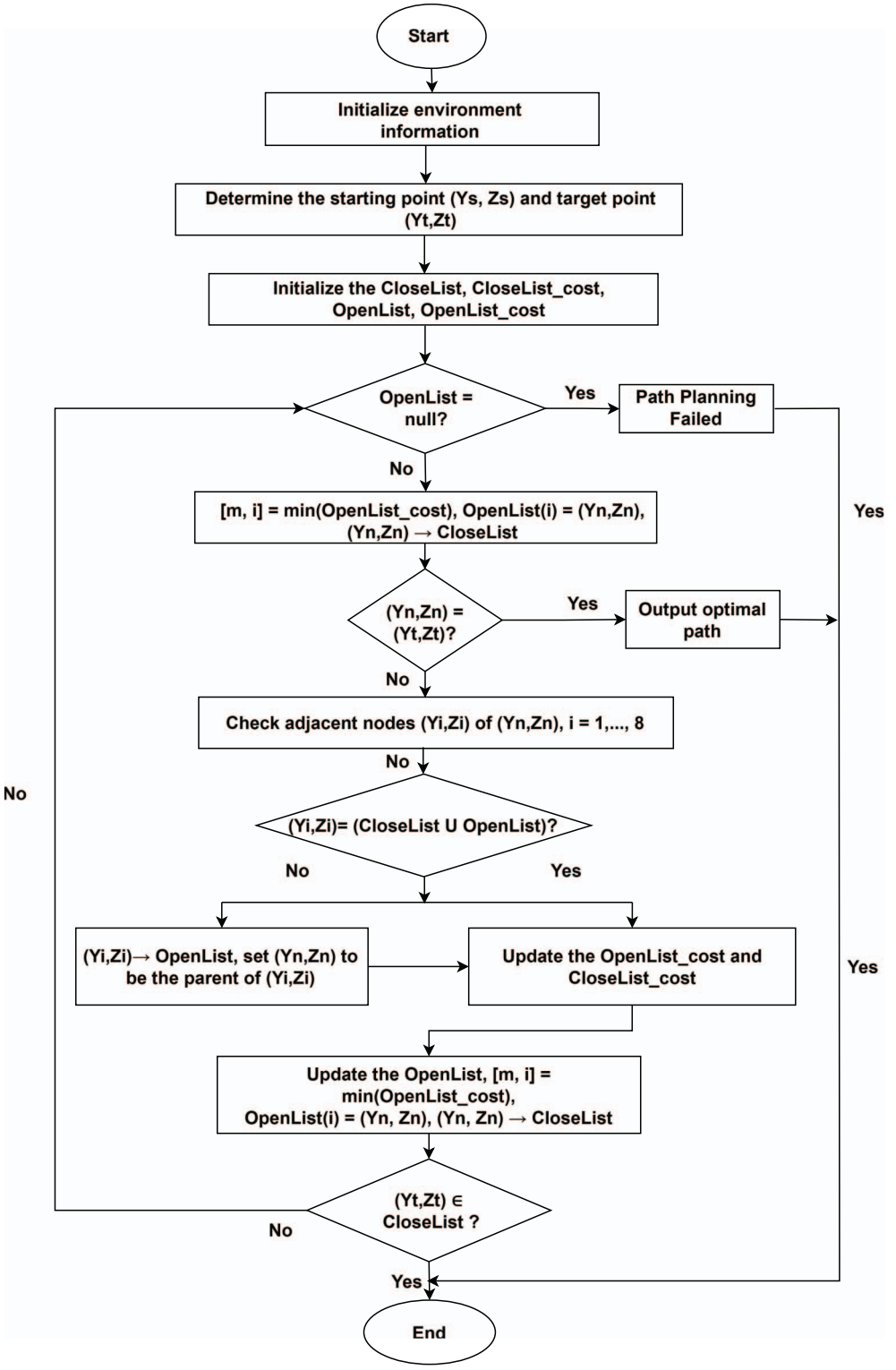


Figure 4. *An illustration of the MA* algorithm.*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

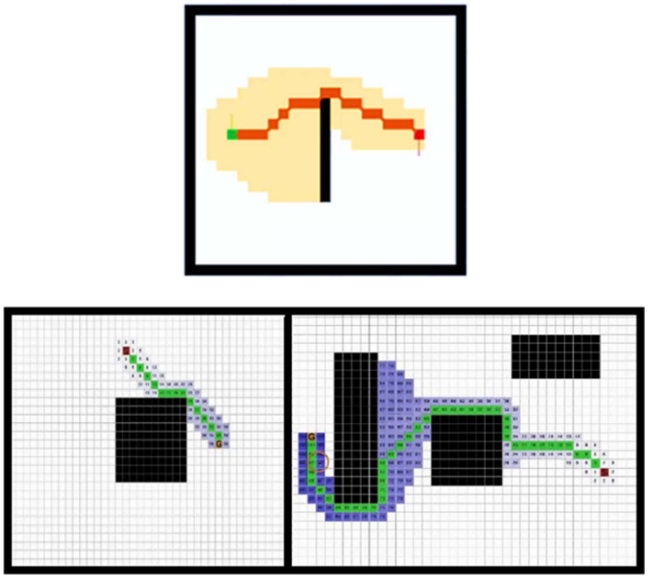


Figure 5. *Simulation results of the MA* algorithm.*

Possible Velocities" generates a list of potential velocities inside the dynamic window. The approach creates candidate velocities before initializing the best velocity to set optimal velocity and score starting points [59]. The program simulates the trajectory and predicts the AGV's path for each candidate velocity and time step. After trajectory simulation, the algorithm scores each trajectory based on how successfully it avoids obstacles and approaches the goal [58]. The system evaluates whether the current score surpasses the best-recorded score at each decision point. If it does, the best velocity and score are updated accordingly; otherwise, the system assesses the next candidate's velocity. After evaluating all candidate velocities, the algorithm returns the best velocity and reaches the end node, completing DWA [60]. The flowchart in Figure 6 shows how DWA chooses an AGV's ideal velocity based on its dynamic environment and goal. Figure 7 shows the MATLAB simulation results of the DWA algorithm.

4.3. Modified A-DWA hybrid optimization algorithm*

The modified A*-DWA hybrid optimization algorithm (ADWA-HO) algorithm, which avoids local barriers in real-time while utilizing global routing, is one of the highly successful control methods for mecanum wheel AGVs. Using the AGV's omnidirectional movement capabilities to their fullest, A* and DWA algorithms enable efficient, safe, and smooth navigation in complex and dynamic settings. The ADWA-HO methodology has several benefits over other algorithms.

By employing a MA* algorithm that eliminates unnecessary nodes and smooths trajectories, the ADWA-HO algorithm provides an optimized approach to global path planning, generating smoother paths than D* [61]. To ensure a more adaptable response to dynamic situations, ADWA-HO integrates DWA for real-time obstacle avoidance, rather than relying on APF or alternative algorithms that are susceptible to local minima [62]. ADWA-HO balances global efficiency and local responsiveness, unlike DWA, which merely focuses on local obstacle avoidance [63]. Compared to PRM & RRT, which have difficulty making real-time adjustments in highly variable environments, its flexible path refinement achieved via the combination of MA* and DWA offers better adaptability [64]. In contrast to D* and PRM & DWA fusion, which frequently generate jerky pathways, the algorithm improves smoothness and stability by preventing sudden directional shifts, making it more suitable for mecanum wheel AGVs [64]. Furthermore, in complex situations where PRM & RRT fusion, as well as PRM & DWA fusion, were unable to handle dynamic changes successfully, ADWA-HO exhibits higher flexibility [64]. Compared

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

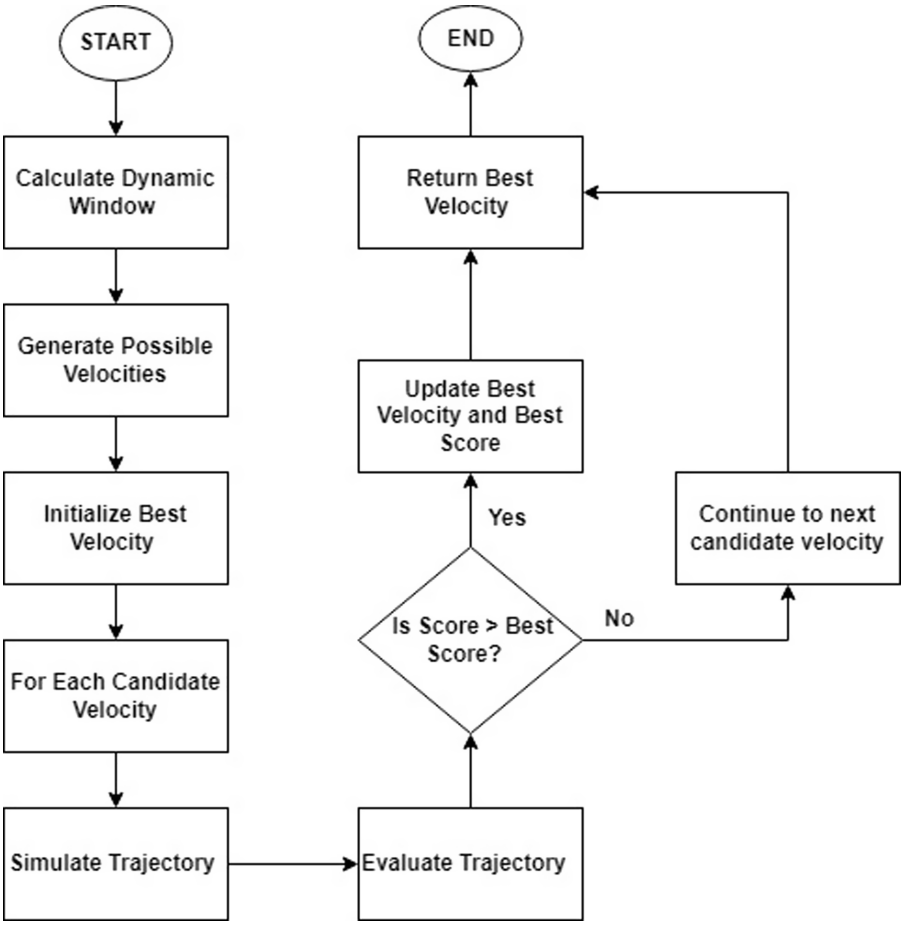


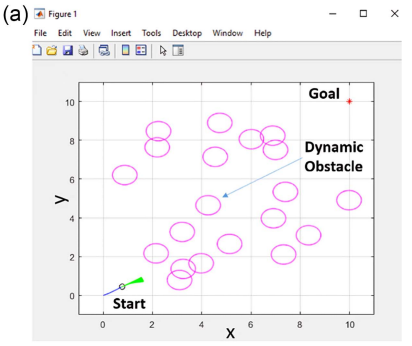
Figure 6. *Flowchart for DWA.*

to techniques like D*, APF, DWA, PRM & RRT fusion, as well as PRM & DWA fusion, ADWA-HO guarantees more accurate and effective navigation by harnessing the full omnidirectional capability of mecanum wheels. The algorithm provides durable and trustworthy solutions for comprehensive AGV control systems while avoiding common failures, such as RRT's inefficiency in high-dimensional environments and APF's local minima issues [65]. The MA* technique avoids fixed barriers and finds a relatively straightforward route from the starting point to the end point for global route planning. The DWA optimizes local routes and dynamically adjusts the AGV's trajectory in real time to steer clear of shifting obstructions and follow kinematic limits [66]. MA* produces a globally optimal strategy, while DWA ensures obstacle avoidance and local adaptability. If the DWA hits a significant barrier not in the initial MA* path, the system replans the global path from the AGV's location [67]. This hybrid technique gives the AGV the durability and flexibility to navigate complex environments. Figure 8 illustrates the flowchart of the hybrid algorithm.

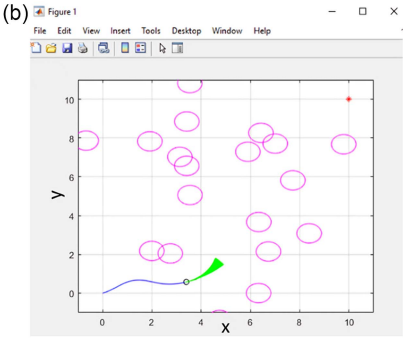
The ADWA-HO combines global path planning with local trajectory optimization for AGVs. The process begins with the initialization of the AGV's state, including its position (x_0,y_0), velocity $v_{0,}$ and heading θ_0, along with the current node position (x_n,y_n), and the goal position (x_{goal},y_{goal}). The key parameters, such as maximum velocity (v_{max}), maximum acceleration (a_{max}), and heading change limits, are set, and the heuristic for the A* algorithm is defined as the Euclidean distance:

$$h\,(n) = \sqrt{\left(x_{\text{goal}} - x_n\right)^2 + \left(y_{\text{goal}} - y_n\right)^2} \tag{4.3a}$$

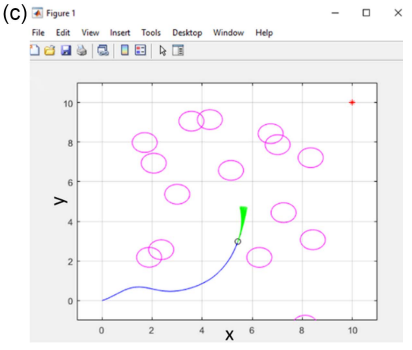
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



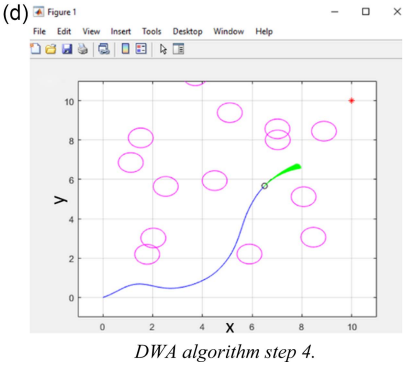
DWA algorithm step 1.



DWA algorithm step 2.



DWA algorithm step 3.



DWA algorithm step 4.

Figure 7. *MATLAB simulation results of the DWA algorithm in a 10 by 10 (in decimeters) raster map.*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

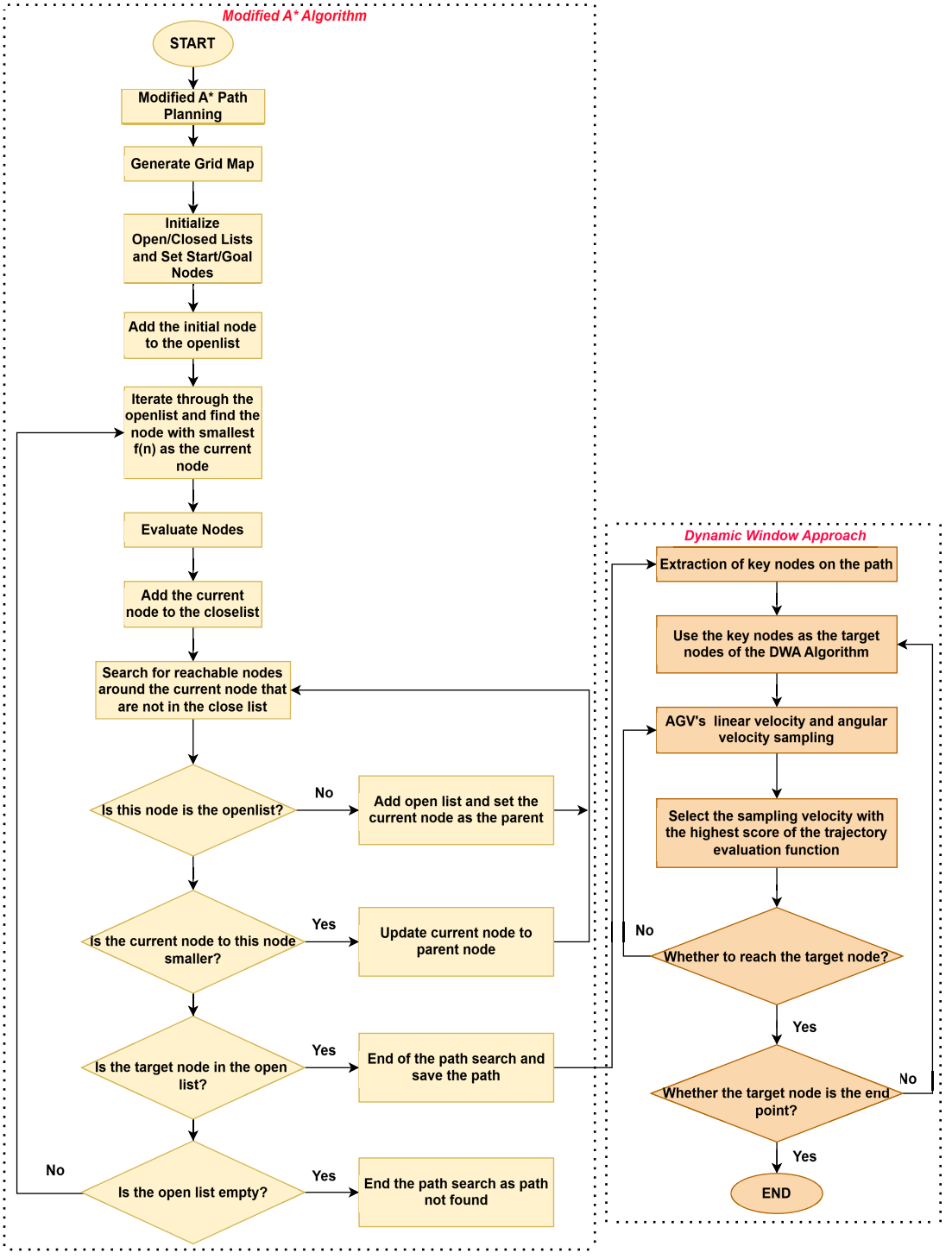


Figure 8. *Modified A* and DWA hybrid optimization algorithm (ADWA-HO).*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

The MA* algorithm then utilizes open and closed lists to determine the optimal route from the start to the end, where the total cost, f(n) = g(n) + h (n), is minimized. Algorithmically, the node with the smallest "f(n)" is selected from the open list and expanded by assessing its neighbors. For each neighbor, the tentative cost "$g_{tentative}$" is calculated as in ref. [16]:

$$g_{\text{tentative}} = g\ (\text{current}) + \text{cost}\ (\text{current, neighbor}) \tag{4.3b}$$

Here, g(current) is the cost from the start node to the current node, and cost (current, neighbor) is the cost to move from the current node to the neighbor. If "$g_{tentative}$" is less than "g(neighbor)," update the neighbor's cost and include it in the open list. Parent pointers are used to trace backward from the target to the beginning or start node once the path has reached the goal node. Once the goal is accomplished, parent pointers are traced to recreate the path. The A* path is divided into intermediate waypoints, which serve as sub-goals for the DWA. The DWA optimizes the local trajectory by calculating a dynamic window of feasible velocities $V_d = [v_{min}, v_{max}]$ and headings $\omega_d = [\omega_{min}, \omega_{max}]$ as in ref. [68], where

$$\left.\begin{aligned} v_{\min} &= v_{\text{current}} - a_{\max} \times \Delta t \\ v_{\max} &= v_{\text{current}} + a_{\max} \times \Delta t \\ \omega_{\min} &= \omega_{\text{current}} - \alpha_{\max} \times \Delta t \\ \omega_{\max} &= \omega_{\text{current}} + \alpha_{\max} \times \Delta t \end{aligned}\right\} \tag{4.3c}$$

The minimum and maximum feasible linear velocities are denoted by v_{min} and v_{max}. The $v_{current}$ is the current linear velocity of the AGV and a_{max} is the maximum linear acceleration/deceleration of the AGV. The term Δt is the time step over which the velocity change is computed. These limits ensure that the AGV's velocity changes are within its physical capabilities. The minimum and maximum feasible angular velocities are denoted by ω_{min} and ω_{max}. The $\omega_{current}$ is the current angular velocity of the AGV. The α_{max} is the maximum angular acceleration/deceleration of the AGV. For each pair of velocity v and angular velocity ω, the trajectory is predicted over a short time horizon T as in ref. [69]:

$$\left.\begin{aligned} x\ (t) &= x_0 + \int_0^T v \cdot \cos(\theta\ (t))\ dt \\ y\ (t) &= y_0 + \int_0^T v \cdot \sin(\theta\ (t))\ dt \\ \theta\ (t) &= \theta_0 + \int_0^T \omega dt \end{aligned}\right\} \tag{4.3d}$$

The $x(t)$ and $y(t)$ are the predicted positions of the AGV at time t; x_0 and y_0 are the initial positions of the AGV at time $t = 0$; v is the linear velocity of the AGV; $\theta(t)$ is the heading angle of the AGV at time t; and $\cos(\theta(t))$ and $\sin(\theta(t))$ are the directional components of the velocity. These equations predict the AGV's position over time based on its velocity and heading. θ_0 is the initial heading angle of the AGV at time $t = 0$; the AGV's angular velocity is denoted by ω. This equation predicts the AGV's heading over time based on its angular velocity. These limits ensure that the AGV's heading changes are within its physical capabilities. Given the AGV's present velocity (v) and angular velocity (ω), these equations forecast its future trajectory (position and heading) over a brief time horizon (T). The trajectory is evaluated using a cost function that includes goal-oriented cost (g_{cost}), obstacle avoidance cost (o_{cost}), and speed cost (v_{cost}), as in refs. [70,71]:

$$g_{\text{cost}} = \text{distance}\ (x\ (T)\ , y\ (T)\ , \text{waypoint}) \tag{4.3e}$$

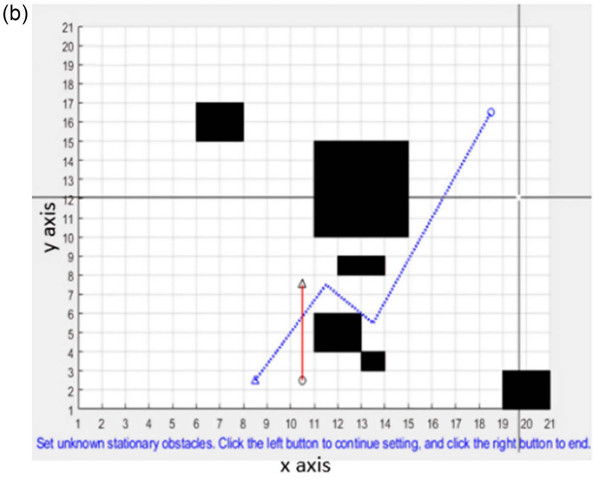
Here, g_{cost} is the cost associated with reaching the goal or sub-goal (waypoint); $x(T)$ and $y(T)$ are the expected location of the time T for AGV based on the current velocity and heading; the A* algorithm's global path is used to determine the waypoint, which is the intermediate target point (sub-goal); distance ($x(T)$, $y(T)$, waypoint) is the distance measured in Euclidean terms between the waypoint and the expected position ($x(T)$ and $y(T)$). This cost encourages the AGV to move toward the goal or sub-goal. A smaller value suggests that the predicted trajectory brings the AGV closer to the target.

$$o_{\text{cost}} = \sum_{\text{obstacles}} \frac{1}{\text{distance}\ \ (x\ (t)\ , y\ (t)\ , \text{obstacle}} \tag{4.3f}$$

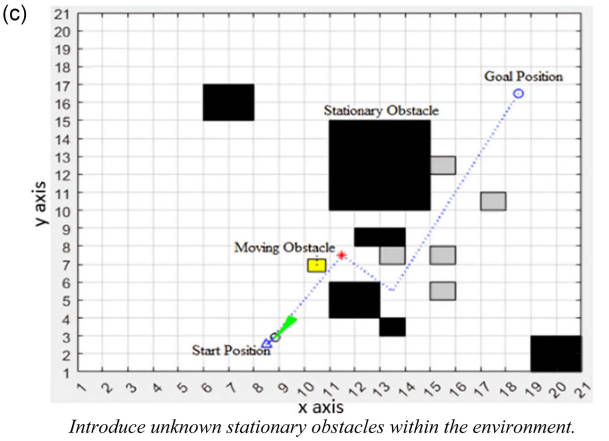
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Define the map, including the start position and the goal location.



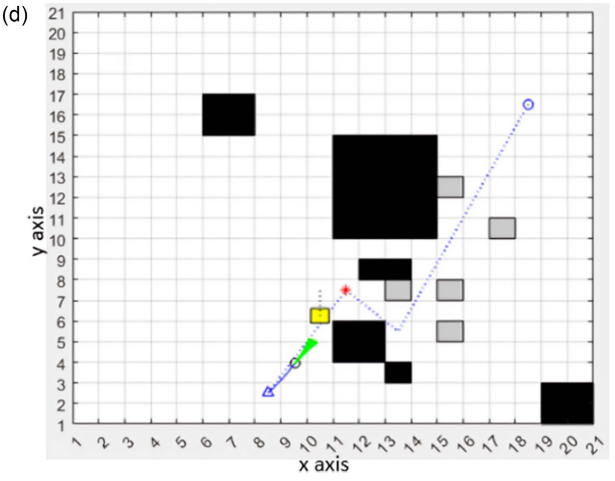
Specify the start and end positions of the moving obstacle.



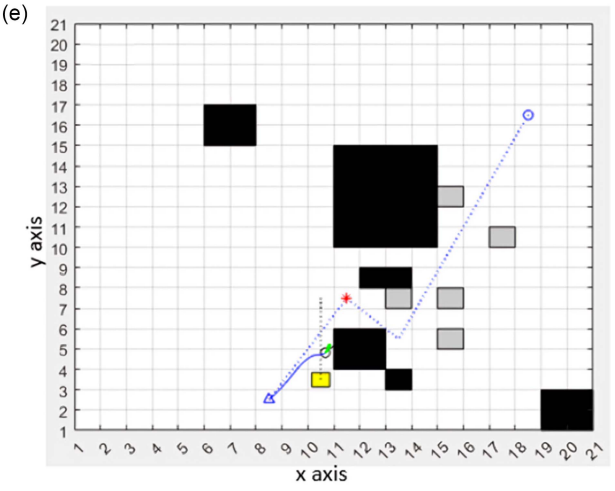
Introduce unknown stationary obstacles within the environment.

Figure 9. *MATLAB simulation testing results of the ADWA-HO algorithm in 21 × 21 raster Map 1 (in decimeters).*

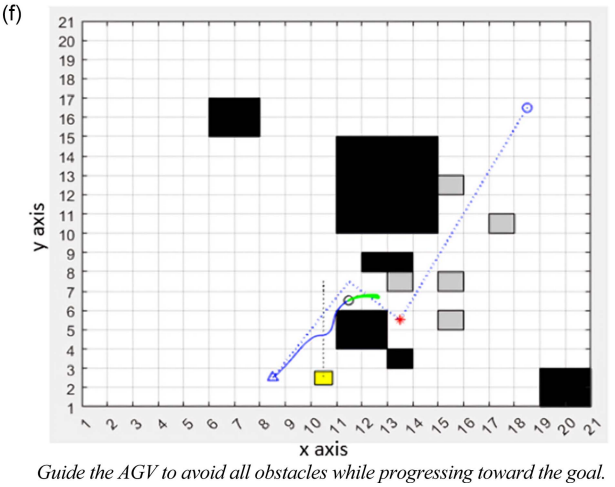
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Ensure the AGV detects and avoids dynamic obstacles while navigating.



Enable the AGV to identify and avoid stationary obstacles.



Guide the AGV to avoid all obstacles while progressing toward the goal.

Figure 9. *Continued.*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

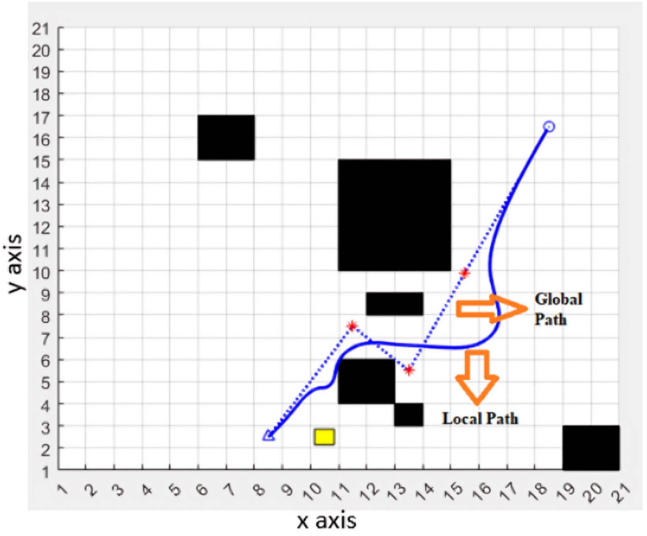


Figure 10. *Global and local path planning of simulation in a 21 × 21 raster Map 1 (in decimeters).*

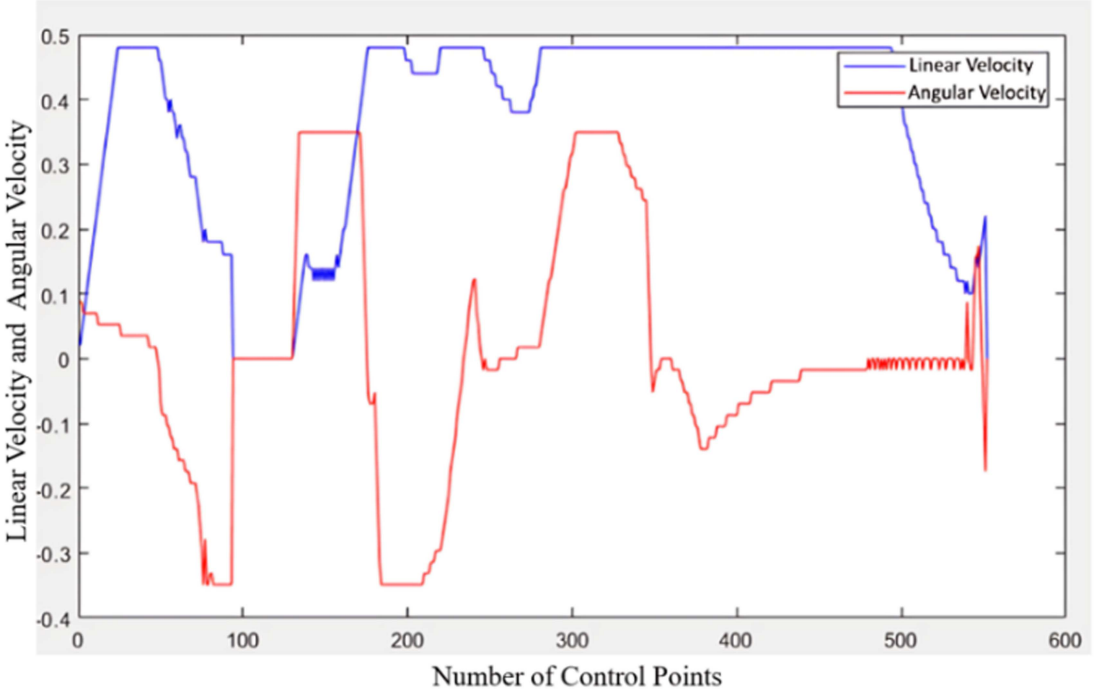


Figure 11. *Variation in linear velocity (m/s) and angular velocity (rad/s) of the simulation in raster Map 1.*

The term o_{cost} represents the cost associated with avoiding obstacles; the projected positions of the AGV at time t along the trajectory are denoted by $x(t)$ and $y(t)$; the location of an obstacle in the surroundings is referred to as an obstacle; the Euclidean distance between the obstacle and the predicted position $(x(t),y(t))$ is represented by the expression distance $(x(t), y(t),$ obstacle); and $\sum$obstacles denotes the

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

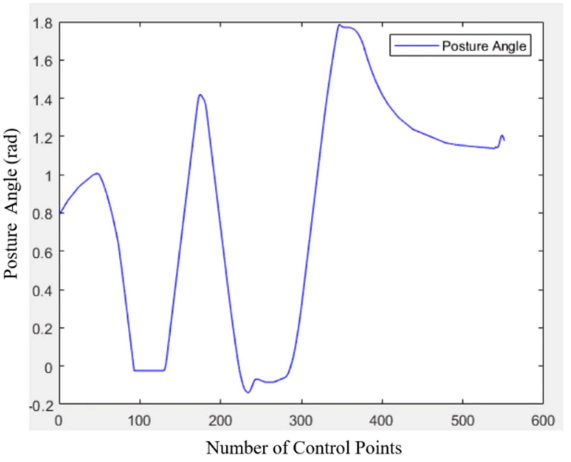


Fig. 12. *Variation in posture angle (rad) of simulation in raster Map 1.*

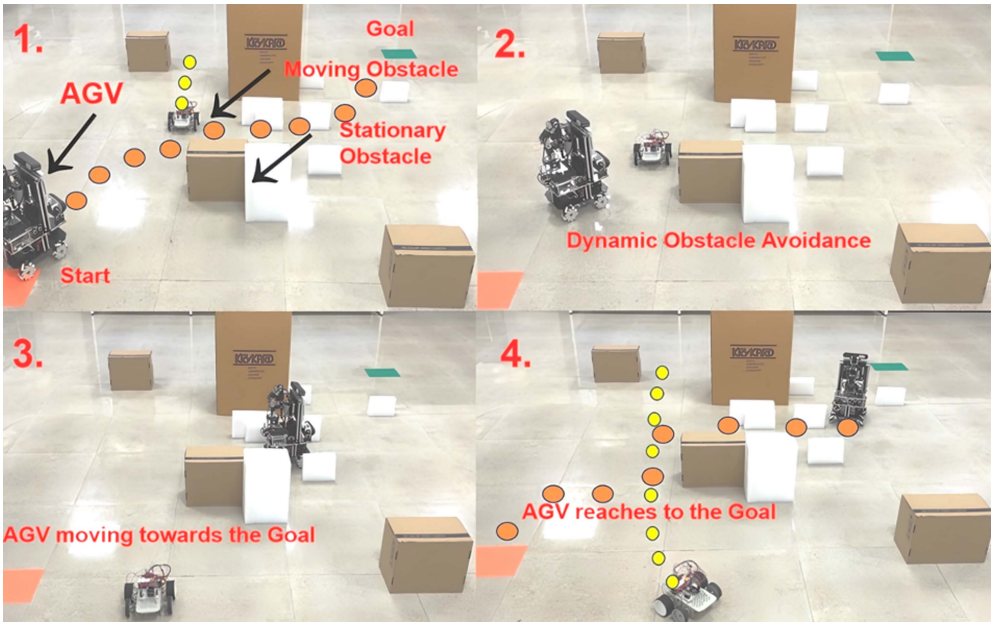


Figure 13. *Real-time testing environment of 21 × 21 Map 1 (in decimeters), that is, 210 cm × 210 cm.*

sum is taken over all obstacles in the environment. This cost penalizes trajectories that approach obstacles too closely. The inverse distance ensures that trajectories passing near obstacles have a higher cost, while those farther away have a lower price.

$$v_{\text{cost}} = v_{\max} - v \tag{4.3g}$$

v_{cost} represents the cost associated with the AGV's speed, v_{max} is the maximum allowable velocity for the AGV, and v is the current velocity of the AGV along the trajectory. This cost encourages the AGV to move at higher velocities. A lower value indicates that the AGV is moving closer to its maximum speed, which is desirable for efficiency.

The total cost is computed as:

$$\text{Total Cost } \lambda_1 \cdot g_{\text{cost}} + \lambda_2 \cdot 0_{\text{cost}} + \lambda_3 \cdot v_{\text{cost}} \tag{4.3h}$$

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

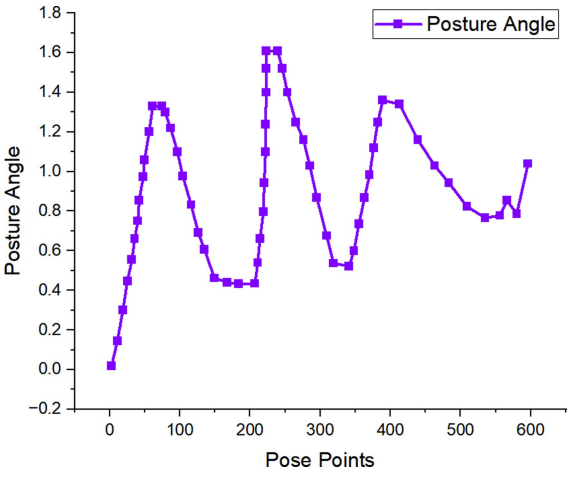


Figure 14. *Posture angle (rad) variation in Map 1.*

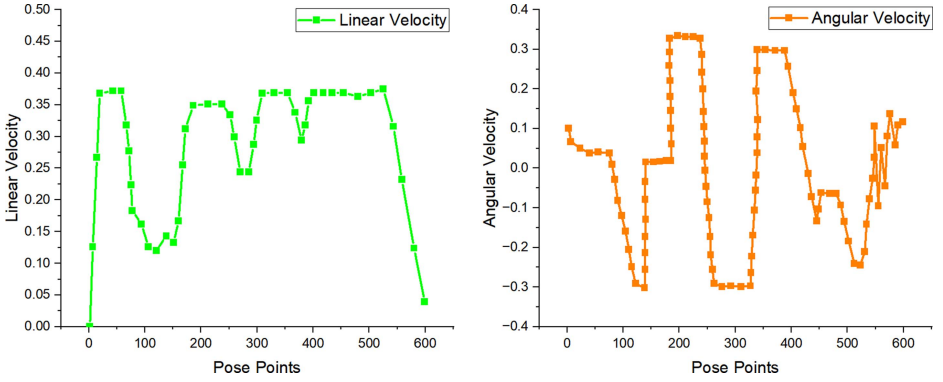


Figure 15. *The variation in linear velocity (m/s) and angular velocity (rad/s) in Map 1.*

where λ_1, λ_2, and λ_3 are weighting factors for the goal-oriented, obstacle avoidance, and speed costs, respectively. The AGV advances to the next waypoint after selecting the trajectory with the lowest overall cost. The total cost function balances the three objectives, that is, reaching the goal, avoiding obstacles, and maintaining high speed. The weights λ_1, λ_2, and λ_3 allow the algorithm to be tuned to prioritize specific behaviors (e.g., safety over speed or vice versa). This procedure is repeated for every waypoint until the objective is accomplished or there are no more feasible routes. The A*-DWA hybrid optimization algorithm leverages the global optimality of A* and the dynamic adaptability of DWA, facilitating safe and effective navigation in complex environments. The results of the ADWA-HO algorithm's simulation in Map 1 are displayed in Figures 9, 10, 11, and 12, while the outcomes of the experiments are shown in Figures 13, 14, and 15. Tables III–VI display the results of the simulations and experiments, respectively.

D* works the least well. At the same time, ADWA-HO surpasses other algorithms, achieving the quickest motion time (17.75 s in simulation and 18.47 s in experiments) and shortest route (192.46 cm in simulation and 202.87 cm in experiments). In addition, ADWA-HO provides the best motion smoothness (84%), the best success rate (95%), and the smallest collision rate (3%). It also has fewer turns (8) and path inflections (2) than D*, which has more turns (4 inflections, 25 turns). These outcomes show ADWA-HO's exceptional reliability and effectiveness in autonomously navigating. The most efficient and reliable algorithm is ADWA-HO, which reduces path inflections and turns while achieving

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

Table III. *Simulation and experimental performance of each algorithm in Map 1.*

AGV matrix in Map 1	Dynamic A* (D*)	APF	DWA	PRM and RRT	PRM and DWA	ADWA-HO
Simulation path length (cm)	258.76	246.15	240.88	230.14	214.05	192.46
Simulation motion time (s)	23.84	21.94	21.36	20.32	19.28	17.75
Experimental path length (cm)	284.29	264.96	258.87	246.63	227.84	202.87
Experimental motion time (s)	25.54	23.46	22.67	21.55	20.33	18.47
RMS error for path length	25.5	18.81	17.97	16.49	13.78	10.41
RMS error for motion time	1.70	1.52	1.31	1.23	1.05	0.72
Rate of collision	10%	9%	8%	6%	5%	3%
Motion smoothness	58%	62%	67%	71%	77%	84%
Rate of reaching goal	72%	75%	81%	85%	89%	95%
Path inflection	4	4	3	3	2	2
Number of turns	25	21	16	12	10	8
Computational load	Low	Low	Moderate	High	Moderate	Moderate

Table IV. *Percentage deviation between experimental and simulated path lengths in Map 1.*

AGV matrix in Map 1	Dynamic A* (D*)	APF	DWA	PRM and RRT	PRM and DWA	ADWA-HO
Simulation path length (cm)	258.76	246.15	240.88	230.14	214.05	192.46
Experimental path length (cm)	284.29	264.96	258.87	246.63	227.84	202.87
Standard deviation	12.75	9.41	9.01	8.25	6.90	5.21
% Deviation in path length	9.86	7.64	7.46	7.16	6.44	5.40

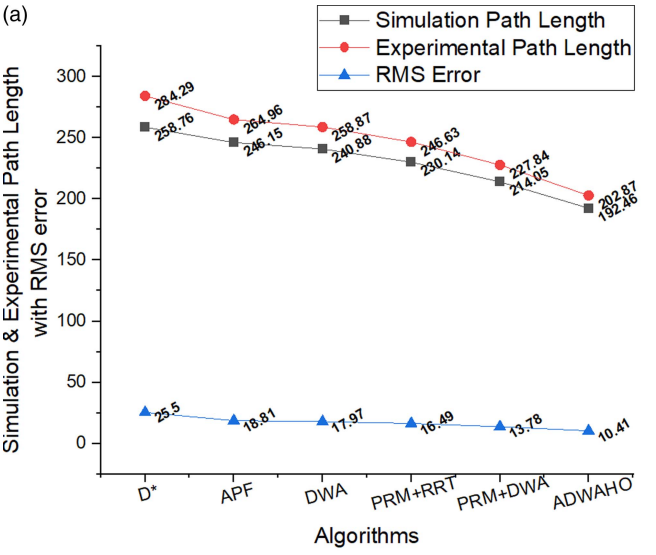
Table V. *Percentage deviation between experimental and simulated motion time in Map 1.*

AGV matrix in Map 1	Dynamic A* (D*)	APF	DWA	PRM and RRT	PRM and DWA	ADWA-HO
Simulation Motion time (s)	23.84	21.94	21.36	20.32	19.28	17.75
Experimental motion time (s)	25.54	23.46	22.67	21.55	20.33	18.47
Standard deviation	0.85	0.76	0.65	0.61	0.52	0.36
% Deviation in motion time	7.13	6.92	6.13	6.05	5.44	4.06

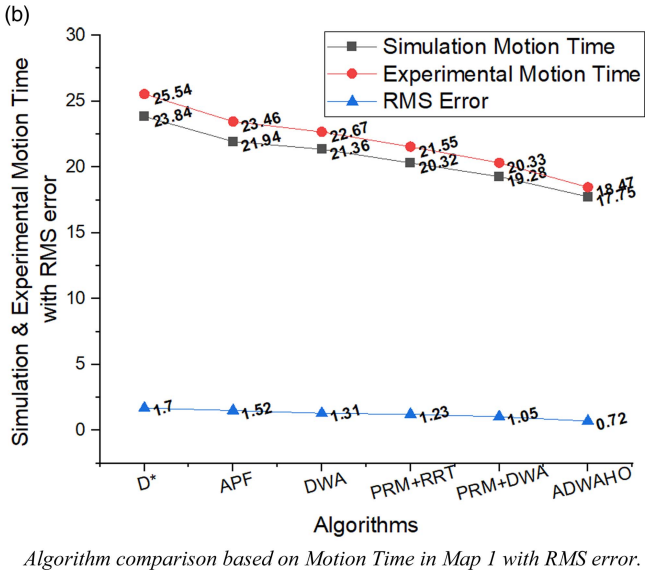
Table VI. *Percentage difference in experimental path length and motion time between ADWA-HO and other algorithms in Map 1.*

AGV matrix in Map 1	D*-ADWAHO	APF-ADWAHO	DWA-ADWAHO	(PRM + RRT)-ADWAHO	(PRM + DWA)-ADWAHO
	%Change in path length				
Experimental path length (cm)	28.63	23.43	21.63	17.74	10.95
Experimental motion time (s)	27.68	21.27	18.52	14.29	9.15

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Algorithm comparison based on Path Length in Map 1 with RMS error.



Algorithm comparison based on Motion Time in Map 1 with RMS error.

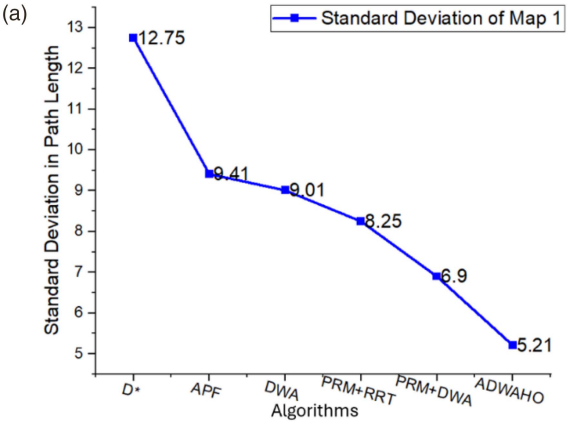
Figure 16. *Algorithms comparison based on path length and motion time in Map 1 with RMS error.*

outstanding outcomes in terms of path length, motion duration, collision rate, smoothness, and goal-reaching rate. On the other hand, D* performs the worst, exhibiting slower motion, longer paths, greater collisions, and a lower success rate. While other algorithms, such as APF, DWA, PRM & RRT fusion, and PRM & DWA fusion, perform effectively in reducing turns and inflections compared to D*, they are still moderately less effective than ADWA-HO.

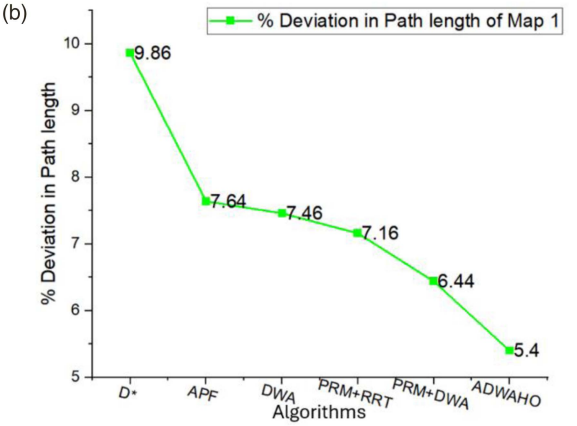
Figure 16 represents the algorithms comparison based on path length and motion time in map 1 with RMS error. Figures 17 and 18 illustrate the algorithm comparison based on standard deviation and percentage deviation in path length and motion time in Map 1.

The results of the ADWA-HO algorithm's simulation in Map 2 are presented in Figures 19, 20, 21, and 22, while Figures 23, 24, and 25 display the experimental results. Tables VII, VIII, IX, and X display the results of the simulations and experiments, respectively.

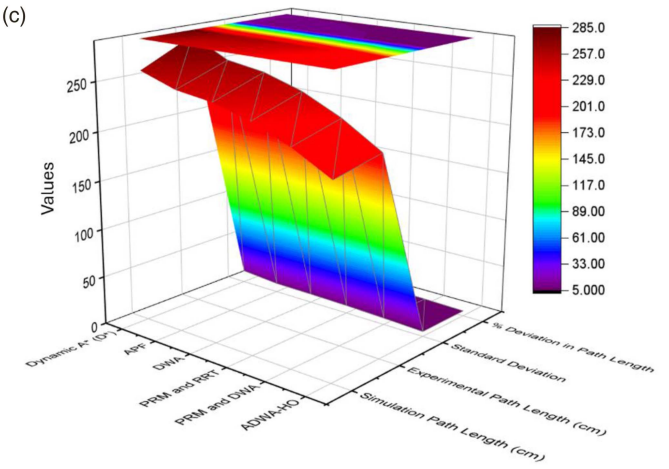
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Standard Deviation in path length of map 1.



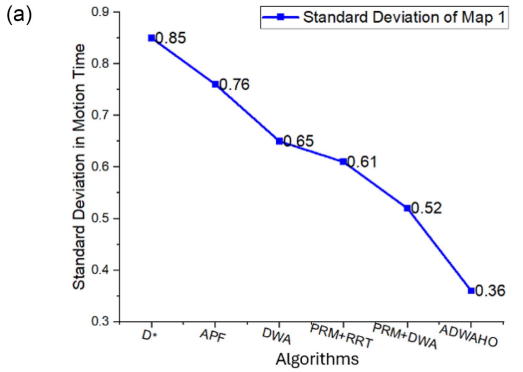
Percentage deviation in the path length of map 1.



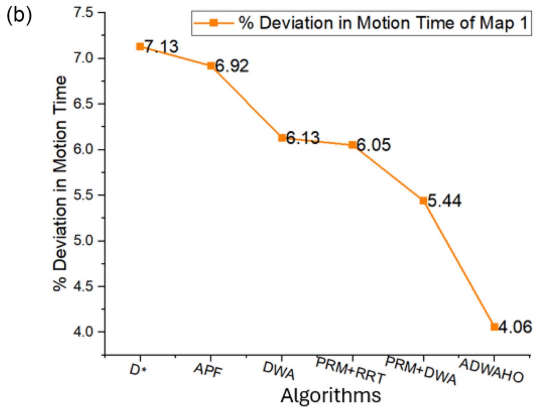
3D Visualization of Simulation Path Length, Experimental Path Length, Standard Deviation, and % Deviation for Different Path Planning Algorithms.

Figure 17. *Algorithms comparison according to standard deviation and % deviation in path length in Map 1.*

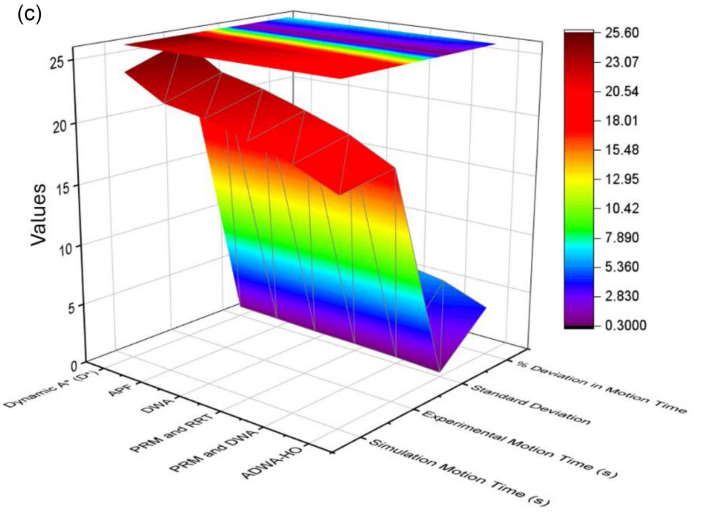
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



a) Standard Deviation in the motion time of map 1.



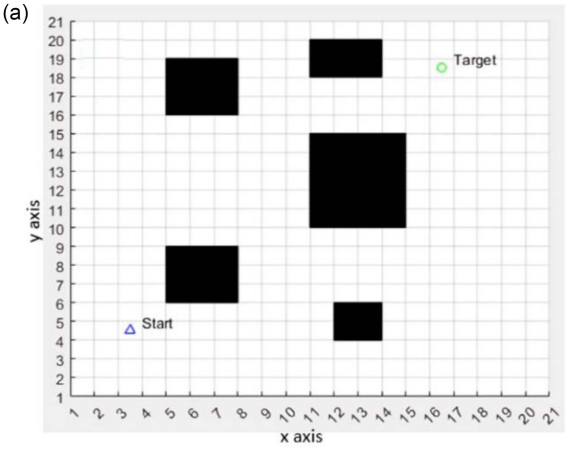
b) Percentage deviation in the motion time of map 1.



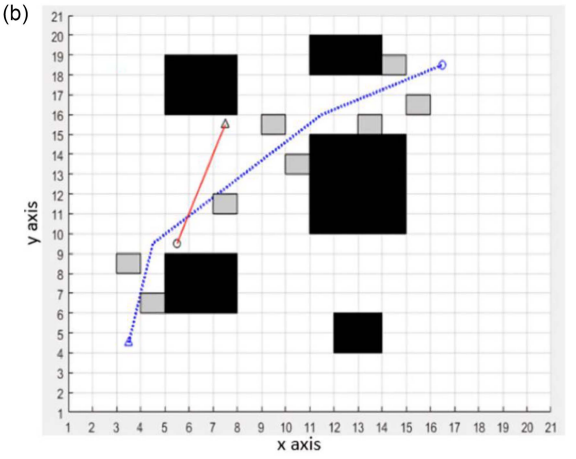
3D Visualization of Simulation Motion Time, Experimental Motion Time, Standard Deviation, and % Deviation for Different Path Planning Algorithms.

Figure 18. *Comparison of algorithms according to standard deviation and % deviation in motion time in Map 1.*

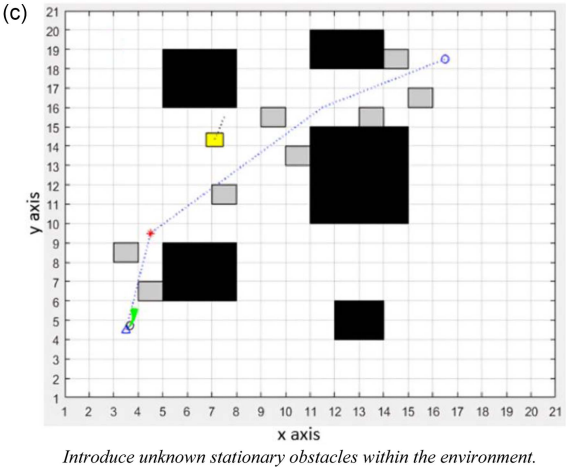
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Define the map, including the start position and the goal location.



Specify the start and end positions of the moving obstacle.

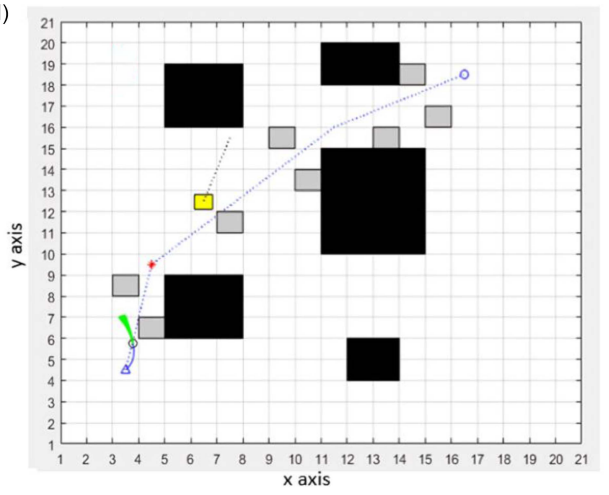


Introduce unknown stationary obstacles within the environment.

Figure 19. *MATLAB simulation testing results of the ADWA-HO algorithm in 21 × 21 raster Map 2 (in decimeters).*

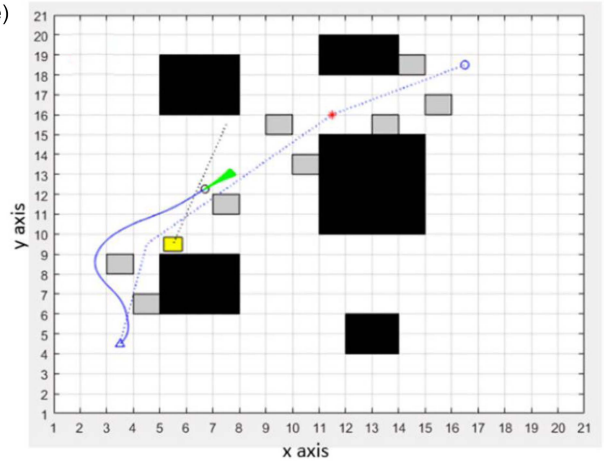
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

(d)



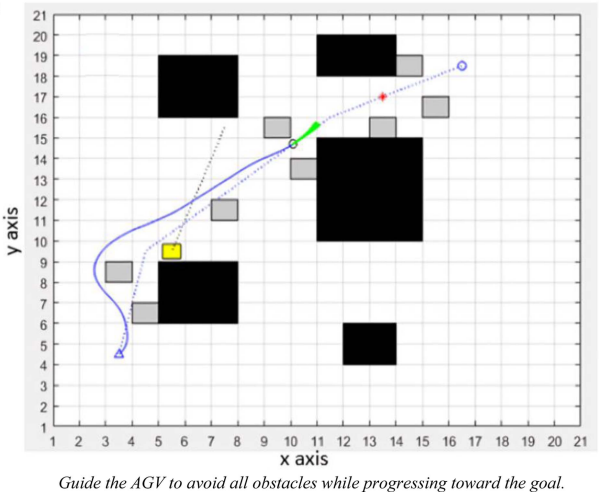
Ensure the AGV detects and avoids dynamic obstacles while navigating.

(e)



Enable the AGV to identify and avoid stationary obstacles.

(f)



Guide the AGV to avoid all obstacles while progressing toward the goal.

Figure 19. *Continued.*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

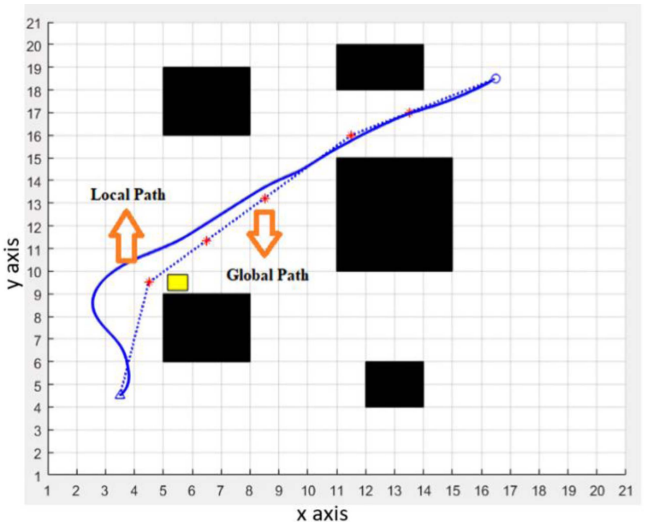


Figure 20. *Global and local path planning of simulation in 21 × 21 raster Map 2 (in decimeters).*

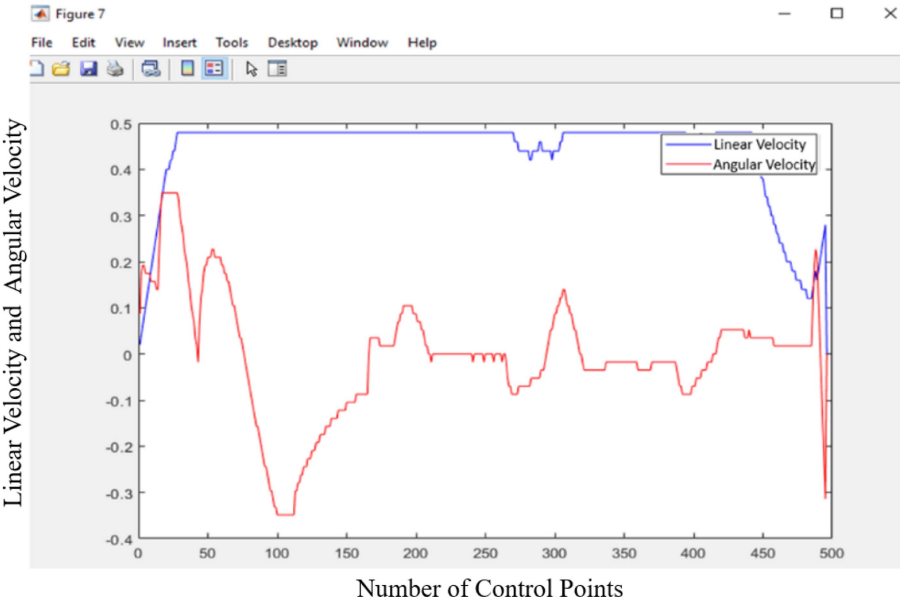


Figure 21. *Variation in linear velocity (m/s) and angular velocity (rad/s) of the simulation in raster Map 2.*

While D* shows the longest paths (269.87 cm in simulation and 295.39 cm in experiment) and the slowest times (33.88 s in simulation and 36.65 s in experiment), ADWA-HO executes better across each of the significant metrics, attaining the shortest route length (203.16 cm in simulation and 213.93 cm in experiments) and its quickest motion time (28.25 s in simulation and 29.58 s in experiments). D* has the largest collision rate (18%) and the lowest smoothness (55%), while ADWA-HO has the least collision rate (4%) and the finest motion smoothness (82%). ADWA-HO has the most successful goal-reach success rate (90%), and D* offers the lowest (69%). Furthermore, D* and APF have the greatest path inflections (5), while ADWA-HO and PRM-DWA fusion have the fewest (2). In addition, D* needs the most turns (27), whereas ADWA-HO needs the fewest (7). These outcomes demonstrate the

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

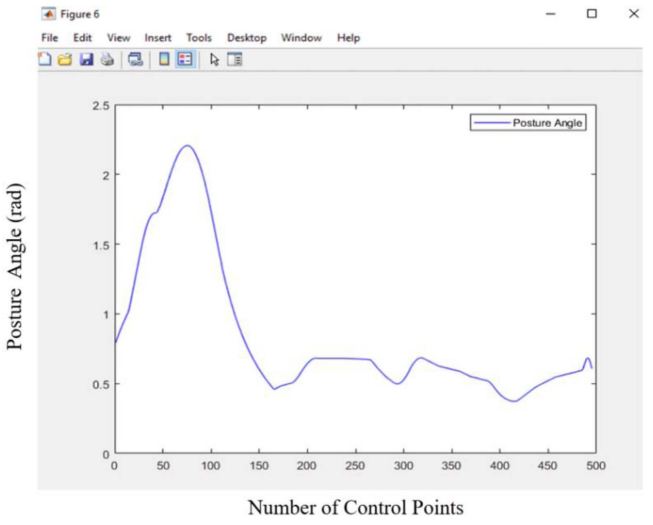


Figure 22. *Variation in posture angle (rad) of simulation in raster Map 2.*

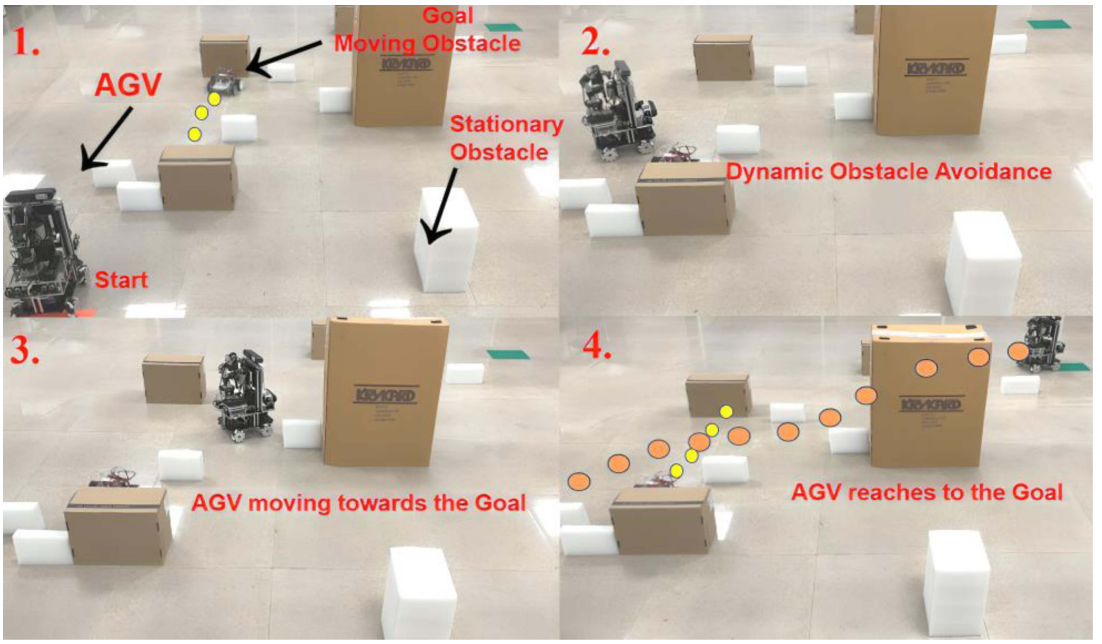


Figure 23. *Real-time testing environment of 21 × 21 Map 2 (in decimeters), that is, 210 cm × 210 cm.*

effectiveness, precision, and reliability of ADWA-HO in autonomous navigation. The most reliable algorithm is ADWA-HO, which minimizes path inflections and turns while achieving outstanding outcomes in terms of path length, motion time, collision rate, motion smoothness, and goal-reaching rate. D* performs the worst, exhibiting slower motion, longer pathways, more collisions, and a lower success rate. Both the fusion of PRM & RRT and PRM & DWA demonstrate strong performance, particularly in minimizing path inflections and reducing the number of turns. However, they remain less efficient compared to the ADWA-HO approach.

Figure 26 represents the algorithms comparison based on path length and motion time in map 2 with RMS error. Figures 27 and 28 illustrate the algorithm comparison based on standard deviation and percentage deviation in path length and motion time in Map 2.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

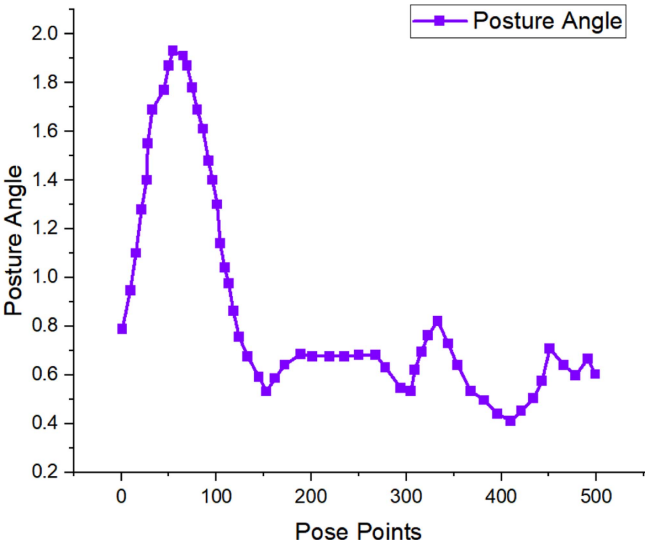


Figure 24. *Posture angle (rad) variation in Map 2.*

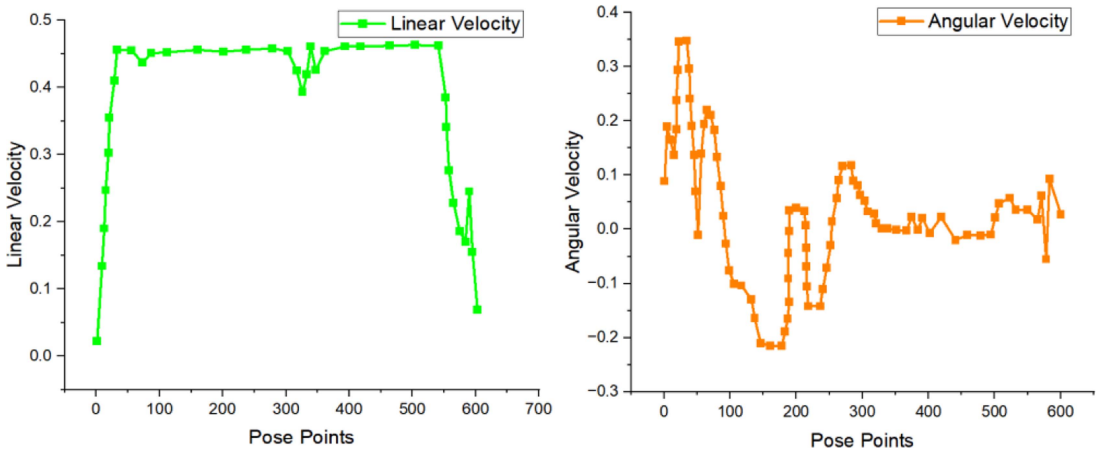


Figure 25. *Linear velocity (m/s) and angular velocity (rad/s) variation in Map 2.*

The average percentage variance in path length for Map 1 is presented in Table XI, while for Map 2, it is presented in Table XII. Table XIII compares ADWA-HO to other algorithms in Maps 1 and 2 in terms of experimental path length and percentage difference in motion time.

Figure 29 represents the percentage change in AGV algorithms, path length, and motion time in map 1 and map 2. Figure 30 represents the algorithm comparison based on average percentage change in map 1 and map 2. Comparing ADWA-HO to various algorithms (D*, APF, DWA, PRM & RRT fusion, and PRM & DWA fusion) on two distinct maps, Table XIII compares the path's length and motion period of time between ADWA-HO and other algorithms. The greatest improvement in percentage difference between ADWA-HO and other algorithms in experimental path length is seen in D* (28.63% in Map 1 and 27.57% in Map 2), while the least improvement is noted against PRM & DWA fusion (10.95% in Map 1 and 10.48% in Map 2). The greatest improvement in percentage difference between ADWA-HO and other algorithms in experimental motion time is seen in D* (27.68% in Map 1 and 19.29 % in Map 2), while the least improvement is noted against PRM & DWA fusion (9.15% in Map 1 and 5.91% in Map 2). ADWA-HO outperforms D*, APF, DWA, PRM & RRT, as well as PRM & DWA, by 28.10%,

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

Table VII. *Simulation and experimental performance of each algorithm in Map 2.*

AGV matrix in Map 2	Dynamic A* (D*)	APF	DWA	PRM and RRT	PRM and DWA	ADWA-HO
Simulation path length (cm)	269.87	257.26	252.88	242.09	226.16	203.16
Simulation motion time (s)	33.88	32.49	31.79	31.03	29.98	28.25
Experimental path length (cm)	295.39	275.97	269.78	257.64	238.98	213.93
Experimental motion time (s)	36.65	34.57	33.78	32.66	31.44	29.58
RMS error for path length	25.52	18.71	16.90	15.55	12.82	10.77
RMS error for motion time	2.77	2.08	1.98	1.63	1.46	1.33
Rate of collision	18%	15%	12%	9%	7%	4%
Motion smoothness	55%	58%	65%	69%	80%	82%
Rate of reaching goal	69%	70%	76%	79%	82%	90%
Path inflection	5	5	4	3	2	2
Number of turns	27	24	19	15	14	7
Computational load	Low	Low	Moderate	High	Moderate	Moderate

Table VIII. *Percentage deviation between experimental and simulated path lengths in Map 2.*

AGV matrix in Map 2	Dynamic A* (D*)	APF	DWA	PRM and RRT	PRM and DWA	ADWA-HO
Simulation path length (cm)	269.87	257.26	252.88	242.09	226.16	203.16
Experimental path length (cm)	295.39	275.97	269.78	257.64	238.98	213.93
Standard deviation	12.74	9.35	8.45	7.78	6.42	5.38
% Deviation in path length	9.45	7.27	6.68	6.42	5.66	5.30

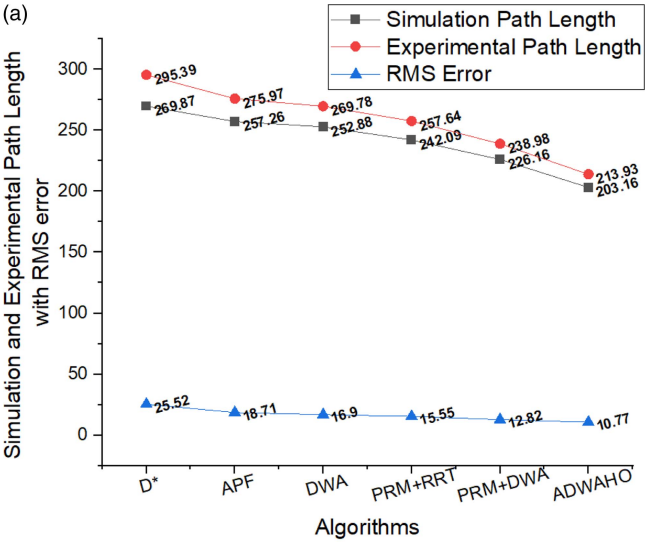
Table IX. *Percentage deviation between experimental and simulated motion time in Map 2.*

AGV matrix in Map 2	Dynamic A* (D*)	APF	DWA	PRM and RRT	PRM and DWA	ADWA-HO
Simulation motion time (s)	33.88	32.49	31.79	31.03	29.98	28.25
Experimental motion time (s)	36.65	34.57	33.78	32.66	31.44	29.58
Standard deviation	1.39	1.04	0.99	0.82	0.73	0.66
% Deviation in motion time	8.17	6.40	6.25	5.24	4.86	4.70

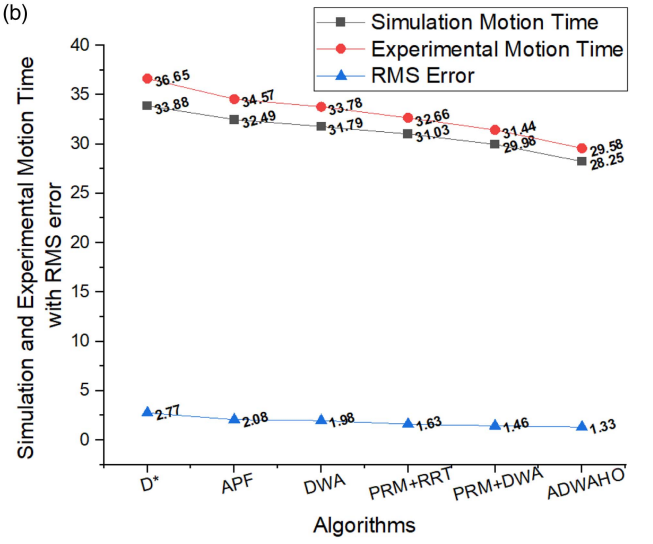
Table X. *Percentage difference in experimental path length and motion time between ADWA-HO and other algorithms in Map 2.*

AGV matrix in Map 2	D*-ADWAHO	APF-ADWAHO	DWA-ADWAHO	(PRM + RRT)-ADWAHO	(PRM + DWA)-ADWAHO
	%Change in path length				
Experimental path length (cm)	27.57	22.48	20.70	16.96	10.48
Experimental motion time (s)	19.29	14.43	12.43	9.43	5.91

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Algorithm comparison based on Path Length in Map 2 with RMS error.

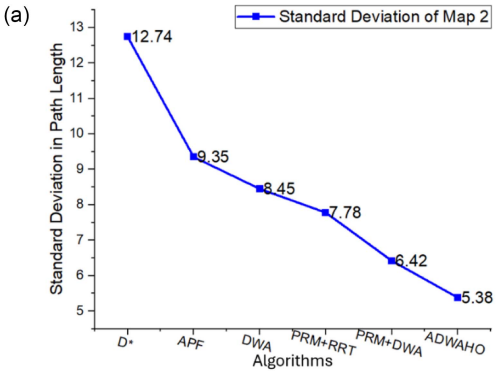


Algorithm comparison based on Motion Time in Map 2 with RMS error.

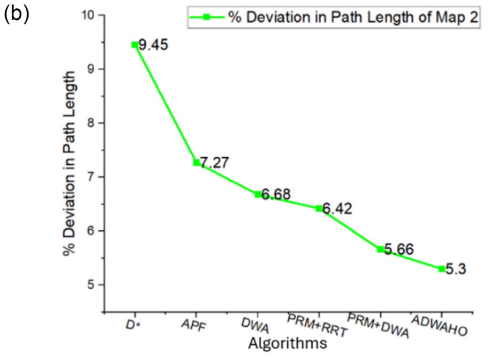
Figure 26. *Algorithms comparison based on path length and motion time in Map 2 with RMS error.*

22.95%, 21.16%, 17.35%, and 10.71% in the average path length of both maps. ADWA-HO outperforms D*, APF, DWA, PRM & RRT, and PRM & DWA by 23.48%, 17.85%, 15.47%, 11.86%, and 7.53% in the average motion time of both maps. These outcomes demonstrate how successfully ADWA-HO improved the variation in motion time and path length, making it a stronger and more well-suited navigation technique compared with alternative strategies. According to Tables III and VII, due to their simpler formulations, the D* and APF approaches require less computing power; however, they are more prone to path inflections and collisions, thereby reducing efficiency. The DWA optimizes short-range obstacle avoidance but not global path planning due to its moderate processing load from real-time velocity sample evaluation. Sample-based approaches, such as PRM and RRT, require more processing due to graph formation and random tree expansion, but they enhance path smoothness and collision avoidance.

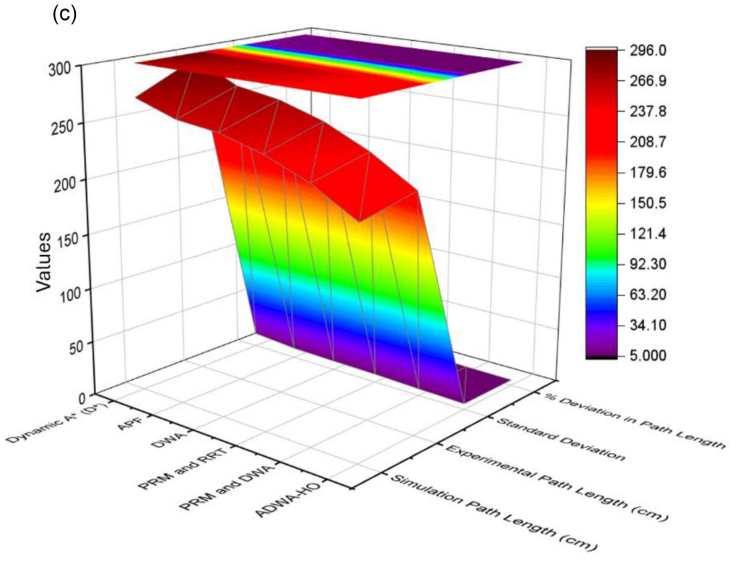
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Standard deviation in the path length of map 2.



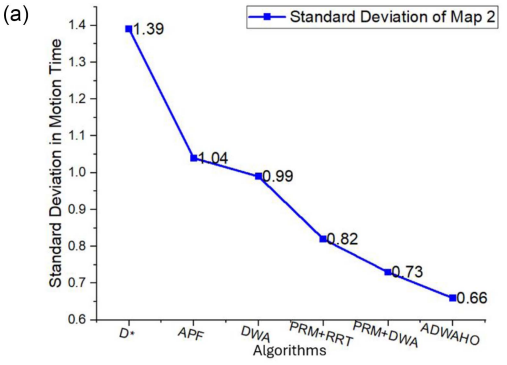
Percentage deviation in the path length of map 2.



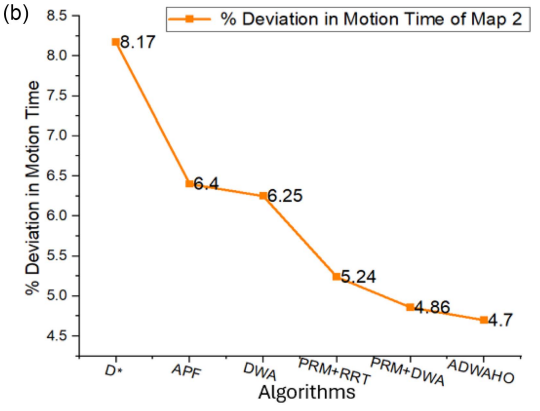
3D Visualization of Simulation Path Length, Experimental Path Length, Standard Deviation, and % Deviation for Different Path Planning Algorithms.

Figure 27. *Algorithms comparison according to standard deviation and % deviation in path length in Map 2.*

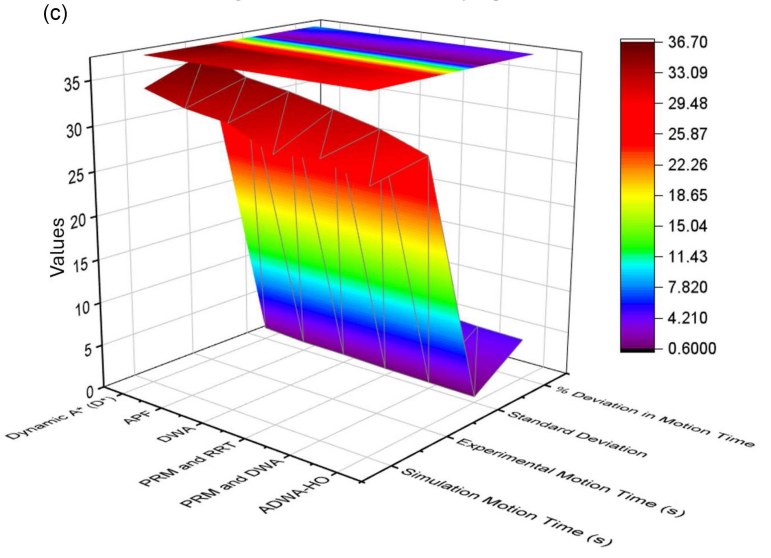
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Standard deviation in the motion time of map 2.



Percentage deviation in the motion time of map 2.



3D Visualization of Simulation Motion Time, Experimental Motion Time, Standard Deviation, and % Deviation for Different Path Planning Algorithms.

Figure 28. *Comparison of algorithms according to standard deviation and % deviation in motion time in Map 2.*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

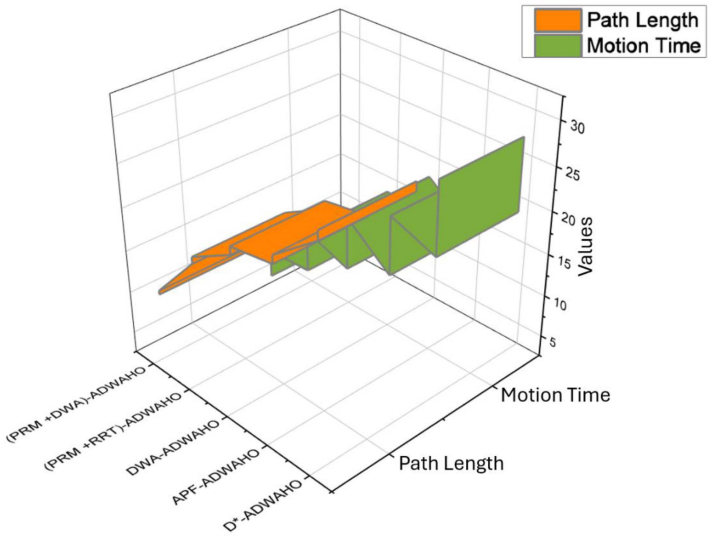


Figure 29. *Percentage change in AGV algorithms, path length, and motion time in Maps 1 and 2.*

Table XI. *Average % deviation in path length in Map 1 and Map 2.*

Average % deviation in path length in Map 1 and Map 2	Dynamic A* (D*)	APF	DWA	PRM and RRT	PRM and DWA	ADWA-HO
% Deviation in path length Map 1	9.86	7.64	7.46	7.16	6.44	5.40
% Deviation in path length Map 2	9.45	7.27	6.68	6.42	5.66	5.30
Average % deviation	9.66	7.46	7.07	6.79	6.05	5.35

Table XII. *Average % deviation in motion time in Map 1 and Map 2.*

Average % deviation in motion time in Map 1 and Map 2	Dynamic A* (D*)	APF	DWA	PRM and RRT	PRM and DWA	ADWA-HO
% Deviation in motion time in Map 1	7.13	6.92	6.13	6.05	5.44	4.06
% Deviation in motion time in Map 2	8.17	6.40	6.25	5.24	4.86	4.70
Average % deviation	7.65	6.66	6.19	5.65	5.15	4.38

By combining global and local planning, the hybrid PRM and DWA algorithm optimizes performance and computing demand. Finally, hybrid optimization makes the suggested ADWA-HO technique more computationally intensive during startup, but once the trajectory is defined, it executes efficiently continuously. ADWA-HO outperforms other approaches in terms of path length, motion duration, smoothness, and goal-reaching rate, while maintaining an acceptable computational complexity for real-time AGV navigation.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

Table XIII. *Comparing the experimental path length and motion time difference percentage between ADWA-HO and other algorithms in Maps 1 and 2.*

AGV matrix in Map 1 and Map 2	D*-ADWAHO	APF-ADWAHO	DWA-ADWAHO	(PRM + RRT)-ADWAHO	(PRM + DWA)-ADWAHO
	%Change in path length				
Experimental path length (cm) in Map 1	28.63	23.43	21.63	17.74	10.95
Experimental path length (cm) in Map 2	27.57	22.48	20.70	16.96	10.48
Average path length of both Map 1 and Map 2	28.10	22.95	21.16	17.35	10.71
Experimental motion time (s) in Map 1	27.68	21.27	18.52	14.29	9.15
Experimental motion time (s) in Map 2	19.29	14.43	12.43	9.43	5.91
Average motion time of both Map 1 and Map 2	23.48	17.85	15.47	11.86	7.53

5. Hardware and software structure

5.1. Hardware structure

The AGV is designed using SolidWorks software and developed from a laser-cut aluminum sheet and 3D-printed mecanum wheels, as shown in Figures A1 and A2 in the appendix. An embedded computer, specifically an NVIDIA Jetson Nano board with 4 GB of LPDDR4 RAM, a quad-core ARM Cortex-A57 CPU, and a 128-core NVIDIA Maxwell GPU, powers the system within the ROS environment. It also features a 40-pin general-purpose input/output (GPIO) header, which connects to sensors, motors, and other equipment. For 360-degree detection, the AGV is equipped with a set of LiDAR sensors connected via USB, enabling it to receive laser data essential for localization and map generation [72]. An STM32 microcontroller is utilized to control the permanent magnet-brushed motors, ensuring precise movement. Additionally, depth cameras facilitate deep learning, system image recognition, and point cloud data collection, thereby enhancing the overall functionality and performance of the AGV [73]. The table AI in the appendix shows the overview of software and hardware. The AGV features two LiDAR sensors, YD LiDAR TG-15 and YD LiDAR X4, for obstacle detection and distance measurement, as well as two vision sensors, an Intel depth camera D455 and an Astra Pro Plus depth camera, for object recognition and ambient awareness. Acceleration, angular velocity, and even direction are among the motion-related data that an IMU measures. By utilizing sound waves, the ultrasonic sensor determines the distance to an object to avoid it.

The NVIDIA Jetson Nano CPU performs complex calculations, including fusing sensor data and processing the ADWA-HO algorithm. Ultrasonic sensors provide proximity data for obstacle avoidance and safety in complex environments, while an STM32 motor controller controls mecanum wheel rotations accurately [74]. Vision sensors deliver high-resolution video feeds over Camera Serial Interface (CSI) ports for object detection and mapping, while LiDAR sensors transfer real-time data via specialized ports to the Jetson Nano. For close-range obstacle detection, ultrasonic sensors utilize GPIO pins, whereas the Jetson Nano employs a universal asynchronous receiver/transmitter (UART) to operate and provide feedback to the STM32 motor controller. The hardware system wiring diagram and flowchart of the motor control process are shown in Figures A3 and A4 in the appendix, respectively. The AGV's integrated design enables it to navigate, locate, and avoid collisions accurately in various experimental settings due to its omnidirectional capabilities. The concept emphasizes indoor, well-lit workplaces for human–AGV collaboration. Outdoor applications present distinct challenges, but the technology

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

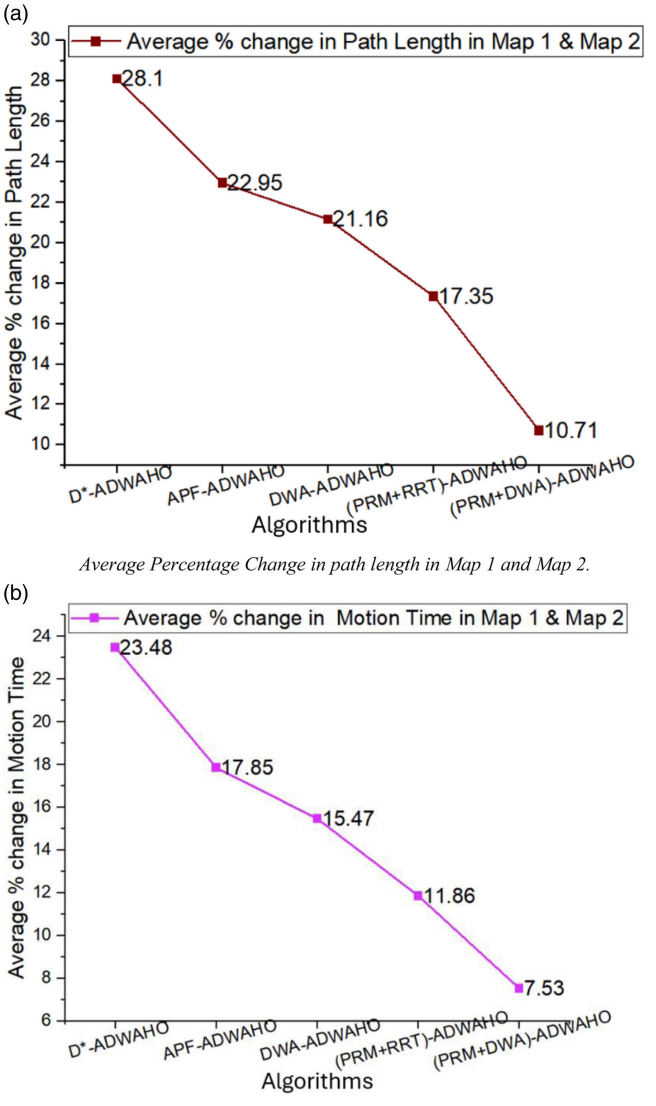


Figure 30. *Algorithm comparison based on average percentage change in Maps 1 and 2.*

has considerable potential for extension and use in human–AGV collaborative scenarios beyond interior environments to enhance data quality. Today, humans and AGVs must cooperate closely together to fulfill industrial activities, such as when a machine operator needs the AGV to collect a piece of work [71]. The AGV needs geometric (location), semantic (description), and search-related activities (contextual logic and algorithmic decision-making). A created map efficiently provides object-specific geometric and semantic information for contextual understanding and decision-making. This work presents a semantic and geometric map-building strategy for improving real-time human–AGV operations in industrial environments. Experimental scenarios I and II demonstrate how this method might avoid unforeseen impediments in a cooperative activity case study where the AGV switches between starting and goal positions. Experimental scenarios III test dynamic outdoor settings. Future industrial workspaces will benefit from combining this map-creation approach with human workers and AGVs for collaborative working. Figure 31 depicts the AGV control structure block diagram.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

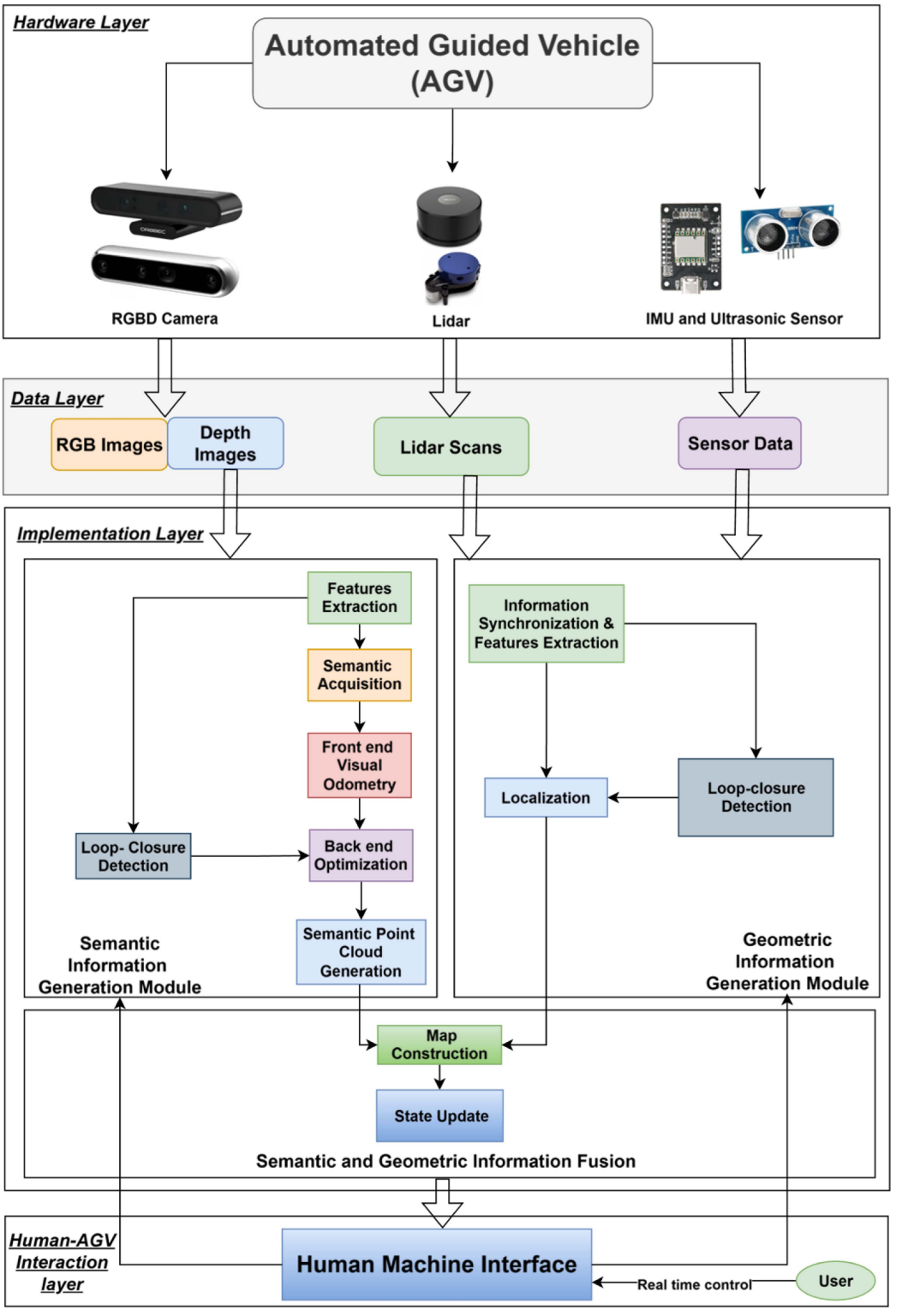


Figure 31. *Block diagram of the AGV control structure.*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

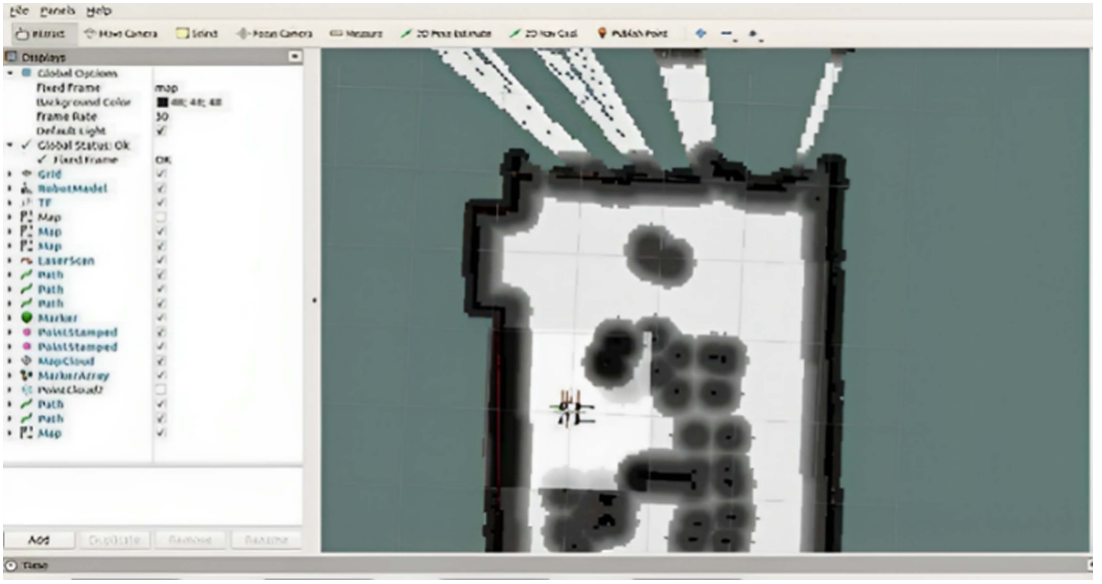


Figure 32. *Robot operating system (ROS)-based real-time environment simulation.*

5.2. Software structure

The software structure of the AGV, which operates on Ubuntu with ROS, is designed to integrate various sensors and controllers, enabling precise navigation, localization, and obstacle avoidance in both dynamic and static environments [75].

5.2.1. ROS Framework on Ubuntu

The entire software stack is powered by Ubuntu, with ROS serving as the middleware and providing crucial functions such as package management, low-level device control, device abstraction, and message passing. All hardware and algorithms are contained within ROS nodes, which exchange messages, services, and topics with one another.

5.2.2 Sensor integration and data processing

ROS drivers for the LiDAR sensors publish laser scan data to topics like /scan or /lidar_points. This data is processed in real time for obstacle detection and distance measurement, which are then fed into the navigation stack for path planning and obstacle avoidance. The vision sensor's ROS nodes stream high-resolution video feeds via the CSI ports to the Jetson Nano. These feeds are utilized for object detection and ambient mapping using deep learning models, such as YOLOv3, integrated within ROS. The processed data is published to topics such as /camera/rgb/image_raw for further use in navigation and decision-making. The ultrasonic sensors communicate with the Jetson Nano via GPIO pins, with data being processed and published on topics like /ultrasonic_data. These sensors provide close-range obstacle detection, enhancing the safety and responsiveness of the AGV in complex environments [76]. The ROS-based simulation is shown in Figure 32.

5.2.3 Navigation and Control

The Jetson Nano executes the ADWA-HO algorithm, as well as other algorithms, all of which are implemented as ROS nodes. The program optimizes route planning, navigation, and dynamic obstacle avoidance by combining data from LiDAR, vision, inertial measurement unit, and ultrasonic sensors. It publishes navigation commands on topics such as/cmd_vel and subscribes to topics for sensor data [77]. Dedicated ROS nodes control the STM32 motor controller using UART. This node receives velocity commands from the Jetson Nano and then controls the mecanum wheels with the necessary motor

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

signals. Posting motor feedback to topics like /motor feedback lets the system fine-tune commands spontaneously for accurate and smooth movement [78].

5.2.4 System Integration and Functionality

To process sensor data in real time and execute control orders precisely, the ROS master coordinates the communication between all ROS nodes. Using inputs from LiDAR, vision, IMU, and ultrasonic sensors, the sensor data fusion is performed by the ROS nodes on the Jetson Nano. The ADWA-HO algorithm, like other algorithms, relies on this fusion to construct a consistent map of its surroundings. With the help of ROS action servers and clients, the AGV's navigation goals can be adjusted autonomously to adapt to new conditions. Logging and monitoring the system's performance are made easier with ROS's features. To ensure the AGV operates successfully in various environments, nodes can post diagnostic data that is recorded and tracked [79–80].

6. Experimental validation

AGVs with mecanum wheels utilize slanted rollers to enable omnidirectional movement (rotating, sideways, or diagonal) without requiring steering. In contrast to traditional wheels, this increases maneuverability in confined spaces. Sophisticated algorithms for intricate path planning, navigation, localization, and collision avoidance are needed to properly utilize this capability. To ensure accurate and smooth movement, navigation algorithms must be developed for optimal trajectory planning. However, to maintain efficiency and safety, path design must take into account dynamic boundaries, such as speed and acceleration limits. Motion modeling and sensors, such as wheel encoders and odometers, can be combined to provide real-time positional data, giving accurate localization. When combined, these systems enable intelligent and efficient operation in changing conditions. AGV collision avoidance algorithms must be tuned to the movement patterns and dynamics of the AGVs for safe operation. This guarantees the safety and efficiency of operations by detecting and responding to obstacles in real time. The hardware and software configuration of the AGV is described in Table AI in the appendix. The tests are conducted in an unfamiliar setting with potential static or dynamic obstacles to verify the efficacy of the recommended approach. All controlled situations are tested in a workplace, corridor, and outdoors, offering a realistic and practical setting to evaluate system performance and functionality. To illustrate the discrepancy between the reference path generated by the global path planner and the actual trajectory attained by the suggested approach, it is noteworthy that we continue to utilize the global path planner. On the other hand, the reference path did not play any part in the navigation of the AGV.

6.1 Experiment scenario I

A scenario of the first experiment will be conducted in an environment with the presence of unknown static obstacles. Figure 33 depicts the original scenario's two-dimensional map, which is free from impediments and was constructed using the SLAM technique. On the other hand, as shown in Figure 34, the actual area being navigated is filled with impediments, such as boxes and sheets. During the navigation process, the AGV is able to sense the barriers because it has no prior knowledge of them. The geographical distribution of these obstacles is designed to significantly obstruct the path of the AGV as it moves toward the destination. Figure 35 shows the actual AGV position recorded during experimental scenario I, as well as the time sequence of AGV navigation as seen in ROS RViz. This comparison demonstrates the effectiveness of the navigation system by highlighting the alignment between the simulated and actual trajectories. Figure 35a provides a visual representation of the reference path, the goal, and the starting point. Figure 35b–d depicts the movements of the AGV as it navigates around the obstacles that have been present, and Figure 35e depicts the actual trajectory that the AGV must take in order to successfully attain the goal. It can be observed that the AGV moves in a flexible manner between the obstacles, and the trajectory is considerably different from the reference path. This indicates that the

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

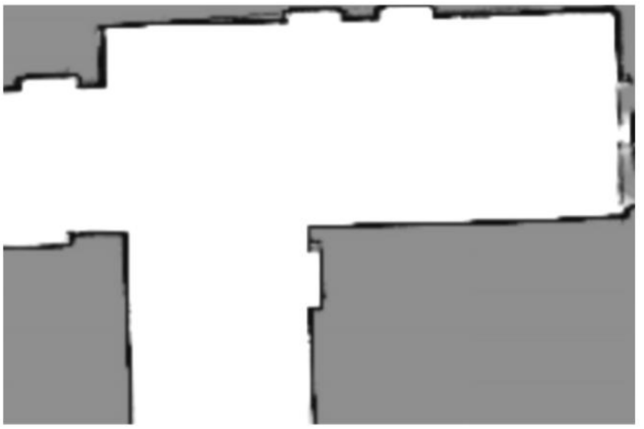


Figure 33. *Original map of experimental scenario I.*

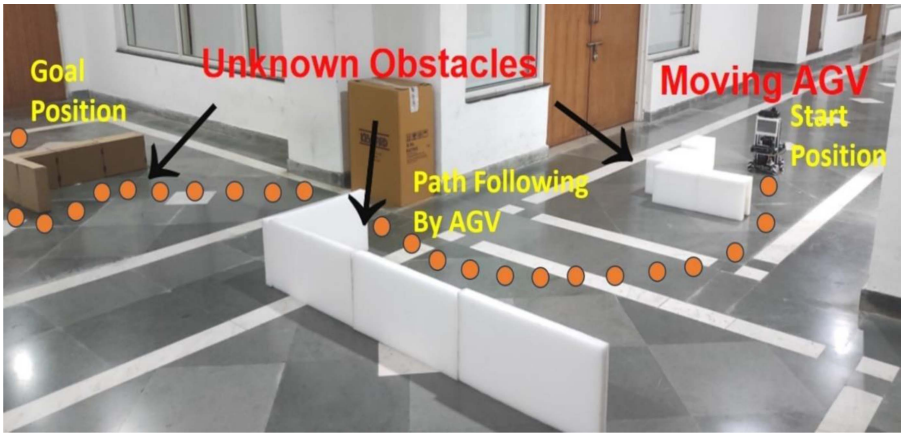


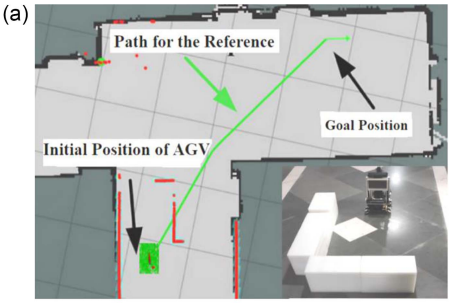
Figure 34. *Actual experiment scenario I.*

suggested approach successfully led the AGV through an environment with static obstacles from the starting point to the target point.

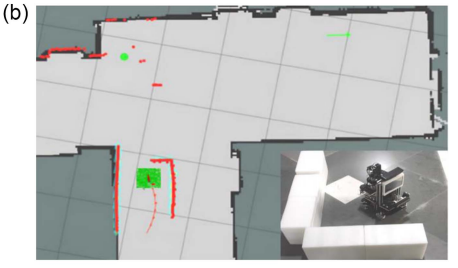
6.2 Experiment scenario II

The experiment's second scenario is conducted in an environment where two little Arduino-based AGVs serve as the dynamic obstacles. It is crucial to remember that the original ADWA-HO has been significantly enhanced to improve navigation performance in dynamic environments as well. Scenario II, in which the mecanum wheel AGV's trajectory is considerably disrupted twice by dynamic obstacles (small differential-wheeled AGVs) in its direction toward the target, will be described. Figure 36 illustrates the original scenario of the two-dimensional map, which is free from impediments and was constructed using SLAM. On the other hand, as shown in Figure 37, the actual area being navigated is filled with two small differential-wheeled AGVs. The temporal sequence of the AGV navigation, visualized in ROS RViz, and the corresponding real AGV position in experiment scenario II are depicted in Figure 38. Figure 38a provides a detailed representation of the reference path, the goal, and the starting point. Because the initial map does not contain any obstructions, the reference path leads directly to the place of interest. Figure 38b–e illustrates the actions of the AGV as it navigates around the two small moving AGVs. In the event that a small AGV prevents the large AGV from moving forward, the large AGV will move away from the small AGV (refer to Figure 38b, c). When the small AGV is a significant distance

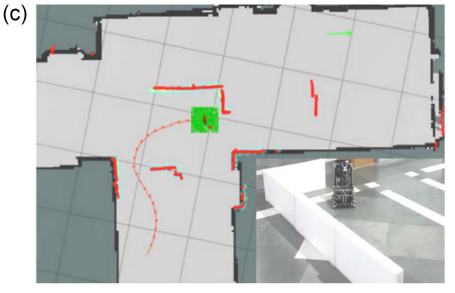
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



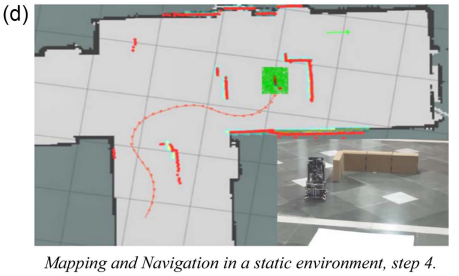
Mapping and Navigation in a static environment, step 1.



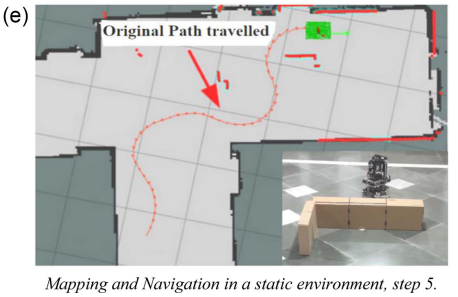
Mapping and Navigation in a static environment, step 2.



Mapping and Navigation in a static environment, step 3.



Mapping and Navigation in a static environment, step 4.



Mapping and Navigation in a static environment, step 5.

Figure 35. *The process of AGV's testing in experiment scenario I from a to e.*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

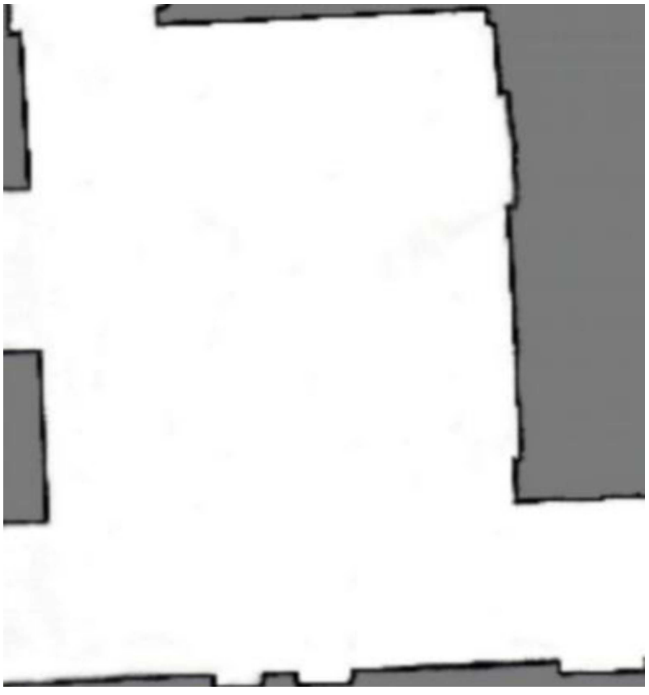


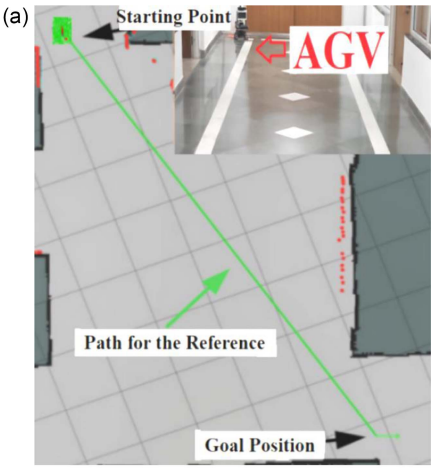
Figure 36. *Original map of experimental scenario II.*



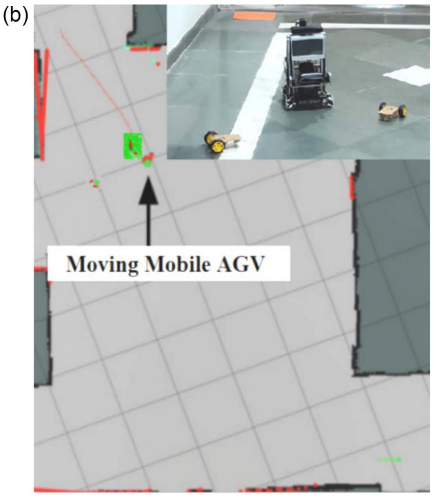
Figure 37. *Actual experiment scenario II.*

away from the large AGV (i.e., more than 0.20 m), the large AGV will begin to move in the target's direction after dynamic obstacle avoidance, that is, the small AGV (see Figure 38). In practical terms, once a dynamic obstacle is detected within a radius of less than 0.20 m, the AGV enters a wait or detour behavior. As soon as the dynamic obstacle moves beyond this threshold, the local costmap is updated, and the AGV resumes path planning toward its target, as illustrated in Figure 38. A buffer zone is established to compensate for sensor update delays and AGV deceleration lag, thereby reducing the risk of collision with fast-approaching or unpredictable obstacles. In Figure 38e, the actual trajectory that the AGV takes when it successfully reaches the objective is depicted. As can be observed, the real trajectory of the large moving AGV is considerably distinct from the reference path. This demonstrates that the suggested method is capable of successfully navigating the AGV in a dynamic environment with two small moving AGVs as obstacles.

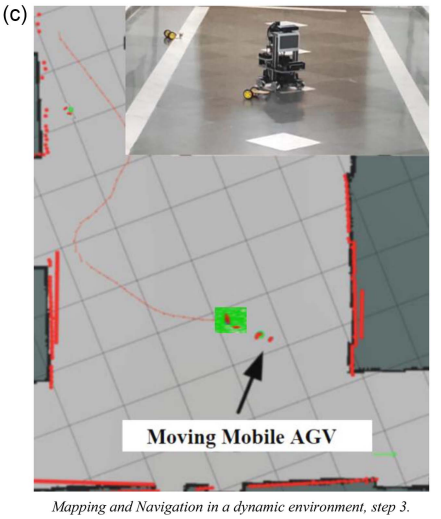
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Mapping and Navigation in a dynamic environment, step 1.



Mapping and Navigation in a dynamic environment, step 2.

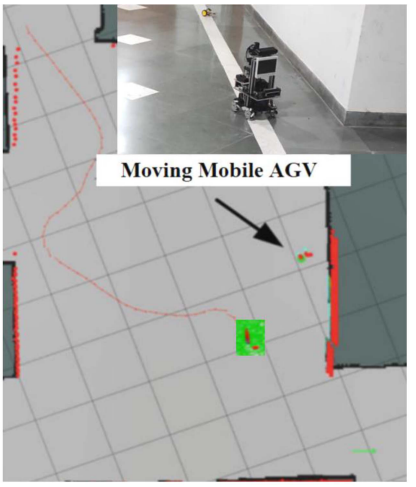


Mapping and Navigation in a dynamic environment, step 3.

Figure 38. *The process of AGV's testing in experiment scenario II from a to e.*

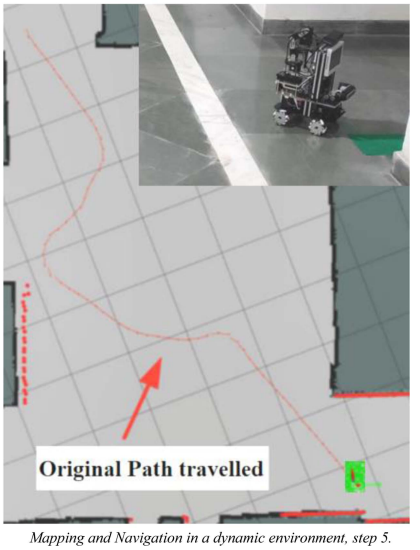
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

(d)



Mapping and Navigation in a dynamic environment, step 4.

(e)



Mapping and Navigation in a dynamic environment, step 5.

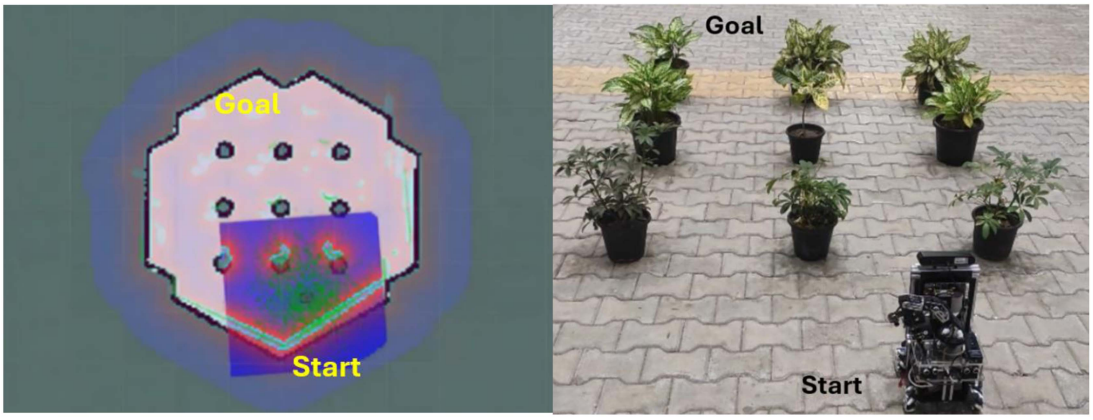
Figure 38. *Continued.*

6.3 Experiment scenario III

The final experiment is conducted outdoors, where the AGV must navigate between the start and goal locations while avoiding both dynamic and static obstacles in real time. Figure 39 displays a mapped environment in RViz, with the workspace denoted by defined bounds and gridlines. The working area is delineated by a hexagonal form, which changes dynamically as the AGV moves from one location to another. The AGV's intended mapping process is shown by the green line. It illustrates how the AGV adjusts its trajectory to steer clear of obstacles as it approaches its target. The map indicates barriers that represent either stationary or moving elements of the surroundings. These are identified by the AGV's sensors, which include depth cameras and LiDAR sensors. The AGV's costmap can be observed by the gradation of red and blue hues surrounding obstacles. The AGV refrains from traveling through red zones, which are areas closer to obstacles and have higher costs. Safe paths are indicated by lower-cost regions, that is, blue zones. The costmap and path are constantly adjusted in real time as the AGV advances to account for environmental changes. This enables the AGV to travel quickly when encountering impediments that are stationary, moving, or have just been detected. The global plan, sensors scan, and local costmap belong to the inputs required to configure the navigation stack. Figures 40 and 41

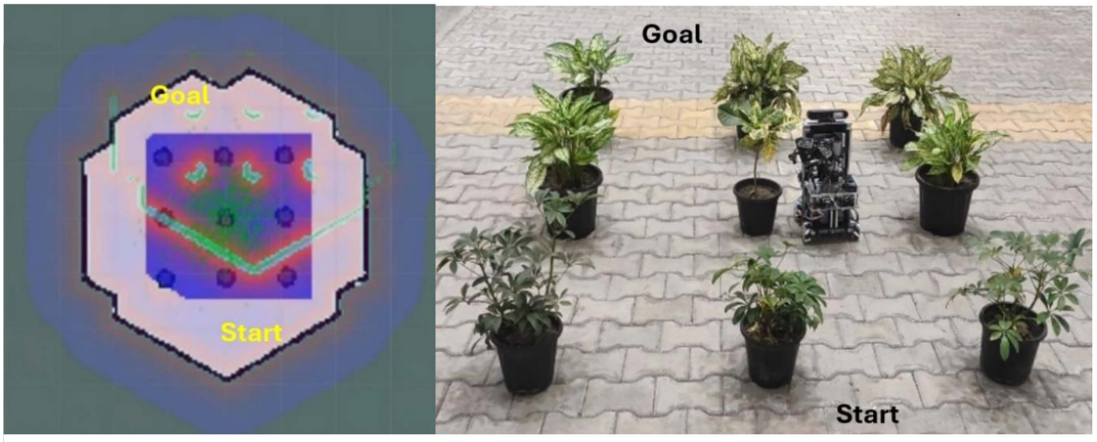
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

(a)



Mapping in the outdoor environment, step 1.

(b)



Mapping in the outdoor environment, step 2.

Figure 39. *Mapping in the outdoor environment in a) and b).*

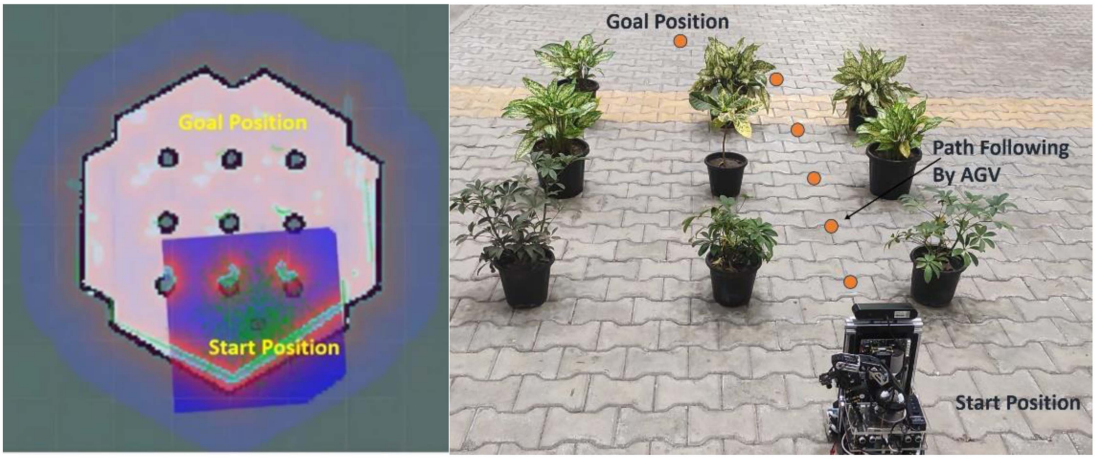
illustrate the steps involved in testing an AGV in experiment scenario III, which includes nine static and three dynamic impediments, respectively, in an outdoor setting. They also provide a thorough depiction of the objectives, the initial point, and the path taken to attain the target.

The ADWA-HO algorithm's real-time adaptability and efficacy were confirmed by the ROS-based outdoor AGV testing. This demonstrates its ability to adapt to both external static and dynamic situations. Using LiDAR and depth cameras, the AGV skillfully avoids both stationary and moving impediments as it moves from its starting point to its destination. With the help of the dynamically updated costmap, it selects safe routes by avoiding more expensive locations. Hexagonal gridlines enhance path planning, while RViz's environment trajectory confirms accurate navigation. The algorithm's resilience in real-time replanning is demonstrated by the AGV's ability to dynamically adjust the route in response to static and moving obstructions.

Moving obstacles are identified via sensor fusion between LiDAR scans and depth images. The velocity vector and distance threshold (approximately 0.20 m) are used to classify an obstacle as dynamic. As soon as a dynamic obstacle is detected within a proximity range of less than 0.20 m, the local costmap assigns a high traversal cost to the immediate vicinity (red zones in the costmap), prompting the planner to seek alternative paths. The ADWA-HO algorithm triggers a local trajectory replanning if obstacles are predicted to intersect the current path. The AGV slows down, waits, or circumvents the obstacle based on proximity and motion trajectory. After successfully avoiding the obstacle, the AGV smoothly

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

(a)



Mapping and Navigation in the outdoor static environment, step 1.

(b)



Mapping and Navigation in the outdoor static environment, step 2.

(c)

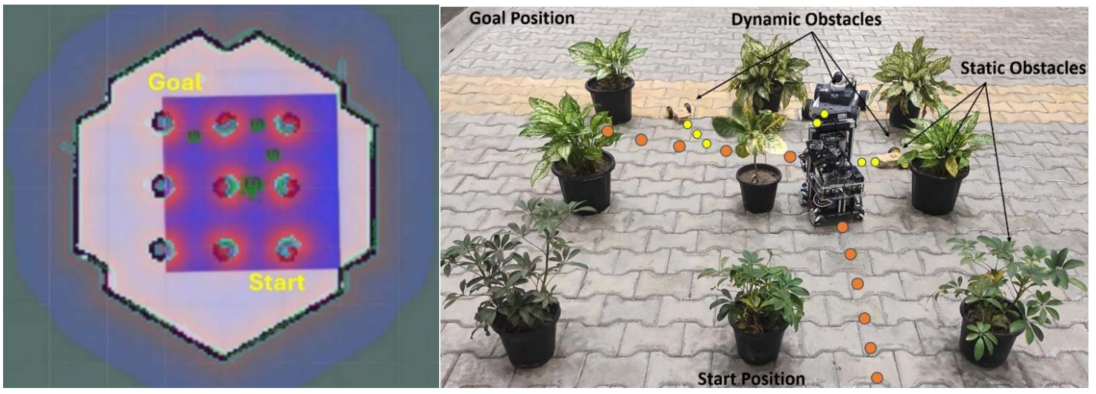


Mapping and Navigation in the outdoor static environment, step 3.

Figure 40. *The process of AGV's testing in experiment scenario III with static obstacles from a to c.*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

(a)



Mapping and Navigation in the outdoor static and dynamic environment, step 1.

(b)



Mapping and Navigation in the outdoor static and dynamic environment, step 2.

Figure 41. *The process of AGV's testing in experiment scenario III with static and dynamic obstacles from a to b.*

returns to a globally optimized path, ensuring goal convergence without significant trajectory drift. The above three figures illustrate the global costmap, sensor scans, and path adaptation (Figure 39), as well as trajectory adjustments made by the AGV to avoid nine static obstacles and account for sensor noise (Figure 40). The AGV's local path deviations around the three dynamic obstacles, along with static obstacles, with visible smooth inflection points and avoidance maneuvers (Figure 41). This dynamic motion modeling of the obstacles and the AGV's responsive strategy demonstrates the robustness of the ADWA-HO algorithm, validating its capability to handle unpredictable real-world navigation challenges in static and dynamic environments.

6.4 Dynamic obstacle avoidance

The AGV employs a hybrid navigation stack that integrates local and global planning to address these dynamics. While the local planner (DWA) continuously modifies the trajectory based on real-time sensor data, the global planner creates an initial path using a map of the environment. Strategies for managing dynamic obstacles include:

Real-time Detection: Continuous scanning of the environment is performed by LiDAR and depth cameras. Based on motion patterns, which are differences between successive scans, dynamic as well as static objects that have been detected are identified.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

Costmap Update: Through the utilization of the ROS navigation stack, dynamic barriers can be added to the local costmap in the form of inflated high-cost zones. Because of this, the AGV is unable to choose routes that bring it close to moving impediments and therefore avoids the obstacles.

Trajectory Replanning: The AGV immediately begins the process of local trajectory replanning using DWA whenever it is anticipated that an obstacle will intersect the current path. This process involves computing velocity commands that avoid collisions while maintaining proximity to the global path.

Predictive Behavior: To anticipate future positions and adjust the path accordingly, a short-term motion prediction model is utilized for obstacles that are in motion. This model involves velocity extrapolation throughout one to two seconds.

Safety Buffering: A dynamic safety buffer is maintained around moving objects, approximately ranging from 0.15 to 0.30 m, and its size grows or decreases depending on the speed of the obstacle and its proximity.

6.5 Observed limitations

Real-world tests were conducted in various indoor and outdoor environments, as well as under different obstacle dynamics, to thoroughly evaluate the effectiveness and long-term reliability of the proposed hybrid navigation system. The AGV's decision-making and path planning endurance was tested in unknown static barriers, dynamic interference from other AGVs, and harsh outside environments with static and dynamic variables. From every scenario, the system's safety and collision prevention strengths, as well as stress-related behaviors such as careful maneuverability, temporary delays, and path recalculations, are revealed. The outcomes and system actions for each experimental scenario are listed below:

- In experiment scenario I (unknown static obstacles), the AGV occasionally exhibited minor deviations and increased lateral clearance while navigating through tightly packed static obstacles. This was due to the cautious behavior of local planners, which prioritizes safety when real-time updates show high obstacle density. In rare cases, this led to path segments that were longer than optimal, reflecting a conservative trade-off between safety and efficiency.
- When two little Arduino-based AGVs concurrently approached the main AGV from convergent directions in experiment scenario II (dynamic barriers), the system initiated a brief pause and replanning cycle. Although collision avoidance was maintained, this behavior caused little delay in achieving the goal. Furthermore, the replanning process occasionally produced zigzag courses when the processor recalculated the best escape routes if the dynamic AGVs suddenly changed direction or hovered nearby.
- Strong ambient lighting occasionally resulted in transient reflections in LiDAR returns from the shiny objects in experiment scenario III (outdoor setting with static and dynamic obstacles). This resulted in too cautious rerouting and exaggerated obstacle bounds. Additionally, because the costmap was updated frequently, short local trajectory oscillations were seen when dynamic and static barriers were grouped together. Nonetheless, the system continuously recovered and stayed clear of accidents, demonstrating the hybrid planner's resilience.

7. Conclusion

To overcome the difficulties associated with path planning and control in omnidirectional mecanum wheel AGVs, we proposed and implemented the ADWA-HO method. Traditional algorithms, despite being effective in theory, frequently face practical challenges such as excessive turning points, redundant nodes, and longer runtimes, which restrict their efficiency in applications used within real-time environments. Through the combination of the MA* algorithm and the DWA, the ADWA-HO method enhances path planning, particularly in situations that are both static and dynamic.

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

- The ADWA-HO algorithm outperformed D*, APF, DWA, PRM & RRT fusion, and PRM & DWA fusion in average path length by 28.10%, 22.95%, 21.16%, 17.35%, and 10.71% and average motion time by 23.48%, 17.85%, 15.47%, 11.86%, and 7.53% in Map 1 and Map 2. ADWA-HO closely matches real-world conditions in maps 1 and 2, with an average path length deviation of 5.35% and motion time deviation of 4.38%. The fact that ADWA-HO has a lower standard deviation for both path length and motion time suggests that the algorithm performs continuously better compared to other algorithms.
- In Map 1, the RMS path length error is 10.41 cm, and the RMS motion time error is 0.72 s for ADWA-HO. In Map 2, the path length error is 10.77 cm, and the RMS motion time error is 1.33 s. The simulated and experimental results are closely correlated, demonstrating the accuracy and consistency of the ADWA-HO algorithm across various map settings.
- D* and APF require less computation but risk path inflections and collisions. DWA handles short-range obstacle avoidance with a moderate load but lacks global planning. PRM and RRT require higher computation for graph/tree expansion, which improves smoothness and safety. Hybrid PRM and DWA balance performance and load, while ADWA-HO achieves superior computation efficiency and accuracy with acceptable real-time complexity, as shown in Tables III and VII.
- With experiment scenarios I, II, and III with unknown static and dynamic barriers, ADWA-HO easily combines geometrical and semantic information for autonomous navigation and path planning. The effectiveness of the suggested approach is strongly supported by experiments utilizing AGV path planning and navigation in both static and dynamic environments, encompassing indoor and outdoor settings. These experiments confirm the proposed algorithm's versatility in a range of real-world situations with multiple static and dynamic obstacles.
- The proposed hybrid AGV navigation system is resilient, adaptable, and safety-focused, as shown by real-world experiments in different and challenging circumstances. The system successfully avoided collisions and navigated safely despite unknown static obstructions, dynamic interference from other AGVs, and severe external conditions. Temporary delays, changes in trajectories, and path extensions while avoiding the obstacles were acceptable trade-offs that showed the planner prioritized safety over optimality in uncertain scenarios.
- The approach developed is capable of functioning in a workplace that is a coexistence of humans and robots, that is, human–robot collaboration, with the objective of producing a real-time map that incorporates semantic and geometric data.
- In future studies, multiple dynamic impediments will be taken into consideration during the testing process, and increasingly complicated barriers will be set up during the experiment. Future studies could use human and swarm-inspired hybrid obstacle avoidance strategies to increase the AGV's capabilities. This could enhance the AGV's adaptability and efficiency in challenging environments.
- Better human–robot collaboration may benefit from combining the recommended map construction method with generative AI models. The map will provide a baseline for scene perception, while generative AI models will enable the AGV to receive dynamic instructions, determine motions, and interact with humans.

The ADWA-HO algorithm generally improves the performance of AGV systems by offering path planning and control that is more efficient, accurate, and precise. As a result, it is ideal for deployment in environments that are both static and dynamic, as well as complex.

By following this link, anyone can access the codes and videos made for this research work: https://github.com/ankurgsb21/Omnidirectional-AGV-Control-in-static-and-dynamic-environments-using-Modified-ADWA-HO-Algorithm

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

Author contributions. Under the supervision of Prof. Mohammad Suhaib and Prof. Ajay K. S. Singholi, Ankur Bhargava undertakes the conceptualization, design, data collection, methodology, representation, and analysis of the study. The supervisors shared insights on how to write, edit, review, and analyze the data.

Financial support. Not applicable.

Competing interests. The authors report no conflicts of interest.

Ethical approval. Not applicable.

References

[1] A. K. S. Singholi, M. Mittal and A. Bhargava, "A Review on IoT-Based Hybrid Navigation System for Mid-sized Autonomous Vehicles," **In:** *Advances in Electromechanical Technologies*. Lecture Notes in Mechanical Engineering (V. C. Pandey, P. M. Pandey, and S. K. Garg, eds.) (Springer, Singapore, 2021). https://doi.org/10.1007/978-981-15-5463-6_65

[2] 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* **39**(2), 250–276 (2021). https://doi.org/10.1017/S0263574720000351

[3] S. Mary Joans, N. Gomathi and P. Ponsudha, "Wireless vision-based digital media fixed-point DSP processor depending robots for natural calamities," *Robotica* **42**(5), 1368–1385 (2024). https://doi.org/10.1017/S0263574724000225

[4] F. Alkhawaja, M. A. Jaradat and L. Romdhane, "Low-cost depth/IMU intelligent sensor fusion for indoor robot navigation," *Robotica* **41**(6), 1689–1717 (2023). https://doi.org/10.1017/S0263574723000073

[5] P. Xiong, Y. Huang, Y. Yin, Y. Zhang and A. Song, "A novel tactile sensor with multimodal vision and tactile units for multifunctional robot interaction," *Robotica* **42**(5), 1420–1435 (2024). https://doi.org/10.1017/S0263574724000286

[6] A. Bhargava, M. Suhaib and A. S. Singholi, "Path planning and collision avoidance algorithms for AGV: A comparative review," *Tech. Sicherh.* **24**(3), 79–92 (2024). https://doi.org/22.8342.TSJ.2024.V24.3.01297

[7] M. Khan, A. Haleem and M. Javaid, "Changes and improvements in Industry 5.0: A strategic approach to overcome the challenges of Industry 4.0," *Green Technol. Sustain.* **1**(2), 100020 (2023). https://doi.org/10.1016/j.grets.2023.100020

[8] A. Singholi, "Impact of manufacturing flexibility and pallets on buffer delay in flexible manufacturing systems," *Int. J. Eng. Manage. Econ.* **5** (3/4), 308–330 (2015). https://doi.org/10.1504/IJEME.2015.072546

[9] C. Huang, M. Zhu, S. Song, Y. Zhao and J. Jiang, "Vision-based adaptive LT sliding mode admittance control for collaborative robots with actuator saturation," *Robotica* **42**(6), 1986–2003 (2024). https://doi.org/10.1017/S0263574724000729

[10] A. Bhargava, M. Suhaib and A. S. Singholi, "A review of recent advances, techniques, and control algorithms for automated guided vehicle systems," *J. Braz. Soc. Mech. Sci. Eng.* **46**(7), 419 (2024). https://doi.org/10.1007/s40430-024-04896-w

[11] A. Singh and J. P. Khatait, "RCM adjustment for a double-parallelogram based RCM mechanism used for MIS robots," *Proc. Inst. Mech. Eng. Pt. C J. Mechan. Eng. Sci.* **238**(6), 2267–2282 (2023). https://doi.org/10.1177/09544062231185518

[12] D. Agarwal and P. S. Bharti, "Implementing modified swarm intelligence algorithm based on Slime moulds for path planning and obstacle avoidance problem in mobile robots," *Appl. Soft. Comput.* **107**, 107372 (2021). https://doi.org/10.1016/j.asoc.2021.107372

[13] D. Agarwal and P. S. Bharti, "Nature inspired evolutionary approaches for robot navigation: Survey," *J. Inform. Optim. Sci.* **41**(2), 421–436 (2020). https://doi.org/10.1080/02522667.2020.1723938

[14] A. K. S. Singholi and A. Sharma, "Finding capabilities of 4D printing," *Int. J. Eng. Adv. Technol.* **8**(5), 1095–1110 (2019).

[15] M. Ding, X. Zheng, L. Liu, J. Guo and Y. Guo, "Collision-free path planning for cable-driven continuum robot based on improved artificial potential field," *Robotica* **42**(5), 1350–1367 (2024). https://doi.org/10.1017/S026357472400016X

[16] B. Zhou, J. Yi, X. Zhang, L. Chen, D. Yang, F. Han and H. Zhang, "An autonomous navigation approach for unmanned vehicle in outdoor unstructured terrain with dynamic and negative obstacles," *Robotica* **40**(8), 2831–2854 (2022). https://doi.org/10.1017/S0263574721001983

[17] M. A. Post, "Probabilistic robotic logic programming with hybrid Boolean and Bayesian inference," *Robotica* **42**(1), 40–71 (2024). https://doi.org/10.1017/S0263574723001339

[18] A. Bircher, K. Alexis, U. Schwesinger, S. Omari, M. Burri and R. Siegwart, "An incremental sampling-based approach to inspection planning: The rapidly exploring random tree of trees," *Robotica* **35**(6), 1327–1340 (2017). https://doi.org/10.1017/S0263574716000084

[19] A. Lonklang and J. Botzheim, "Mobile Robot Path Planning for Unknown Static Obstacle Avoidance by Improved RRT* Algorithm," **In:** *2024 10th International Conference on Automation, Robotics and Applications (ICARA)*, Athens, Greece (2024) pp. 155–159. https://doi.org/10.1109/ICARA60736.2024.10553042

[20] C. Zheyi and X. Bing, "AGV Path Planning Based on Improved Artificial Potential Field Method," **In:** *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, (2021) pp. 32–37. https://doi.org/10.1109/icpeca51329.2021.9362

[21] R. M. Jans, A. S. Green and L. J. Koerner, "Characterization of a miniaturized IR depth sensor with a programmable region-of-interest that enables hazard mapping applications," *IEEE Sens. J.* **20**(10), 5213–5220 (2020). https://doi.org/10.1109/JSEN.2020.2971595

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

[22] V. G. R. Caitité, D. M. G. dos Santos, I. C. Gregório, W. B. da Silva and V. F. Mendes, "Diffusion of Robotics through Line Follower Robots," **In:** *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, João Pessoa, Brazil, 2018) pp. 604–609. https://doi.org/10.1109/LARS/SBR/WRE.2018.00109

[23] A. E. Setiawan, A. Rusdinar, S. Rizal, R. Mardiati, A. Wasik and E. A. Z. Hamidi, "Fuzzy Logic Control for Modeling Multi Robot AGV Maneuver Based on Inverted Camera," **In:** *2022 16th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, Lombok, Indonesia (2022) pp. 1–7. https://doi.org/10.1109/TSSA56819.2022.10063872

[24] C. Gu, S. Liu, H. Li, K. Yuan and W. Bao, "Research on hybrid path planning of underground degraded environment inspection robot based on improved A* algorithm and DWA algorithm," *Robotica* **43**(3), 1–22 (2025). https://doi.org/10.1017/S0263574725000037

[25] M. Lazreg and N. Benamrane, "Hybrid system for optimizing the robot mobile navigation using ANFIS and PSO," *Robot. Auton. Syst.* **153**, 104114 (2022). https://doi.org/10.1016/j.robot.2022.104114

[26] Q. Jiang and H. Yan, "Leader–follower Motion Control of Two Line Following AGVs using PID Algorithm," **In:** *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Chongqing, China (2022) pp. 445–449. https://doi.org/10.1109/IMCEC55388.2022.10020081

[27] M. A., P. K., A. S. k. and N. N., "Design and Development of Automated Guided Vehicle with Line Follower Concept using IR," **In:** *2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Erode, India (2023) pp. 1–11. https://doi.org/10.1109/ICECCT56650.2023.10179618

[28] A. Bhargava, A. S. Singholi and M. Suhaib. A Study on Design and Control of Omni-directional Mecanum Wheels Based AGV System," **In:** *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Delhi, India (2023) pp. 1–6. https://doi.org/10.1109/ICCCNT56998.2023.10306665

[29] M. Kokot, D. Miklić and T. Petrović, "Path continuity for multi-wheeled AGVs," *IEEE Robot. Autom. Lett.* **6**(4), 7437–7444 (2021). https://doi.org/10.1109/LRA.2021.3099086

[30] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib and A. J. Humaidi, "Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm," *Appl. Soft. Comput.* **89** 106076 (2020). ISSN 1568-4946. https://doi.org/10.1016/j.asoc.2020.106076

[31] R. Koyama, S. Katsumata and N. Kimura, "Rapid Determination of Transportation Line Layout by Combination of AGVs Line Searching and Conveyor's Neighbor Recognition," **In:** *2021 26th International Conference on Automation and Computing (ICAC)*, Portsmouth, United Kingdom (2021) pp. 1–6. https://doi.org/10.23919/ICAC50006.2021.9594156

[32] J. Zheng and Z. Zhang, "Research on AGV Visual Perception Dynamic Exposure Algorithm Based on Gray Entropy Threshold Difference Value," **In:** *2022 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*, Suzhou, China (2022) pp. 1–6. https://doi.org/10.1109/CTISC54888.2022.9849781

[33] S. Qin, J. Jian, B. Chen and X. Jian, "Research and Application of Image Processing Technology in the AGV System Based on Smart Warehousing," **In:** *2020 5th International Conference on Control, Robotics and Cybernetics (CRC)*, Wuhan, China (2020) pp. 123–128. https://doi.org/10.1109/CRC51253.2020.9253500

[34] C. Wang and J. Mao, "Summary of AGV Path Planning," **In:** *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, Xiamen, China (2019) pp. 332–335. https://doi.org/10.1109/EITCE47263.2019.9094825

[35] J. H. Li, Y. S. Ho and J. J. Huang, "Line Tracking with Pixy Cameras on a Wheeled Robot Prototype," **In:** *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, Taichung, Taiwan (2018) pp. 1–2. https://doi.org/10.1109/ICCE-China.2018.8448948

[36] C. Kim, J. Suh and J.-H. Han, "Development of a hybrid path planning algorithm and a bio-inspired control for an omni-wheel mobile robot," *Sensors* **20**(15), 4258 (2020). https://doi.org/10.3390/s20154258

[37] J. Yan, Z. Liu, T. Zhang and Y. Zhang, "Autonomous Decision-making Method of Transportation Process for Flexible Job Shop Scheduling Problem Based on Reinforcement Learning," **In:** *2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, Chongqing, China (2021) pp. 234–238. https://doi.org/10.1109/MLISE54096.2021.00049

[38] D. Habermann, A. Hata, D. Wolf and F. S. Osorio, "3D Point Clouds Segmentation For Autonomous Ground Vehicle," **In:** *2013 III Brazilian Symposium on Computing Systems Engineering*, (2013) pp. 143–148. https://doi.org/10.1109/sbesc.2013.43

[39] Z. Miljković, N. Vuković, M. Mitić and B. Babić, "New hybrid vision-based control approach for automated guided vehicles," *Int. J. Adv. Manuf. Technol.* **66**(1-4) 231–249 (2013). https://doi.org/10.1007/s00170-012-4321-y

[40] X. Gao, R. Gao, P. Liang, Q. Zhang, R. Deng and W. Zhu, "A hybrid tracking control strategy for nonholonomic wheeled mobile robot incorporating deep reinforcement learning approach," *IEEE Access* **9**, 15592–15602 (2021). https://doi.org/10.1109/ACCESS.2021.3053396

[41] R. S. Bisht, P. M. Pathak and S. K. Panigrahi, "Modelling, simulation and experimental validation of wheel and arm locomotion based wall-climbing robot," *Robotica* **41**(2), 433–469 (2023). https://doi.org/10.1017/S026357472200025X

[42] R. Tarvirdilu-Asl, R. Zeinali and H. B. Ertan, "FEM-based Design Modifications and Efficiency Improvements of a Brushed Permanent Magnet DC Motor," **In:** *2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP)*, Brasov, Romania (2017) pp. 401–407. https://doi.org/10.1109/OPTIM.2017.7975003

[43] H. Khan, S. Khatoon and P. Gaur, "Stabilization of wheeled mobile robot by social spider algorithm based PID controller," *Int. j. inf. tecnol.* **16**(3), 1437–1447 (2024). https://doi.org/10.1007/s41870-023-01438-w

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

[44] B. I. Adamov and G. R. Saypulaev, "A Study of the Dynamics of an Omnidirectional Platform, Taking into Account the Design of Mecanum Wheels and Multicomponent Contact Friction," **In:** *2020 International Conference Nonlinearity, Information and Robotics (NIR), Innopolis*, Russia (2020) pp. 1–6. https://doi.org/10.1109/NIR50484.2020.9290193

[45] M. Abdelrahim, A. Hassan, D. Ojo, M. Hosny, H. H. Ammar and M. El-Samanty, "A Novel Design of a T-Model Three Mecanum Wheeled Mobile Robot," **In:** *2022 International Conference on Advanced Robotics and Mechatronics (ICARM)*, Guilin, China (2022) pp. 509–513. https://doi.org/10.1109/ICARM54641.2022.9959189

[46] C. Chaichumporn, P. Ketthong, N. Thi Mai, K. Hashikura, M. A. S. Kamal and I. Murakami, "The Dynamical Modeling of Four Mecanum Wheel Mobile Robot on Typical Unstructured Terrain," **In:** *2024 21st International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) 21st International Conference on Electrical Engineering/Electronics*, Khon Kaen, Thailand (2024) pp. 1–5. https://doi.org/10.1109/ECTI-CON60892.2024.10594952

[47] H. Xu, D. Yu, Q. Wang, P. Qi and G. Lu. Current Research Status of Omnidirectional Mobile Robots with Four Mecanum Wheels Tracking based on Sliding Mode Control," **In:** *2019 IEEE International Conference on Unmanned Systems and Artificial Intelligence (ICUSAI)*, Xi'an, China (2019) pp. 13–18. https://doi.org/10.1109/ICUSAI47366.2019.9124796

[48] P. I. Chang and C. K. Lin, "Sensor Fusion Estimation for Omni-Directional Vehicle with Mecanum Wheel," **In:** *2023 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, Taipei, Taiwan (2023) pp. 1–6. https://doi.org/10.1109/ARIS59192.2023.10268507

[49] P. Atlantis, C. P. Bechlioulis, G. Karras, G. Fourlas and K. J. Kyriakopoulos, "Fault Tolerant Control for Omni-directional Mobile Platforms with 4 Mecanum Wheels," **In:** *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden (2016) pp. 2395–2400. https://doi.org/10.1109/ICRA.2016.7487389

[50] J. Zhang and X. Liu-Henke, "Model-based Design of the Vehicle Dynamics Control for an Omnidirectional Automated Guided Vehicle (AGV)," **In:** *2020 International Conference Mechatronic Systems and Materials (MSM)*, Bialystok, Poland (2020) pp. 1–6. https://doi.org/10.1109/MSM49833.2020.9202248

[51] A. Bhargava, A. S. Singholi and M. Suhaib, "Design and Development of a Visual - SLAM based Automated Guided Vehicle," **In:** *2023 IEEE Fifth International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, Bengaluru, India (2023) pp. 1–7. https://doi.org/10.1109/ICAECC59324.2023.10560331

[52] A. V. Le, V. Prabakaran, V. Sivanantham and R. E. Mohan, "Modified A-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor," *Sensors* **18**, 2585 (2018). https://doi.org/10.3390/s18082585

[53] A. Pal, R. Tiwari and A. Shukla, "Modified A* Algorithm for Mobile Robot Path Planning," **In:** *Soft Computing Techniques in Vision Science*. Studies in Computational Intelligence (S. Patnaik and Y. M. Yang, eds.), vol. **395** (Springer, Berlin, Heidelberg, 2012). https://doi.org/10.1007/978-3-642-25507-6_16

[54] D. Xiang, H. Lin, J. Ouyang and D. Huang, "Combined improved A* and greedy algorithm for path planning of multi-objective mobile robot," *Sci. Rep.* **12**, 13273 (2022). https://doi.org/10.1038/s41598-022-17684-0

[55] A. Bhargava, M. Suhaib and A. S. Singholi, "Comparative study of localization techniques for indoor mobile robots," (2024). Technische Sicherheit, 2024. pp. 49–62. https://doi.org/22.8342.TSJ.2024.V24.3.01295

[56] M. Javaid, A. Haleem, R. P. Singh, R. Suman, S. Khan and S. Rab, "Adoption of internet of things for smart city-enabled smart hospitals," *Intell. Hosp.* **1**(2) 100011 (2025). https://doi.org/10.1016/j.inhs.2025.100011

[57] C. Ju, Q. Luo and X. Yan, "Path Planning Using an Improved A-star Algorithm," **In:** *2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan)*, Jinan, China (2020) pp. 23–26. https://doi.org/10.1109/PHM-Jinan48558.2020.00012

[58] Y. Xinyi, Z. Yichen, L. Liang and O. Linlin, "Dynamic window with virtual goal (DW-VG): A new reactive obstacle avoidance approach based on motion prediction," *Robotica* **37**(8), 1438–1456 (2019). https://doi.org/10.1017/S0263574719000043

[59] L. Guo, S. Zhang, W. Zhao, J. Liu and R. Liu, "Obstacle avoidance control of UGV based on adaptive-dynamic control barrier function in unstructured terrain," *Robotica* **42**(9), 2991–3004 (2024). https://doi.org/10.1017/S026357472400122X

[60] P. Li, L. Hao, Y. Zhao, J. Lu and M. Ghadiri Nejad, "Robot obstacle avoidance optimization by A* and DWA fusion algorithm," *PLoS One* **19**(4), e0302026 (2024). PMID: 38683853; PMCID: PMC11057753. https://doi.org/10.1371/journal.pone.0302026

[61] K. Piemngam, I. Nilkhamhang and P. Bunnun, "Development of Autonomous Mobile Robot Platform with Mecanum Wheels," **In:** *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, Bangkok, Thailand (2019) pp. 90–93. https://doi.org/10.1109/ICA-SYMP.2019.8646085

[62] N. Chivarov, S. Yovkov, S. Chivarov, P. Stoev and D. Chikurtev, "Teleoperation and Autonomous Mode of Transport Mobile Robot with Mecanum Wheels," **In:** *2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC)*, Crete, Greece (2022) pp. 310–315. https://doi.org/10.1109/CSCC55931.2022.00060

[63] Y. Gao, Q. Han, S. Feng, Z. Wang, T. Meng and J. Yang, "Improvement and fusion of D*Lite algorithm and dynamic window approach for path planning in complex environments," *Machines* **12**(8), 525 (2024). https://doi.org/10.3390/machines12080525

[64] J. Xu, Z. Tian, W. He and Y. Huang, "A Fast Path Planning Algorithm Fusing PRM and P-Bi-RRT," **In:** *2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan)*, Jinan, China (2020) pp. 503–508. https://doi.org/10.1109/PHM-Jinan48558.2020.00098

[65] S. Fu, "Robot Path Planning Optimization Based on RRT and APF Fusion Algorithm," **In:** *2024 8th International Conference on Robotics and Automation Sciences (ICRAS)*, Tokyo, Japan (2024) pp. 32–36. https://doi.org/10.1109/ICRAS62427.2024.10654464

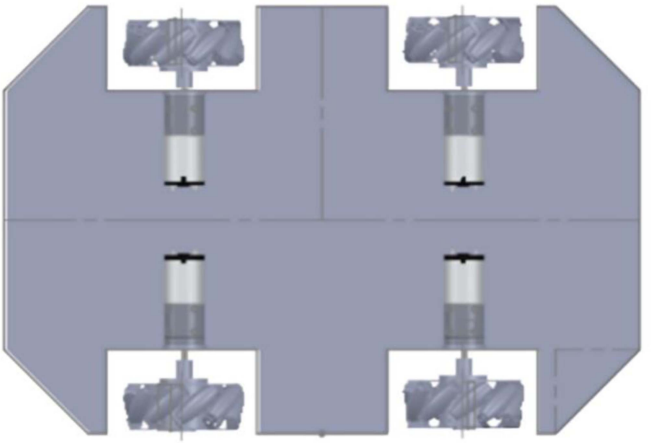
https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

[66] Y. Liang, H. Tan, J. Wei, Z. Zhang, Y. Wang and X. Huang, "Mobile Robot Path Planning with Improved DWA Algorithm under Dynamic Complex Scene," **In:** *2024 14th Asian Control Conference (ASCC)*, Dalian, China (2024) pp. 344–349.
[67] A. Bhargava, M. Suhaib and A. K. S. Singholi, "An omnidirectional mecanum wheel automated guided vehicle control using hybrid modified A* algorithm," *Robotica* **43**(2) 449–498 (2025). https://doi.org/10.1017/S0263574724001954
[68] S. A. Wani, I. Nasiruddin, S. Khatoon and M. Shahid, "Particle Swarm Optimization-Based Optimal Controller for an Autonomous System," **In:** *International Conference on Signal, Machines, Automation, and Algorithm. SIGMAA 2023. Advances in Intelligent Systems and Computing* (H. Malik, S. Mishra, Y. R. Sood, F. P. García Márquez and T. S. Ustun, eds.), vol. **1460** (Springer, Singapore, 2024). https://doi.org/10.1007/978-981-97-6349-8_22
[69] F. Pfeiffer, "Projecting robot dynamics onto trajectories," *Robotica* **41**(7), 2122–2138 (2023). https://doi.org/10.1017/S026357472300036X
[70] H. Houshyari and V. Sezer, "A new gap-based obstacle avoidance approach: Follow the obstacle circle method," *Robotica* **40**(7), 2231–2254 (2022). https://doi.org/10.1017/S0263574721001624
[71] N. Hasan, H. Khan, S. Khatoon and M. Sajid, "Intelligent controller design of an autonomous system using a social spider optimizer for path navigation and obstacle avoidance," *Automatika* **65**(4), 1432–1446 (2024). https://doi.org/10.1080/00051144.2024.2391246
[72] D. Tiozzo Fasiolo, L. Scalera and E. Maset, "Comparing LiDAR and IMU-based SLAM approaches for 3D robotic mapping," *Robotica* **41**(9), 2588–2604 (2023). https://doi.org/10.1017/S026357472300053X
[73] H. Kang, W. Zhang, Y. Ge, H. Liao, B. Huang, J. Wu, R.-J. Yan and I.-M. Chen, "A high-accuracy hollowness inspection system with sensor fusion of ultra-wide-band radar and depth camera," *Robotica* **41**(4), 1258–1274 (2023). https://doi.org/10.1017/S0263574722001692
[74] M. Waseem, K. S. Reddy, T. R. Rao, M. Suhaib and M. A. Khan, "Failure modes, safety concerns, testing protocol, and advancement in lithium-ion battery technology," *Future Batteries* **8**, 100113 (2025). https://doi.org/10.1016/j.fub.2025.100113. ISSN 2950-2640.
[75] W. Tang, Y. Zhou, T. Zhang, Y. Liu, J. Liu and Z. Ding, "Cooperative collision avoidance in multirobot systems using fuzzy rules and velocity obstacles," *Robotica* **41**(2), 668–689 (2023). https://doi.org/10.1017/S0263574722001515
[76] N. Nfaileh, K. Alipour, B. Tarvirdizadeh and A. Hadi, "Formation control of multiple wheeled mobile robots based on model predictive control," *Robotica* **40**(9), 3178–3213 (2022). https://doi.org/10.1017/S0263574722000121
[77] M. Javaid and A. Haleem, "Evolution of Industry 5.0 from Industry 4.0," **In:** *Photonics and Optoelectronics in Industry 5.0. Advances in Optics and Optoelectronics* (V. Bhardwaj, S. Khurana and R. Bhardwaj, eds.), (Springer, Singapore, 2025). https://doi.org/10.1007/978-981-96-7594-4_1
[78] M. Waseem, G. S. Lakshmi, M. Ahmad and M. Suhaib, "Energy storage technology and its impact in electric vehicle: Current progress and future outlook," *Next Energy* **6**, 100202 (2025). https://doi.org/10.1016/j.nxener.2024.100202. ISSN 2949-821X.
[79] A. Bhargava and A. Kumar, "Arduino Controlled Robotic Arm," **In:** *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India (2017) pp. 376–380. https://doi.org/10.1109/ICECA.2017.8212837
[80] H. Khan, S. Khatoon, P. Gaur, M. Abbas, C. A. Saleel and S. A. Khan, "Speed control of wheeled mobile robot by nature-inspired social spider algorithm-based PID controller," *Processes* **11**(4), 1202 (2023). https://doi.org/10.3390/pr11041202

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

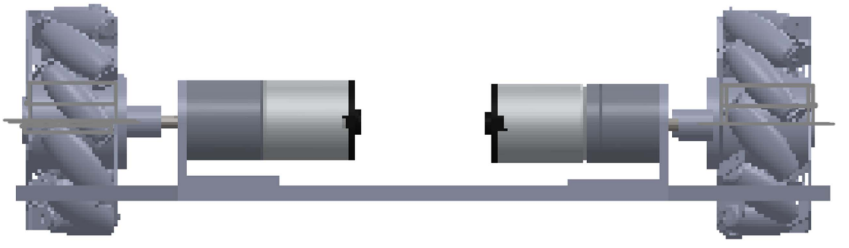
A. Appendix

(a)



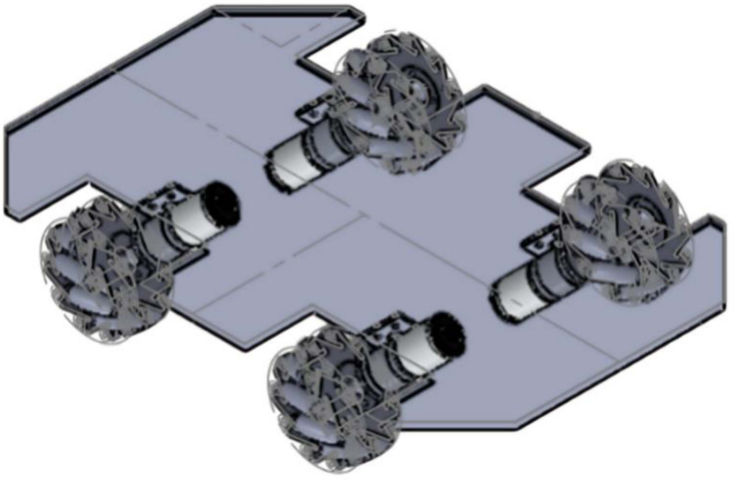
Top view (plan view) of a four-wheeled Mecanum-based AGV.

(b)



Front view of the four-wheeled Mecanum-based AGV.

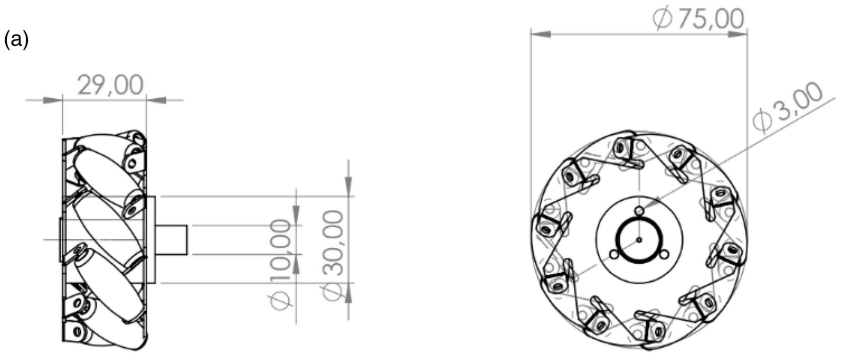
(c)



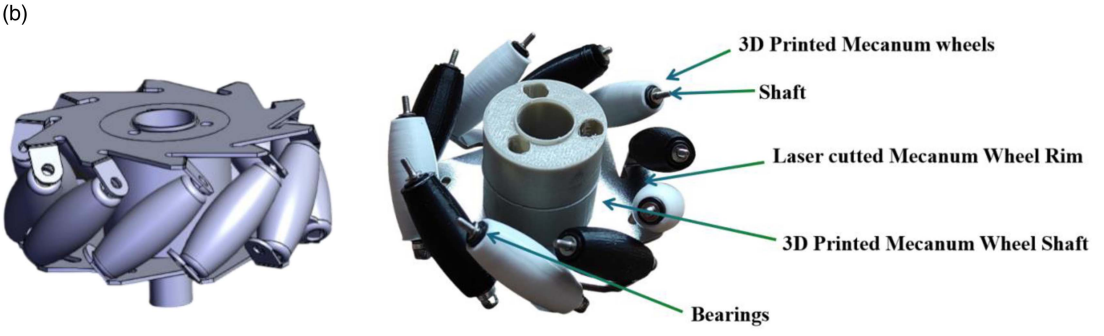
Isometric view of the four-wheeled Mecanum-based AGV.

Figure A1. *Chassis design of the Mecanum wheel AGV.*

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press



Dimensions of the Mecanum wheel.



3D printed Mecanum wheel.

Figure A2. *Overall Mecanum wheel structure.*

Table AI. *Overview of software and hardware.*

CHARACTERISTICS	SPECIFICATIONS FOR TECHNOLOGY
AGV DIMENSIONS	300*200*450 mm
MOTOR TYPE	Brushless motor with a permanent magnet
WEIGHT OF AGV	6 KG
MAXIMUM PAYLOAD	10 KG
TOP SPEED	2.8 m/s
SENSORS & FEEDBACK	Depth Camera, RP LiDAR, and Inertial Measurement Unit
PATH PLANNING ALGORITHMS	D*, APF, DWA, PRM-RRT fusion, PRM-DWA fusion, ADWA-HO.
SIMULATION PLATFORMS	MATLAB, ROS, RViz, and Gazebo
PROCESSING UNITS	The Nvidia Jetson Nano with the B01 model and 4 GB of memory
CONTROL SYSTEM	Neotic in Ubuntu 20.04
MOTOR RATED CURRENT	360mA
RATED TORQUE	2.2 kgf.cm
MOTOR VOLTAGE RATE	10V
RATING OF MOTOR POWER	5W
WHEEL	Omnidirectioanl Mecanum Wheel
DEPTH CAMERA	Astra Pro and Intel D455

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press

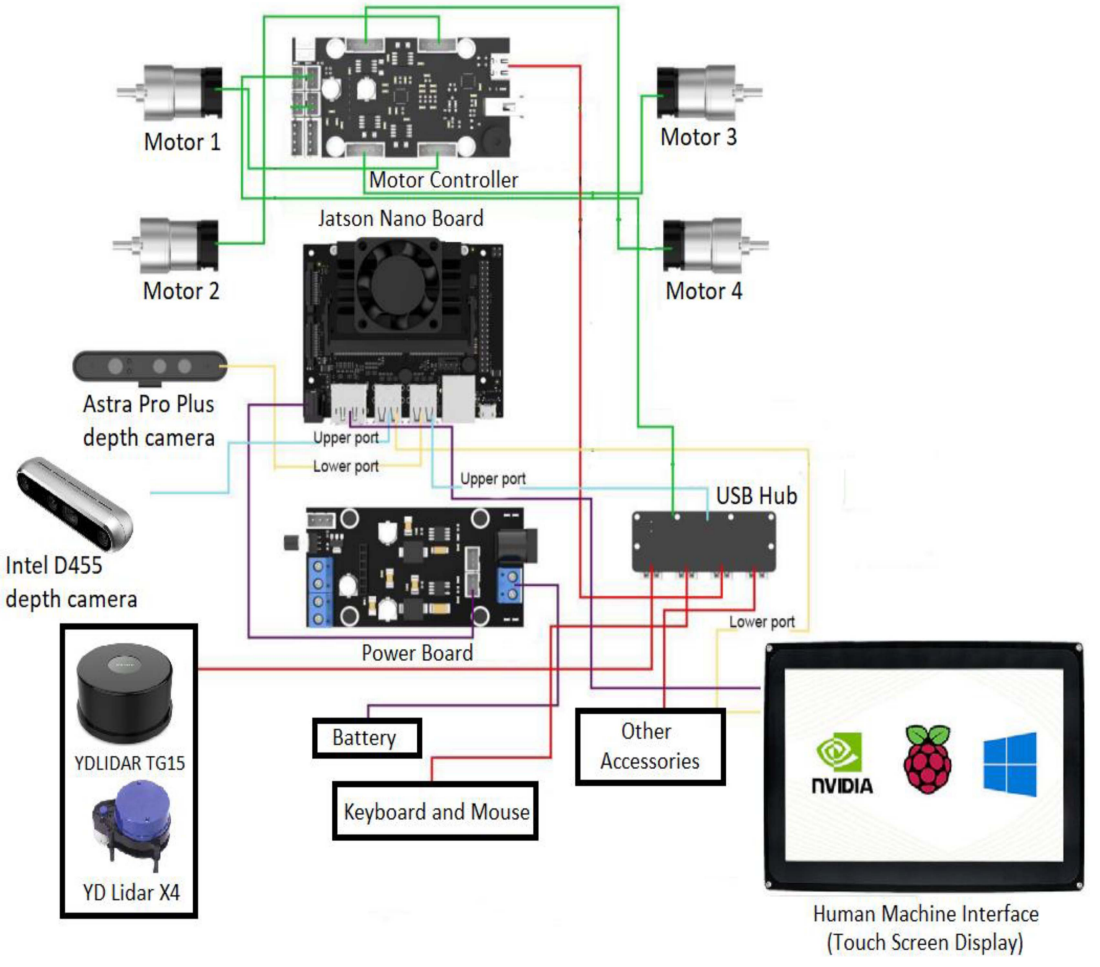


Figure A3. *Hardware system wiring diagram.*

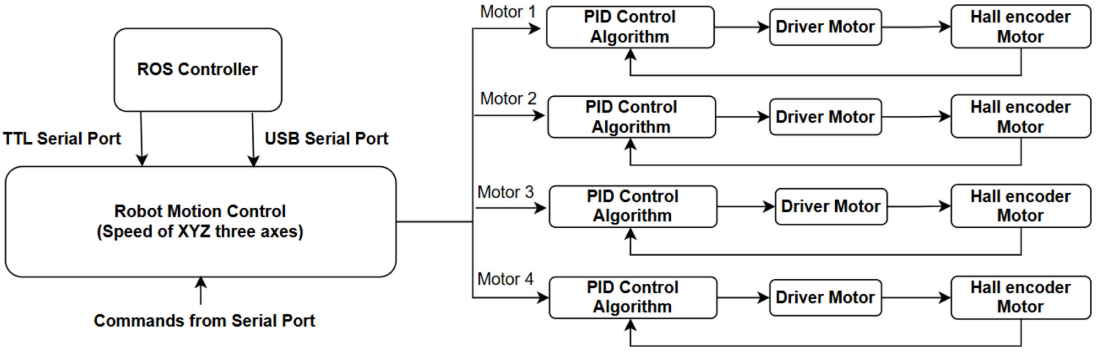


Figure A4. *Flowchart of the motor control process.*

Cite this article: A. Bhargava, M. Suhaib and A. K. S. Singholi (2026). "A hybrid optimization algorithm combining A* and the dynamic window approach for automated guided vehicle control in static and dynamic environments", Robotica **44**, 408–470. https://doi.org/10.1017/S0263574725102944

https://doi.org/10.1017/S0263574725102944 Published online by Cambridge University Press