



OPEN

Research on autonomous navigation of mobile robots based on IA-DWA algorithm

Quanling He^{1,2}, Zongyan Wang^{1,2}, Kun Li¹, Yuting Zhang^{1,2} & Menglong Li¹

To improve the efficiency of mobile robot movement, this paper investigates the fusion of the A* algorithm with the Dynamic Window Approach (DWA) algorithm (IA-DWA) to quickly search for globally optimal collision-free paths and avoid unknown obstacles in time. First, the data from the odometer and the inertial measurement unit (IMU) are fused using the extended Kalman filter (EKF) to reduce the error caused by wheel slippage on the mobile robot's positioning and improve the mobile robot's positioning accuracy. Second, the prediction function, weight coefficients, search neighborhood, and path smoothing processing of the A* algorithm are optimally designed to incorporate the critical point information in the global path into the DWA calculation framework. Then, the length of time and convergence speed of path planning are compared and simulated in raster maps of different complexity. In terms of path planning time, the algorithm reduces by 23.3% compared to A*-DWA; in terms of path length, the algorithm reduces by 1.8% compared to A*-DWA, and the optimization iterations converge faster. Finally, the reliability of the improved algorithm is verified by conducting autonomous navigation experiments using a ROS (Robot Operating System) mobile robot as an experimental platform.

Keywords Mobile robots, Autonomous navigation system, A* algorithm, DWA algorithm, ROS

Path planning technology is crucial for ensuring that mobile robots can navigate safely and efficiently through complex and dynamic environments¹. In this process, the robot must determine a precise and adaptable path based on its surroundings². The planned route should minimize travel distance while ensuring that the path is smooth and unobstructed to prevent collisions and enhance operational safety³. Among the various algorithms, the A* algorithm is one of the most widely used and holds a central role in path planning. However, there is still considerable room for improvement and optimization. Yang et al.⁴ have innovatively combined the A* search algorithm with the Artificial Potential Field (APF) method to enhance safety and ensure smoother global path planning for autonomous vehicles, particularly addressing vehicle turning constraints. Zhang et al.⁵ developed a path fitting and navigation technique that combines the improved A-Star algorithm and Support Vector Machine Regression (SVR) to meet the needs of autonomous navigation robots for accurate autonomous navigation in an orchard environment. Then, a minimum radar cross-section (RCS) tactic is proposed for path replanning against robber threats. Duan et al.⁶ integrated the DWA with the enhanced A* algorithm for robot navigation. This approach adaptively adjusts the cost function to enhance search efficiency by incorporating environmental data into the augmented A* algorithm. To optimize the global path, critical nodes are prioritized for planning, unnecessary nodes are removed, and only essential path points are retained. The DWA evaluation function is then based on data from these critical nodes, with the local path defined through dynamic window regulation, focusing on critical nodes as midpoint targets to improve path smoothness while achieving global path optimization and real-time obstacle avoidance⁷. In environments that are variable and unpredictable, relying solely on global path-planning algorithms often proves inadequate, as these methods cannot anticipate or avoid unknown obstacles⁸. Therefore, during navigation, mobile robots must employ more detailed and specific local planning strategies to promptly avoid obstacles. Common local path planning strategies include APF⁹, DWA¹⁰, and temporal elastic band algorithms¹¹.

The DWA algorithm uses path information generated by global path planning¹² to perform local path planning and then outputs the robot's velocity information to drive the robot. Jiang et al.¹³ applied a two-delay depth deterministic strategy gradient algorithm to an automated guided vehicle (AGV) system. This approach generates auxiliary candidate trajectories through reinforcement learning in the prediction domain and

¹North University of China, School of Mechanical Engineering, Taiyuan 030051, Shanxi, China. ²Key Laboratory of Digital Design and Manufacturing in Shanxi Province, Taiyuan 030051, Shanxi, China. email: wangzongyan321@sina.com

introduces a variable sampling mechanism to enhance the flexibility and adaptability of trajectory generation. Liu et al.¹⁴ focused on optimizing the trajectory evaluation function of the DWA, improving it to make local path planning smoother and more coherent. This enhancement significantly reduces the difficulty the robot faces in executing motion commands and improves overall navigation efficiency. Wang et al.¹⁵ redefined obstacle avoidance as an optimal learning incentive problem and optimized the DWA obstacle avoidance trajectory evaluation function by integrating the reward incentive mechanism of deep deterministic policy gradient with experience replay techniques. This approach not only addresses the limitation of the DWA algorithm's tendency to fall into local optima in complex environments but also solves challenges related to the robot's motion control during continuous speed changes and steering angle selections, thereby achieving a more intelligent and flexible obstacle avoidance strategy.

Numerous studies mentioned above have achieved good planning path results for planning local or global paths for mobile robots. However, the traditional A* and DWA algorithms have slow iteration speeds and poor robustness for autonomous navigation of mobile robots in multiple indoor environments. Simple serial superposition of relations may lead to the deterioration of path planning efficiency, which leads to the problem of long task execution time for mobile robots. Therefore, to address the above problems, this study proposes an IA-DWA algorithm, which makes it suitable for fast response of global and local path planning for mobile robots, improves the efficiency of obstacle avoidance, and reduces the response time. The paper makes the following contributions:

1. The A* global path planning algorithm is improved by refining aspects such as heuristic function selection, weight coefficients, search neighborhoods, and search strategies. This optimized algorithm is then integrated with the DWA algorithm.
2. Improved IA-DWA simulates path planning through raster maps of varying complexity. Evaluate the improved method, the original A* algorithm, and the DWA algorithm to verify their reliability.
3. A mobile robot experimental platform is constructed for SLAM and autonomous navigation experiments, verifying the feasibility of the proposed research.

Methodology

Odometer and IMU data fusion

In this paper, raster maps are employed to construct the map model, and IMU sensors are incorporated to enhance map construction accuracy. The common EKF algorithm is used to fuse information from the IMU and odometer, reducing errors caused by wheel slippage and providing more precise positioning for the mobile robot. The IMU, which includes a gyroscope and an accelerometer, measures an object's angular velocity and acceleration in three-dimensional space¹⁶. The fusion of IMU and odometer data using EKF is illustrated in Fig. 1.

Robot global path planning

This paper delves deeply into optimization strategies for the A* algorithm, significantly enhancing its performance through multi-dimensional refinements. A new path planning and obstacle avoidance algorithm is developed by integrating the optimized A* algorithm with the DWA. To demonstrate the effectiveness of this integrated approach, simulations are conducted on the MATLAB platform, showcasing the path planning and obstacle avoidance processes in complex environments.

1) Selection of the heuristic function

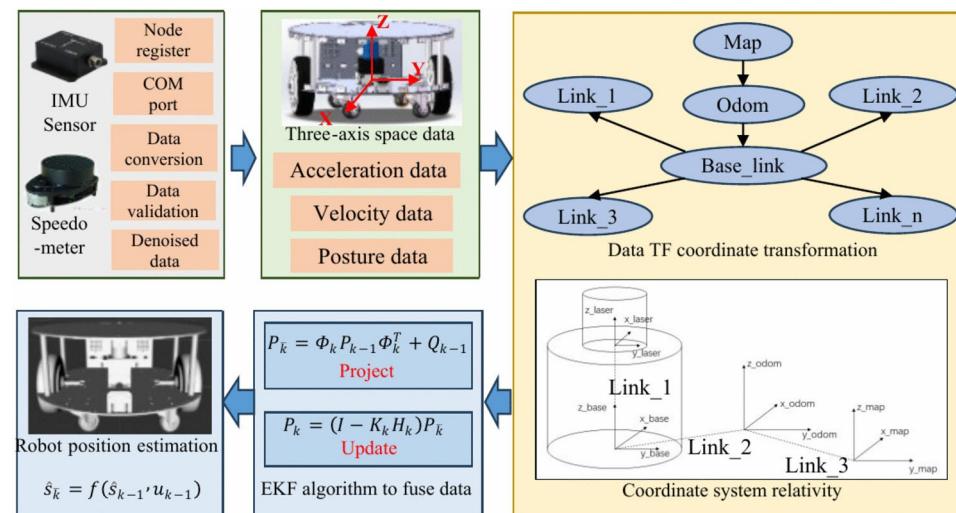
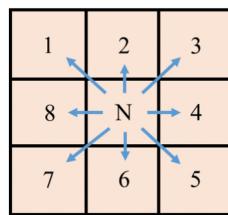


Fig. 1. Flow chart of IMU and odometer data fusion.

**Fig. 2.** Graph of search nodes.

α	Direction of reservations	Direction of abandonment
$[337.5^\circ, 360^\circ) \cup [0^\circ, 22.5^\circ)$	1,2,3,4,8	5,6,7
$[22.5^\circ, 67.5^\circ)$	1,2,3,4,5	6,7,8
$[67.5^\circ, 112.5^\circ)$	2,3,4,5,6	1,7,8
$[112.5^\circ, 157.5^\circ)$	3,4,5,6,7	1,2,8
$[157.5^\circ, 202.5^\circ)$	4,5,6,7,8	1,2,3
$[202.5^\circ, 247.5^\circ)$	1,5,6,7,8	2,3,4
$[247.5^\circ, 292.5^\circ)$	1,2,6,7,8	3,4,5
$[292.5^\circ, 337.5^\circ)$	1,2,3,7,8	4,5,6

Table 1. Direction of search rules.

The heuristic function has a great impact on the performance of the A* algorithm. Its main role is to guide the A* algorithm towards the right goal. In this paper, the European distance formula is used.

2) Weighting factor

Unlike traditional A* algorithms¹⁷, path prediction of the improved method incorporates arcs and straight-line segments instead of relying solely on straight lines. Dynamic constraints and various factors influence the duration of both straight and arc trajectories differently. For enhanced accuracy, the prediction process is segmented into steering and straight-line travel components. Consequently, the prediction function equations have been revised to Eqs. (1–3).

$$f(n) = g(n) + \left(1 + \frac{d}{D}\right) (h_1(n) + h_2(n)) \quad (1)$$

$$h_1(n) = \left\lceil \frac{\Delta\varphi_n}{\varphi_{\max}} \right\rceil * dT \quad (2)$$

$$h_2(n) = \frac{\sqrt{(x_t - x_{on})^2 + (y_t - y_{on})^2 - R^2}}{v_{\max}} \quad (3)$$

where d is the distance from the current state to the target, D is the distance from the start state to the target, $\Delta\varphi_n$ is the deviation of the current state of the robot from the direction of the steering target; φ_{\max} is the maximum deviation of the robot's direction within the stroke dT ; and the coordinates of the target. x_{on} and y_{on} are the coordinates of the turning center; v_{\max} is the maximum speed of the robot.

3) Search neighborhoods

Based on the relative positions of the endpoints and the robot¹⁸, the direction search without results is canceled, thus reducing the waste of time and improving the robot's pathfinding efficiency. In the process of searching the path, the angle between the line connecting the current point and the target point and the direction of the number 2 is set to α . Figure 2 shows the search node graph. In this paper, new direction search rules in Table 1 are developed.

4) Search strategy

The A* algorithm primarily aims to select the path with the shortest distance during the search process¹⁹. However, this approach results in traversing an excessive number of nodes, which adversely impacts search efficiency. To address this issue, we propose a bidirectional A* search strategy. This approach mitigates redundant

node computations by simultaneously searching paths from both the start and end points in both directions. A detailed flowchart of the bidirectional A* search algorithm is shown in Fig. 3.

5) Path smoothing

To ensure continuity and smoothness of the path and avoid wheel slippage, three quasi-uniform B-spline curves are employed to enhance the optimal path's smoothness. To tackle issues such as inadequate B-spline curve fitting and excessive curvature, we propose an improved path-smoothing method for these three quasi-uniform B-spline curves. This method adaptively adjusts the path control points to meet the desired constraints. By increasing the number of control points, smoother curves that more closely approximate the optimal path are achieved. Various coefficients are applied to adjust the number of control points based on the mobile robot's corner dimensions. Equation (4) provides the formula for selecting the control point distances.

$$x = \frac{d}{3} + \left(\frac{d}{5} - \frac{d}{3} \right) \cdot \frac{90 - \theta}{90} \quad (4)$$

where the turning angle is θ , the turning distance is d , and the control point distance is x . When $\theta \geq 90$, $x = \frac{d}{3}$; when $\theta < 90$, x is between $\frac{d}{5}$ and $\frac{d}{3}$, depending on the turning angle.

The polynomial form of the triple quasi-uniform B-spline curve is Eq. (5).

$$P(u) = [P_0 P_1 \cdots P_n] \begin{bmatrix} B_{0,3}(u) \\ B_{1,3}(u) \\ \vdots \\ B_{n,3}(u) \end{bmatrix} = \sum_{i=0}^n P_{i,3}(u) \quad (5)$$

When $n = 3$, the basis function is Eq. (6).

$$\begin{cases} B_{0,3}(u) = \frac{1}{6}(1-t)^3 \\ B_{1,3}(u) = \frac{1}{6}(3u^3 - 6u^2 + 4) \\ B_{2,3}(u) = \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1) \\ B_{3,3}(u) = \frac{1}{6}u^3 \end{cases} \quad (6)$$

The expression for the curvature of the triple-fitted uniform b-sample curve is Eq. (7).

$$\rho(u) = \frac{p''(u) * p'(u)}{p'(u)^3}, \quad \rho(u) \leq \rho_{\max} \quad (7)$$

When the road curvature at a corner fails to meet the curvature constraints, the smoothing section is updated by resetting the control points. This involves replacing the node that does not satisfy the curvature constraint with

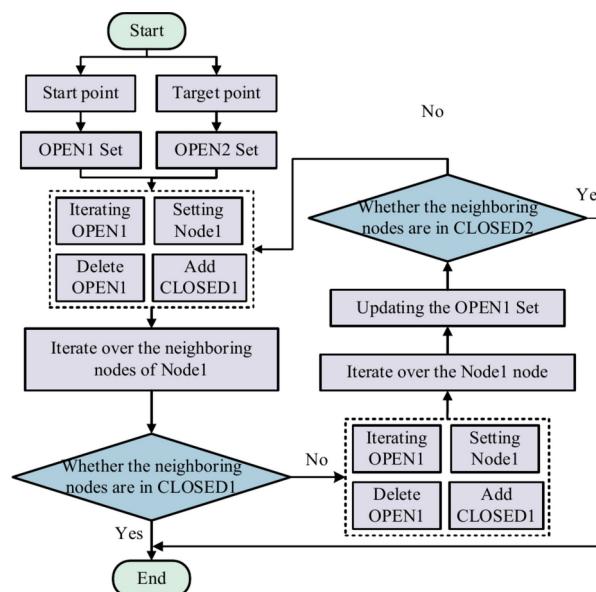
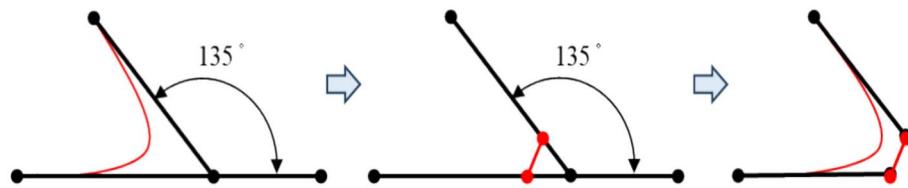
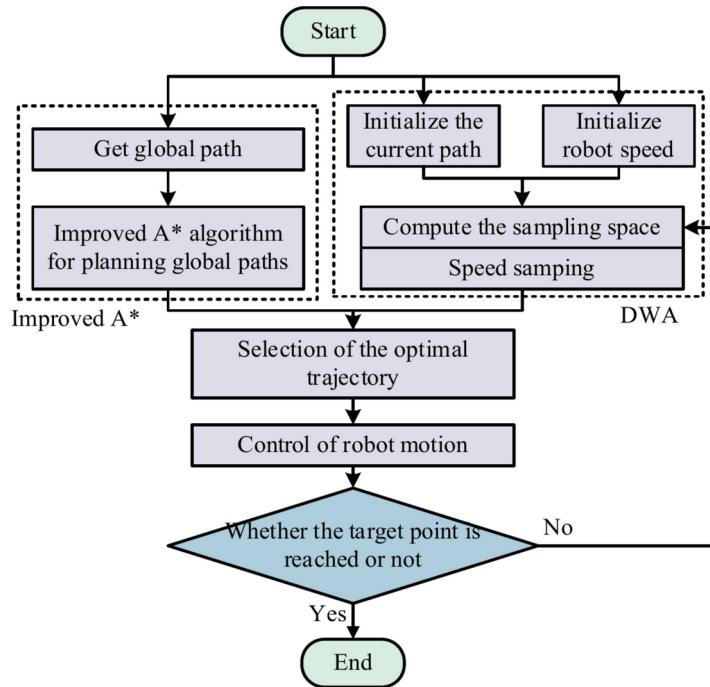


Fig. 3. Flow of the bi-directional search A* algorithm.

**Fig. 4.** Improved smoothing method operation diagram.**Fig. 5.** Integration of processes to improve A* and DWA.

two new nodes. As shown in Fig. 4, the two control points, as determined by Eq. (4), together with the corner points, create an isosceles triangle. This adjustment transforms the original obtuse angle into two acute angles, resulting in a refitted path that complies with the maximum curvature requirement.

Fusion of global and local path planning algorithms

In this study, a novel path-planning algorithm is developed by integrating the improved A* algorithm with the DWA path-planning technique. This integration significantly enhances the global search capability of the DWA algorithm, allowing the robot to consistently follow the optimal path by incorporating key point information from the global path into its computational framework. This approach not only improves the accuracy of path selection but also ensures that the robot's trajectory steadily converges towards the global optimum, thereby effectively enhancing the robot's efficiency and stability, as optimized in Eq. (8).

$$g(\partial, \lambda) = \eta[\alpha * p_dist(\partial, \lambda) + \lambda * g_dist(\partial, \lambda) + \beta * dist(\partial, \lambda) + \gamma * vel(\partial, \lambda)] \quad (8)$$

Equation (10), $p_dist(\partial, \lambda)$ is the distance from the simulated target point to the global path and $g_dist(\partial, \lambda)$ is the distance from the target point to the critical point. The flow of the fusion algorithm is shown in Fig. 5.

Simulation and analysis of results

To test the feasibility of the fusion algorithm in autonomous navigation, comparative experiments with different types of raster maps were conducted in MATLAB R2023b. Figure 6 shows three maps with different levels of complexity. Simulating the path planning algorithm in different map environments can better verify the effectiveness and robustness of the algorithm²⁰. Figure 6a is for a simple obstacle map, Fig. 6b is for an irregular obstacle map, and Fig. 6c is for a complex map with multiple obstacles. The three maps mentioned are referred to as Map-1, Map-2, and Map-3.

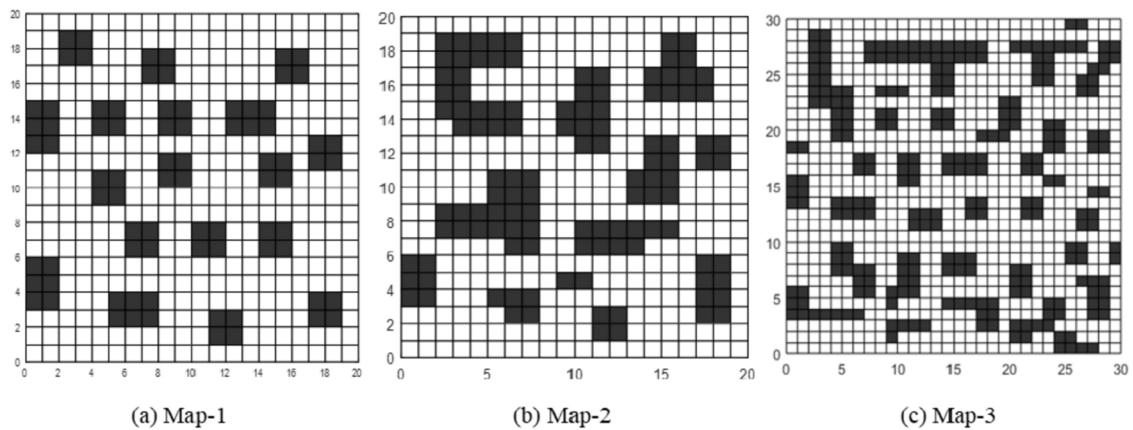
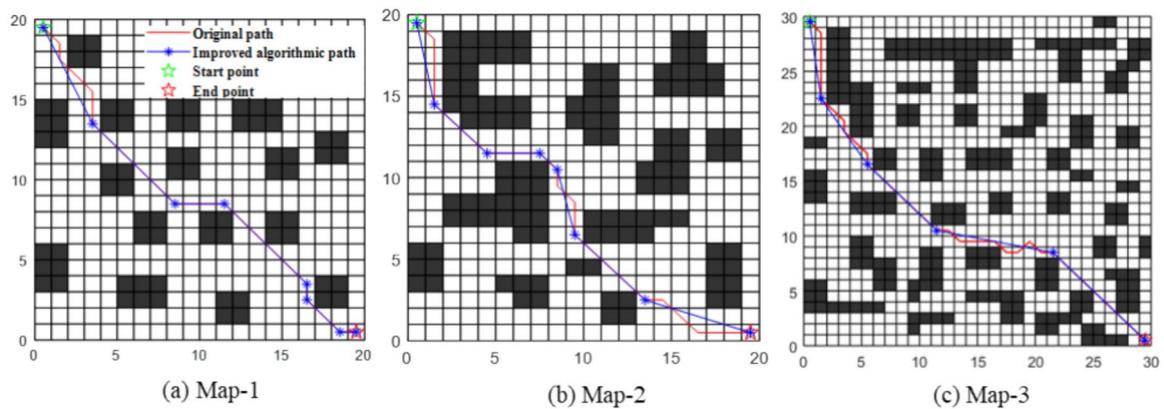
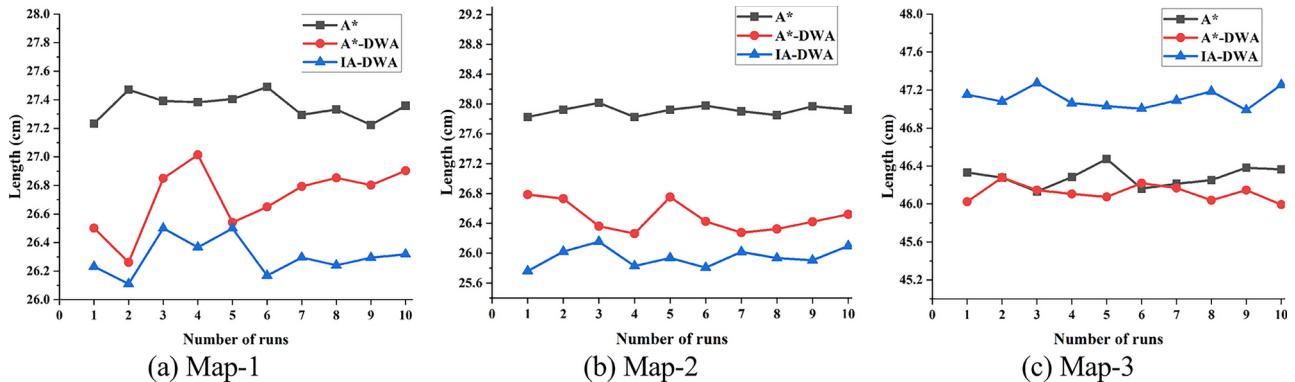
**Fig. 6.** Different types of raster maps.**Fig. 7.** The effect of different map path planning.**Fig. 8.** Comparison of autonomous navigation path lengths.

Figure 7a shows path planning in a low-complexity map. Its obstacle distribution is sparse, and the improved algorithm path planning is straighter. Figure 7b shows the corresponding medium complexity map with diverse obstacle distribution the improved algorithm path removes redundant nodes and avoids dense regions. Figure 7c shows a high-complexity map where the improved algorithm's planned path is shorter than the original algorithm and no redundant nodes are generated.

Figure 8 compares the path length results of automatic path navigation. In Fig. 8a, the IA-DWA algorithm requires an average path length of 26.3 cm, 1.03 cm shorter than the original A* algorithm and 0.42 cm shorter than the A*-DWA algorithm. In Fig. 8b, the IA-DWA algorithm requires an average path length of 25.9 cm,

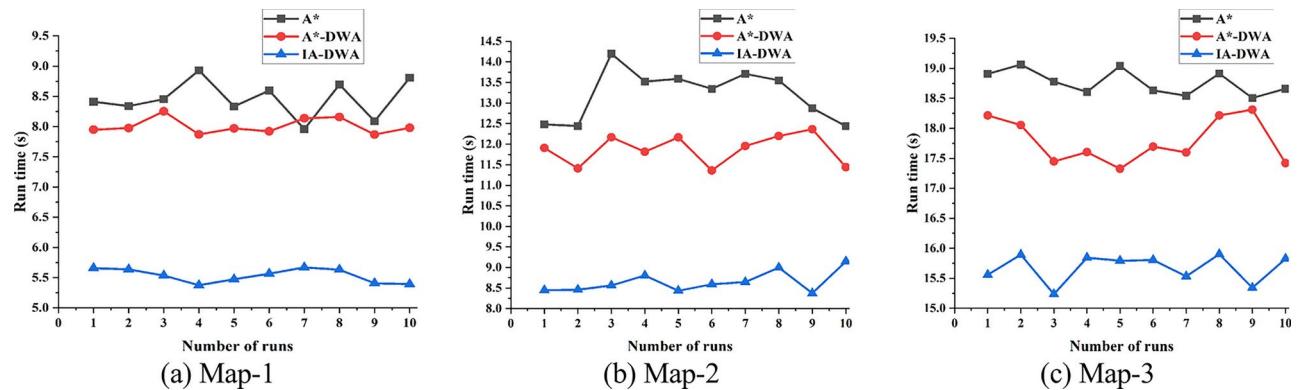


Fig. 9. Comparison of autonomous navigation turning times.

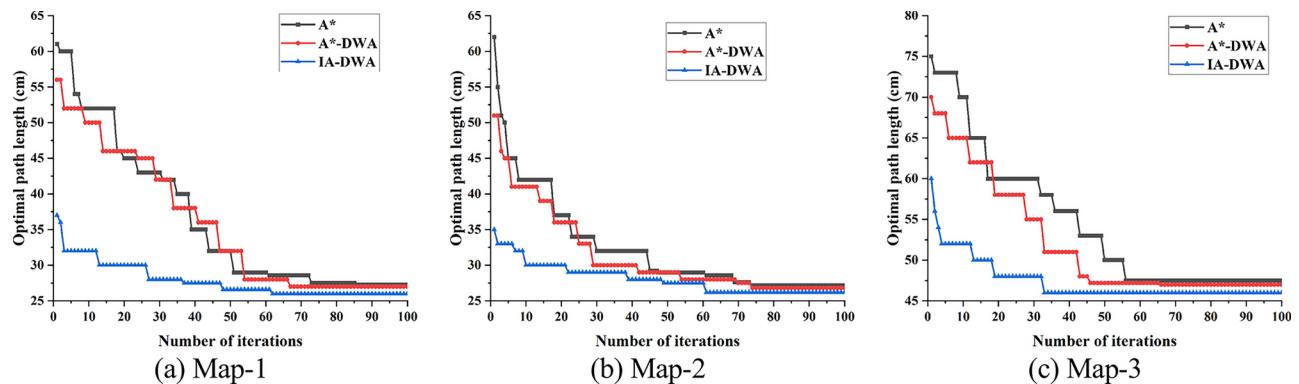


Fig. 10. Comparison of autonomous navigation iteration process.

1.63 cm shorter than the original A* algorithm and 0.49 cm shorter than the A*-DWA algorithm. In Fig. 8c, the IA-DWA algorithm has an average path length of 46.12 cm, 1.04 cm shorter than the original A* algorithm and 0.25 cm shorter than the A*-DWA algorithm. The IA-DWA algorithm shows the most significant improvement on complex maps.

The rotation times of the autonomous navigation cart are illustrated in Fig. 9. In Fig. 9a and b, the IA-DWA algorithm achieves the shortest rotation times, with reductions of 2.54 s and 3.18 s compared to the original A* and A*-DWA algorithms, respectively. In Fig. 9c, due to the increased map size, the IA-DWA algorithm still demonstrates superior performance, requiring 3.54 s less rotation time than the original A* algorithm and 3.98 s less than the A*-DWA algorithm.

The iterative process of the optimized path algorithm for the autonomous navigation cart is shown in Fig. 10. In Fig. 10a and b, the IA-DWA algorithm achieves the fastest iteration speed, obtaining the optimal path after 60 iterations with the shortest path length. In Fig. 10c, the IA-DWA algorithm converges after 33 iterations, whereas the A* and A*-DWA algorithms converge more slowly, averaging 59 iterations. This demonstrates the reliability of the improved algorithms in path planning.

Experiment

Hardware platform construction

This section presents the construction of an experimental platform using a microcomputer mainframe and LIDAR, among other components. The platform is then utilized to analyze the navigation system. A two-wheel differential mobile robot realizes autonomous navigation experiment in real environments. The robot hardware platform is shown in Fig. 11. The built mobile robot motion control substrate mainly consists of modules such as a microcomputer host, a one-two dual-drive hub driver, two 400 motors, and a power supply. One of the key feedback elements is the wheel encoder of the MDR400 dual motor drive, which is mainly used to obtain information such as wheel speed, displacement, and direction for precise control. Specific MDR400 motor drive parameters are shown in Table 2.

Experiments in SLAM mapping for mobile robots

To evaluate the effectiveness of the mobile robot in mapping after integrating the IMU data, we chose a corridor. Obstacles were placed to assess the ability of the mobile robot to localize the map. The experimental setup is shown in Fig. 12.



Fig. 11. Hardware platforms for mobile robots.

Functional parameters	Value
Input voltage	DC12-55 V
Communication link	RS232
Maximum continuous current	13A*2
Encoder resolution	1000 PPR

Table 2. MDR400 motor drive parameter.

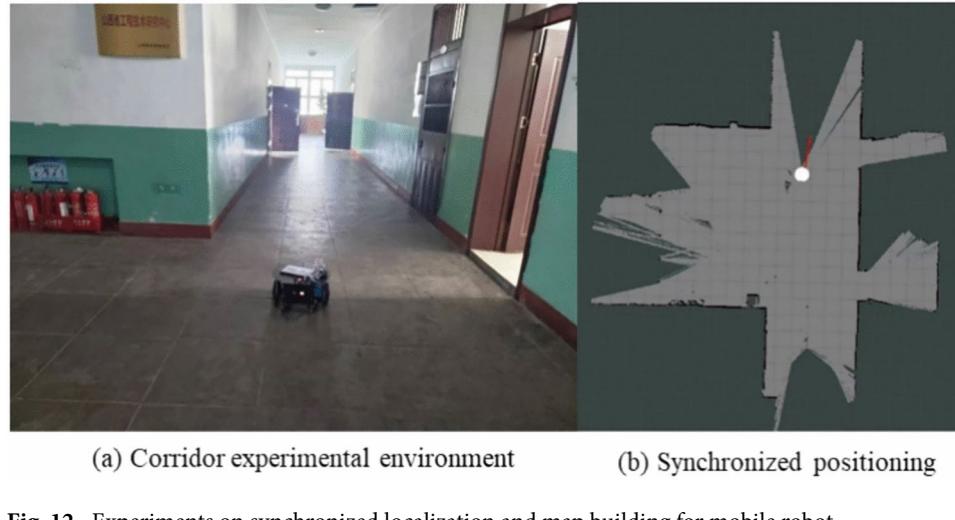
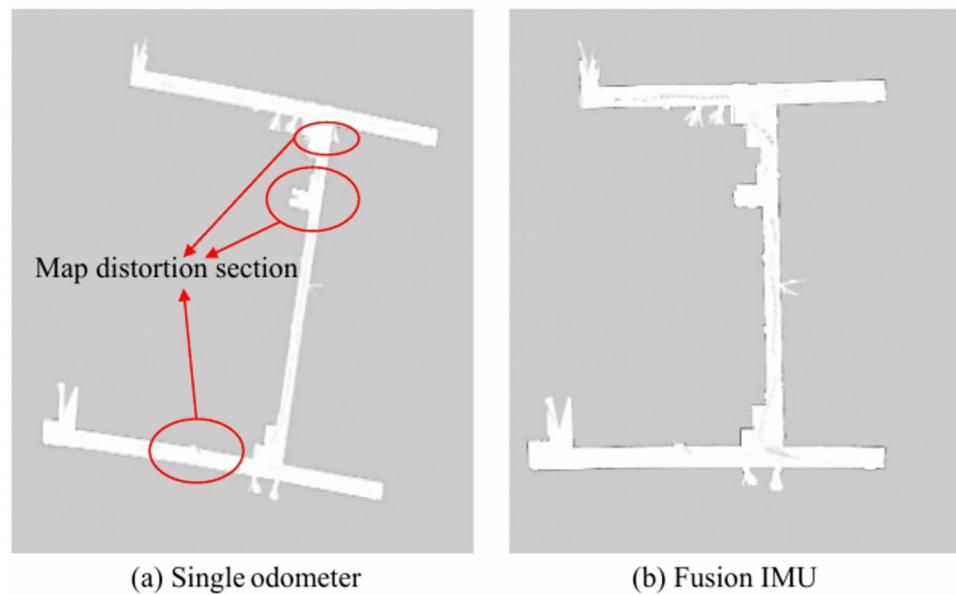


Fig. 12. Experiments on synchronized localization and map building for mobile robot.

Figure 13 shows a map of the corridor. Initially, the ends of the corridor and the middle section were selected as measurement points. At each designated location, the width was measured three times using a tape measure, and the average width was calculated. Subsequently, by analyzing the number of raster and their dimensional proportions in the real environment, the corridor's width at each measurement point on the map was accurately determined, as shown in Tables 3 and 4. The tables indicate that while the initial error of the map is minimal, the error in the robot's map building increases as the distance traveled by the mobile robot grows. Comparison results show that the maps generated by integrating odometer and IMU data have less error and reflect the real environment more accurately.

The basic framework of the Gmapping algorithm is built in a physical prototype of a mobile robot. In order to ensure the reliability of subsequent autonomous navigation and avoid the influence of navigation and localization errors on trajectory planning, therefore, multiple SLAM methods are compared, and the SLAM with the highest accuracy is used as the basic framework, as shown in the Table 5 for the comparison results.

As shown in Table 5, the RBPF-SLAM algorithm has the worst localization effect. The Hector-SLAM algorithm has a worse localization effect in the corridor environment, with a relative error of 6.3%. The Kart SLAM algorithm relies on the accuracy of the odometer, with an absolute error of 0.04 m and a relative error of

**Fig. 13.** Map of the corridor.

Position	Measured value/m	Map value/m	Measurement error/m	Percentage of error/%
1	2.75	2.70	0.05	1.8
2	2.10	2.01	0.09	4.3
3	2.76	2.58	0.18	6.5

Table 3. Accuracy of single odometer raster maps.

Position	Measured value/m	Map value/m	Measurement error/m	Percentage of error/%
1	2.75	2.73	0.02	0.7
2	2.10	2.07	0.03	1.4
3	2.76	2.70	0.06	2.2

Table 4. Accuracy of raster maps incorporating odometer and IMU.

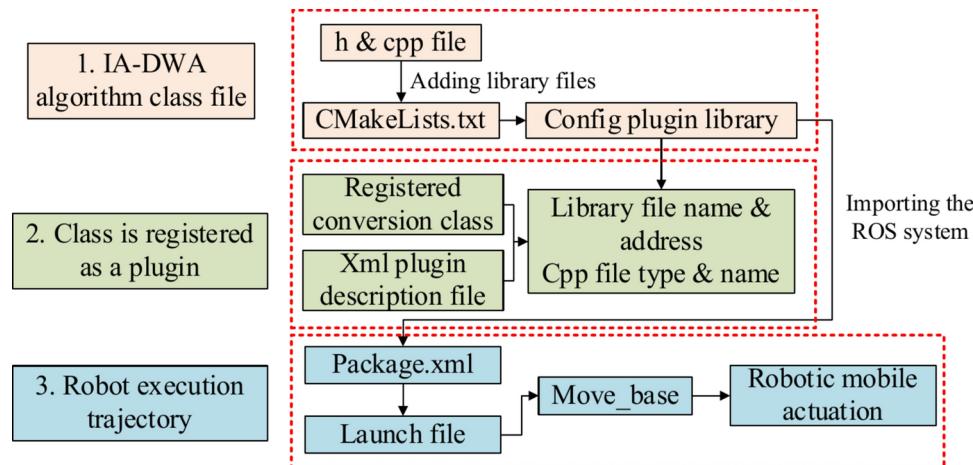
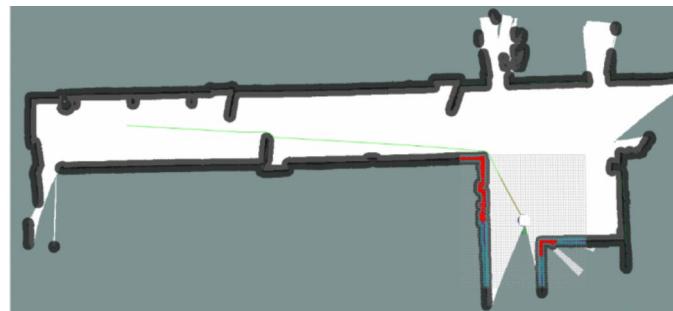
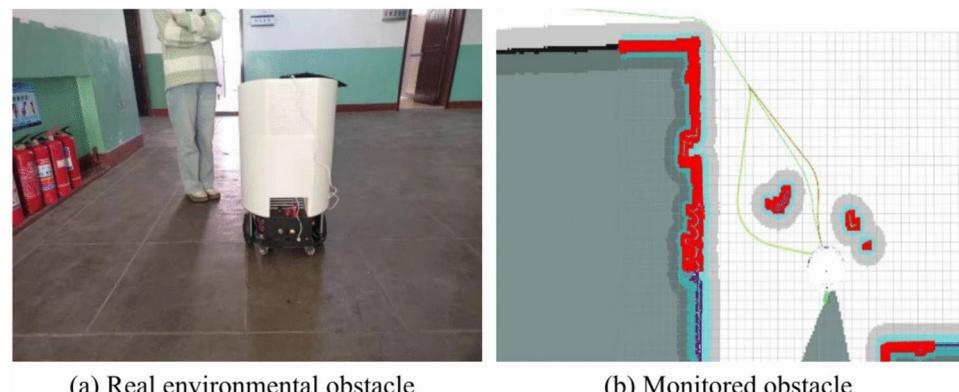
Algorithm categories	Absolute positioning errors/m	Relative error/%	Runtime/ms
RBPF-SLAM	0.06	6.3	15.98
Gmapping ²¹	0.03	3.2	13.56
Hector-SLAM	0.04	3.5	22.70
Karto SLAM ²²	0.03	3.2	15.72

Table 5. Comparison results of different SLAM algorithms.

3.5%. The Gmapping and Karto SLAM algorithms have the best effect in indoor map building, but to ensure real-time operation, Gmapping with a shorter running time is chosen as the localization and map building method.

Mobile robot autonomous navigation experiment

The IA-DWA algorithm is deployed in the ROS system to make it applicable in the current trajectory path planning and execution of mobile robots. The algorithm deployment and packaging process is shown in Fig. 14. First build the IA-DWA class file, add the cpp and h file to the library of CMakeLists, register the class to convert it to a structured Xml file, which contains the library name, address, and type. Import the library file into the ROS system, use the Launch execution file to command the robot to perform basic mobile operations, including path position movement, rotation, and obstacle avoidance operations, and finally realize the path planning process of the mobile robot entity.

**Fig. 14.** The algorithm deployment and packaging process.**Fig. 15.** Global path planning for mobile robot.**Fig. 16.** Robot encounters a dynamic pedestrian situation.

When an endpoint is designated in the Rviz software, the robot can employ this algorithm to devise a comprehensive global path, as illustrated in Fig. 15. The results show that the fusion algorithm presented in this study notably elevates the efficacy of path planning, demonstrating superior adaptability and stability.

Mobile robots may encounter both static and dynamic unknown obstacles while navigating in real-world environments. Thus, static unknown barriers are introduced, and dynamic, unpredictable obstacles with pedestrians are modeled. This complex setup enables a comprehensive evaluation of the performance of the obstacle avoidance strategies and algorithms employed by the mobile robot when facing unknown obstacles. Figure 16 shows a scenario where the robot encounters dynamic pedestrians during navigation. As the robot begins its journey from the starting point, it can promptly scan for pedestrians as they appear, allowing it to plan a path that avoids these dynamic obstacles promptly.

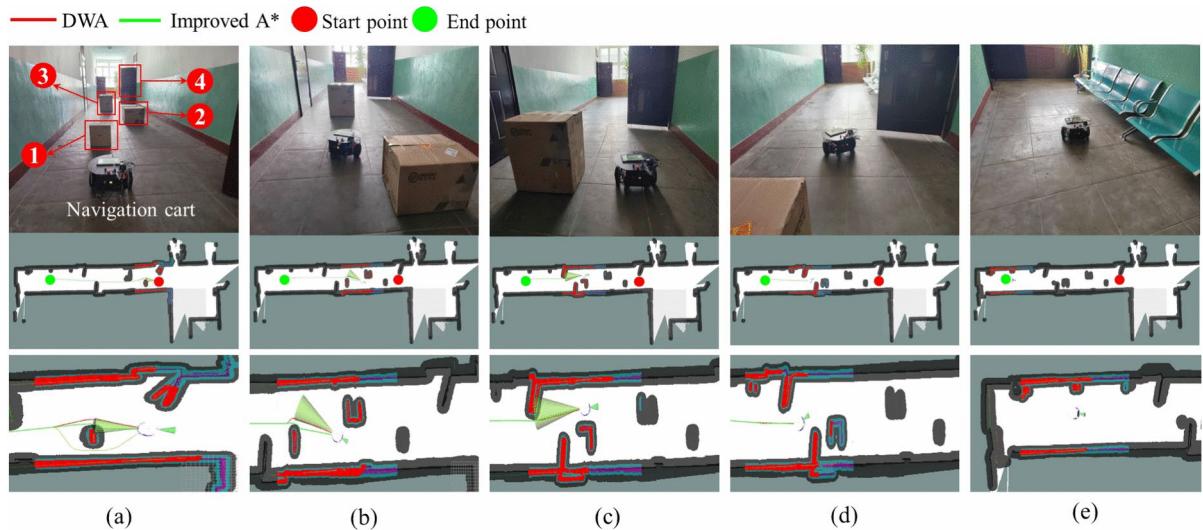


Fig. 17. Motion process of a robot avoiding an unknown static obstacle **(a)** scanning for obstacles; **(b)** dodge the first obstacle; **(c)** dodge the second obstacle; **(d)** dodge the third obstacle; **(e)** get to the end point.

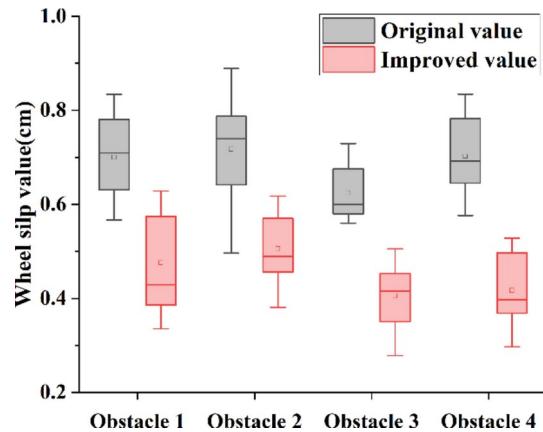


Fig. 18. Wheel slip before and after using the improved algorithm.

Figure 17 shows the robot navigating toward the target point while avoiding obstacles. Global paths are presented as green curves and local paths are presented as red curves. Figure 17 demonstrates that the robot avoids obstacles and continues along the optimal path. The local path consistently converges towards the global optimal state during local obstacle avoidance. In this autonomous navigation experiment, the improved algorithm enables the mobile robot to swiftly identify a smooth, collision-free path with the shortest distance, allowing it to reach the designated target point safely, the experimental results verify the feasibility of the algorithm in this paper.

In above Fig. 17, there are four obstacles. The IA-DWA algorithm employs a new path-smoothing strategy to improve the path execution accuracy. Therefore, wheel slip is used as an important evaluation index. This experiment monitors the wheel slip phenomenon when passing through four obstacles. The wheel slip before and after using the improved algorithm is shown in Fig. 18.

As shown in Fig. 18, 20 sets of trajectories are navigated using the original method with improved algorithms, and data collection is performed on the slip. The average error of autonomous navigation of the mobile robot through four obstacles in an indoor hallway is between 0.56 and 0.89 cm. Using the improved method, path smoothing resulted in a decrease in wheel slip, and the average error of the mobile robot passing four obstacles was between 0.24–0.62 cm, and the overall slip decreased by 0.29 cm, which verified the reliability of trajectory smoothing in the improved algorithm for reducing wheel slip.

Conclusion

In this paper, IMU sensors are integrated with the Gmapping algorithm to enhance robot localization, and the odometer and IMU data are fused using the EKF to achieve more accurate position estimation. The improved A* algorithm and DWA algorithm are combined for path planning, and the effectiveness of this

fusion in path planning and obstacle avoidance is demonstrated through simulation, achieving full autonomous navigation of the mobile robot. An experimental platform for the mobile robot is then constructed, and map construction and autonomous navigation experiments are conducted in various indoor environments. In the map construction experiment, maps generated before and after fusing odometer and IMU data are compared, showing that the fused data significantly improves map accuracy. In the autonomous navigation experiments, the robot successfully plans a smooth, collision-free path with the shortest distance to reach the target point safely. These experiments demonstrate that the proposed autonomous navigation system significantly enhances the intelligence and automation of robots.

The current work lacks algorithmic trajectory optimization for mobile robots under kinematic and dynamic constraints, and there is a lack of research on trajectory planning for dynamic obstacles. In future work, the improvement method should be carried out in a variety of factory or indoor environments for operational experiments and consider dynamic obstacles and human–robot collaboration for further performance enhancement of the IA-DWA algorithm. The mobile robot based on the IA-DWA algorithm can be applied in the future to multiple complex environments such as multi-task point path handling under factories or indoor cleaning and disinfecting, to achieve global and local path optimization and to improve the efficiency of executing tasks.

Data availability

The data generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Received: 27 August 2024; Accepted: 27 December 2024

Published online: 15 January 2025

References

- Li, F. F., Du, Y. & Jia, K. J. Path planning and smoothing of mobile robot based on improved artificial fish swarm algorithm. *Sci. Rep.* **12**, 659. <https://doi.org/10.1038/s41598-021-04506-y> (2022).
- Wakchaure, M., Patle, B. K. & Mahindrakar, A. K. Application of AI techniques and robotics in agriculture: A review. *Artif. Intell. Life Sci.* **3**, 100057. <https://doi.org/10.1016/j.ailsci.2023.100057> (2023).
- Yang, J., Ni, J., Xi, M., Wen, J. & Li, Y. Intelligent path planning of underwater robot based on reinforcement learning. *IEEE Trans. Autom. Sci. Eng.* **20**(3), 1983–1996. <https://doi.org/10.1109/TASE.2022.3190901> (2023).
- Yang, Y. et al. AAPF*: A safer autonomous vehicle path planning algorithm based on the improved A* algorithm and APF algorithm. *Clust. Comput.* <https://doi.org/10.1007/s10586-024-04287-9> (2024).
- Zhe Zhang, Ju, Jiang, J. W. & Zhu, X. Efficient and optimal penetration path planning for stealth unmanned aerial vehicle using minimal radar cross-section tactics and modified A-Star algorithm. *ISA Trans.* **134**, 42–57. <https://doi.org/10.1016/j.isatra.2022.07.032> (2023).
- Auh, E. et al. Unloading sequence planning for autonomous robotic container-unloading system using A-star search algorithm. *Eng. Sci. Technol. Int. J.* **50**, 101610. <https://doi.org/10.1016/j.jestch.2023.101610> (2024).
- Chang, Lu., Shan, L., Jiang, C. & Dai, Y. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. *Auton. Robot* **45**, 51–76. <https://doi.org/10.1007/s10514-020-09947-4> (2021).
- Chuanbo, Wu., Wangneng, Yu., Li, G. & Liao, W. Deep reinforcement learning with dynamic window approach based collision avoidance path planning for maritime autonomous surface ships. *Ocean Eng.* **284**, 115208. <https://doi.org/10.1016/j.oceaneng.2023.115208> (2023).
- Li, Q., Ma, Q. & Weng, X. Dynamic path planning for mobile robots based on artificial potential field enhanced improved multiobjective snake optimization (APF-IMOSO). *J. Field Robot.* **41**, 1843–1863. <https://doi.org/10.1002/rob.22354> (2024).
- He, Y., Hou, T. & Wang, M. A new method for unmanned aerial vehicle path planning in complex environments. *Sci. Rep.* **14**, 9257. <https://doi.org/10.1038/s41598-024-60051-4> (2024).
- Macenski, S., Moore, T., Lu, D. V., Merzlyakov, A. & Ferguson, M. From the desks of ROS maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2. *Robot. Auton. Syst.* **168**, 104493. <https://doi.org/10.1016/j.robot.2023.104493> (2023).
- Li, X. et al. Dynamic path planning of mobile robots using adaptive dynamic programming. *Expert Syst. Appl.* **235**, 121112. <https://doi.org/10.1016/j.eswa.2023.121112> (2024).
- Jiang, Da. et al. An improved dynamic window approach based on reinforcement learning for the trajectory planning of automated guided vehicles. *IEEE Access* **12**, 36016–36025. <https://doi.org/10.1109/ACCESS.2024.3373446> (2024).
- Liu, Y., Wang, C., Wu, H. & Wei, Y. Mobile robot path planning based on kinematically constrained A-star algorithm and DWA fusion algorithm. *Mathematics* **11**, 4552. <https://doi.org/10.3390/math11214552> (2023).
- Wang, S., Hu, Y., Liu, Z. & Ma, L. Research on adaptive obstacle avoidance algorithm of robot based on DDPG-DWA. *Comput. Electr. Eng.* **109**(Part A), 108753. <https://doi.org/10.1016/j.compeleceng.2023.108753> (2023).
- Zeng, Q., Ou, B., Lv, C., Scherer, S. & Kan, Y. Monocular visual odometry using template matching and IMU. *IEEE Sensors J.* **21**, 17207–17218. <https://doi.org/10.1109/JSEN.2021.3078847> (2021).
- Li, G., Liu, C., Lei, Wu. & Xiao, W. A mixing algorithm of ACO and ABC for solving path planning of mobile robot. *Appl. Soft Comput.* **148**, 110868. <https://doi.org/10.1016/j.asoc.2023.110868> (2023).
- Petrović, M., Jokić, A., Miljković, Z. & Kulesza, Z. Multi-objective scheduling of a single mobile robot based on the grey wolf optimization algorithm. *Appl. Soft Comput.* **131**, 109784. <https://doi.org/10.1016/j.asoc.2022.109784> (2022).
- Lin, Xu., Cao, M. & Song, B. A new approach to smooth path planning of mobile robot based on quartic Bezier transition curve and improved PSO algorithm. *Neurocomputing* **473**, 98–106. <https://doi.org/10.1016/j.neucom.2021.12.016> (2022).
- Yin, X. et al. Route planning of mobile robot based on improved RRT star and TEB algorithm. *Sci. Rep.* **14**, 8942. <https://doi.org/10.1038/s41598-024-59413-9> (2024).
- Grisetti, G., Stachniss, C. & Burgard, W. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Trans. Robot.* **23**, 34–46. <https://doi.org/10.1109/TRO.2006.889486> (2007).
- Konolige, K., Grisetti, G., Kümmeler, R., Burgard, W., & Vincent, R. Efficient sparse pose adjustment for 2D mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 18–22, 2010, (Taipei, IEEE, 2010). <https://doi.org/10.1109/IROS.2010.5649043>.

Author contributions

Q.W. and K.Y. wrote the main manuscript text and M.Q. prepared Figs. 1–16. All authors reviewed the manu-

script.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Z.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024