Path Planning of Two-Wheeled Self-Balancing Robots in Static Obstacle Scenarios Based on Improved A* Algorithm

Chen Luo

{mnl2930@mail.ncut.edu.cn}

School of Electrical and Control Engineering, North China University of Technology, Beijing, 100144, China

Abstract. Two-wheeled self-balancing robots have great application potential in indoor inspection and home services due to their compact size and high flexibility. However, their mechanical and kinematic characteristics—narrow wheel track and high center of gravity causing tipping instability—impose stringent requirements on path safety and smoothness that the traditional A* algorithm cannot meet. To address this, this study proposes an improved A* algorithm integrating obstacle expansion and moving average smoothing, solving collision risks from body size and motion instability from unsmooth paths. This study built a Python-based static obstacle simulation platform, with comparative verification in simple and complex scenarios. Results show the improved algorithm ensures a minimum 10 cm safe distance from obstacles, generates smoother paths with fewer inflection points and continuous curvature changes, and avoids collisions. This study provides a safe, smooth path planning solution for two-wheeled self-balancing robots in indoor static obstacle environments.

Keywords: Two-wheeled self-balancing robot; Improved A* algorithm; Static obstacle; Moving average smoothing

1 Introduction

With the increasing demand for autonomous mobile devices in scenarios such as indoor inspection and home services, two-wheeled self-balancing robots have become a preferred vehicle due to their compact size and high flexibility. However, the collision risk of manual control caused by static obstacles and the adaptability problem of path planning in autonomous navigation restrict their scenario implementation. The A* algorithm is the core path planning technology for the autonomous navigation of two-wheeled self-balancing robots. Due to the robots' characteristics of "wide body and easy imbalance", there are more urgent requirements for path safety and smoothness, which the traditional A* algorithm is difficult to meet [1]. Therefore, aiming at the static obstacle scenario of two-wheeled self-balancing robots, this study improves the A* algorithm to solve the two core pain points of carrier size adaptation and path

ICMEEA 2026, April 24-26, Singapore, Singapore

DOI 10.4108/eai.24-4-2026.2364938

smoothness. Verifying the algorithm's effectiveness through Python simulation is of great value for promoting the application of small mobile robots.

From the perspective of the research status of A* algorithm improvement, current studies can be divided into three directions: First, research on improving search efficiency. By introducing Jump Point Search (JPS) [2, 3], pruning key nodes to reduce invalid extended nodes [4], or optimizing search guidance through adaptive direction strategies [5], dynamic heuristic function optimization [6], and reverse search strategies [7], the computational overhead is significantly reduced. However, this kind of method mostly targets general mobile robots, focuses simply on improving the search speed, and fails to consider the coupled constraints of "wide body" and "easy imbalance" of two-wheeled self-balancing robots; the second type of research optimizes path smoothness, using methods such as dynamic circle smoothing [7], key point selection [8], and Bezier curve fitting to reduce inflection points [9]. However, these methods do not consider the motion constraints under the differential drive model, which may generate paths with drastic curvature changes that do not meet the minimum turning radius of the self-balancing robot, and lacks targeted verification on this platform; the third category focuses on safety improvement, constructing obstacle avoidance constraints through means such as local obstacle avoidance algorithms [10, 11], virtual obstacles [12], and extended distances [4]. However, they do not simultaneously adapt to the dual needs of body width and easy imbalance, which makes it difficult to meet the actual movement requirements of two-wheeled self-balancing robots. In summary, the existing improvements mostly revolve around the characteristics of general mobile robots or four-wheeled vehicles, and fail to comprehensively adapt to the unique dynamic constraints of two-wheeled self-balancing robots. Therefore, how to design an A* algorithm that can ensure safe distance and movement smoothness at the same time to adapt to the unique dynamic characteristics of two-wheeled self-balancing robots remains an unresolved challenge.

In response to the above problems, this study proposes a two-stage improved A* algorithm framework to solve the shortcomings of the traditional algorithm: the pre-stage designs an obstacle expansion strategy based on the actual size of the self-balancing robot, marks the "safety restricted areas" in map modeling, and transforms the body width into grid safety constraints to ensure driving safety; the post-stage adopts the moving average smoothing method to process the polyline paths generated by the A* algorithm, reduce path inflection points, avoid safety risks, and lower the risk of body shaking [8]. The research will establish two types of typical static obstacle simulation environments using Python, compare the path differences between the traditional A* algorithm and the improved A* algorithm, verify the safety and smoothness of the algorithm, and provide a practical solution for the static scene path planning of two-wheeled self-balancing robots.

2 Research methods

2.1 Problem modeling and experimental platform

Two-wheeled self-balancing robot model. The parameters of the simulated two-wheeled self-balancing robot in this experiment are based on the measured parameters of the physical prototype in the laboratory to build the simulation model. Its body dimensions are 10 cm (length) × 20 cm (width) × 20 cm (height). Among them, the 20 cm width (wheel track) has a greater

impact on the safety of path planning and serves as the main basis for the obstacle expansion strategy. The robot adopts a two-wheeled differential drive mode, and the minimum turning radius is estimated to be 30 cm based on its structural characteristics, which is consistent with the actual kinematic constraints of the self-balancing robot.

Environment modeling. With reference to the hardware dimensions of the self-balancing robot, this modeling adopts the 0-1 binary encoding rule to construct a grid map for indoor static obstacle scenarios [9]. "0" represents a passable grid, "1" represents an obstacle grid, and "2" and "3" mark the start and end points respectively. The grid resolution is set to 5 cm to ensure high-precision mapping of physical space. Two types of scenarios are designed: the simple scenario is a 30×30 grid (150 cm×150 cm) with the start point (2,2) and end point (28,28), where obstacles are concentrated without continuous occlusion; the complex scenario is a 50×50 grid (250 cm×250 cm) with the start point (5,5) and end point (45,45), where obstacles are scattered and contain multiple obstacle clusters, and both the start and end points are set away from obstacles.

Simulation platform. The hardware adopts an Intel Core i7-1165G7 CPU, 16GB RAM, and Windows 10 operating system to ensure operational stability. The software is developed based on Python 3.12, with the numpy 2.3.5 library for numerical calculations, matplotlib 3.10.8 for result visualization, and scipy 1.16.3 for obstacle expansion operations.

2.2 Algorithm design

The improved A* algorithm adopts a two-stage framework of "pre-map optimization + post-path optimization". Through the collaborative design of obstacle expansion and moving average smoothing, it adapts to the characteristics of two-wheeled self-balancing robots, and its overall framework flow is shown in Fig. 1. This flowchart illustrates the two-stage optimization process of the improved A* algorithm: first, create a grid map and mark safety restricted areas through obstacle expansion (pre-map optimization); then, after searching for the path using the traditional A* algorithm, optimize path inflection points through moving average smoothing (post-path optimization); finally, output a safe and smooth path suitable for two-wheeled self-balancing robots.

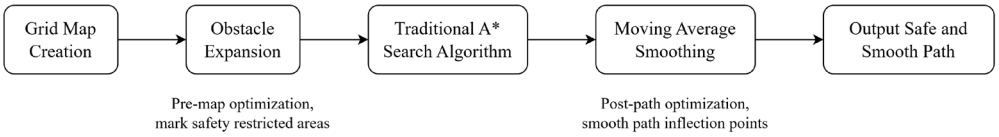

Fig. 1. Overall framework flowchart of the improved A* algorithm.

Traditional A* algorithm. This algorithm retains the functional form of the traditional A* algorithm:

$$f(n)=g(n)+h(n)\ . \tag{1}$$

Among them, g(n) is the actual cost from the start point to the current node, with a movement cost of 1 for orthogonal directions (up, down, left, right) and $\sqrt{2}$ for diagonal directions; h(n) is the estimated cost from the current node to the target node, using Manhattan distance:

$$h(n)=\left|x_i-x_j\right|+\left|y_i-y_j\right|\ . \tag{2}$$

Path planning is implemented by a traditional A* algorithm program, which supports 8-neighborhood search in up, down, left, right and four diagonal directions. Through built-in functions, it not only judges whether a node is out of bounds or an obstacle, but also judges whether it is a safety restricted area in the improved algorithm to avoid the risk of path being close to obstacles.

Aiming at the shortcomings of the traditional A* algorithm, the improved A* algorithm proposed in this study effectively solves the adaptability defects of the traditional A* algorithm in the scenario of two-wheeled self-balancing robots through two major links: "pre-map optimization + post-path optimization".

Obstacle expansion strategy. As the core of pre-map optimization for the improved A* algorithm, this strategy integrates grid map modeling and expansion logic, and the algorithm expands the created grid map [3]. The design principle of this obstacle expansion algorithm is based on the 20 cm body width of the self-balancing robot, with a 10 cm safe distance threshold set. It forms safety restricted areas by expanding obstacle grids to prevent the robot's side from colliding with obstacles. The core formula for the expansion radius is:

$$S_{base}=\frac{W_{body}/2+D_{safe}}{\delta} . \tag{3}$$

Among them, W_{body} is the body width, D_{safe} is the safe distance, and δ is the grid resolution. The base number of expanded grids is calculated to ensure safety redundancy. The implementation logic is as follows: a 3×3 expansion kernel is used to expand obstacle grids (value 1), and the newly added area is marked as safety restricted areas (value 4). During expansion, directions outside the map are automatically ignored, and the expansion range is constrained to be no less than the base value. This enables the A* algorithm to naturally avoid dangerous areas during search, without the need for additional size adaptation logic.

Moving average smoothing strategy. As the core of post-path optimization for the improved A* algorithm, this part addresses the "easy imbalance" characteristic of two-wheeled self-balancing robots during movement, ensuring more stable operation of the robot. Its design principle targets the problem of dense inflection points in the polyline path generated by the traditional A* algorithm: it calculates the local average of inflection point coordinates through a sliding window to reduce the impact of sudden direction changes, while preserving the overall trend of the path, thus balancing smoothing effect and computational efficiency.

For each intermediate inflection point in the path, the method takes the average of the coordinates of k adjacent inflection points before and after the target point as the new coordinate, using local smoothing to reduce the impact of sudden direction changes from a single inflection point. Let the sequence of path inflection points be $P_0,P_1,\ldots,P_n$ (where $P_i=(x_i,y_i)$ represents grid coordinates). With the sliding window size set to k=3, the smoothed coordinate of the intermediate inflection point $P_i(0 < i < n)$ is:

$$P_i'=\left(\frac{x_{i-1}+x_i+x_{i+1}}{3},\frac{y_{i-1}+y_i+y_{i+1}}{3}\right) . \tag{4}$$

The core mechanism of the algorithm is collision detection and fallback: the smoothed coordinate P_i' needs to be verified to check if it falls into obstacle grids (value 1), safety restricted areas (value 4), or beyond the map boundary. If there is a safety risk, the original inflection point P_i is retained to ensure that smoothing does not compromise safety. The start

and end points are directly retained to avoid overall path deviation, ultimately forming an executable path with continuous curvature.

2.3 Experimental design

To systematically verify the effectiveness of the improved A* algorithm, the experiment realizes full-process automated integration through a Python main function, connecting links such as grid map creation, obstacle expansion, path planning, and smoothing verification. No manual intervention is required, ensuring high efficiency and reproducibility of the simulation. The experiment builds two types of static obstacle scenarios, conducts three groups of comparative experiments, and defines clear evaluation indicators to quantify the algorithm's performance.

All experimental scenarios are built based on a resolution of 5 cm per grid: the simple scenario is a 30×30 grid (150 cm×150 cm) with the start point (2,2) and end point (28,28), containing two centrally distributed square obstacles without continuous occlusion; the complex scenario is a 50×50 grid (250 cm×250 cm) with the start point (5,5) and end point (45,45), including square obstacles, single-point obstacles, and horizontal obstacles. These obstacles are scattered and complex, closer to the actual indoor environment.

The unified process of the three groups of comparative experiments in both scenarios is as follows: (a) Traditional A* algorithm group: Perform 8-neighborhood path search based on the original grid map to verify the basic planning effect; (b) A* + obstacle expansion group: Introduce pre-map expansion optimization before path search to verify the effect of improving safe distance; (c) Improved A* algorithm group (A* + expansion + smoothing): Add moving average smoothing optimization on the basis of path search in the expanded map to verify the improvement effect of the full process. Each group of experiments outputs path planning result graphs, and intuitive analysis is carried out based on the visualized path results: judge whether there are collision risks by observing whether the path adheres to or invades obstacle areas; evaluate path smoothness by combining the intuitive characteristics of the path's polyline trend and the number of inflection points, and comprehensively complete the judgment of experimental results.

3 Experimental results

3.1 Simulation results of the simple scenario (30×30 Grid)

Three groups of comparative experiments are conducted in this scenario, and the experimental results are as shown in Figs. 2-4:

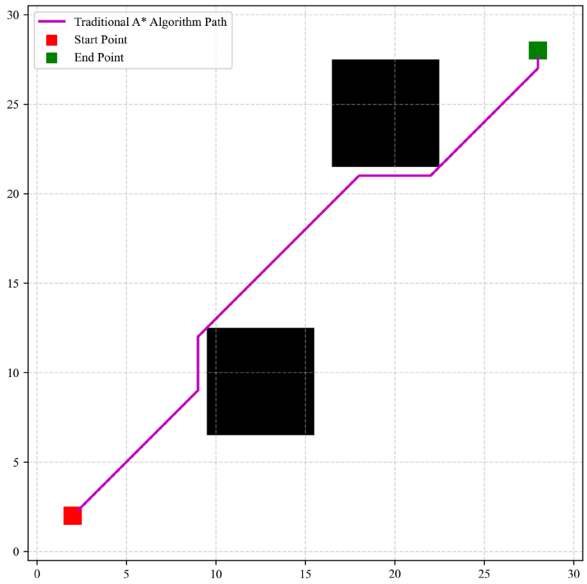

Fig. 2. Path planning diagram of the traditional A* algorithm in the simple scenario.

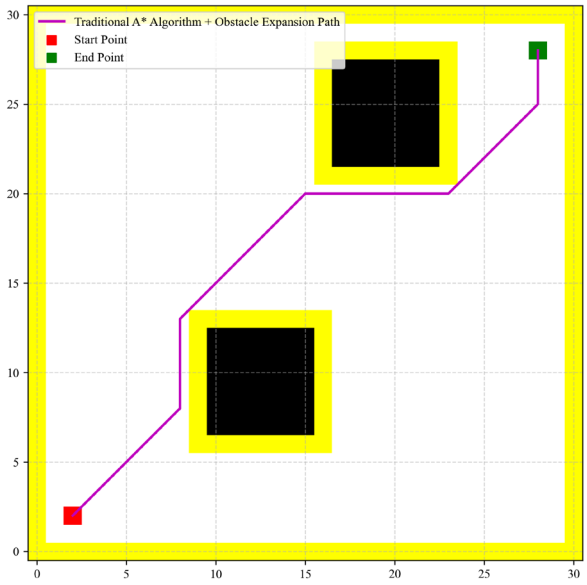

Fig. 3. Path planning diagram of the traditional A* algorithm + obstacle expansion in the simple scenario.

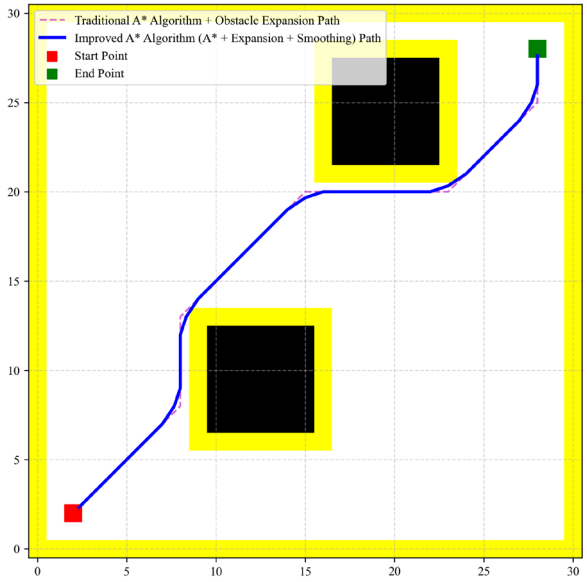

Fig. 4. Path planning diagram of the improved A* algorithm (A* + expansion + smoothing) in the simple scenario.

When (a) the traditional A* algorithm is adopted, the path shows an obvious polyline shape, with large sudden direction changes at inflection points. Some sections are less than 10 cm away from obstacles, but the path length is relatively short, as shown in Fig. 2. When (b) the A* + obstacle expansion group is adopted, the path successfully avoids the safety restricted areas expanded from obstacles, and the distance between each section and obstacles is ≥10 cm. However, the path still maintains a polyline shape with many inflection points, as shown in Fig. 3. When (c) the improved A* algorithm group (A* + expansion + smoothing) is adopted, the path presents a continuous and gentle curve shape, with sharp inflection points eliminated. Compared with group (b), the path length is reduced by 4 cm, as shown in Fig. 4. This reduction is attributed to the smoothing of path inflection points.

3.2 Simulation results of the complex scenario (50×50 Grid)

Three groups of comparative experiments are also conducted in this scenario, and the experimental results are as shown in Figs. 5-7:

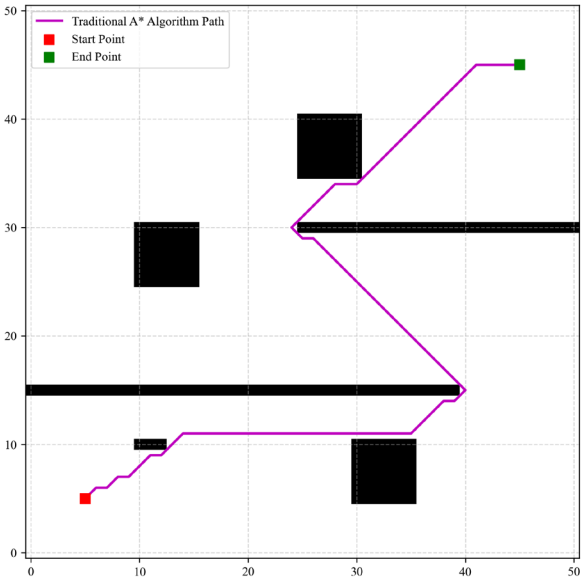

Fig. 5. Path planning diagram of the traditional A* algorithm in the complex scenario.

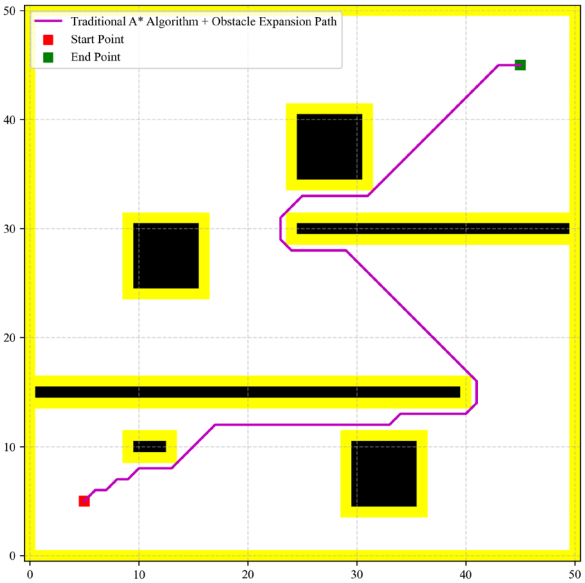

Fig. 6. Path planning diagram of the traditional A* algorithm + obstacle expansion in the complex scenario.

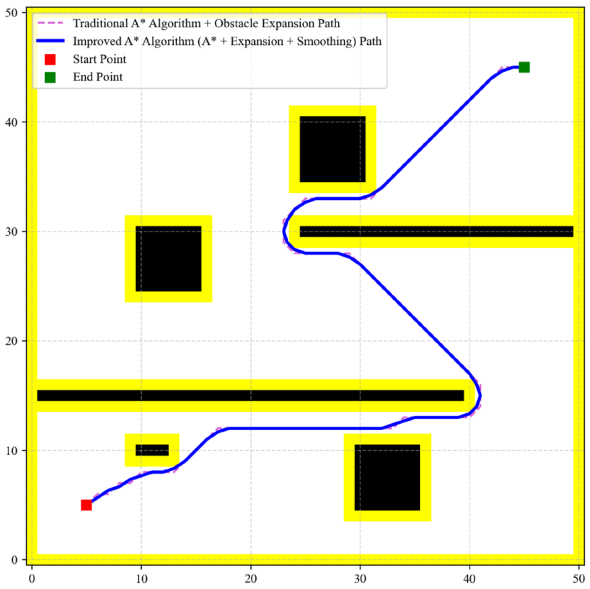

Fig. 7. Path planning diagram of the improved A* algorithm (A* + expansion + smoothing) in the complex scenario.

When (a) the traditional A* algorithm is adopted, the path has a higher density of polylines. During the avoidance of obstacle clusters, the frequency of sudden direction changes increases, and some obstacle-avoiding sections are less than 10 cm away from obstacles, as shown in Fig. 5. When (b) the A* + obstacle expansion group is adopted, the path successfully avoids all safety restricted areas expanded from obstacles, and the distance between each section and obstacles is ≥10 cm. However, the polyline shape remains unchanged with many inflection points, as shown in Fig. 6. When (c) the improved A* algorithm group (A* + expansion + smoothing) is adopted, the original polyline obstacle-avoiding sections are converted into continuous and gentle curves, with sharp inflection points eliminated. Compared with group (b), the path length is reduced by 13.9 cm, and the absolute value of path curvature is relatively reduced by approximately 26.77%, as shown in Fig. 7.

4 Analysis and discussion

4.1 Result analysis

The obstacle expansion strategy converts the 20 cm body width of the self-balancing robot and 10 cm safe distance into the expansion size of the grid map through the core formula. Mark the dangerous areas around the obstacle as safety restricted areas, so that the A* algorithm naturally avoids the area when searching, and directly embeds the physical dimension constraints from the map modeling level. It can ensure a safe distance of ≥10 cm without additional adaptation logic, which effectively eliminates the collision hazards.

The moving average smoothing uses the local mean calculation of the sliding window to perform weighted integration of the coordinates of the path's inflection points, so as to weaken the influence of the directional mutation of a single inflection point, while retaining the overall direction of the path. This method does not require complex curve fitting. Through simple calculation, the polyline can be transformed into a curvature-continuous trajectory, so as to adapt to the small-angle continuous turning movement constraints in the differential drive mode of the two-wheeled self-balancing robot and reduce the risk of body shaking.

Smoothing may lead to a slight lengthening of the path, but in exchange for a significant improvement in movement stability and safety. For the two-wheeled self-balancing robot that is prone to imbalance, the smoothness of the path directly determines the stability of the movement and avoids the posture loss caused by frequent inflection points. This cost has clear practical value.

4.2 Limitation analysis

At present, there are still shortcomings in this study: First, the experiment is only aimed at static obstacle scenarios, and the real-time obstacle avoidance needs of dynamic obstacles are not considered, which exhibits limitations in scenario applicability. Second, the sliding window size of the moving average is selected by experience, and the optimal value is not determined through the comparison of multiple sets of parameters, leading to insufficient parameter optimization. Third, no performance comparison has been conducted with more sophisticated smoothing methods such as spline curves, and the superiority of the algorithm has not been fully verified. Finally, the validity of the algorithm is only verified through Python simulation, and physical experiments have not been carried out on the real two-wheeled self-balancing robot platform, and the feasibility of practical deployment needs to be further verified.

4.3 Future work prospects

In the future, this research can be expanded in the following directions: first, combine the Dynamic Window Approach (DWA) to expand the algorithm's dynamic obstacle adaptation ability, so as to achieve full coverage of static and dynamic obstacle scenes; second, combine path smoothing with a Model Predictive Control (MPC) tracker to improve the two-wheeled self-balancing robot's tracking accuracy of the planned path; third, conduct multiple groups of comparative experiments on window size parameters to determine the optimal parameter configuration under different scenarios; finally, establish a physical experimental platform with a real two-wheeled self-balancing robot, carry out physical verification, and further verify the engineering practicality of the algorithm.

5 Conclusion

This study focuses on the path safety and smoothness needs of indoor static obstacle scenarios, aiming at the core characteristics of "wide body and easy imbalance" of two-wheeled self-balancing robots. A two-stage improved A* algorithm that integrates obstacle expansion and moving average smoothing is proposed. The physical size of the body is transformed into grid safety constraints through pre-map expansion, and post-path smoothing optimizes the inflection points and curvature, forming a complete solution that takes into account safety and motion

adaptability. The Python simulation verification of simple and complex scenarios shows that the algorithm can stably achieve ≥10 cm safe distance redundancy and effectively avoid collision hazards; at the same time, it reduces path inflection points, generates gentle trajectories with continuous curvature, and adapts to the minimum turning radius constraint of two-wheeled self-balancing robots, exhibiting excellent adaptability and stability in both scenarios. This research provides practical and efficient technical support for the indoor static obstacle path planning of two-wheeled self-balancing robots and similar small mobile robots, laying the foundation for the practical deployment of such equipment. In the future, the dynamic obstacle adaptation ability of the algorithm will be further expanded, combined with model predictive control to improve path tracking accuracy, and through physical experiments to verify its engineering practicality, promoting technology from simulation to practical application.

References

[1] Y. Huang, S. Guo, Path planning of mobile robots based on improved A* algorithm, in Proceedings of the 2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering, Qingdao, China, August 26-28 (2022), 133

[2] S. Ji, J. Chen, H. Liu, H. Yang, G. Wu, The indoor robot path planning based on improved A* algorithm. Mech. Des. Manuf. Eng. **54**, 48 (2025)

[3] X. Zhao, Z. Wang, C. Huang, Y. Zhao, Mobile robot path planning based on an improved A* algorithm. Robot. **40**, 903 (2018)

[4] H. Wang, X. Qi, S. Lou, J. Jing, H. He, W. Liu, An efficient and robust improved A* algorithm for path planning. Symmetry **13**, 2213 (2021)

[5] Y. Yao, L. Hu, L. Qin, Z. Zhou, Mobile robot path planning based on adaptive direction A* algorithm. IoT Technol. **15**, 62 (2025)

[6] H. Wang, C. Gan, F. Wang, Improved A* algorithm for unmanned vehicle path planning. Mech. Des. Manuf. **43**, 117 (2025)

[7] F. Liu, S. Qiu, Path planning of indoor mobile robot based on improved A* algorithm, in Proceedings of the 2021 2nd International Conference on Artificial Intelligence and Information Systems, Chongqing, China, May 28-30 (2021), 1-4

[8] C. Huang, T. Han, X. Wu, R. Feng, Robot path planning and obstacle avoidance technology based on improved A* algorithm. Electron. Des. Eng. **32**, 24 (2024)

[9] R. Jin, P. Wang, Y. Qian, D. Li, X. Gong, W. Zhao, J. Fu, Improved A* Path Planning Algorithm Integrating Bezier Curves, in Proceedings of the 2024 International Conference on Advanced Robotics and Mechatronics, Anhui, China, May 10-12 (2024), 885

[10] Y. Li, R. Jin, X. Xu, Y. Qian, H. Wang, S. Xu, Z. Wang, A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach. IEEE Access **10**, 57736 (2022)

[11] Z. Shan, J. Yu, C. Qian, Path planning of mobile robots based on fusion of improved A* and APF algorithms. Light Ind. Mach. **43**, 69 (2025)

[12] H. Liang, X. Du, Global path planning of unmanned vehicle based on improved A* algorithm, in Proceedings of the International Conference on Algorithms, Software Engineering, and Network Security, Nanchang, China, April 26-28 (2024), 9