Robotica (2025), **43**, pp. 887–908
doi:10.1017/S0263574725000037

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Research on hybrid path planning of underground degraded environment inspection robot based on improved A* algorithm and DWA algorithm

Congcong Gu[1,2], Songyong Liu[1,2], Hongsheng Li[3], Kewen Yuan[1,2] and Wenjiie Bao[1,2]

[1]School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou, China
[2]National Key Laboratory of Intelligent Mining Equipment Technology, Xuzhou, China
[3]School of Mechatronic Engineering, Xuzhou University of Technology, Xuzhou, China
Corresponding author: Songyong Liu; Email: lsycumt@163.com

Received: 3 August 2024; **Revised:** 28 November 2024; **Accepted:** 6 January 2025; **First published online:** 30 January 2025

Keywords: inspection robot; path planning; autonomous driving

Abstract

Aiming at the problems of many path inflection points, unsmooth paths, and poor local obstacle avoidance in path planning of inspection robots in static-dynamic scenes under complex geological conditions in coal mine roadways, a hybrid path planning method based on the improved A* algorithm and dynamic window approach (DWA) algorithm is proposed. First, the inspection robot platform and system model are constructed. An improved heuristic function that incorporates target weight information is proposed based on the A* global path planning algorithm. Additionally, redundant nodes are eliminated, and the path is smoothed using the Floyd algorithm and B-spline curves. Second, the global path planning A* algorithm and the local path planning DWA algorithm are fused. The dynamic path planning is carried out by setting the key node information of the global path extracted from the improved A* algorithm as the local target point of the DWA algorithm. On this basis, a grid map is established to simulate and analyze the proposed path planning algorithm. Finally, the autonomous path planning and walking experiment of inspection robot in simulated roadway environment are carried out. The results show that the hybrid path planning method based on improved A* algorithm and DWA algorithm proposed in this paper is more efficient and safer, which can meet the motion requirements of inspection robot in coal mine roadway.

1. Introduction

Safe and efficient tunneling of coal mine roadways plays a very important role in coal mining, but the roadway environment is complex and has a large number of safety hazards, so regular inspections are essential in order to prevent downtime due to abnormal equipment conditions. The use of traditional inspection methods is inefficient, time-consuming, and requires manual in-depth on-site measurements, so the use of robots to participate in the inspection of coal mine equipment is the future direction of development [1–3]. Compared with traditional manual inspection methods, these robots not only have high inspection efficiency and can meet the requirements of high-risk environments but also reduce production costs for enterprises. In the actual inspection operation, the inspection robot must plan the inspection path to ensure economy and rationality. This planning ultimately reduces the inspection time and minimizes the energy consumption of the robot [4–6]. Path planning is an important research area for inspection robots. The goal is to design an optimal path that meets specific constraints, avoids collisions, and guides the robot from the starting point to the goal point. This ensures that the inspection robot completes its operations safely and efficiently [7–9].

The current algorithms concerning path planning can basically be divided into three categories: graph-based algorithms, sampling-based algorithms, and other algorithms [10]. Graph-based search

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

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

algorithms can be mainly categorized into Breadth First Search (BFS) and Depth First Search (DFS) [11]. Although Breadth First Search algorithms have the advantages of simple implementation and easy to understand, the number of nodes grows exponentially with increasing depth, which may lead to problems such as combinatorial explosion, higher space complexity, and the possibility of not finding a solution. The depth-first algorithm requires less space for searching compared to the breadth-first algorithm, and the algorithmic idea is simpler and easier to understand and implement, but it may be trapped in a dead loop in some cases and may not be able to find a solution when the depth limit is unreasonable. Besides, these algorithms are based on traversal or improved traversal logic. In the case of high-dimensional space and complex constraints, it is also prone to exponential explosion, that is, the path search process will be very complex, and the time required to find the path will be greatly increased [12]. Sampling-based planning methods mainly include probabilistic path graph method (PRM) and rapid search random tree (RRT) [13]. Although PRM algorithm can plan paths using very few sampling points in some cases, the planned paths are hardly guaranteed to be optimal, and when the environment is very complex, a large number of sampling points may be required, so its complexity is not easy to be determined [14]. The RRT algorithm is a relatively efficient algorithm that can better deal with path planning problems with incomplete constraints, but because the sampling points are randomly selected, the path planned by the RRT algorithm is not necessarily optimal. In addition to graph-based search and sampling algorithms, there are several other path planning algorithms inspired by natural phenomena or organisms, such as genetic algorithms, artificial potential field methods, ant colony optimization, and dynamic window approaches (DWA). On this basis, many scholars have conducted extensive research on the path planning of inspection robots. Zhang [15] proposed a real-time look-ahead algorithm for machining complex curved workpieces with high precision and speed, and verified the effectiveness of this algorithm through experiments and simulations. Luo [16] introduced a novel planning method known as the Stepwise Follow-the-Leader (SFTL) algorithm, which utilizes sliding control points to facilitate sequential movement along the planned path. Tutsoy [17] developed a multidimensional particle swarm optimization algorithm and a multidimensional genetic algorithm to plan paths for translational, rotational, and Euler angles, generating the corresponding control signals. This method demonstrated that the quadrotor accurately follows the generated path while utilizing the maximum available rotor speeds. Wang [18] proposed SLAM application systems that integrate multi-line laser radar and visual perception for various scenarios. This approach effectively addresses the issue of path planning becoming trapped in local optima within complex environments, thereby enhancing the inspection efficiency of robots. Bogaerts [19] developed a novel gradient-based optimization strategy for robotic inspection planning. The algorithm begins with an initial measurement path and iteratively refines this path to minimize its length while ensuring adequate sensor coverage. Chaves [20] introduced a path planning algorithm that integrates visual SLAM, specifically designed to plan loop-closure paths to reduce navigation uncertainty. This approach demonstrated several advantages over existing solutions for managing navigation uncertainty in long-term autonomous operations. Tan [21] combined the ant colony algorithm with the particle swarm algorithm and uses the particle swarm algorithm to train the initial parameters of the ant colony algorithm to plan an optimal path for spot welding robots. Jiang [22] introduced an adaptive sampling collision-free detection path generation algorithm based on a CAD computer-aided detection planning system and a coordinate measuring machine.

Despite significant advancements in the research of path planning algorithms, several challenges and limitations remain in the practical application of these algorithms. This is particularly true in the complex environment of coal mines. As shown in Figure 1, compared with the surface environment, the complexity of the coal mine tunnel environment is characterized by high temperature and high humidity, dangerous gas, more dust, poor air circulation, and poor lighting environment. At the same time, due to the narrow operating space and numerous equipment in the underground, the roadway contour, the surrounding environment obstacles (such as trenches, ramps, and steps) and other elements will greatly limit the safe and smooth movement of the inspection robot. Therefore, this puts forward more demanding requirements on the environmental adaptability and performance of coal mine roadway inspection robots.

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

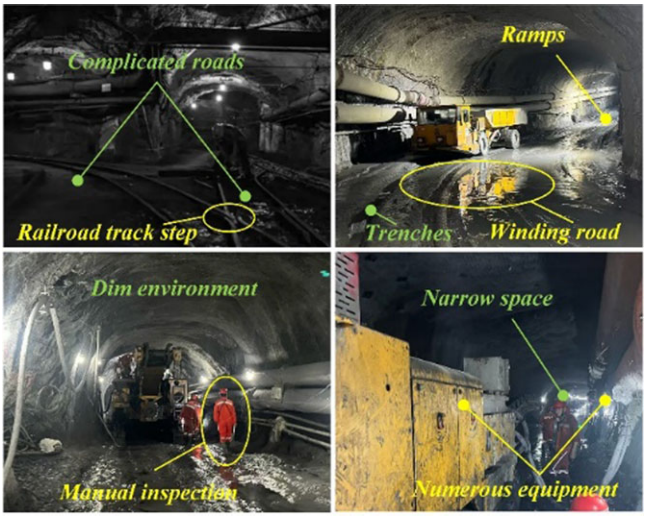


Figure 1. *Coal mine tunnel environment characteristics.*

Combining the environmental characteristics of coal mine roadways and the research results of existing path planning algorithms, it can be seen that the current algorithms cannot be effectively applied to coal mining operations. Furthermore, a single global or local planning algorithm is insufficient to address the challenges of real-time and high-precision obstacle avoidance in both static and dynamic scenarios. Therefore, integrating two path planning methods should be considered to effectively tackle these issues.

In this paper, according to the environmental characteristics of the coal mine roadway, the test platform of the inspection robot is built, and the analysis and establishment of the inspection system model are completed. A hybrid path planning algorithm combining improved A * algorithm and DWA algorithm is proposed. The new heuristic function containing target weight information is used to eliminate redundant nodes and smooth the path to improve the effect of global path planning. On this basis, guided by the global path information planned by the improved A * algorithm, the real-time path update and obstacle avoidance of the robot are realized by combining the DWA local path planning algorithm. The experimental results show that in the complex coal mine environment, the algorithm can analyze the local target points in real time and plan the global path, and at the same time perform effective obstacle avoidance maneuver. This method significantly improves the accuracy and efficiency of the detection robot path planning.

2. System platform and model development

2.1. System platform

Aiming at the underground degraded roadway environment, an efficient, accurate, and robust inspection robot test platform is constructed based on lidar and inertial measurement unit, as shown in Figure 2.

The system comprises LiDAR and inertial sensors to gather external data, execute the hybrid path planning algorithm for the inspection robot on the control board, and transmit motion control commands to the lower computer control board via the serial port. The lower computer module primarily consists of an STM32 control board, a motor driver, and an encoder motor. The STM32 control board communicates with the upper computer control board through the serial port, receives motor speed control instructions from the upper computer, adjusts the encoder motor speed via the motor driver, and receives encoder speed feedback to complete the closed-loop motion control. The hardware system utilizes a 24 V lithium battery to provide a consistent power supply. It powers the LiDAR, upper computer control board, lower computer control board, and motor drive through a step-down module. The inspection robot employs

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

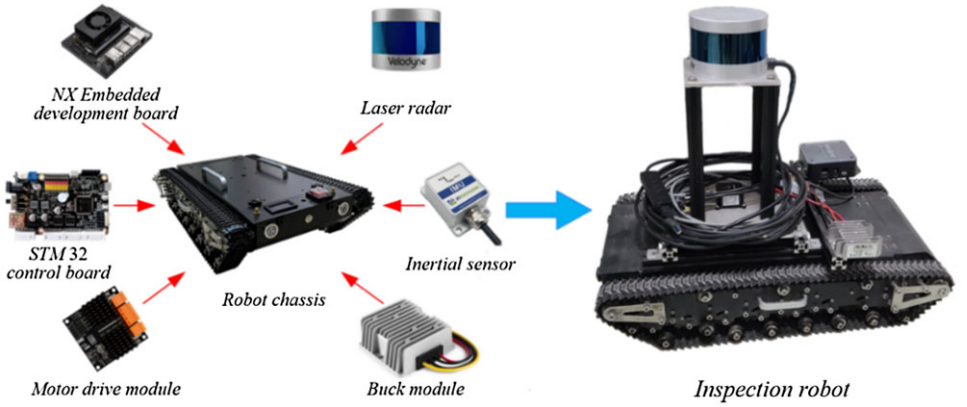


Figure 2. *Inspection robot system platform.*

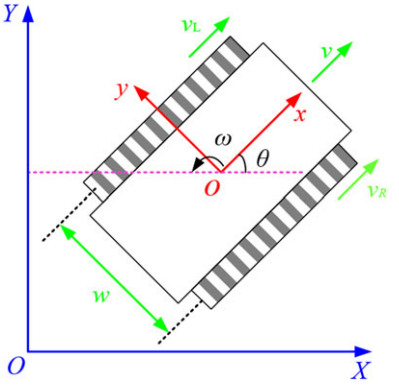


Figure 3. *Inspection robot motion model.*

LiDAR to detect external environmental information and transmits the collected radar point cloud data to the upper decision-making planning system, facilitating the completion of autonomous navigation operations.

2.2. System model establishment

The inspection robot operates on a tracked mobile chassis, with motors connected to the active wheels through a rotary axis. To facilitate the development of the kinematic model, the target speed of the robot is converted into the target speed of the motors, based on the following assumptions:

The overall weight of the inspection robot is uniformly distributed, and its center of gravity coincides with the geometric center.

The inspection robot operates on a two-dimensional plane, with no slipping occurring on either side of the track.

The tracks on both sides of the inspection robot are in full contact with the ground and share identical parameters, including length, width, and thickness.

The motion model of the tracked inspection robot is established within the world coordinate system, and its movement is illustrated in Figure 3.

Where *XOY* is the world coordinate system, *xoy* is the robot coordinate system, ω is the distance between the tracks on both sides of the robot, *o* is the geometric center of the robot, v_L and v_R are the

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

forward velocities of the tracks on both sides of the robot, and θ is the angle between the x-axis of the robot coordinate system and the positive direction of the X-axis of the world coordinate system.

Then the forward velocity and steering angular velocity of the robot can be expressed as

$$v = \frac{1}{2}(v_L + v_R) \tag{1}$$

$$\omega = \frac{v_R - v_L}{w} \tag{2}$$

Taking the geometric center o of the inspection robot as the reference point, the linear velocity and the normal angular velocity of point o of the inspection robot at any given moment can be expressed as follows:

$$x_o' = \frac{1}{2}(v_L + v_R)\cos\theta \tag{3}$$

$$y_o' = \frac{1}{2}(v_L + v_R)\sin\theta \tag{4}$$

$$\theta_o' = \frac{v_R - v_L}{w} \tag{5}$$

At any moment t, the horizontal and vertical coordinates of the geometric center o of the inspection robot in the world coordinate system are $x_o(t)$ and $y_o(t)$, respectively, and the position of the robot under the world coordinate system can be expressed as

$$x_o(t) = \frac{1}{2}\int_0^t [v_L(t) + v_R(t)]\cos\theta_o(t)dt \tag{6}$$

$$y_o(t) = \frac{1}{2}\int_0^t [v_L(t) + v_R(t)]\sin\theta_o(t)dt \tag{7}$$

$$\theta_o(t) = \frac{\int_0^t [v_R(t) - v_L(t)]\,dt}{w} \tag{8}$$

The turning radius R of the inspection robot under the current position is

$$R = \frac{v_o}{\omega_o} = \frac{v_R + v_L}{2} \cdot \frac{w}{v_R - v_L} \tag{9}$$

The rotation angle of the inspection robot at a specific point is denoted as $d\alpha$, and the corresponding arc length is represented by ds. Therefore, the curvature of the trajectory at this point is given by

$$\kappa = \frac{d\alpha}{ds} = \frac{x'(t)y''(t) - x''(t)y'(t)}{\left(\sqrt{x'(t)^2 + y'(t)^2}\right)^3} \tag{10}$$

According to Eq. (10), the steering curvature of the inspection robot is

$$\kappa = \frac{\omega_o}{v_o} = \frac{2(v_R - v_L)}{w(v_R + v_L)} \tag{11}$$

When the inspection robot's right track reaches its maximum speed, denoted as $v_R = v_{\max}$, the maximum steering curvature of the robot is

$$\kappa = \frac{2(v_R - v_L)}{w\,(v_R + v_L)} = \frac{1}{w}\left(\frac{v_{\max} - (2v - v_{\max})}{v}\right) = \frac{2}{w}\left(\frac{v_{\max}}{v} - 1\right) \tag{12}$$

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

3. Hybrid path planning algorithm of inspection robot

3.1. A algorithm problem analysis*

The A* algorithm conducts a path search from the starting point to the goal point using an evaluation function. This heuristic path search method significantly reduces redundant node calculations, enhances the efficiency of the path search, and aims to identify the optimal path. The evaluation function denoted as $f(n)$, which is defined as follows:

$$f(n) = g(n) + h(n) \tag{13}$$

where $f(n)$ denotes the total estimated cost from the initial node to the target node, $g(n)$ denotes the actual cost from the initial node to the current node n, and $h(n)$ denotes the estimated cost from the current node n to the target node, which is known as the heuristic function.

The heuristic function distance models commonly used in traditional A* algorithm are mainly Manhattan distance and Euclidean distance.

(1) Manhattan distance

The Manhattan distance is defined as the sum of the absolute differences between the coordinates of the current node and the target node in a coordinate system. It can be expressed as follows:

$$h(n) = abs(x_n - x_g) + abs(y_n - y_g) \tag{14}$$

(2) Euclidean distance

The Euclidean distance is the distance between the current node and the target node, which can be expressed as

$$h(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \tag{15}$$

where x_n and y_n represent the horizontal and vertical coordinates of the current node, respectively, x_g and y_g represent the horizontal and vertical coordinates of the target node, respectively.

Although the A* algorithm offers high efficiency, strong robustness, and low complexity, its inherent algorithmic principles lead to the presence of numerous redundant nodes and inflection points during the path search process, ultimately diminishing the efficiency of path planning. Furthermore, the design of the heuristic function significantly impacts the efficiency of path node searches when employing the A* algorithm for path planning [23, 24]. As the path search space expands, the cost of path searching escalates rapidly, which poses challenges for inspection robots navigating unstructured environments, such as in coal mine roadways.

3.2. A algorithm problem optimization*

3.2.1. Heuristic function optimization

According to the characteristics of the heuristic function distance model commonly used in the traditional A* algorithm, the Manhattan distance is typically greater than the actual distance, while the Euclidean distance is usually less than the actual distance. Therefore, to bring the value of $h(n)$ closer to the true cost and to ensure the optimal path while enhancing the efficiency of the path search, the median of the Manhattan distance and the Euclidean distance is selected as the distance model for $h(n)$.

The optimized distance model is shown in Figure 4, where d_1 and d_2 represent the horizontal and vertical distances from the initial node S to the target node G, respectively. The sum of d_1 and d_2 is the Manhattan distance, while the straight-line distance from point S to point G is the Euclidean distance. The dotted line segment $h\hat{}(n)$ is the distance model of the optimized heuristic function, and the θ is the angle between $h\hat{}(n)$ and the right-angled side of the triangle bounded by the points S and G and the coordinate axes.

In order to verify the influence of θ on the performance of the A* algorithm, comparative experiments were conducted at angles of 30°, 45°, and 60°. As shown in Figure 5, when θ is set to 45°,

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

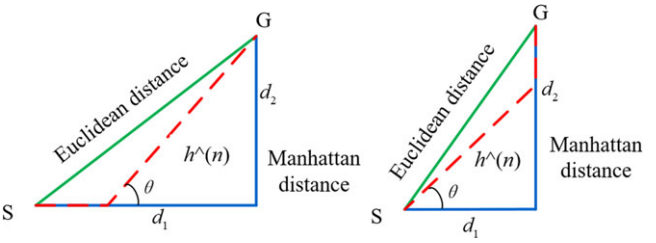


Figure 4. *Optimal distance model of heuristic function.*

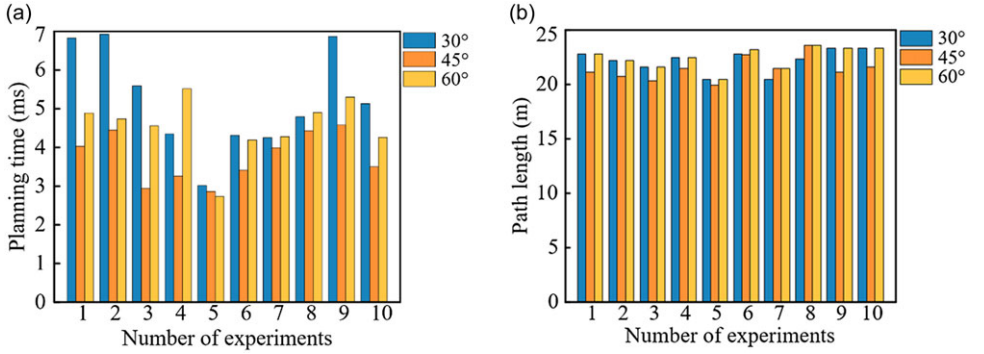


Figure 5. *Comparison of path planning effects under different θ. (a) Comparison of path planning time, (b) comparison of path planning length.*

the A* algorithm demonstrates optimal performance in terms of both time and path length for path planning. Therefore, considering the impact on path planning time and path length, θ is selected as 45° for subsequent analysis.

The optimized distance model $h\hat{}(n)$ is derived as follows:

When $d_1 \geq d_2$:

$$h\hat{}(n) = \sqrt{2d_2^2} + d_1 - d_2 \tag{16}$$

When $d_1 \leq d_2$:

$$h\hat{}(n) = \sqrt{2d_1^2} + d_2 - d_1 \tag{17}$$

In addition, to enhance the efficiency of heuristic search, the objective tendency function is incorporated into the heuristic function. The principle of the objective tendency function is expressed as follows:

$$cross = |dx_1 \times dy_2 - dx_2 \times dy_1| \tag{18}$$

where dx_1 and dy_1 represent the horizontal and vertical distances between the current node and the target node, dx_2 and dy_2 represent the horizontal and vertical distances between the initial node and the target node, and cross represents the target tendency function value.

In order to dynamically adjust the path search effectiveness of the target tendency function cross, a weight factor k is introduced. Consequently, the heuristic function of the final improved A* algorithm can be expressed as follows:

$$h^*(n) = \begin{cases} \left(\sqrt{2d_2^2} + d_1 - d_2\right) \times cross \times k, & d_1 \geq d_2 \\ \left(\sqrt{2d_1^2} + d_2 - d_1\right) \times cross \times k, & d_1 < d_2 \end{cases} \tag{19}$$

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

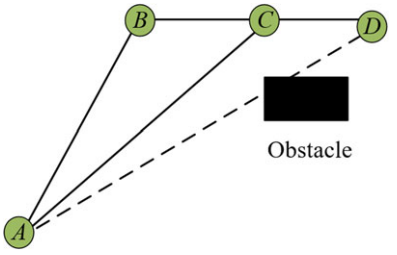


Figure 6. *Floyd algorithm principle diagram.*

3.2.2. Elimination of redundant nodes

The A* algorithm, which optimizes the heuristic function, can efficiently identify a viable path. However, it may still include some redundant nodes in the planned route. To address this issue, the Floyd algorithm is employed to eliminate redundant nodes during the path search process of the improved A* algorithm, thereby optimizing the path trajectory [25–27]. The principle of the Floyd algorithm is illustrated in Figure 6, which effectively determines the shortest distance between two points. By integrating the Floyd algorithm with the A* algorithm, redundant nodes can be removed, and collinear nodes can be merged, resulting in a path that is not only shorter but also better suited to real-world application scenarios.

According to Figure 6, let L (A, D) represent the shortest distance from point A to point D. If there is an obstacle between points A and D that cannot be traversed, then L (A, D) $= +\infty$, and the shortest path from point A to point D is expressed as R (A, D) = A $\rightarrow$ D.

When the path from point A to point D cannot be passed, insert point B between point A and point D. If the condition of the following equation is satisfied:

$$L(A,B) + L(B,D) < L(A,D) \tag{20}$$

Then the distance and path between point A and point D can be expressed as

$$\begin{cases} L(A,D) = L(A,B) + L(B,D) \\ R(A,D) = A \rightarrow B \rightarrow D \end{cases} \tag{21}$$

At this time, insert point C between point A and point D, if the conditions are met:

$$L(A,C) + L(C,D) < L(A,B) + L(B,D) \tag{22}$$

Then the distance and path between point A and point D can be expressed as

$$\begin{cases} L(A,D) = L(A,C) + L(C,D) \\ R(A,D) = A \rightarrow C \rightarrow D \end{cases} \tag{23}$$

Finally, point B is removed to obtain the optimal path from point A to point D, represented as A$\rightarrow$C$\rightarrow$D. Consequently, by applying Floyd's algorithm, the A* algorithm can search for the path more efficiently, reducing unnecessary node processing. This approach not only shortens the robot's driving path but also enhances overall work efficiency.

3.2.3. Path smoothing

Although the path optimized by Floyd's algorithm eliminates redundant nodes and some inflection points, there are still several inflection points present in the overall driving path of the coal mine roadway inspection robot. In the unstructured environment of coal mine tunnels, the presence of inflection points is highly detrimental to the movement of the inspection robot. Therefore, it is essential to smooth these inflection points along the path. The path planned by the improved A* algorithm is smoothed using a

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

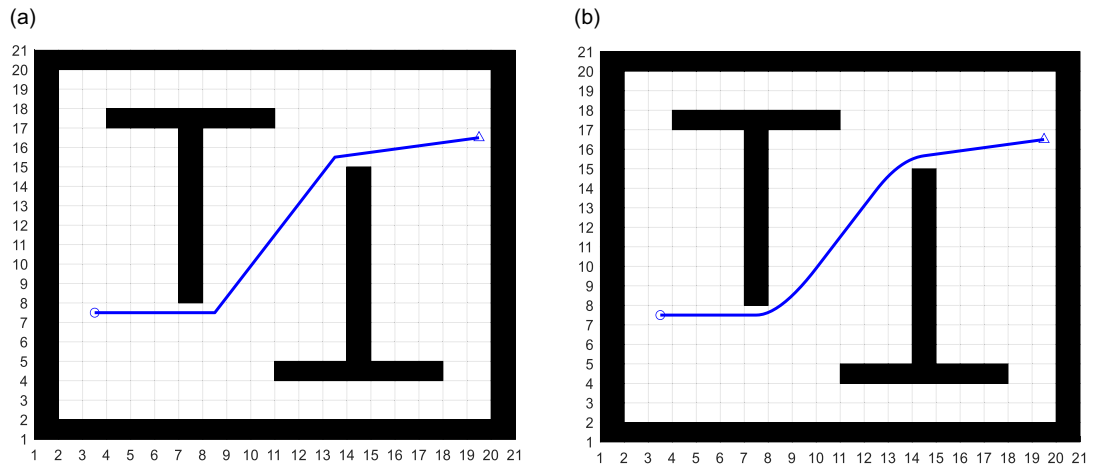


Figure 7. *The schematic diagram of the planned path before and after the improved A* algorithm is processed by B-spline. (a) The original path (b) after B-spline curve processing.*

B-spline curve. The basis function of an n-th degree B-spline curve can be expressed as follows [28]:

$$F_{k,n}(u) = \frac{1}{n!}\sum_{j=0}^{n-k}\left[(-1)^j C_{n+1}^j (u+n-k-j)^n\right] (k=0,1,2\ldots,n) \tag{24}$$

of which:

$$C_{n+1}^j = \frac{(n+1)!}{j!\,(n+1-j)!} \tag{25}$$

Then the n-th B-spline curve of the ith segment can be expressed as

$$H_{i,n}(u) = \sum_{k=0}^{n} G_{i+k} F_{k,n}(u)\ (i=1,2,\ldots,m-n, u\in[0,1]) \tag{26}$$

where n denotes the order of the B-spline curve, m denotes the number of control points, and G_{i+k} denotes the coordinate value of the $i+k$-th control point.

From the path information presented in Figure 7, it is evident that after processing the B-spline curve, the breakpoints in the original path have been replaced with a smooth curve. As a result, the overall path is smoother and better aligns with the trajectory requirements of the coal mine roadway inspection robot.

3.3. Local path planning based on DWA algorithm

After the inspection robot moves for a period of time, its position changes are shown in Figure 8.

It is assumed that during a period of time Δt, according to the inspection robot motion characteristics, the motion model of the robot during that period of time can be expressed as

$$\begin{cases} x_t = x + v\Delta t\cos(\theta) \\ y_t = y + v\Delta t\sin(\theta) \\ \theta_t = \theta + \omega\Delta t \end{cases} \tag{27}$$

where x and y represent the coordinate values of the robot in the world coordinate system, respectively, v represents the moving speed of the robot, and θ represents the rotation angle of the robot.

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

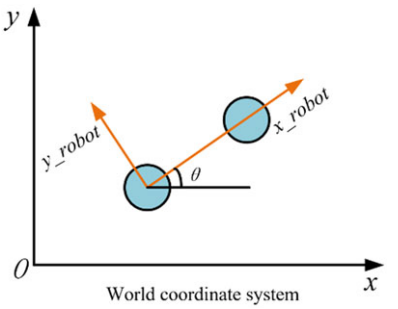


Figure 8. *The schematic diagram of position change of inspection robot.*

3.3.1. Sampling speed

In the DWA algorithm, the robot generates simulated trajectories by sampling in velocity space. The linear and angular velocities are limited during the velocity sampling in order to prevent the robot from skidding or not being able to drive due to too large or too small traveling speeds, etc. The sampling speed of the robot is limited by its own linear and angular velocities and needs to satisfy the following constraints:

$$V_1 = \{v \in [v_{\min}, v_{\max}] \cap \omega \in [\omega_{\min}, \omega_{\max}]\} \tag{28}$$

Where $v_{\max}$ and $v_{\min}$ represent the maximum and minimum linear velocities, respectively, $\omega_{\max}$ and $\omega_{\min}$ represent the maximum and minimum angular velocities, respectively.

Because the inspection robot platform is driven by a DC motor, affected by the performance of the motor, the speed that the robot can reach is limited to a window within a time interval. Therefore, there is a range constraint on the acceleration of the robot, that is, the following constraints are satisfied:

$$V_2 = \{(v, \omega) \,|v \in [v_c - a_{v\min}\Delta t, v_c + a_{v\max}\Delta t] \cap \omega \in [\omega_c - a_{\omega\min}\Delta t, \omega_c + a_{\omega\max}\Delta t]\} \tag{29}$$

where v_c and ω_c represent the linear velocity and angular velocity of the robot, respectively, $a_{v\max}$ and $a_{v\min}$ are the maximum acceleration and minimum acceleration of linear motion, respectively, and $a_{\omega\max}$ and $a_{\omega\min}$ are the maximum acceleration and minimum acceleration of rotation, respectively.

During the robot's movement, it must avoid colliding with obstacles in its environment. By establishing a safety distance threshold, a sampling speed space that ensures the motion safety of the inspection robot is proposed. The constraints are as follows:

$$V_3 = \left\{(v, \omega) \left| v \le \sqrt{2 \times dist(v, \omega) \times a_v} \cap \omega \le \sqrt{2 \times dist(v, \omega) \times a_\omega} \right.\right\} \tag{30}$$

where $dist(v, \omega)$ represents the distance between the robot and the nearest obstacle.

Based on the three constraints mentioned above, the speed range of the robot can be determined. Specifically, the sampling speed range of the robot's dynamic window must satisfy the following conditions:

$$V = V_1 \cap V_2 \cap V_3 \tag{31}$$

3.3.2. Evaluation function optimization

After the inspection, robot generates simulated trajectories through multiple sets of sampling in the velocity space, these trajectories represent collision-free, passable paths. However, it is not guaranteed that each trajectory is the most suitable for the robot's movement. Therefore, it is essential to select the most appropriate trajectories using an evaluation function. The evaluation function is expressed as follows:

$$G(v, \omega) = \alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot vel(v, \omega) \tag{32}$$

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

where heading (v, ω) is the azimuth evaluation function, which scores the current trajectory based on the angular difference between the end of the inspection robot's current trajectory and the target point in the facing direction. $dist(v, \omega)$ denotes the distance evaluation function, which scores the current trajectory based on the magnitude of the distance between the inspection robot and the nearest obstacle. $vel(v, \omega)$ denotes the velocity evaluation function, which scores the current trajectory based on the velocity vector corresponding to the current trajectory. α, β, γ are the weight coefficients corresponding to each evaluation factor.

Considering that the dynamic window method simulates the motion trajectory after sampling multiple sets of speeds and selects the optimal motion speed based on the results of the evaluation function. In the actual process, the parameter results of each factor in the evaluation function are large or small, which may directly cover the effect of other factor terms because of the large parameter results of a certain factor, so the robot's motion is made smoother by normalizing each factor of the evaluation function in order to select the simulated trajectory at the optimal speed. The normalization process is as follows:

$$\begin{cases} my_heading(i) = \dfrac{heading(i)}{\sum_{i=1}^{n} heading(i)} \\ my_dist(i) = \dfrac{dist(i)}{\sum_{i=1}^{n} dist(i)} \\ my_vel(i) = \dfrac{vel(i)}{\sum_{i=1}^{n} vel(i)} \end{cases} \tag{33}$$

where *my_heading* (i), *my_dist* (i), and *my_vel* (i) denote the weights of each evaluation factor normalized to each evaluation factor of the trajectory i to be evaluated, respectively.

3.3.3. Hybrid path planning algorithm

The DWA algorithm demonstrates effective obstacle avoidance capabilities; however, it is prone to becoming trapped in local optima, which can hinder its ability to reach the target point [29–31]. The enhanced A* algorithm presented in this paper is capable of planning an optimal path when the global state is known. To prevent the DWA algorithm from succumbing to local optima while facilitating local dynamic obstacle avoidance, this paper introduces a hybrid path planning algorithm that integrates the improved A* algorithm with the DWA algorithm. This approach not only preserves global optimality but also addresses the requirements for dynamic obstacle avoidance in robots.

The specific steps involved in the fusion process of the hybrid path planning algorithm are as follows:

(1) Initialize the map information and the robot's initial position, then plan a globally optimal path on the map using the improved A* algorithm.

(2) Based on the global path information, key nodes are extracted as the local target points for the DWA algorithm.

(3) The DWA algorithm is employed to sample the speed and generate a simulated trajectory. Subsequently, based on the new evaluation function, the speed corresponding to the optimal trajectory is selected to guide the robot toward the target point.

(4) Update the robot's motion state. First, determine whether the robot has reached the local target point. If it has not, drive the robot to the local target point. If it has reached the local target point, assess whether the position is the global target point. If the position is not the global target point, continue with step (3). Conversely, if the robot has reached the global target point, control the robot to stop moving.

For the combination of the improved A* and DWA algorithms, the following path fusion function has been constructed:

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

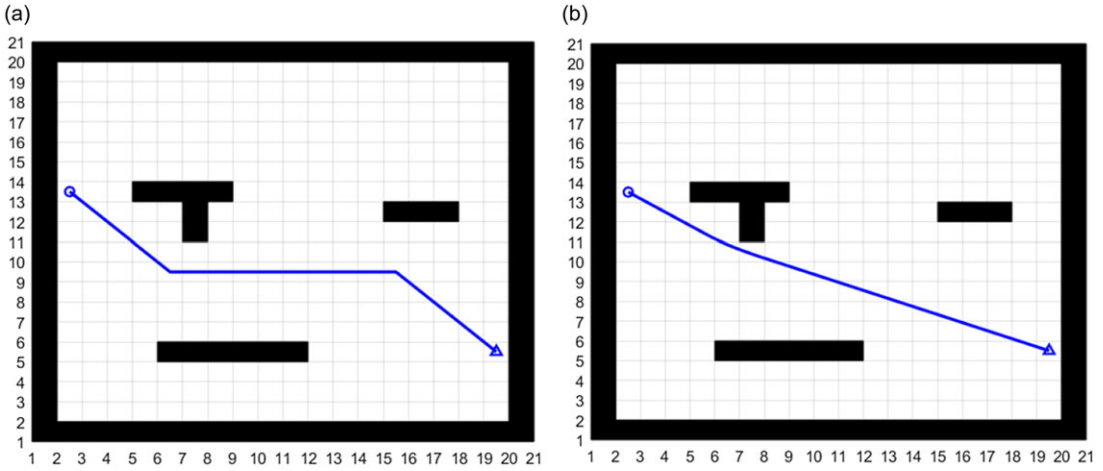


Figure 9. *Environment 1. (a) Traditional A* algorithm, (b) Improved A* algorithm.*

$$Combine(v, \omega) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{34}$$

where (x_1, y_1) denotes the end coordinates of the motion trajectory simulated by sampling the velocity, and (x_2, y_2) denotes the coordinates of the critical nodes of the global path obtained based on the improved A* algorithm.

In order to better combine the improved A* algorithm with the DWA algorithm, a new evaluation function based on global path information is designed based on the evaluation function of the previous DWA algorithm:

$$G(v, \omega) = \alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot vel(v, \omega) + \lambda \cdot Combine(v, \omega) \tag{35}$$

where λ is the weight factor of the path fusion function Combine (v, ω), the smaller the value of the path fusion function takes, the closer the end of the local path planning trajectory is to the global path trajectory.

By establishing a grid map, the simulation experiments of both the improved and unimproved partitioning algorithms are analyzed. First, three distinct environmental scenarios are constructed using the grid map, and then the simulation tests are conducted based on these scenarios. The results are presented in Figures 9 to 11. The symbol '◯' represents the starting point, 'Δ' denotes the target point, the black area indicates the obstacle zone, and the white area signifies the passable region.

In three distinct environmental scenarios, the path generated by the improved A* algorithm exhibits fewer inflection points and is smoother compared to the traditional A* algorithm. Additionally, at the inflection points of the path, there is a maintained safety distance from obstacles, which better aligns with the motion requirements of the robot.

Table 1 presents a comparative analysis of the improved A* algorithm and the traditional A* algorithm regarding path planning length and time across different environments. As shown in Table 1, the improved A* algorithm proposed in this paper consistently outperforms the traditional A* algorithm in both time and path length across three distinct environments. The combined results indicate that the path planning time is reduced by 56.08%, while the path length is decreased by 3.05% compared to the pre-improved path planning. This enhancement significantly improves the efficiency of the robot's autonomous navigation.

According to the hybrid path planning of the aforementioned fusion A* algorithm and DWA algorithm, the simulation experiment is carried out. The azimuth evaluation function coefficient in the evaluation function is set to 0.05, the distance evaluation function coefficient is 0.25, the speed evaluation function coefficient is 0.20, and the path fusion function coefficient is 0.15; the initial direction angle

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

Table 1. *Comparison of path planning length and time of A* algorithm before and after improvement in different environments.*

Environment types	Path planning algorithm	Length (m)	Time (ms)
Environment 1	Traditional A* algorithm	20.31	19.69
	Improved A* algorithm	18.85	6.82
Environment 2	Traditional A* algorithm	20.34	18.91
	Improved A* algorithm	20.10	5.20
Environment 3	Traditional A* algorithm	26.56	21.81
	Improved A* algorithm	26.21	14.51

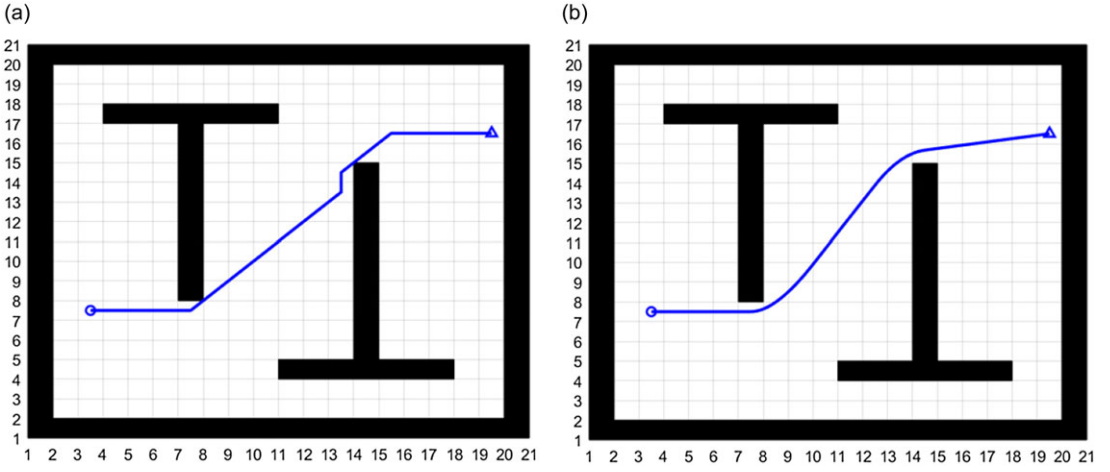


Figure 10. *Environment 2. (a) Traditional A* algorithm, (b) Improved A* algorithm.*

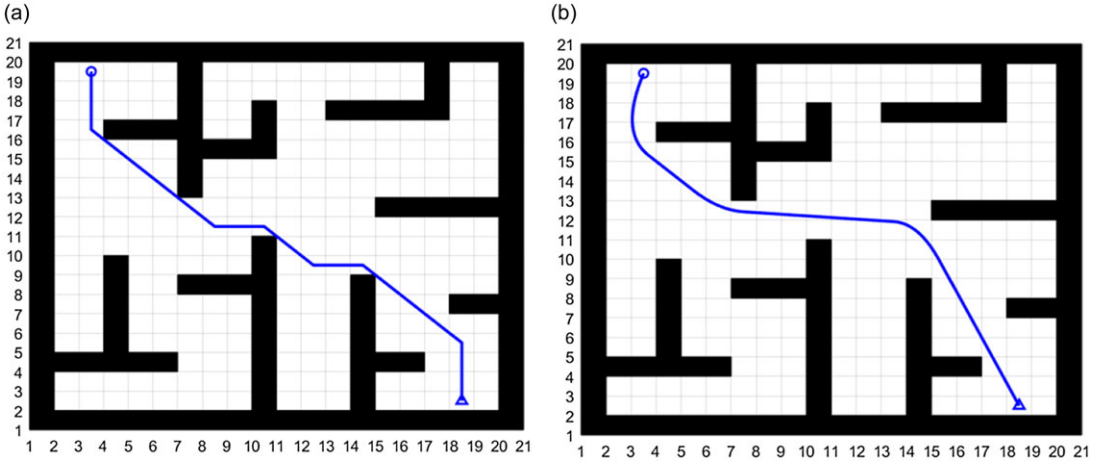


Figure 11. *Environment 3. (a) Traditional A* algorithm, (b) Improved A* algorithm.*

of the robot is π/4, the maximum forward speed is 2 m/s, and the maximum rotation speed is 0.2 rad/s. In order to better verify the path planning and real-time obstacle avoidance ability of the hybrid path planning algorithm, the path planning effects in the known static environment and the unknown dynamic environment are compared and analyzed. Figure 12 is the experimental results of hybrid path planning in three different scenarios in a static environment. The blue solid line represents the path planned by

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

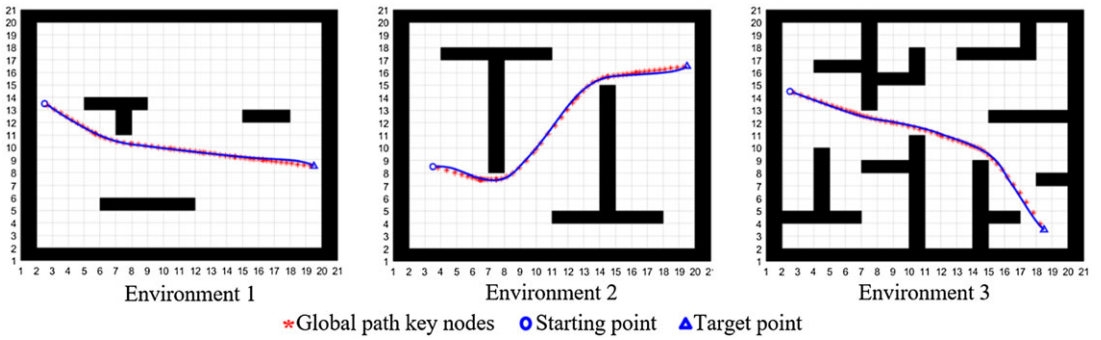


Figure 12. *Simulation results of hybrid path planning algorithm in static environment.*

the hybrid path planning algorithm, and the red '∗' represents the key node information of the global path.

From Figure 12, it is evident that in three distinct environmental scenarios, DWA algorithm is directed by the global path information generated by the enhanced A* algorithm. This allows the robot to navigate along the globally optimal path, effectively minimizing the risk of becoming trapped in local optima. Additionally, the path is smoother, facilitating the movement of the inspection robot.

Figure 13 illustrates the results of a dynamic real-time obstacle avoidance simulation using a hybrid path planning algorithm across three distinct scenarios in an unknown dynamic environment. In each scene, 1, 2, and 3 unknown dynamic obstacles are introduced, respectively.

From the information presented in Figure 13, it is evident that the hybrid path planning algorithm effectively avoids dynamic obstacles, even when faced with varying numbers of them. In addition to navigating around known obstacles, the algorithm is capable of avoiding newly emerging unknown obstacles. This functionality meets the autonomous navigation requirements for coal mine roadway inspection robots.

4. Experiment results and discussion

The inspection robot is primarily utilized in the coal mine roadway environment. To effectively validate the autonomous path planning and obstacle avoidance capabilities of the inspection robot in this setting, this paper first constructs a simulated roadway scene tailored to specific requirements. Subsequently, it conducts both dynamic and static environment path planning and autonomous navigation experiments, culminating in the collection of simulated roadway data. As illustrated in Figure 14, to accurately replicate the roadway environment, a scene characterized by low light and uneven terrain is selected, with several obstacles introduced in the simulated roadway to assess the inspection robot's autonomous obstacle avoidance capabilities.

The grid map created using LiDAR scanning of the environment is displayed in Figure 15. In this map, the white areas represent the passable zones for the inspection robot, while the black areas indicate impassable obstacles.

4.1. Static environment

In a static environment, the A* algorithm plays a crucial role in enabling the inspection robot to complete the autonomous driving task, as outlined in the hybrid path planning algorithm described above. To evaluate the reliability of the improved A* algorithm, we compare the global path planning effectiveness of the A* algorithm before and after the enhancement through experimental results. The findings are illustrated in Figure 16.

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

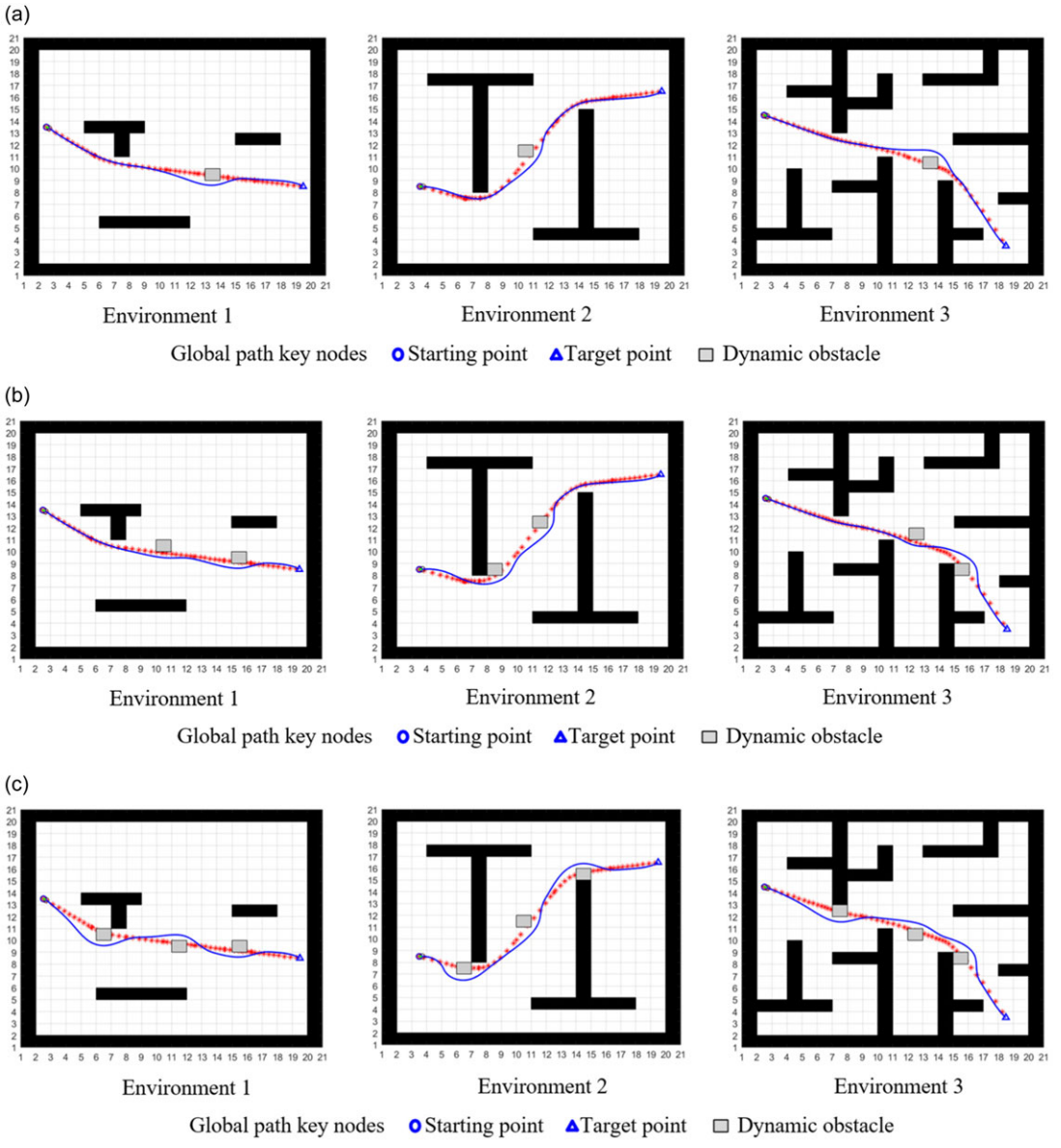


Figure 13. *Simulation results of hybrid path planning algorithm under different dynamic obstacles. (a) 1 dynamic obstacle, (b) 2 dynamic obstacle, (c) 3 dynamic obstacle.*

It can be found that the path planned by the improved A* algorithm is smoother compared to the traditional A* algorithm and has a certain safety distance from the obstacles, which is more favorable for the robot to walk.

In order to further verify the performance of the improved A* algorithm, five groups of different target points are randomly set up in the experimental environment to compare the path planning time and the number of inflection points between the traditional A* algorithm and the improved A* algorithm, and the experimental results are shown in the Table 2. From the table, it can be found that the path planning time of the improved A* algorithm is significantly less than that of the traditional A* algorithm, and the time of path planning is shortened by 49.60 % compared with that of the pre-improvement path

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

Table 2. *Comparison of path planning results between traditional A* algorithm and improved A* algorithm.*

Path types	Path planning algorithm	Path planning time (ms)	Number of inflection points
Path 1	Traditional A* algorithm	68.77	13
	Improved A* algorithm	35.14	5
Path 2	Traditional A* algorithm	62.70	10
	Improved A* algorithm	32.15	4
Path 3	Traditional A* algorithm	52.53	6
	Improved A* algorithm	21.15	2
Path 4	Traditional A* algorithm	29.14	4
	Improved A* algorithm	15.43	1
Path 5	Traditional A* algorithm	28.70	5
	Improved A* algorithm	18.02	0

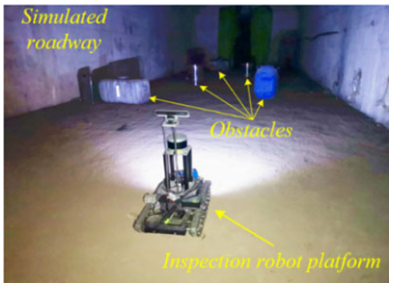


Figure 14. *Experimental environment.*

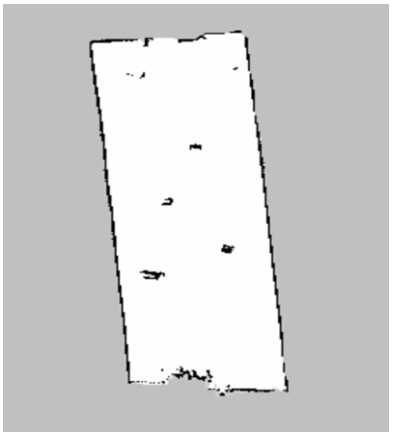


Figure 15. *Simulated roadway map.*

planning, and at the same time, the number of path inflection points is less, and the smoothing degree is higher.

Guided by the global path planning established by the improved A* algorithm, the DWA algorithm is employed to dynamically adjust the path in response to various obstacles in the environment, ensuring real-time obstacle avoidance. The actual movement of the inspection robot, based on the hybrid path planning algorithm in a known static environment, is illustrated in Figure 17. The left half of

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

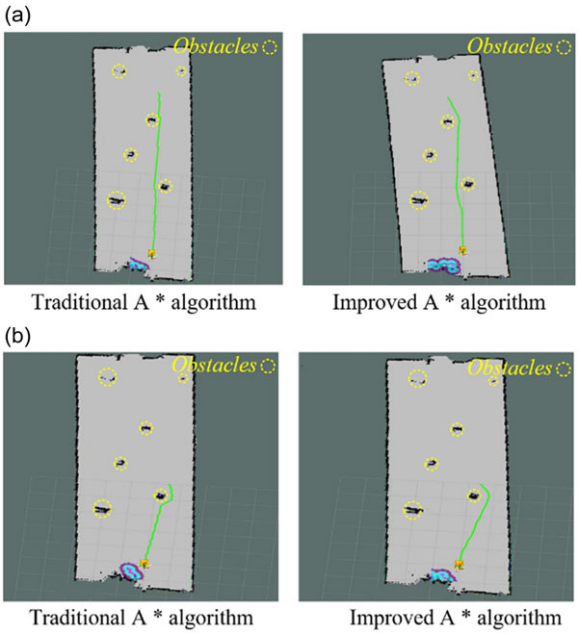


Figure 16. *Comparison of path planning between traditional A* algorithm and improved A* algorithm. (a) Path 1, (b) Path 2.*

each diagram displays the path planning information of the inspection robot alongside the visual interface showing the robot's current position. Conversely, the right half depicts the robot's position within the actual scene. The green solid line represents the path information generated by the path planning algorithm, while the blue line indicates the path that the inspection robot has actually traversed.

Based on the information presented in Figure 17, it is evident that the inspection robot can effectively plan its driving path and navigate safely by employing a hybrid path planning method that combines the improved A* algorithm with the DWA algorithm. Taking the pose deviation as the standard, the initial pose of the robot is defined as the origin. The position deviation of the actual endpoint and the target point in the x direction, y direction, and angle is measured multiple times using odometer information to assess the effectiveness of the inspection robot's autonomous driving.

In the known static environment, 10 autonomous driving experiments were conducted, and the experimental data were recorded. The results of these experiments are presented in Table 3, where Δx represents the absolute position deviation in the x direction, and Δy represents the absolute position deviation in the y direction.

It can be observed from Table 3 that after the inspection robot reaches the designated target point along the path planned by the algorithm described in this paper, the maximum positional deviation in the x-axis direction is 0.101 m, the minimum positional deviation is 0.034 m, and the absolute average positional deviation is 0.075 m. The maximum position deviation along the y-axis is 0.090 m, while the minimum position deviation is 0.013 m. The absolute average position deviation is 0.055 m. In terms of angular position deviation, the maximum is 0.234 radians (13.41°), the minimum is 0.007 radians (0.401°), and the absolute average position deviation is 0.059 radians (3.38°).

4.2. Unknown dynamic environment

As shown in Figure 18, the validation of hybrid path planning incorporating the improved A* algorithm and the DWA algorithm is carried out based on unknown or partially unknown environments by setting unknown dynamic obstacles on the traveling path of the inspection robot.

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

Table 3. *The experimental results of autonomous driving of inspection robot in static environment are known.*

Experimental times	Δx (m)	Δy (m)	Angle deviation (*rad*)
1	0.070	0.068	0.034
2	0.062	0.082	0.031
3	0.101	0.013	0.181
4	0.067	0.031	0.007
5	0.094	0.090	0.021
6	0.085	0.025	0.039
7	0.088	0.070	0.020
8	0.034	0.088	0.013
9	0.059	0.067	0.012
10	0.094	0.018	0.234

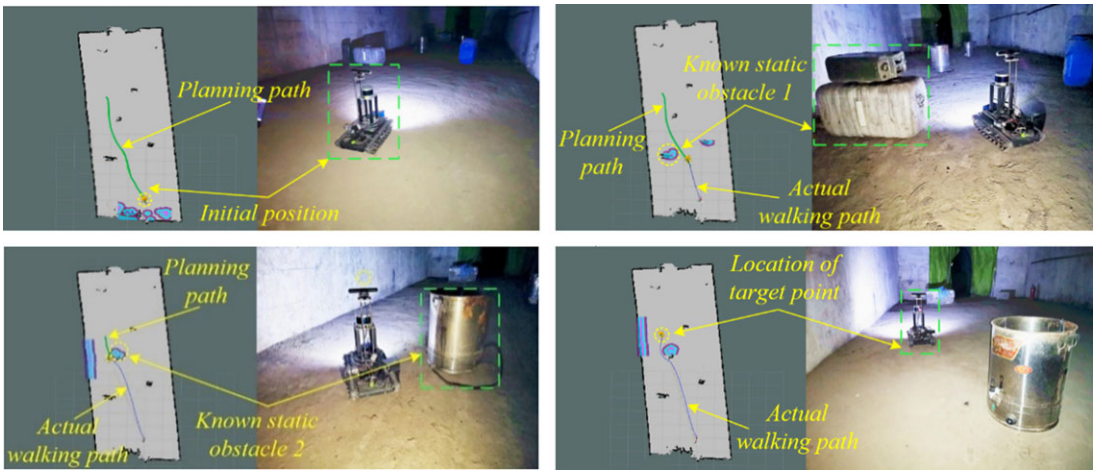


Figure 17. *The actual driving process in a known static environment.*

From Figure 18(a), it is evident that the global path planning A* algorithm plays a crucial role in path planning when there are no unknown dynamic obstacles in the forward direction of the inspection robot. However, when unknown dynamic obstacles appear in the vicinity of the inspection robot, the local path planning DWA algorithm becomes essential. The robot promptly adjusts its local path to avoid obstacles based on the detected obstacle information, as illustrated in Figure 18(b). When an unknown dynamic obstacle moves around the inspection robot, the DWA algorithm recalibrates the local path according to the position information of the obstacle within the global map, thereby fulfilling the obstacle avoidance requirement. This adjustment is depicted in Figure 18(c). Consequently, the experimental results presented in Figure 18 demonstrate that the hybrid path planning approach, which combines the improved A* algorithm with the DWA algorithm, can effectively adapt the path in response to environmental changes surrounding the inspection robot. This method enables the planning of a safer and more reliable path that meets practical requirements.

In addition, in order to verify the practical application effect of the hybrid path planning algorithm that integrates the improved A* algorithm and DWA algorithm, pedestrians are added as unknown dynamic obstacles in the simulated alleyway scenario, and then autonomous obstacle avoidance driving experiments are carried out based on the inspection robot platform in the unknown dynamic environment.

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

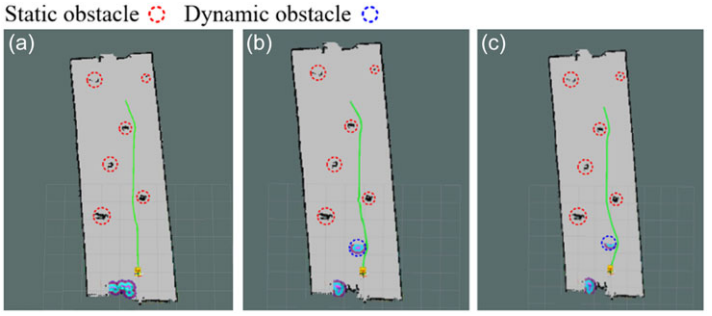


Figure 18. *Mixed path planning effect diagram. (a) No unknown dynamic obstacles appeared, (b) unknown dynamic obstacles appeared, (c) path planning results.*

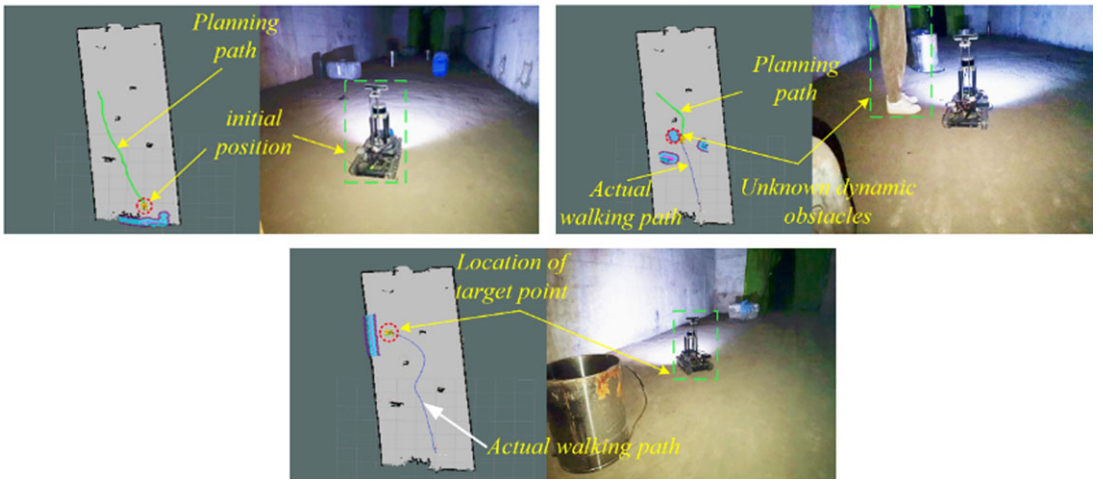


Figure 19. *The actual driving process in unknown dynamic environment.*

As shown in Figure 19, the path planning results without adding unknown pedestrians and the initial position of the inspection robot in the actual scene are shown. After the start point and target point are set for the inspection robot in the completed map model, the inspection robot plans a path from the start point to the target point based on the hybrid path planning algorithm.

Similar to the known static environment, in order to verify the effect of dynamic obstacles on the inspection robot's travel to the target point, autonomous driving experiments were conducted in an environment containing unknown dynamic obstacles, and the experimental results are shown in Table 4.

As shown in Table 4, after the inspection robot reaches the target point along the planned path in the unknown dynamic environment, the maximum position deviation in the x axis direction is 0.148 m, the minimum position deviation is 0.020 m, and the absolute average position deviation is 0.081 m; the maximum position deviation in the y axis direction is 0.143 m, the minimum position deviation is 0.011 m, and the absolute average position deviation is 0.072 m; the maximum position deviation in the angular direction is 0.543 radians (31.11°), the minimum position deviation is 0.001 radians (0.057°), and the absolute average position deviation is 0.099 radians (5.67°). Compared to the known static environment, the inspection robot occasionally has a larger angular deviation from the target position angle due to the influence of dynamic obstacles, but the position deviation occurring in the x axis direction and y axis direction is not much different from the experimental results in the static environment. Combined with the working principle of LiDAR, the autonomous driving function of the inspection robot used in this paper can also meet the requirements in an unknown dynamic environment.

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

Table 4. *Experimental results of autonomous driving of inspection robot in unknown dynamic environment.*

Experimental times	**Δx(m)**	**Δy(m)**	**Angle deviation (*rad*)**
1	0.136	0.033	0.072
2	0.079	0.036	0.178
3	0.111	0.011	0.040
4	0.148	0.028	0.543
5	0.054	0.143	0.001
6	0.084	0.011	0.018
7	0.023	0.305	0.037
8	0.086	0.065	0.089
9	0.073	0.045	0.013
10	0.020	0.052	0.006

5. Conclusion

Aiming at the problem of incomplete state estimation of inspection robots in degraded environments, such as underground roadways, and the difficulty of realizing automated and unmanned monitoring of coal mine roadways, research on the autonomous driving technology of inspection robots has been carried out, and the main conclusions obtained are as follows:

(1) The inspection robot system platform has been constructed, and the analysis and establishment of the inspection robot system model have been completed. This provides theoretical support for the realization of autonomous navigation and environmental scanning by the inspection robot.

(2) A hybrid path planning algorithm that integrates an improved A* algorithm with DWA algorithm is proposed. The global path generated by the improved A* algorithm serves as a reference, effectively mitigating the limitations of the DWA algorithm, which is susceptible to local optima. This method also allows the robot to dynamically avoid obstacles.

(3) A raster map simulating the roadway environment has been established, and experiments on autonomous path planning and obstacle avoidance have been conducted in both known static and unknown dynamic environments. These experiments aim to evaluate the autonomous navigation and real-time obstacle avoidance capabilities of the inspection robot. The results indicate that the proposed path planning algorithm effectively meets the autonomous navigation requirements of the inspection robot in the coal mine roadway environment.

Although the proposed algorithm has been validated in a simulated coal mine tunnel environment, the path planning algorithm presented in this paper is based on a 2D environmental map. Recognizing that 3D information encompasses additional environmental features that enhance robot decision-making, Therefore, in the subsequent research, we will carry out the research on the path planning and autonomous operation algorithm of the inspection robot in the three-dimensional environment, so as to improve the autonomous operation capability of the inspection robot as well as the level of intelligence.

Author contributions. Congcong Gu: Software, writing – original draft. Songyong Liu: Funding acquisition, project administration, writing – review and editing. Hongsheng Li: Writing – review and editing. Kewen Yuan: Formal analysis and writing – review and editing. Wenjie Bao: Funding acquisition and formal analysis.

Financial support. This work was supported by the Funded by Basic Research Program of Jiangsu (Grant Nos. BK20211531), National Natural Science Foundation of China (Grant Nos. 52405143), Postdoctoral Fellowship Program of CPSF (Grant Nos.

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

GZC20233008), Funded by Basic Research Program of Jiangsu (Grant Nos. BK20241639), and the Priority Academic Program Development of Jiangsu Higher Education Institute of China.

Competing interests. The authors declare no competing interests exist.

Ethical approval. Not applicable under the heading.

References

[1] J. Katrasnik, F. Pernus and B. Likar, "A survey of mobile robots for distribution power line inspection," *IEEE Trans. Power Deliver.* **25**(1), 485–493 (2010).
[2] Z. P. Wang, B. He, Y. M. Zhou, K. Liu and C. H. Zhang, "Design and implementation of a cable inspection robot for cable-stayed bridges," *Robotica* **39**(8), 1417–1433 (2021).
[3] J. H. Yin, X. M. Liu, Y. Q. Wang and Y. C. Wang, "Design and motion mechanism analysis of screw-driven in-pipe inspection robot based on novel adapting mechanism," *Robotica* **42**(4), 1297–1319 (2024).
[4] Q. Jiang, Y. D. Liu, Y. J. Yan, P. Xu, L. Pei and X. C. Jiang, "Active pose relocalization for intelligent substation inspection robot," *IEEE Trans. Ind. Electron.* **70**(5), 4972–4982 (2023).
[5] R. K. Dubey, S. S. Sohn, T. Thrash, C. Hölscher, A. Borrmann and M. Kapadia, "Cognitive path planning with spatial memory distortion," *IEEE Trans. Visual. Comput. Graphics* **29**(8), 3535–3549 (2023).
[6] S. H. Li, J. S. Gu, Z. J. Li, S. F. Li, B. X. Guo, S. B. Gao, F. Zhao, Y. W. Yang, G. X. Li and L. F. Dong, "A visual SLAM-based lightweight multi-modal semantic framework for an intelligent substation robot," *Robotica* **42**(7), 2169–2183 (2024).
[7] L. Chen, Y. Ma, Y. Zhang and J. G. Liu, "Obstacle avoidance and multitarget tracking of a super redundant modular manipulator based on Bezier curve and particle swarm optimization," *Chin. J. Mech. Eng.* **33**(1), 71 (2020).
[8] S. N. Hu, S. P. Tian, J. S. Zhao and R. Q. Shen, "Path planning of an unmanned surface vessel based on the improved A-star and dynamic window method," *J. Marine Sci. Eng.* **11**(5), 1060 (2023).
[9] T. Wang, Z. L. Xue, X. Q. Dong and S. L. Xie, "Autonomous intelligent planning method for welding path of complex ship components," *Robotica* **39**(3), 428–437 (2021).
[10] B. Hao, J. S. Zhao and Q. Wang, "A review of intelligence-based vehicles path planning," *SAE Int. J. Commer. Veh.* **16**(4), 329–339 (2023).
[11] X. M. Yan, H. Huang, Z. F. Hao and J. H. Wang, "A graph-based fuzzy evolutionary algorithm for solving two-echelon vehicle routing problems," *IEEE Trans. Evol. Comput.* **24**(1), 129–141 (2020).
[12] F. D. Frederick, M. S. Marlim and D. Kang, "Optimization of chlorine injection schedule in water distribution networks using water age and breadth-first search algorithm," *Water* **16**(3), 486 (2024).
[13] E. Plaku, K. E. Bekris, B. Y. Chen, A. M. Ladd and L. E. Kavraki, "Sampling-based roadmap of trees for parallel motion planning," *IEEE Trans. Robot.* **21**(4), 597–608 (2005).
[14] X. Yin, P. Cai, K. W. Zhao, Y. Zhang, Q. Zhou and D. J. Yao, "Dynamic path planning of AGV based on kinematical constraint A* algorithm and following DWA fusion algorithms," *Sensors - Basel* **23**(8), 4102 (2023).
[15] S. Zhang, X. J. Liu, B. K. Yan, J. Bi and X. D. Han, "A real-time look-ahead trajectory planning methodology for multi small line segments path," *Chin. J. Mech. Eng.* **36**(1), 59 (2023).
[16] M. R. Luo, Y. N. Tian, E. Li, M. H. Chen and M. Tan, "A local obstacle avoidance and global planning method for the follow-the-leader motion of coiled hyper-redundant manipulators," *IEEE Trans. Ind. Inform.* **20**(4), 6591–6602 (2024).
[17] O. Tutsoy, K. Ahmadi, D. Asadi, S. Y. Nabavi-Chashmi and J. Iqbal, "Minimum distance and minimum time optimal path planning with bioinspired machine learning algorithms for faulty unmanned air vehicles," *IEEE Trans. Intel. Transp.* (2024).
[18] X. H. Wang, X. Ma and Z. W. Li, "Research on SLAM and path planning method of inspection robot in complex scenarios," *Electronics* **12**(10), 2178 (2023).
[19] B. Bogaerts, S. Sels, S. Vanlanduit and R. Penne, "A gradient-based inspection path optimization approach," *IEEE Robot. Autom. Lett.* **3**(3), 2646–2653 (2018).
[20] S. M. Chaves, A. Kim, E. Galceran and R. M. Eustice, "Opportunistic sampling-based active visual SLAM for underwater inspection," *Auton. Robot.* **40**(7), 1245–1265 (2016).
[21] Y. S. Tan, J. Ouyang, Z. Zhang, Y. L. Lao and P. J. Wen, "Path planning for spot welding robots based on improved ant colony algorithm," *Robotica* **41**(3), 926–938 (2023).
[22] P.Y. Jiang, R.J. Zhao and Q.Q. Chu. Inspecter - A Cad Directed Intelligent Inspection Planning System for Coordinate Measuring Machines. **In:** *Proceedings of The International Conference on Computer Integrated Manufacturing, ICCCIM 91: Manufacturing Enterprises of The 21st Century* (B.S. Lim, ed.) (International Conf on Computer Integrated Manufacturing (ICCIM 91) (1991), pp. 598–600.
[23] J. Q. Chen, M. Y. Li, Y. S. Su, W. Q. Li and Y. Z. Lin, "Direction constraints adaptive extended bidirectional A* algorithm based on random two-dimensional map environments," *Robot. Auton. Syst.* **165**, 104430 (2023).
[24] Y. J. Liu, C. Wang, H. Wu and Y. L. Wei, "Mobile robot path planning based on kinematically constrained A-star algorithm and DWA fusion algorithm," *Mathematics -Basel* **11**(21), 4552 (2023).
[25] Y. Zhang, J. F. Tang, S. M. Lv and X. G. Luo, "Floyd-A* algorithm solving the least-time itinerary planning problem in urban scheduled public transport network," *Math. Probl. Eng.* **2014**(1), 185383 (2014).

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press

[26] D. Zhang, R. J. Ju and Z. C. Cao, "Reinforcement learning-based motion control for snake robots in complex environments," *Robotica* **42**(4), 947–961 (2024).
[27] G. S. Tsitsiashvili and A. S. Losev, "Application of the Floyd algorithm to the asymptotic analysis of networks with unreliable ribs," *Automat. Remote Control* **69**(7), 1262–1265 (2008).
[28] M. Ruchanurucks, "Humanoid robot upper body motion generation using B-spline-based functions," *Robotica* **33**(4), 705–720 (2015).
[29] S. Q. Wang, Y. Y. Hu, Z. N. Liu and L. J. Ma, "Research on adaptive obstacle avoidance algorithm of robot based on DDPG-DWA," *Comput. Electr. Eng.* **109**, 108753 (2023).
[30] C. J. Liu, J. Zhang, N. Li, Q. H. Zhuo, K. H. Yu and K. X. Lin, "Investigation on the optimization strategy of DWA algorithm for path planning of the USVs with the consideration of environmental factors," *Int. J. Pattern. Recogn.* (2024).
[31] X. Yin, P. Cai, K. W. Zhao, Y. Zhang, Q. Zhou and D. J. Yao, "Dynamic path planning of AGV based on kinematical constraint A* algorithm and following DWA fusion algorithms," *Sensors-Basel* **23**(8), 4102 (2023).

Cite this article: C. Gu, S. Liu, H. Li, K. Yuan and W. Bao (2025). "Research on hybrid path planning of underground degraded environment inspection robot based on improved A* algorithm and DWA algorithm", Robotica **43**, 887–908. https://doi.org/10.1017/S0263574725000037

https://doi.org/10.1017/S0263574725000037 Published online by Cambridge University Press