Robotica (2024), 1–16
doi:10.1017/S0263574724001930

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Research on the path planning algorithm and obstacle-crossing motion planning strategy for a cable trench inspection robot

Linjie Dong[1], Renfei Zhang[1], Junfei Wang[1], Jie Li[2], Shanshan Wang[3] and Xingsong Wang[1]

[1]School of Mechanical Engineering, Southeast University, Nanjing, China
[2]College of Automation & College of Artificial Intelligence, Nanjing University of Posts and Telecommunications, Nanjing, China
[3]Nanjing Forestry University, Nanjing, China
Corresponding author: Xingsong Wang; Email: xswang@seu.edu.cn

Received: 13 August 2024; **Revised:** 25 September 2024; **Accepted:** 15 October 2024

Keywords: inspection robot; path planning; obstacle-crossing motion planning; improved A* algorithm

Abstract

The path planning and obstacle-crossing motion planning of cable trench inspection robots are essential for achieving automated inspection. To improve path planning efficiency and obstacle navigation in complex environments, an enhanced global path planning algorithm based on the A* algorithm has been developed, combined with an improved Dynamic Window Approach (DWA) for local path planning. For unavoidable obstacles, a specific obstacle-crossing motion planning strategy has been formulated. The enhanced A* algorithm improves efficiency and safety through adaptive neighborhood expansion and the elimination of redundant path points. The improved DWA algorithm enables real-time dynamic obstacle avoidance in local path planning. The simulation results on a 20 × 20 grid map indicate that the improved A* algorithm reduces the number of nodes by 58.4% and shortens the path length by 6.1% compared to the traditional A* algorithm, demonstrating significant advantages over other conventional path planning algorithms. In the simulation experiments integrating global and local path planning, the enhanced A* algorithm combined with the improved DWA algorithm reduces the path length by 3.2% on the 20 × 20 grid map compared to the integration with the traditional DWA algorithm. On the 30 × 30 grid maps with different obstacle configurations, the path lengths are reduced by 3.5% and 3.6%, respectively. In the obstacle-crossing experiments, the robot successfully overcame obstacles of 10 cm and 20 cm in height. The proposed path planning algorithm and obstacle-crossing motion planning strategy hold substantial application potential in complex environments, offering reliable technical support for cable trench inspection robots.

1. Introduction

Cable trenches are critical components of the power transmission system [1]. To ensure the safe operation of the power grid, regular inspection and maintenance of cable trenches are essential. Currently, these inspections are primarily performed manually, which is labor-intensive and challenging due to the potential presence of toxic and hazardous gases within the trenches [2]. This not only complicates manual inspection but also poses significant safety risks.

Replacing manual inspection with robots can significantly improve the quality and efficiency of inspections [3, 4], aligning with future development trends. Given the complex environment of cable trenches, robots must be equipped with mechanical arms capable of interacting with cables or equipment within the trenches and must possess strong obstacle-crossing capabilities. In our previous work, we designed a tracked inspection robot for cable trenches, equipped with a six-degree-of-freedom

https://doi.org/10.1017/S0263574724001930 Published online by Cambridge University Press

mechanical arm and high climbing and obstacle-crossing performance [5]. Additionally, to achieve automated and intelligent inspection, the robot must possess robust path planning and trajectory tracking capabilities [6].

Robot path planning includes both global and local path planning. Common algorithms for global path planning include bio-inspired algorithms, sampling-based algorithms, and grid-based search algorithms [7]. Bio-inspired algorithms are inspired by animal self-organization behavior, natural evolution, and biological neural systems [8], such as genetic algorithms (GAs) [9] and ant colony algorithms [10]. These are suitable for solving and optimizing complex problems but are limited by slow processing speeds and susceptibility to local optima [11, 12]. Sampling-based algorithms, like the well-known Rapidly-exploring Random Tree (RRT) algorithm [13], iteratively add random nodes to draw edges and use collision detection to validate edge effectiveness, performing well in high-dimensional configuration spaces. However, traditional RRT algorithms face limitations such as slow convergence and fluctuating path quality [14]. Grid-based search algorithms, typically represented by A* [15], offer advantages in real-time performance and optimization [16]. O. A. R. Abdul Wahhab et al. [17] employed three optimization algorithms, including A*, to address the issues of obstacle avoidance and minimizing path length in path planning, achieving promising results. Building on the minimum cost equation of the A* algorithm, Liu et al. [18] further enhanced the fitness function of the Improved Gray Wolf Optimization algorithm. This resulted in a fast iterative path planning method tailored for parking lots. To enhance the performance of the A* algorithm, many researchers have proposed improvements. Wang et al. [19] introduced distance expansion from obstacles, bidirectional search, and smoothing processes to increase search efficiency, reduce redundant path points, and decrease path turns. Li et al. [20] introduced a bidirectional search strategy with weighted heuristic functions using exponential decay and incorporated path node filtering, improving search efficiency, reducing redundant nodes, and minimizing turning angles. To address the issues of excessive node searches and low computational efficiency in end-effector path planning for robotic arms using the traditional A* algorithm, Tang et al. [21] proposed an improved A* algorithm. This method incorporates a novel node search strategy and local path optimization technique, significantly reducing the number of searched nodes and enhancing search efficiency. Further considering robot motion constraints, Chen et al. [22] guided hybrid A* for secondary path planning, optimizing the algorithm path, eliminating redundant nodes, and ensuring safe and smooth robot movement. However, the standard A* algorithm often encounters difficulties when dealing with dynamic obstacles and may generate suboptimal paths in certain environments.

Common methods for local path planning include the Artificial Potential Field (APF) [23, 24] and Dynamic Window Approach (DWA) algorithms. The APF algorithm is prone to local optima and oscillations, insufficiently considers dynamic constraints, and handles dynamic obstacles poorly. The DWA algorithm samples the current environment, including robot speed, motion parameters, and position. Although the DWA algorithm is capable of handling dynamic obstacles, it may become trapped in local optima and has limitations in generating smooth paths. To improve the performance of the DWA algorithm, many researchers have proposed enhancements. Kobayashi et al. [25] combined DWA with a virtual manipulator (VM) to create DWV, generating path candidates using variable speeds adjusted by the VM and predicted positions of static and dynamic obstacles. Sun et al. [26] optimized the evaluation function's sub-functions using fuzzy control logic by dynamically modifying weight coefficients and selecting critical points on the global path as key sub-targets in the local motion planning phase, reducing planning time and shortening the planned path length. Dobrevski et al. [27] utilized neural networks to model the motion trajectories of dynamic obstacles and predict corresponding adjustments to the DWA weights, thereby enabling safe local navigation for robots in dynamic environments.

Building upon prior research [5], this study further optimizes robot path planning by developing an enhanced integrated global and local path planning algorithm based on traditional A* and DWA methods. Compared to traditional algorithms, the enhanced A* algorithm improves the smoothness of the robot's path and accelerates global exploration efficiency, while the optimized DWA enhances the robot's ability to avoid both static and dynamic obstacles in local environments. The integration of these two algorithms results in significant improvements in planning efficiency, safety, and path smoothness. Additionally, a

https://doi.org/10.1017/S0263574724001930 Published online by Cambridge University Press

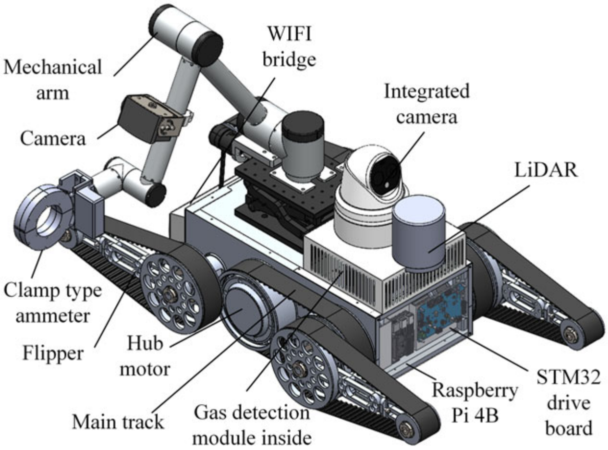


Figure 1. *Cable trench inspection robot.*

robot obstacle-crossing motion planning strategy is devised specifically for convex obstacle scenarios where avoidance is not feasible. Simulation and experimental results demonstrate that the proposed robot path planning algorithm reduces path length and enhances both path search efficiency and path smoothness compared to other common algorithms. The obstacle-crossing motion planning strategy effectively guides the robot in overcoming convex obstacles.

2. Robot path planning and obstacle-crossing strategy

2.1. Cable trench inspection robot

The path planning algorithm and obstacle-crossing motion planning strategy designed in this paper are applied to the cable trench inspection robot, as shown in Fig. 1. The robot features a structure comprising a six-degree-of-freedom robotic arm and a quadrupedal six-track walking mechanism. It is equipped with sensors including a 32-line laser radar and integrated infrared and visible light cameras. Detailed design parameters of the robot and its control system can be found in our previous work [5].

2.2. Improved global path planning algorithm based on A*

Automated inspection in cable trenches requires robots to autonomously plan a safe and efficient path on a predefined grid map to reach their destinations. The A^* algorithm provides advantages in search efficiency and path length optimization. Traditional A^* integrates optimal path search principles from Dijkstra algorithm with heuristic strategies akin to greedy algorithms [28]. Equation (1) defines the evaluation function used to estimate the anticipated total distance from the starting node to the target node, guiding the search toward nodes closer to the projected goal.

$$F(n) = G(n) + H(n) \tag{1}$$

In this context, $F(n)$ represents the estimated total cost from the initial node through node n to the target node; $G(n)$ is the actual cost from the initial node to node n; and $H(n)$ is the heuristic estimate cost from node n to the target node, typically calculated using Euclidean distance, Manhattan distance, or Chebyshev distance. The traditional A^* algorithm is constrained by its inherent algorithmic nature and the discrete nature of grid-based maps, resulting in planned paths that may include unnecessary turns and risks approaching obstacles diagonally. To address the inefficiencies, lack of smoothness, and safety concerns in traditional A^* path planning, this study optimizes the traditional A^* algorithm based on its evaluation function as follows.

https://doi.org/10.1017/S0263574724001930 Published online by Cambridge University Press

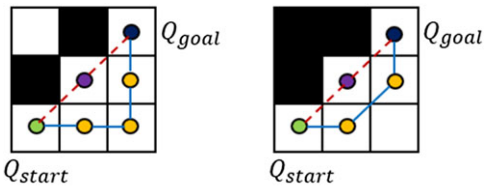


***Figure 2.** Optimizing child node selection using artificial potential fields.*

2.2.1. Adaptive neighborhood expansion

To accommodate the uneven distribution of obstacles in the environment and enhance the accuracy of shortest path estimation, we introduce the concept of adaptive "step size." This approach involves dynamically adjusting the obstacle probability P to effectively manage the complexity of the current map environment.

P is defined as the proportion of obstacle grid units in the local environment formed by the current point and the target point. Assuming N is the number of obstacle grid units in this local environment, with current coordinates (x_{cur}, y_{cur}) and goal coordinates (x_{goal}, y_{goal}), the expression for obstacle probability P is presented in Eq. (2).

$$P = \frac{N}{(|x_{cur} - x_{goal}| + 1) \times (|y_{cur} - y_{goal}| + 1)} \tag{2}$$

Based on obstacle probability P, dynamically adjust the node neighborhood expansion method, and search step size: in densely populated obstacle areas, employ a conservative neighborhood expansion strategy to decrease the search step size and minimize collision risks; in sparsely populated obstacle areas, employ a more aggressive neighborhood expansion strategy to increase the search step size and enhance search speed.

2.2.2. Optimization of evaluation function

The search performance of the traditional A* algorithm is influenced by the heuristic function $H(n)$, where varying weights of $H(n)$ may increase the planned path length [29]. By incorporating obstacle probability P into $F(n)$, dynamically adjusting the weight of $H(n)$ based on obstacle probability, we adapt the A* search process accordingly. The enhanced evaluation function is presented in Eq. (3).

$$F(n) = G(n) + \left(1 - \ln^{(P)}\right) H(n) \tag{3}$$

When obstacles are scarce, lowering obstacle probability P increases the weight of $H(n)$, thereby accelerating the A* algorithm's search process. Conversely, in densely populated obstacle areas, raising obstacle probability P decreases the weight of $H(n)$, improving the accuracy of the A* algorithm's search. Integrating obstacle probability P for search preference adjustment enhances the efficiency and adaptability of the A* algorithm across diverse map environments.

2.2.3. Optimization of child node selection

During the A* algorithm's search process, it is critical to avoid paths that diagonally traverse through obstacle vertices, as this increases collision risks. Therefore, it is essential to evaluate the spatial relationship between child nodes and obstacles during their generation to prevent paths from diagonally crossing obstacle vertices.

The specific improvement in child node selection is depicted in Fig. 2. When encountering obstacles along the search path, an obstacle repulsive field, inspired by the artificial potential field method, is introduced. This mechanism causes nodes approaching obstacles to experience repulsive forces, prompting the algorithm to reevaluate and adjust path nodes, thereby ensuring the discovery of safer paths.

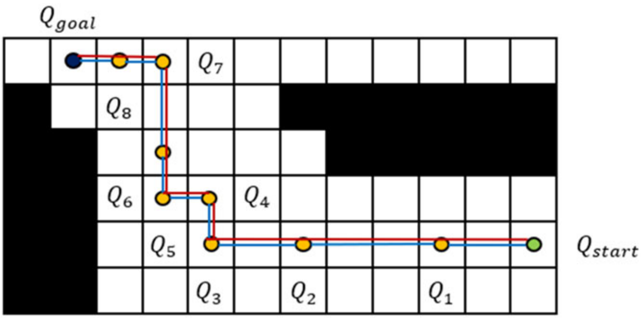


Figure 3. *Extracting key points of the path.*

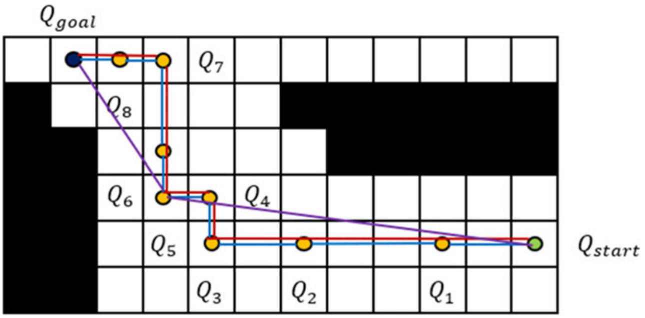


Figure 4. *Removing redundant points from the path.*

2.2.4. Path smoothing

Paths generated by traditional A* algorithms often lack optimality and include multiple turning points, which can hinder actual robot movement. Therefore, additional path smoothing is necessary. This process involves extracting key path points and eliminating redundant path segments to improve the overall smoothness of the planned path, thereby reducing energy consumption during robot motion.

The specific process of extracting key points from the path is illustrated in Fig. 3: Suppose the initial path planned by the A* algorithm is $Q_{start} - Q_1 - Q_2 - Q_3 - Q_4 - Q_5 - Q_6 - Q_7 - Q_8 - Q_{goal}$. This process involves removing intermediate points along the straight segments of the path, retaining only the starting point, path turning points, and the endpoint. This results in a streamlined path such as $Q_{start} - Q_3 - Q_4 - Q_5 - Q_7 - Q_{goal}$. This approach significantly reduces pauses in robot motion, stabilizes movement, and prevents excessive acceleration and deceleration that could lead to impact loads.

The specific process of removing redundant path points is depicted in Fig. 4: Starting from the initial point, the path following the extraction of key points is evaluated. It checks whether there are obstacles between Q_{start} and Q_4. If no obstacles exist, Q_4 is removed. The process continues to evaluate whether obstacles exist between Q_{start} and Q_5. If obstacles are present, this node is retained as an intermediate point, and the search continues from this node as the new starting point to find the next intermediate node until reaching the endpoint. This iterative process results in a streamlined path such as $Q_{start} - Q_5 - Q_{goal}$, effectively reducing the path length.

2.3. Improved local path planning algorithm based on DWA

Global path planning provides the robot with a path to the target point in a static map. However, the environment in cable trenches may change, potentially obstructing previously accessible paths with obstacles. Therefore, enhancing the robot's capability for local path planning is crucial. This paper

https://doi.org/10.1017/S0263574724001930 Published online by Cambridge University Press

improves upon the traditional DWA algorithm to achieve effective local path planning for the robot. The principle of traditional DWA [30] involves sampling multiple sets of velocities (v, ω) at a specific resolution in velocity space and simulating the trajectories of these velocities over a specified time period. After obtaining feasible trajectories, these trajectories are evaluated using the scoring function in Eq. (4), and the optimal trajectory corresponding to (v, ω) is selected to drive the robot's motion.

$$G(v, \omega) = \sigma(\alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot velocity(v, \omega)) \tag{4}$$

Here, $heading(v, \omega) = 180^\circ - |\theta_{da}|$ represents the heading evaluation function. θ_{da} denotes the robot's orientation angle, which is the angle between the direction of motion and the line connecting the robot's geometric center to the target point. The distance function $dist(v, \omega) = \sqrt{(x_{pred} - x_{obs})^2 + (y_{pred} - y_{obs})^2}$ serves as a safety metric, indicating the distance between the predicted path endpoint and the nearest obstacle, where (x_{pred}, y_{pred}) are the coordinates of the predicted path endpoint and (x_{obs}, y_{obs}) are the coordinates of the nearest obstacle. The velocity function $velocity(v, \omega) = v$ represents the speed, indicating the linear velocity of the robot in the current sampling space. σ, α, β, and γ are weight coefficients that influence the overall effectiveness of the algorithm.

To prevent discontinuities caused by a single function dominating the overall evaluation function, it is necessary to normalize the evaluation function as shown in Eq. (5).

$$\begin{cases} heading_{normal}(m) = \dfrac{heading(m)}{\sum_{i=1}^{n} heading(m)} \\ dist_{normal}(m) = \dfrac{dist(m)}{\sum_{i=1}^{n} dist(m)} \\ velocity_{normal}(m) = \dfrac{velocity(m)}{\sum_{i=1}^{n} velocity(m)} \end{cases} \tag{5}$$

where n represents the number of velocity and angular velocity combinations that satisfy the constraints and m denotes the current path being evaluated.

Although the DWA algorithm achieves path planning by selecting the optimal velocity combinations, it may fall into local optima and is highly dependent on the choice of evaluation function weights. To improve path smoothness, reduce robot impacts, and minimize energy consumption, the path length constraints, as shown in Eq. (6), are integrated into the evaluation function.

$$\begin{aligned} DIST(v, \omega) = & \sqrt{\left(y_{pred} - y_{start}\right)^2 + \left(x_{pred} - x_{start}\right)^2} \\ & - \sqrt{(y_{cur} - y_{start})^2 + (x_{cur} - x_{start})^2} \end{aligned} \tag{6}$$

where (x_{cur}, y_{cur}) denotes the current coordinates of the robot's position.

Introducing the $DIST(v, \omega)$ function accelerates the robot's search speed toward the target point, thereby shortening the path length. Therefore, by incorporating path length constraints, the evaluation function of the DWA algorithm can be enhanced as shown in Eq. (7).

$$\begin{aligned} G(v, \omega) = & \sigma(\alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) \\ & + \gamma \cdot velocity(v, \omega) + \delta \cdot DIST(v, \omega)) \end{aligned} \tag{7}$$

2.4. Obstacle-crossing motion planning for the robot

Path planning algorithms enable the automated operation of robots. However, when encountering convex obstacles that cannot be avoided along the path, the robot must cross them to continue its inspection tasks. Obstacles of different heights require different strategies for crossing actions. This paper categorizes obstacles into three height classes and develops corresponding crossing strategies, detailed in Table I.

***Table I.** Obstacle classification and obstacle-crossing strategies.*

Obstacle classification	**Obstacle-crossing strategies**
$h < h_{max}/2$	Strategy 1
$h_{max}/2 < h < h_{max}$	Strategy 2
$h > h_{max}$	Unable to pass through

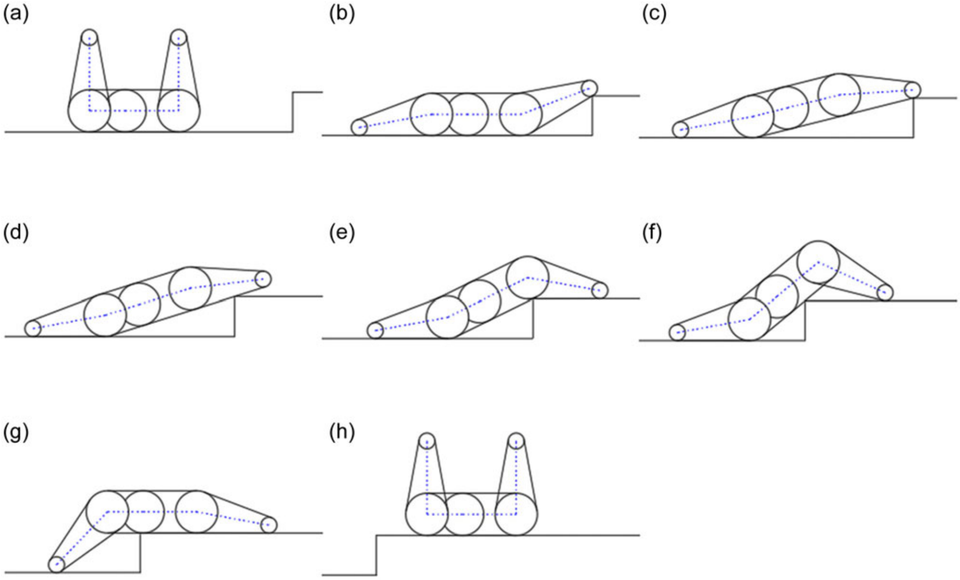


***Figure 5.** Obstacle-crossing motion planning for strategy 1.*

In Table I, h represents the height of obstacles, and h_{max} denotes the maximum obstacle height the robot can traverse. According to the analysis and testing documented in [5], the maximum obstacle height h_{max} that the robot designed in this study can traverse is 300 mm.

This paper is based on the principle of smooth obstacle-crossing, aiming to maintain stable variation of the robot's center of mass during the obstacle-crossing process.

When $h < h_{max}/2$, strategy 1 is employed for obstacle-crossing motion planning, with specific steps detailed in Fig. 5. The process unfolds as follows:

(a) The robot halts its forward movement at an appropriate distance from the vertical face of the obstacle.

(b) Lower the front flipper until its end rests on the edge of the convex surface, simultaneously pressing down the rear flipper to maintain ground contact.

(c) Further lower the front flipper to elevate the robot until the lower edge of the front flipper's track is level.

(d) Proceed by advancing until the robot's front wheels are positioned on the edge of the convex surface, then stop.

(e) Lower the front flipper to maintain contact with the ground, reducing the risk of the robot tipping as its center of gravity crosses the vertical face of the convex surface.

(f) Continue moving forward until the robot's center of gravity completely crosses the convex surface and then stop.

https://doi.org/10.1017/S0263574724001930 Published online by Cambridge University Press

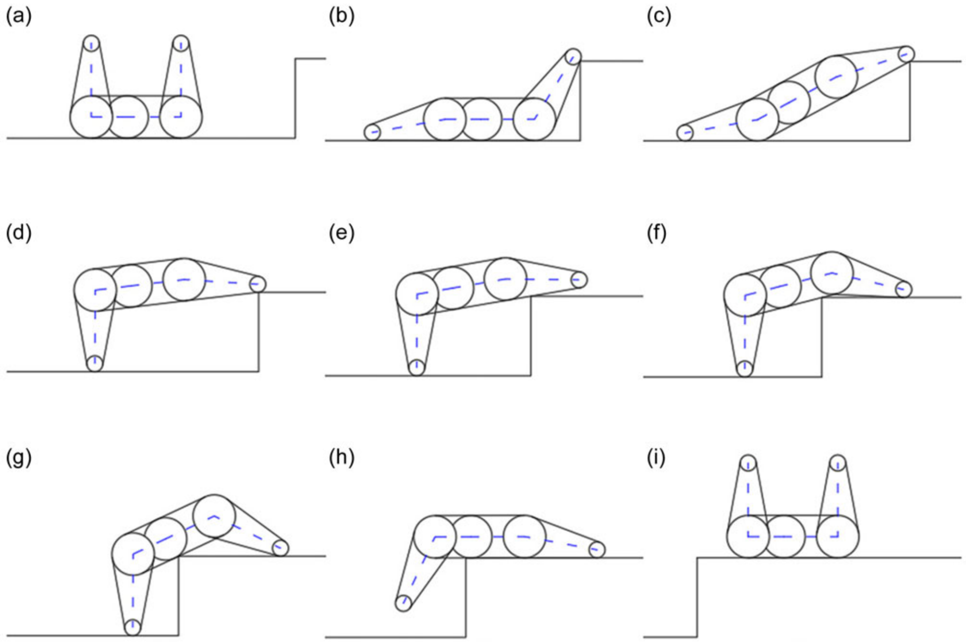


Figure 6. *Obstacle-crossing motion planning for strategy 2.*

(g) Raise the front flipper to level the robot, ensuring full ground contact with its tracks while pressing down the rear flipper.

(h) Finally, continue forward motion until no longer suspended, retracting both front and rear flippers to complete the obstacle-crossing action.

When $h_{max}/2 < h < h_{max}$, strategy 2 is employed for obstacle-crossing motion planning, with specific steps detailed in Fig. 6:

(a) The robot halts its forward movement at an appropriate distance from the vertical face of the convex surface.

(b) Lower the front flipper until its end rests on the edge of the convex surface, while simultaneously lowering the rear flipper to maintain ground contact.

(c) Continue lowering the front flipper to lift the robot until the lower edge of the front flipper's track is level with the robot.

(d) Lower the rear flipper to further lift the robot, ensuring it remains perpendicular to the ground, achieving a greater obstacle-crossing height.

(e) The robot then advances until its front wheels are on the edge of the convex obstacle, then stops.

(f) Lower the front flipper to maintain contact with the ground, minimizing the risk of the robot tipping as its center of gravity crosses the vertical face of the convex surface.

(g) Continue forward until the robot's center of gravity completely crosses the convex surface, then stop.

(h) Lift the front flipper to level the robot, ensuring full ground contact with its tracks.

(i) Finally, continue forward while retracting both front and rear flippers to complete the obstacle-crossing action.

https://doi.org/10.1017/S0263574724001930 Published online by Cambridge University Press

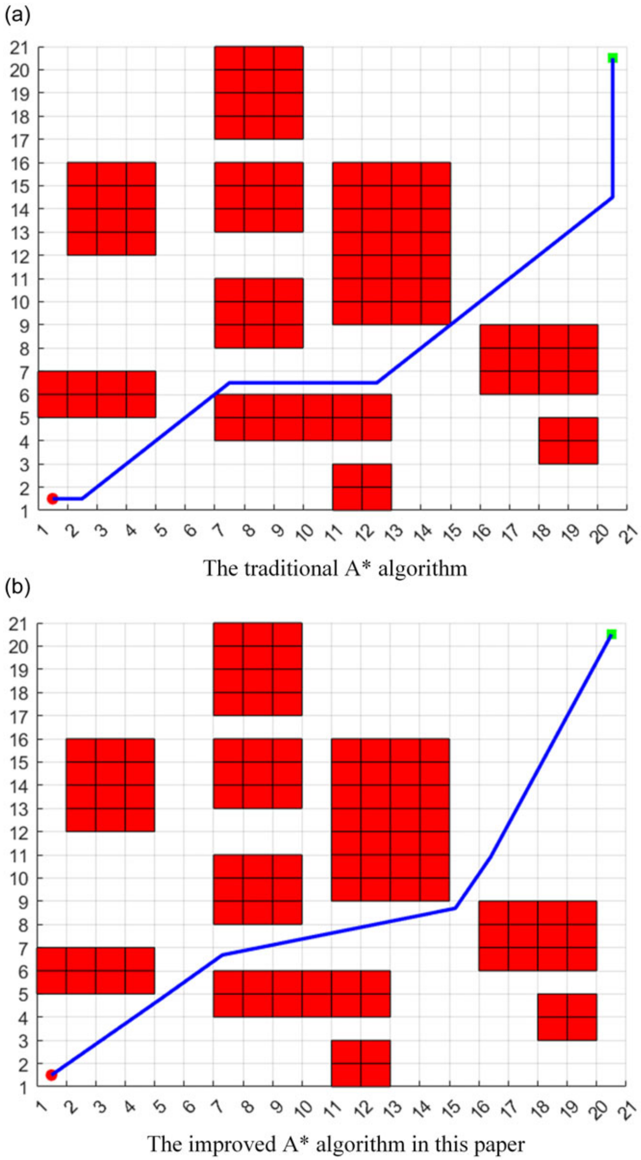


Figure 7. *Path planning using improved A* compared to traditional A*.*

3. Simulation and experimental testing

To evaluate the effectiveness of the improved robot path planning algorithm and obstacle-crossing motion planning strategy proposed in this paper, simulations of the path planning algorithm and experimental tests of the robot's obstacle-crossing maneuvers were conducted.

3.1. Simulation of global path planning based on A algorithm*

Initially, a 20 × 20 grid map is constructed, where red squares represent obstacles and white squares denote navigable areas for the robot. The complexity of the obstacle distribution on this map is designed to reflect the real working environment of a robot at the interface between cable trenches and substations, making the simulation results applicable to real-world scenarios. On this map, the improved A*

Table II. *Performance comparison between traditional A* algorithm and improved A* algorithm proposed in this paper.*

Comparison items	A* algorithm	The improved A* algorithm	Percentage reduction
The number of nodes	202	84	58.4%
The length of the path(*grids*)	30.3848	28.5241	6.1%
The number of turns	4	2	50%
Total degree of turns(°)	180	81.0578	55.0%

algorithm proposed in this paper is compared with the traditional A* algorithm for path planning. The simulation results are depicted in Fig. 7, where the red dot denotes the starting position, the green dot denotes the target position, and the solid blue line illustrates the globally planned path by the algorithm.

We compared the performance of the path planning algorithms across four aspects: the number of nodes, path length, number of turns, and total degree of turns. The results are presented in Table II.

Table II shows that the improved A* algorithm in this paper exhibits significant improvements over the traditional A* algorithm in terms of node count, path length, number of turns, and degree. Specifically, the node count has been reduced by 58%, the path length has been shortened by 6.1%, the number of turns decreased by 50%, and the total degree of turns reduced by 55%. To further validate the performance of the improved A* algorithm proposed in this study, simulations were conducted to compare its results with those of other algorithms such as Ant Colony Optimization (ACO), ACO combined with GA (ACO + GA), Sparrow Search Algorithm (SSA), and Improved Sparrow Search Algorithm (ISSA), as depicted in Fig. 8. Performance comparison of path planning algorithms is shown in Table III.

From Table III, it is evident that the improved A* algorithm presented in this study exhibits the shortest search time among the five tested algorithms, reducing the search duration by 97% compared to the ISSA algorithm, which had the shortest search time among the other methods. Additionally, the path length is reduced by 0.8% compared to ISSA. This demonstrates that the enhanced A* algorithm developed in this study possesses superior search efficiency. Although the search path length of the improved A* algorithm is approximately 1% longer than that of ACO + GA, its search time is significantly shorter. Therefore, the proposed improved A* algorithm demonstrates excellent performance and holds substantial practical value.

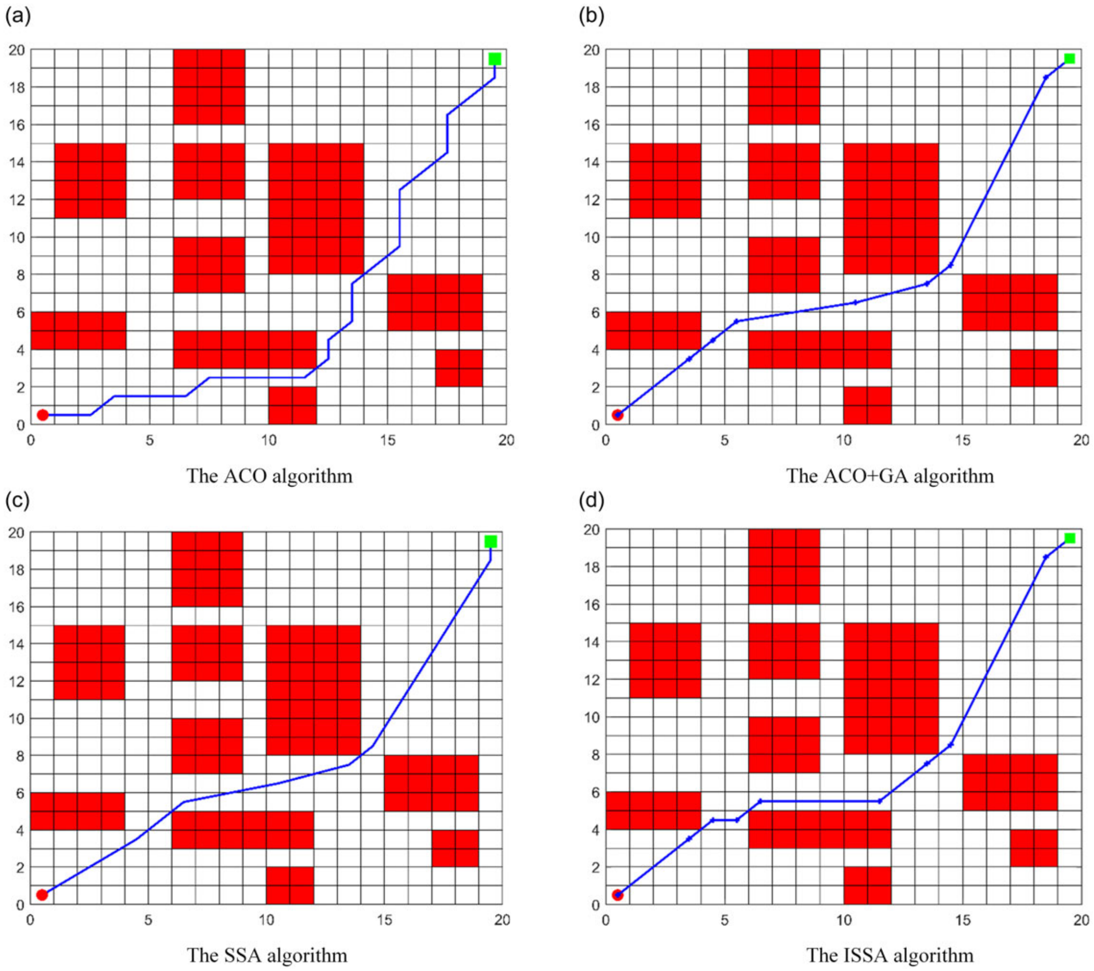
3.2. Simulation of integrated global and local path planning

To enhance the robot's navigation through cable trenches and its resilience to disturbances, we utilize a fusion algorithm that integrates the improved A* algorithm with enhanced DWA for path planning. The fusion algorithm is implemented by first employing the improved A* algorithm to discover an initial path. Subsequently, the nodes from this path serve as critical waypoints for real-time dynamic obstacle avoidance using the enhanced DWA algorithm. The improved DWA algorithm utilizes the robot's perceived surrounding information along with its kinematic and dynamic models to plan the robot's path within a local range, thereby enabling it to effectively avoid dynamic obstacles. Consequently, the simulation experiments conducted in this study are based on the assumption that the robot can accurately acquire environmental information. In the subsequent simulation experiments, the robot's maximum speed is set to 1 m/s, the maximum angular velocity is set to 0.23 rad/s, the acceleration is set to 0.1 m/s^2, the angular acceleration is set to 0.57 rad/s^2, the velocity resolution is set to 0.03 m/s, and the angular velocity resolution is set to 0.0175 rad/s.

To validate the effectiveness of the path planning approach based on the improved A* algorithm and enhanced DWA algorithm, we conducted fusion path planning simulations on three different grid maps: map1, a 20 × 20 grid with simple obstacles; map2, a 30 × 30 grid with complex obstacles; and map3, which features a different obstacle configuration from map2. In these maps, red squares represent

Table III. *Comparison of simulation results from multiple algorithms.*

Algorithm	**Search duration(s)**	**The length of the path($grids$)**
ACO	3.114	33.2174
ACO + GA	13.085	28.2372
SSA	1.062	29.1789
ISSA	0.904	28.7602
The improved A* algorithm	0.025	28.5241
Comparison of the improved A* and ISSA	Reduced by 97%	Reduced by 0.8%



The ACO algorithm

The ACO+GA algorithm

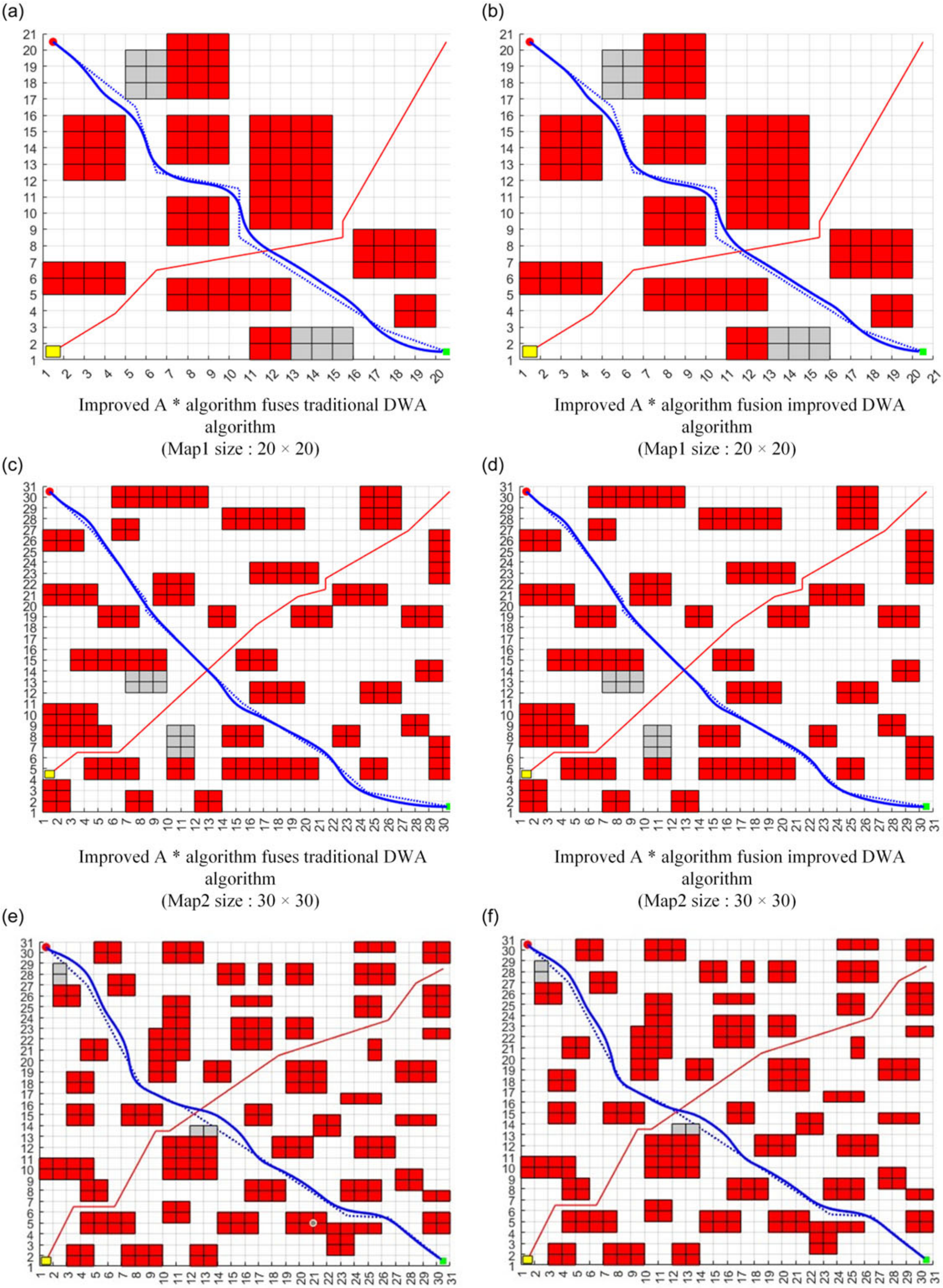
The SSA algorithm

The ISSA algorithm

Figure 8. *Path planning results of ACO, ACO + GA, SSA, and ISSA algorithms.*

known obstacles, white squares denote navigable areas, gray squares indicate unknown static obstacles, and yellow squares represent dynamic obstacles. The complexity of the obstacle distribution on these maps is slightly higher than that of the robot's actual working environment at the interface between cable trenches and substations, making the test results applicable to real-world robot scenarios. Each of the three maps was tested using both the improved A* algorithm combined with traditional DWA and the improved A* algorithm combined with the enhanced DWA. In the enhanced DWA, the coefficients for the heading evaluation function, obstacle distance evaluation function, and current velocity evaluation

https://doi.org/10.1017/S0263574724001930 Published online by Cambridge University Press



(a) Improved A * algorithm fuses traditional DWA algorithm (Map1 size : 20 × 20)

(b) Improved A * algorithm fusion improved DWA algorithm (Map1 size : 20 × 20)

(c) Improved A * algorithm fuses traditional DWA algorithm (Map2 size : 30 × 30)

(d) Improved A * algorithm fusion improved DWA algorithm (Map2 size : 30 × 30)

(e) Improved A * algorithm fuses traditional DWA algorithm (Map3 size : 30 × 30)

(f) Improved A * algorithm fusion improved DWA algorithm (Map3 size : 30 × 30)

Figure 9. *Integrated global and local path planning.*

Table IV. *Length comparison of integrated global and local path planning.*

Algorithm	**Map1 size :** ***20 × 20 (grids)***	**Map2 size :** ***30 × 30 (grids)***	**Map3 size :** ***30 × 30 (grids)***
Improved A* and traditional DWA	28.9792	44.1697	44.4365
Improved A* and improved DWA	28.0476	42.6296	42.8380
Percentage reduction in path length	3.2%	3.5%	3.6%

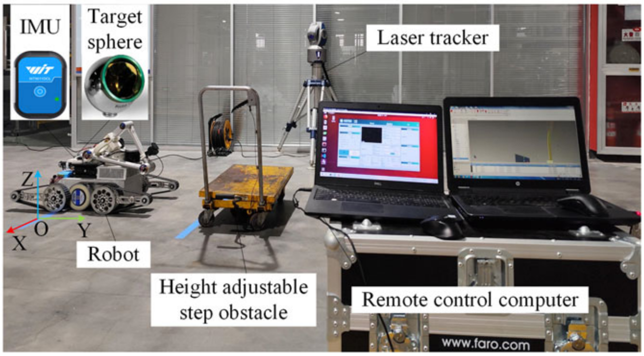


Figure 10. *Experimental validation of robot obstacle-crossing motion planning strategies.*

function were set to 0.25, 0.14, and 0.23, respectively, with the prediction time set to 3 s. Additionally, the minimum safe distance between the robot and obstacles was set to 0.65 m. The simulation results are shown in Fig. 9, where the globally planned path is represented by a blue dashed line, the robot's actual trajectory by a blue solid line, and the trajectories of obstacles by red solid lines. The fusion path planning algorithms were evaluated based on the length of the final planned path, with the results detailed in Table IV.

From the simulation results presented in Table IV, it is evident that our improved A* algorithm, when integrated with the enhanced DWA algorithm, reduces the robot's path length by 3.2% in the 20 × 20 obstacle map (map1) compared to integration with the traditional DWA algorithm. Additionally, in the 30 × 30 obstacle maps (map2 and map3) with different obstacle configurations, the path lengths are reduced by 3.5% and 3.6%, respectively. The simulation results across different map sizes and varying obstacle configurations demonstrate the superior global and local path planning capabilities of our improved A* algorithm when combined with the enhanced DWA algorithm.

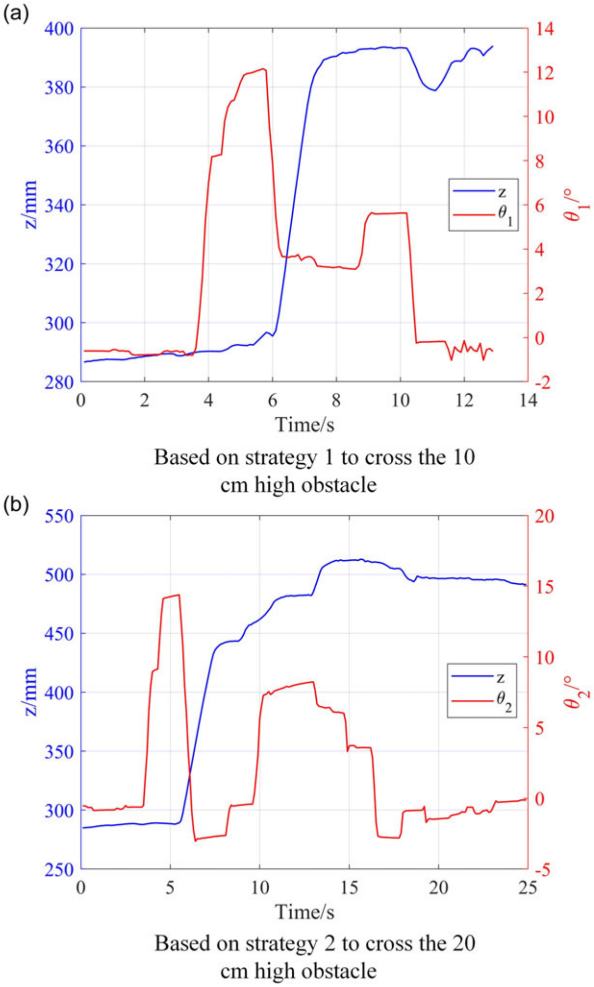
3.3. Experimental validation of robot obstacle-crossing motion planning strategies

To validate the obstacle-crossing motion planning strategies proposed in this paper, as depicted in Fig. 10, the robot was operated by trained personnel to test obstacle crossing over convex obstacles of heights 10 cm and 20 cm. Strategy 1 was employed for overcoming the 10 cm obstacles, while strategy 2 was applied for the 20 cm obstacles.

During the obstacle-crossing process, the robot's pitch angle θ_1 and the absolute displacement (x, y, z) of a target ball (placed above the robot's rear axle with the ground as the xy-plane) were real-time collected using IMU and a laser tracker. The relationship between the robot's pitch angle θ_1 and the z-direction displacement over time is illustrated in Fig. 11.

From Fig. 11, it can be observed that the robot successfully navigates 10 cm and 20 cm convex obstacles using strategy 1 and strategy 2, respectively. This demonstrates that the obstacle-crossing motion planning strategies designed in this study meet the robot's obstacle-crossing requirements, suitable for assisting in the automated inspection of cable trenches. Additionally, Fig. 11(b) reveals that during the

https://doi.org/10.1017/S0263574724001930 Published online by Cambridge University Press



Based on strategy 1 to cross the 10 cm high obstacle

Based on strategy 2 to cross the 20 cm high obstacle

Figure 11. *Experimental results of robot obstacle-crossing motion planning strategies.*

traversal of a 20 cm obstacle, despite efforts by the operator to maintain smooth obstacle-crossing, the robot's pitch angle tends to increase with the height of the convex obstacle.

4. Conclusion

To address the challenges of automated inspection for cable trench inspection robots, we have researched and designed an integrated global and local path planning algorithm, along with obstacle-crossing motion strategies. First, we enhanced the traditional A* algorithm to achieve global path planning. Compared to the traditional A* algorithm, the improved A* algorithm reduces the number of search nodes by 58% and the path length by 6.1% in complex obstacle environments, while also minimizing the number of turns and turning angles. Second, we integrated an improved DWA for dynamic obstacle avoidance. Simulation tests confirmed the superiority of this integrated algorithm. Results indicate that the improved A* algorithm combined with the enhanced DWA reduces path lengths by 3.2%, 3.5%, and 3.6% on the 20 × 20 grid map and two 30 × 30 grid maps with different obstacle configurations, respectively, compared to integration with the traditional DWA. This integrated algorithm significantly enhances the efficiency of robot navigation. Furthermore, to tackle the challenges posed by convex

https://doi.org/10.1017/S0263574724001930 Published online by Cambridge University Press

obstacles during real inspections, we proposed practical scenario-based obstacle-crossing motion planning strategies. Experimental tests on convex obstacles of 10 cm and 20 cm in height confirmed the robot's ability to successfully cross these obstacles, demonstrating the effectiveness and feasibility of the proposed strategies.

Future research will focus on optimizing the robot's path planning algorithm performance, particularly in environments with varying terrain and dynamic obstacles. The research outcomes are expected to be applicable to industrial inspection in complex environments, such as oil refineries, cable trenches, or urban infrastructure where obstacles and challenging terrain are common. Integrating the path planning algorithm and obstacle-crossing motion planning strategies into the robot's autonomous control system will enhance its adaptability and operational efficiency in these scenarios. Additionally, to facilitate the deployment of the algorithm on the robot, future work will incorporate advanced sensing technologies, such as LiDAR and 3D vision systems, to enable more accurate obstacle detection and path planning. Finally, further research will explore the application of these strategies in dynamic environments, particularly in scenarios where multiple obstacles may move unpredictably, thereby enhancing the robot's autonomy and flexibility.

Author contributions. Linjie Dong and Junfei Wang designed and improved the path planning algorithm. Renfei Zhang and Jie Li conducted simulation tests and data collection. Junfei Wang and Shanshan Wang performed simulation result analysis. Linjie Dong, Junfei Wang, and Renfei Zhang conducted experimental tests. Linjie Dong and Xingsong Wang wrote the article.

Financial support. This work is supported by the National Key R&D Program of China (grant no. 2021YFF0500900), the Natural Science Foundation of Jiangsu Province, China (grant no. BK202300367), the Natural Science Foundation of the Jiangsu Higher Education (grant no. 23KJB510016), and Science and Technology Project of the State Administration for Market Regulation, China (grant no. 2023MK040).

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. None.

References

[1] J. Xie, T. Sun, J. Zhang, L. Ye, M. Fan and M. Zhu, "Research on cable defect recognition technology based on image contour detection," **In:** *2021, 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, IEEE (2021) pp. 387–391.

[2] A. A. Khan, N. Malik, A. Al-Arainy and S. Alghuwainem, "A review of condition monitoring of underground power cables," **In:** *2012 IEEE International Conference on Condition Monitoring and Diagnosis*, IEEE (2012) pp. 909–912.

[3] L. Dong, N. Chen, J. Liang, T. Li, Z. Yan and B. Zhang, "A review of indoor-orbital electrical inspection robots in substations," *Ind. Robot: Int. J. Robot. Res. Appl.* **50**(2), 337–352 (2022).

[4] Z. Jia, Y. Tian, Z. Liu and S. Fan, "Condition assessment of the cable trench based on an intelligent inspection robot," *Front. Energy Res.* **10**, 860461 (2022). https://doi.org/10.3389/fenrg.2022.860461

[5] L. Dong, R. Zhang, X. Liu, J. Li, X. Wang and T. Mengqian, "High climbing and obstacle-crossing performance intelligent tracked inspection robot for cable trenches," *Ind. Robot: Int. J. Robot. Res. Appl.* (2024). https://doi.org/10.1108/IR-04-2024-0184

[6] A. S. Al-Araji, "Development of kinematic path-tracking controller design for real mobile robot via back-stepping slice genetic robust algorithm technique," *Arab. J. Sci. Eng.* **39**(12), 8825–8835 (2014).

[7] J. Ding, Y. Zhou, X. Huang, K. Song, S. Lu and L. Wang, "An improved rrt* algorithm for robot path planning based on path expansion heuristic sampling," *J. Comput. Sci.* **67**, 101937 (2023). https://doi.org/10.1016/j.jocs.2022.101937

[8] J. Li, S. X. Yang and Z. Xu, "A survey on robot path planning using bio-inspired algorithms," **In:** *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE (2019) pp. 2111–2116.

[9] A. Ismail, A. Sheta and M. Al-Weshah, "A mobile robot path planning using genetic algorithm in static environment," *J. Comput. Sci.* **4**(4), 341–344 (2008).

[10] Y. Tan, J. Ouyang, Z. Zhang, Y. Lao and P. Wen, "Path planning for spot welding robots based on improved ant colony algorithm," *Robotica* **41**(3), 926–938 (2023).

[11] C. Miao, G. Chen, C. Yan and Y. Wu, "Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm," *Comput. Ind. Eng.* **156**, 107230 (2021). https://doi.org/10.1016/j.cie.2021.107230

[12] M. Elhoseny, A. Tharwat and A. E. Hassanien, "Bezier curve based path planning in a dynamic field using modified genetic algorithm," *J. Comput. Sci.* **25**, 339–350 (2018). https://doi.org/10.1016/j.jocs.2017.08.004

[13] J. Qi, H. Yang and H. Sun, "Mod-rrt*: A sampling-based algorithm for robot path planning in dynamic environment," *IEEE T. Ind. Electron.* **68**(8), 7244–7251 (2020).

[14] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access* **2**, 56–77 (2014). https://doi.org/10.1109/access.2014.2302442

[15] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineer.* **96**, 59–69 (2014). https://doi.org/10.1016/j.proeng.2014.12.098

[16] B. Li, C. Dong, Q. Chen, Y. Mu, Z. Fan, Q. Wang and X. Chen, "Path planning of mobile robots based on an improved A* algorithm," **In:** *Proceedings of the 2020 4th High Performance Computing and Cluster Technologies Conference & 2020 3rd International Conference on Big Data and Artificial Intelligence*, (2020) pp. 49–53.

[17] O. A. Razzaq Abdul Wahhab and A. S. Al-Araji, "Path planning and control strategy design for mobile robot based on hybrid swarm optimization algorithm," *Int. J. Intell. Engineer. Sys.* **14**(3), 565–579 (2021).

[18] S. Liu, S. Liu and H. Xiao, "Improved gray wolf optimization algorithm integrating A* algorithm for path planning of mobile charging robots," *Robotica* **42**(2), 536–559 (2024).

[19] H. Wang, S. Lou, J. Jing, Y. Wang, W. Liu and T. Liu, "The EBS-A* algorithm: An improved A* algorithm for path planning," *PloS one* **17**(2), e0263841 (2022).

[20] C. Li, X. Huang, J. Ding, K. Song and S. Lu, "Global path planning based on a bidirectional alternating search A* algorithm for mobile robots," *Comput. Ind. Eng.* **168**, 108123 (2022). https://doi.org/10.1016/j.cie.2022.108123

[21] X. Tang, H. Zhou and T. Xu, "Obstacle avoidance path planning of 6-DOF robotic arm based on improved A* algorithm and artificial potential field method," *Robotica* **42**(2), 457–481 (2024).

[22] Y. Chen., W. Jiang., L. Yang. and Z. Luo., "Path planning based on motion constraints for mobile robot," *Comput. Integr. Manuf. Sys.* **29**(4), 1186 (2023).

[23] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.* **5**(1), 90–98 (1986).

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

[25] M. Kobayashi and N. Motoi, "Local path planning: Dynamic window approach with virtual manipulators considering dynamic obstacles," *IEEE Access* **10**, 17018–17029 (2022). https://doi.org/10.1109/ACCESS.2022.3150036

[26] Y. Sun, W. Wang, M. Xu, L. Huang, K. Shi, C. Zou and B. Chen, "Local path planning for mobile robots based on fuzzy dynamic window algorithm," *Sensors* **23**(19), 8260 (2023).

[27] M. Dobrevski and D. Skočaj, "Dynamic adaptive dynamic window approach," *IEEE T. Robot.* **40**, 3068–3081 (2024). https://doi.org/10.1109/TRO.2024.3400932

[28] L. Liu, X. Wang, X. Yang, H. Liu, J. Li and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Syst. Appl.* **227**, 120254 (2023). https://doi.org/10.1016/j.eswa.2023.120254

[29] C. Min, Y. Pan, W. Dai, I. Kawsar, Z. Li and G. Wang, "Trajectory optimization of an electric vehicle with minimum energy consumption using inverse dynamics model and servo constraints," *Mech. Mach. Theory* **181**, 105185 (2023). https://doi.org/10.1016/j.mechmachtheory.2022.105185

[30] L. Yang, L. Fu, P. Li, J. Mao and N. Guo, "An effective dynamic path planning approach for mobile robots based on ant colony fusion dynamic windows," *Machines* **10**(1), 50 (2022).

Cite this article: L. Dong, R. Zhang, J. Wang, J. Li, S. Wang and X. Wang, "Research on the path planning algorithm and obstacle-crossing motion planning strategy for a cable trench inspection robot", Robotica. https://doi.org/10.1017/S0263574724001930