Journal of Electronic Research and Application, 2025, Volume 9, Issue 1
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

Path Planning for Thermal Power Plant Fan Inspection Robot Based on Improved A* Algorithm

Wei Zhang*, Tingfeng Zhang

College of Electrical Engineering, Liaoning University of Technology, Jinzhou, 121001, China

Corresponding author: Wei Zhang, 1326001933@163.com

Abstract: To improve the efficiency and accuracy of path planning for fan inspection tasks in thermal power plants, this paper proposes an intelligent inspection robot path planning scheme based on an improved A* algorithm. The inspection robot utilizes multiple sensors to monitor key parameters of the fans, such as vibration, noise, and bearing temperature, and upload the data to the monitoring center. The robot's inspection path employs the improved A* algorithm, incorporating obstacle penalty terms, path reconstruction, and smoothing optimization techniques, thereby achieving optimal path planning for the inspection robot in complex environments. Simulation results demonstrate that the improved A* algorithm significantly outperforms the traditional A* algorithm in terms of total path distance, smoothness, and detour rate, effectively improving the execution efficiency of inspection tasks.

Keywords: Power plant fans; Inspection robot; Path planning; Improved A* algorithm

Online publication: February 17, 2025

1. Introduction

At present, thermal power plants require staff to inspect equipment at multiple points. This process is physically demanding and inefficient [1,2]. Intelligent inspection robots can replace manual inspections [3]. They collect and monitor equipment data in real-time, reducing labor and improving efficiency. Path planning is key to enabling autonomous robot inspections [4,5]. Commonly used algorithms include A* and Dijkstra. The A* algorithm uses heuristic search to quickly find optimal paths in finite graphs, making it widely studied and applied [6]. However, in complex environments, A* often gets stuck in local optima, leading to inefficient paths and excessive turns [7,8]. Researchers have proposed improvements, including optimizing heuristic functions, integrating local planning algorithms such as Dynamic Window Approach (DWA), and combining intelligent search methods like genetic or ant colony algorithms [9–15]. This paper enhances the A* algorithm with obstacle penalties, path reconstruction, and smoothing techniques to improve path planning efficiency.

2. Design scheme of an intelligent inspection robot for thermal power plants

This intelligent inspection robot is controlled by an STM32 microcontroller. It uses Mecanum wheel drive technology for flexible and efficient motion control. The system design includes multiple functional modules. These modules cover temperature detection, gas detection, navigation, video monitoring, and image processing. The main functional modules are shown in **Figure 1**.

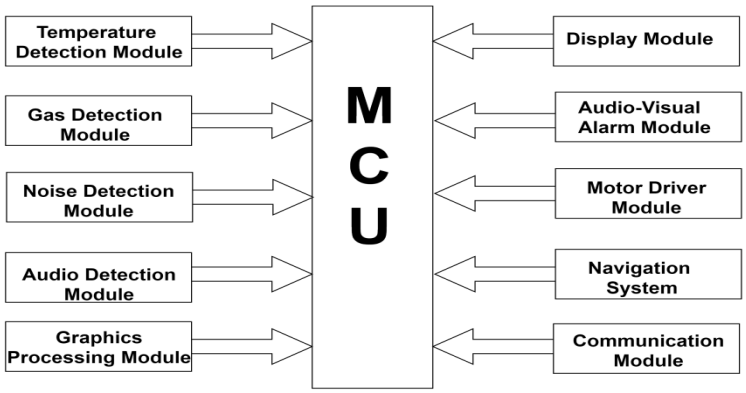

Figure 1. Overall system design

3. Research on path planning algorithm

3.1. Operating principle of traditional A* algorithm

The A* algorithm combines the advantages of Breadth-First Search and Greedy Best-First Search. It determines the search order by calculating the total cost *f(n)* for each node. This ensures the optimal path from the start to the goal can be found in the shortest time. The total cost *f(n)* is calculated using the following formula:

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

In the formula, *g(n)* represents the actual cost from the start node to the current node *n*. It is the total length of the path traveled so far. *h(n)* represents the estimated remaining cost from the current node *n* to the goal node. The heuristic function *h(n)* is expressed using the Manhattan distance as follows:

$$h(n) = |x_1 - x_2| + |y_1 - y_2| \tag{2}$$

In the formula, x_1 and x_2 represent the horizontal coordinates of two points in the plane, while y_1 and y_2 represent the vertical coordinates corresponding to x_1 and x_2 respectively.

3.2. Working principle of improved A* algorithm

To address the aforementioned issues, this paper introduces obstacle penalty terms and employs the Bresenham algorithm for path reconstruction based on the traditional A* algorithm. Additionally, gradient descent is utilized to achieve path smoothing.

3.2.1. Improvement and optimization of heuristic function

The improved heuristic function incorporates an obstacle penalty term on top of the traditional heuristic value *h(n)* to enhance the flexibility and obstacle avoidance capability of path planning. The specific formula is as follows:

$$H(n) = h(n) + \lambda \frac{1}{d_{obstacle} + \varepsilon} \tag{3}$$

In the formula, λ is the weight of the hazard penalty factor. $d_{obstacle}$ is the distance from the current node to the nearest obstacle. ε is the minimum positive number to prevent dividing zero errors. *h(n)* is an estimate of the residual cost from the current node *n* to the target node. *H(n)* is an improved heuristic function. To adapt to the complexity of different environments, the penalty factor λ is dynamically adjusted. The specific adjustment formula is as follows:

$$\lambda = K \times \frac{\text{number of obstacles}}{\text{total number of grids}} \tag{4}$$

In the formula, *K* is a scaling constant with a value range of 5 to 10. It is used to amplify or reduce the impact of the obstacle penalty. To adapt to complex environments in practical inspection tasks, the improved A* algorithm adopts the Euclidean distance to enhance the accuracy of path planning. The Euclidean distance is a standard method for calculating the straight-line distance between the current node and the goal node. Its formula is as follows:

$$h(n) = \sqrt{\left(x_{goal} - x_n\right)^2 + \left(y_{goal} - y_n\right)^2} \tag{5}$$

In the formula, $\left(x_{goal}, y_{goal}\right)$ is the coordinate of the target node. $\left(x_n, y_n\right)$ is the coordinates of the current node.

3.2.2. Path reconstruction and path optimization

After the A* algorithm completes the path search, the initial path often contains redundant nodes and sharp turns. This affects the simplicity of the path and the robot's movement efficiency. To address this, this paper introduces the Bresenham algorithm and gradient descent to optimize the path. These methods eliminate redundant nodes and smooth the path.

(1) The Bresenham algorithm is used to reconstruct the path and remove redundant nodes. The path reconstruction process is illustrated in **Figure 2**.

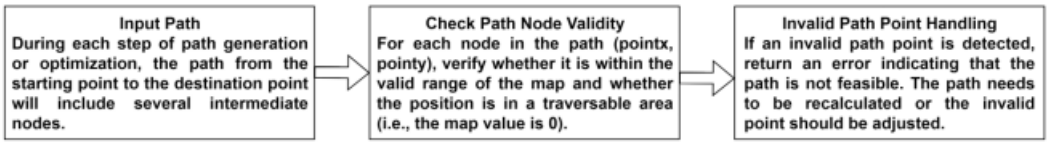

Figure 2. Path reconstruction process

(2) Gradient descent is used to smooth the path. It adjusts the positions of intermediate nodes to make the path as close to a straight line as possible. This reduces turns and improves path smoothness. The path-smoothing steps are as follows:

(i) Initial path input: The path optimized by the Bresenham algorithm is used as the input for gradient descent optimization.

(ii) Smoothing optimization: Intermediate nodes (excluding the start and end nodes) are selected. Their positions are adjusted using the following formula:

$$\text{current} = \text{prev} + \alpha \times (\text{next} - \text{prev}) \quad (6)$$

In the formula, *current* refers to the current point on the path, which is the point to be optimized, *prev* is the previous point of the current point, *next* is the next point of the current point, and α (alpha) is the learning rate, which controls the step size.

3.2.3. Path planning legality check

Ensure that every point on the path lies within the valid area. This avoids the path-crossing obstacles or exceeding the map boundaries. The validity of each path point is determined by the following formula:

$$0 < point_x \leq rows \quad (7)$$

$$0 < point_y \leq cols$$

$$matrix(point_x, point_y) = 0$$

In the formula, $point_x$ and $point_y$ are the coordinates of the current path point, *rows* and *cols* represent the number of rows and columns in the map matrix, defining the valid range of the map, $matrix(point_x, point_y)$ is the value of the corresponding point in the map.

3.3. Implementation process of the improved A* algorithm

This paper improves the traditional A* algorithm by introducing obstacle penalty terms to optimize the heuristic function, combining path reconstruction and optimization techniques, and implementing a dynamic update mechanism. These enhancements enable more efficient and precise path planning in complex environments. Below is the complete implementation process of the improved A* algorithm.

(1) Step 1: Initialize the start point. Define the open list and closed list. Set the cost of the start node to $g(n)$ = 0 and its parent node to null. Add the start node to the open list.
(2) Step 2: Search and expand. Select the node with the smallest $f(n)$ from the open list and move it to the closed list. If it is the goal node, proceed to path reconstruction. Otherwise, expand its neighboring nodes.
(3) Step 3: Process neighboring nodes. Skip nodes that are outside the map boundaries, obstacles, or in the closed list. Calculate the cost for the remaining nodes.
(4) Step 4: Path reconstruction and optimization. Trace back from the goal node to generate the initial path. Use the Bresenham algorithm to straighten the path and remove redundant points. Further optimize the path using gradient descent.
(5) Step 5: Legality check. Ensure all points on the path are within the map boundaries and not obstacles. If any point fails the check, recalculate the path.
(6) Step 6: Return the optimized path. If the open list is empty and the goal node is not found, report that the path is unreachable.

3.4. Simulation experiments and performance analysis of traditional A* algorithm vs improved A* algorithm

3.4.1. Establishment of the map model

In the autonomous navigation of inspection robots, grid maps are a commonly used method for environmental representation. They are typically generated by combining Light Detection and Ranging (LiDAR) sensor data with Simultaneous Localization and Mapping (SLAM) algorithms to create a 2D grid map. The resulting grid map of

the power plant fans is shown in **Figure 3**.

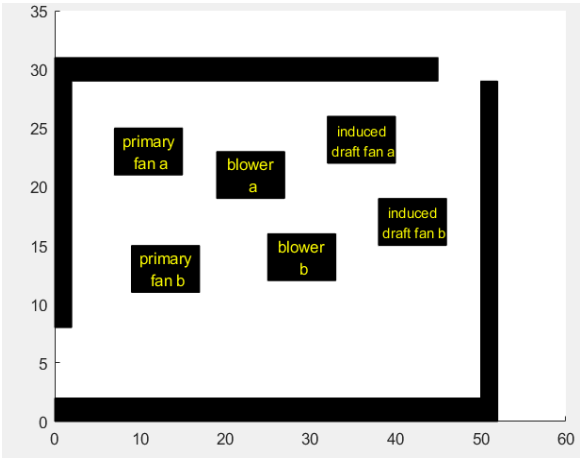

Figure 3. Grid map

3.4.2. Simulation comparison

The effectiveness of the improved A* algorithm for path planning is verified based on the generated 2D grid map. The path planning results of the traditional A* algorithm and the improved A* algorithm for the inspection robot are shown in **Figure 4**.

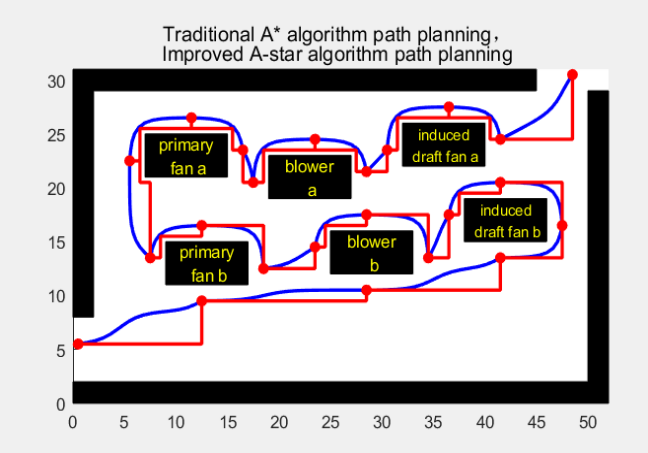

Figure 4. Path planning comparison between traditional A* algorithm and improved A* algorithm

The differences between the traditional A* algorithm and the improved A* algorithm in key metrics are statistically analyzed. These metrics include obstacle avoidance capability and path optimization performance. The results are shown in **Table 1**.

Table 1. Data comparison of path planning between traditional A* algorithm and improved A* algorithm

Metrics	Traditional A* algorithm	Improved A* algorithm
Total steps	211	211
Total distance (m)	200.3m	198.7m
Computation time (s)	0.1234s	0.1402s
Smoothness	2.95	1.85
Detour rate	1.25	1.05
Obstacle proximity	3.72	4.23
Directional angle change (°)	320.5	180.2

The experimental results show that the improved A* algorithm significantly outperforms the traditional A* algorithm. It achieves better performance in path smoothness, detour rate, and obstacle avoidance. This enhances the efficiency and safety of path planning for inspection robots in complex environments.

4. Conclusion

This paper proposes an intelligent inspection robot path planning scheme for fans based on an improved A* algorithm. The goal is to enhance the efficiency and accuracy of path planning for fan inspection tasks in thermal power plants. By introducing obstacle penalty terms, path reconstruction, and smoothing optimization techniques, the improved A* algorithm increases path flexibility, obstacle avoidance capability, and smoothness in complex environments. Simulation results show that the improved A* algorithm outperforms the traditional A* algorithm in total path distance, smoothness, and detour rate. It significantly improves path planning efficiency and safety. This research provides a theoretical foundation and technical support for the design of intelligent inspection systems. It also promotes the development of unmanned inspection systems.

Disclosure statement

The authors declare no conflict of interest.

References

[1] Zhou Y, Wang W, Li Z, et al., 2020, Application Research on Path Planning of Mobile Robots Based on A Algorithm. Computer Knowledge and Technology, 2020, 16(13): 1–3 + 10.

[2] Delobel L, Aufrere R, Debain C, et al., 2019, A Real-Time Map Refinement Method Using a Multi-Sensor Localization Framework. IEEE Transactions on Intelligent Transportation Systems, 20(5): 1644–1658.

[3] Lin H, Dan X, Jian O, et al., 2021, Review of Path Planning Algorithms for Mobile Robots. Computer Engineering and Applications, 57(18): 38–48.

[4] Ouyang M, Ma Y, 2020, Path Planning for Gravity Aided Navigation Based on Improved A* Algorithm. Chinese Journal of Geophysics, 63(12): 4361–4368.

[5] Lai R, Dou L, Wu Z, et al., 2024, Fusion of Improved A* and Dynamic Window Approach for Mobile Robot Path Planning. Journal of System Simulation, 36(08): 1884–1894.

[6] Chen X, Ren G, 2020, Key Technologies and Development Trends of Intelligent Manufacturing and Robot Application. IOP Conference Series: Earth and Environmental Science, 461(1): 1–4.

[7] Patle BK, Babu LG, Pandey A, et al., 2019, A Review: On Path Planning Strategies for Navigation of Mobile Robot. Defence Technology, 15(4): 582–606.

[8] Jin S, Kou Z, Wu J, 2022, Research on Path Planning and Tracking Algorithm for Coal Mine Fan Inspection Robot. Coal Science and Technology, 50(5): 253–262.

[9] Zhao J, Feng S, Sun T, et al., 2020, Functional Design and Application of Intelligent Robot Technology in Coal-Fired Smart Power Plants. Energy Technology, 2020(4): 35–42.

[10] Liu X, Li X, Wang J, 2018, Research on Mobile Robot Path Planning Based on Improved A* Algorithm. Computer Applications and Software, 35(10): 194–199.

[11] Cai X, Xu J, Zhao F, 2019, Research on Path Planning for Intelligent Robots Based on Improved A* Algorithm. Robotics, 41(6): 809–818.

[12] Wang X, Wang S, Wang X, 2017, Path Planning Optimization Based on Genetic Algorithm and A* Algorithm. Computer Engineering and Design, 38(7): 1679–1684.

[13] Han J, Zhang Y, Sun X, 2020, Path Planning Research Based on Improved Dijkstra Algorithm. Automation Technology and Application, 39(2): 102–106.

[14] Chen X, Zhu D, Tian F, 2021, A Robot Path Planning Method Based on Dynamic Weight A* Algorithm. Robotics Technology and Applications, 46(2): 40–47.

[15] Wang J, Zhang H, Li Y, 2021, Optimization of Path Planning Based on Genetic Algorithm and A* Algorithm. Computer Science and Exploration, 15(6): 1122–131.