IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 19, 2022, accepted May 25, 2022, date of publication May 30, 2022, date of current version June 6, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3179397

A Mobile Robot Path Planning Algorithm Based on Improved A* Algorithm and Dynamic Window Approach

YONGGANG LI[1], RENCAI JIN[2], XIANGRONG XU[1], (Member, IEEE), YUANDI QIAN[2,3], HAIYAN WANG[4], SHANSHAN XU[4], AND ZHIXIONG WANG[5]

[1]School of Mechanical Engineering, Anhui University of Technology, Maanshan 243032, China
[2]China MCC17 Group Company Ltd., Maanshan 243000, China
[3]School of Electrical and Information Engineering, Anhui University of Technology, Maanshan 243032, China
[4]School of Osaka Medical Engineering, Maanshan University, Maanshan 243032, China
[5]School of Medicine, Osaka University, Osaka 565-0871, Japan

Corresponding authors: Rencai Jin (1195154491@qq.com) and Xiangrong Xu (xuxr@ahut.edu.cn)

This work was supported in part by the Anhui Provincial Natural Science Foundation under Grant 2108085MF225, in part by the Open Project of China International Science and Technology Cooperation Base on Intelligent Equipment Manufacturing in Special Service Environment under Grant ISTC2021KF07, and in part by the China National Key Research and Development under Project 2017YFE0113200.

ABSTRACT The traditional A* algorithm has several problems in practical applications, such as many path turning points, redundant nodes, and long running time. it is sometimes impossible to plan the theoretical optimal route. To solve the above problem, this paper presents an optimized A* algorithm, the adaptive adjustment step algorithm and the three-time Bezier curve are used to solve the problems of many turning points, large turning angles, and long running time in the search path. Moreover, aiming at the path planning problem of mobile robots facing dynamic obstacle interference in complex environments, an algorithm that integrates the improved A* algorithm with the dynamic window method is proposed, which not only solves the shortcomings of the A* algorithm in which the dynamic obstacles cannot be avoided, but also prevents the mobile robot from falling into local optimization. The results show that the fusion algorithm of the improved A* algorithm and the dynamic window method with the traditional A* algorithm reduces the number of turns by 50% and the path length by 3.62% compared with the original algorithm. In the same environment, compared with the traditional algorithm, the hybrid algorithm in this paper reduces the average time consumption by 10.27%, the number of path inflection points by 57.14%, and the accuracy is higher than 33.33%, which is more effective in complex dynamic environments.

INDEX TERMS Path planning, hybrid algorithms, improved A* algorithm, improved DWA.

I. INTRODUCTION

In recent years, mobile robots have received widespread attention from all over the world, and because of their autonomous and flexible characteristics, they are widely used in many important fields such as national defense science and technology, industrial manufacturing, life services, medical and health. With the continuous popularization of mobile robots, path planning problems have become the primary problems that need to be solved urgently, and the driving efficiency of robots and whether the travel route is optimal will seriously affect the walking of robots [1]–[3].

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

According to the different working environments where path planning is applicable, there are global static path planning and local dynamic path planning, of which global static path planning is only suitable for solving the path planning problem of moving robots in the static environment known for the obstacles in the surrounding environment, and the common methods are the Dijkstra algorithm, fast random tree search algorithm, A* algorithm [4], [5], etc.; local dynamic path planning solves the path planning problem of mobile robots known in the environment within a range. The dynamic window approach, the ant colony algorithm, and other methods are common. Among them, Dijkstra is a breadth-first search algorithm. The search mode is relatively simple, although it can achieve global path planning; in the

case of a more complex environment, the algorithm calculates more nodes, occupies more memory, the search is slow and inefficient, and it is difficult to plan a smooth and safe optimal path in a short period of time [6]. The fast random tree search algorithm is a sample-based search algorithm that has fast search speed and strong ability and occupies an important position in high-dimensional environments, but the search accuracy of the algorithm is low, the path smoothness is poor, and it is difficult to plan the optimal path [7]. The standard A* algorithm is based on the Dijkstra algorithm to introduce heuristic functions. Through the evaluation of the node generation value, the optimal path is finally planned. Because of its fast calculation speed, path optimization and other advantages, it is widely used in global path planning.

A vast number of experts have researched the classic A* algorithm because of its flaws, such as too many inflection points and node redundancy [8]–[10]. The heuristic function of the standard A* algorithm is improved by using the Manhattan distance-Euclidean distance hybrid method, which improves the search efficiency and saves search time [11]. The two-way algorithm is used to search in both positive and negative directions at the same time to improve the search efficiency [12]. The dynamic window method is a local dynamic path planning algorithm. The path is relatively smooth, but it is easy to fall into the local optimal, and it is impossible to reach the target position according to the global optimal path [13]–[15]. The ant colony algorithm is robust and easy to combine with other algorithms, but converges slowly and takes longer search times [16]–[18].

In the face of complex and dynamic environments, relying only on the A* algorithm is not enough. The task cannot be completed with a single global or local path planning. Considering the advantages and disadvantages of these algorithms, global path planning and local path planning are combined. The literature [19]–[22] proposes that a hybrid algorithm combining the A* algorithm and the artificial potential field method realizes the path planning problem in the dynamic environment, but the artificial potential field method cannot better plan the local optimal path, which reduces the overall efficiency [23], [24]. Farhad Bayat deals with the mobile robot path planning problem in the presence of scattered obstacles in a visually known environment. So it is practical and can be applied to static and dynamic environments [25].

None of the above algorithms can solve the problems of traditional A* algorithm and dynamic window method programming path inflection, low smoothness, and easy fall into local optimization. Therefore, this paper proposes an improved A* and dynamic window approach method fusion algorithm, using the improved A* algorithm to plan the global path, and then combining it with the improved dynamic window method to complete the local path planning, to achieve real-time dynamic obstacle avoidance, and finally plan the safe trajectory with optimal path and high smoothness.

II. ENVIRONMENT MODEL DESCRIPTION

A. RASTER MODELING

Create an environment model in route planning using a grid method that divides the environment space into equal, continuous, disjoint grids of a defined granularity. According to the actual environmental information in the route planning, the grids are set as free and occupied, where the free grid is represented by white and the occupied grid is represented by black. The coordinate origin is chosen in the lower left corner of the two-dimensional plane Cartesian coordinate system; the horizontal axis of the grid is represented by the x-axis, and the values are incremented sequentially from left to right; the vertical axis is represented by the y-axis, the values are incremented sequentially from bottom to top, and the specific location of each raster in the raster map is represented by $p(x_i, y_i), (i, j = 1, 2, 3, \cdots\cdots, n)$.

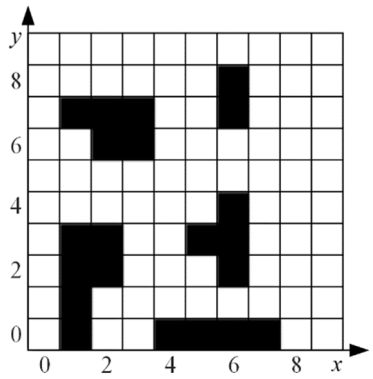


FIGURE 1. Environment model raster.

B. THE RASTER GRAIN SIZE IS DETERMINED

The basic element of the grid method is that the grid granularity is the smallest. If the grid particles are too small, the path search process will be more difficult, consume a lot of computing resources and time, and will not achieve the expected goal; if the grid particles are too large, the environment model will be different from the real environment. The path search algorithm will be unable to avoid obstacles or even finish the desired path planning if it is too vast. Therefore, grid granularity is extremely important for environmental model establishment and path planning.

III. A* ALGORITHM IMPROVEMENTS

A. TRADITIONAL A* ALGORITHM

The A* algorithm is a heuristic path exploration algorithm that enables global path planning, inherits the principles of the classic Dijkstra algorithm and the BFS algorithm, and improves the shortcomings of slower search speeds. The A* algorithm sets the evaluation function, searches around from the starting point, selects the node with the smallest total generation value as the next extension node, and stops the search until the end point, completing the search for the optimal path. The cost function is

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

where: n represents the current node; $f(n)$ is the cost function of the current node; $g(n)$ is the actual generation value of the mobile robot from the current node to the target node n; $h(n)$ is the estimated generation value that will be consumed from the current node to the target node. Common methods of calculating generation value are Manhattan distance, Euclidean distance, and Chebyshev distance. This article selects Euclidean distance as the $h(n)$ cost function, and its calculation formula is expressed as

$$h(n) = [(x_n - x_m)^2 + (y_n - y_m)^2]^{\frac{1}{2}} \tag{2}$$

where: (x_n, y_n) represents the current path node coordinates, (x_m, y_m) represents the target node coordinates.

The following is a simple proof of the convergence optimal point of the A* algorithm.

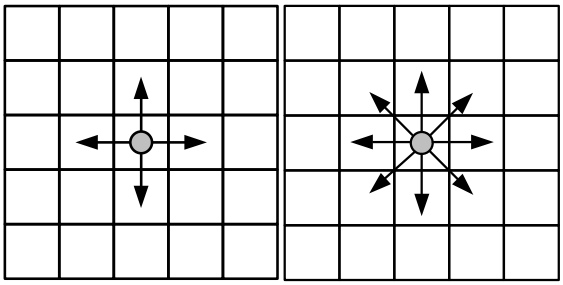
Assumption: The secondary advantage G_2 is a node generated in the open table, n is a node (it is the node with the closest distance to the optimal point G).

Proof:

$f(G_2) = g(G_2)$ because $h(G_2) = 0$;
$g(G_2) > g(G)$ because G_2 is the secondary advantage;
$f(G) = g(G)$ because $h(G) = 0$;
$f(G_2) > f(G)$ from the above;
$h(n) \leq h^*(n)$ basic requirement;

$$g(n) + h(n) \leq g(n) + h^*(n)$$
$$f(n) \leq f(G)$$

Remark: If there is an optimal point, then the A* algorithm will always find the optimal point first.



a) Four-neighborhood search b) Eight neighborhood searches

FIGURE 2. Environment model raster.

The traditional A* algorithm mainly searches for four domains, as shown in Figure 2a, and Eight Neighborhood Search as shown in Figure 2b, and the search neighborhood indicates the direction in which the robot can move. In Figure 2, the gray dot represents the current node position of the robot, and the solid arrow represents the robot's search direction. Use four neighborhoods to search for $\pi/2$ corners per turn and eight neighborhoods to search for $\pi/4$ corners per turn. When there are more search neighborhoods, the direction of the search becomes more important, and the overall length of the planned path is smaller, but the search is less efficient.

The flow chart of the A* algorithm in the actual path search process is shown in Figure 3. Traditional A* algorithms are confined to finding a single optimal path from beginning to end by first splitting the surrounding search space into measurable nodes. The A* algorithm creates two lists when executed; the open list and the closed list. Unexpanded nodes are placed in the open list, and expanded nodes are stored in the closed list. When adding a node to the Open list, it is added directly to the end, regardless of the value. On each expansion, the sizes of all the nodes in the Open list are compared; the node with the smallest value is obtained; the node with the smallest value is removed from it and added to the Close list. If these nodes are not in the Open list, add them all to the Open list and choose the smallest node as the current node, in which case continue searching for the remaining nodes. If these extended nodes are in the Open list, use the current node as the parent node, use the cost function to evaluate, recalculate the value, loop the above steps until the target is found, and finally arrange the nodes in the Open list in reverse order to get the optimal path.

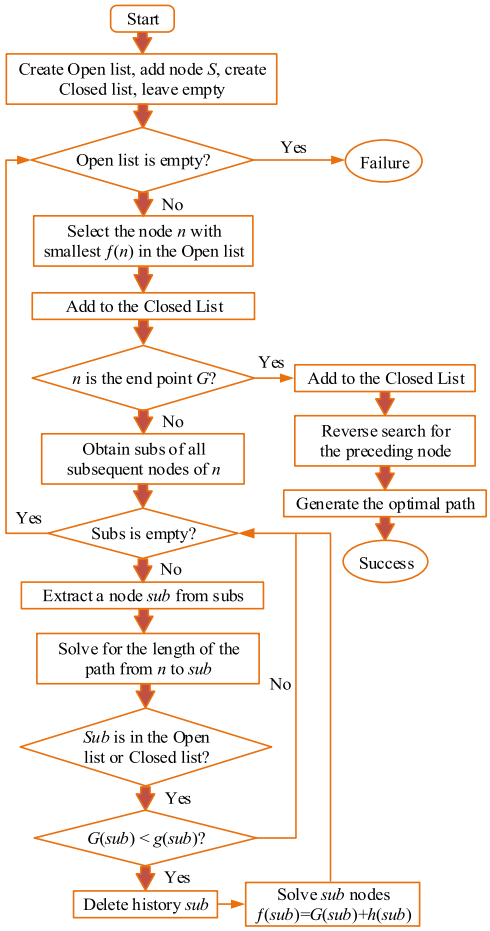


FIGURE 3. A* algorithm flowchart.

B. ADAPTIVE ADJUSTMENT STEP SIZE ALGORITHM

In the A* algorithm, the step length is one of the important parameters affecting the mobile robot, and the fixed step

size makes the mobile robot have defects such as low safety performance, poor obstacle avoidance effect and insufficient flexibility. Therefore, this paper proposes an adaptive adjustment step algorithm, when there are more obstacles in the surrounding environment, reducing the step size increases the number of nodes per search, and the search path is safer and more detailed; when there are fewer obstacles in the surrounding environment, increasing the step size speeds up and improves the efficiency of search. According to the distribution of obstacles, the step size is automatically adjusted to enhance the flexibility of the robot.

The distribution of obstacles in the surrounding environment is split into two groups when considering the elements impacting step size: whether they are dynamic barriers or not, and the quantity and position distribution of obstacles within a specific range.

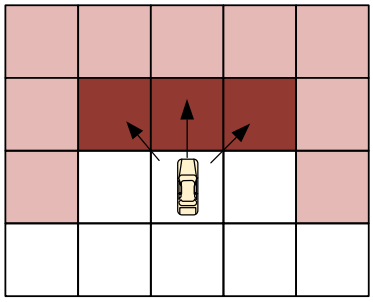


FIGURE 4. Robot movement direction and obstacle threat weight diagram.

In Figure 4, the robot uses eight-neighborhood search, and the trolley model replaces the mobile robot, which has eight directions of motion. The number of static obstacles in the dark red area in the figure is x_1, the number of static obstacles in the light red area is x_2, the number of dynamic obstacles in the direction of motion is d, and the closer to the trolley model, the greater the threat of obstacles in the area to the mobile robot, so the threat function $f(x_1, x_2)$ is defined as:

$$f(x_1, x_2) = \begin{cases} \dfrac{1}{k_1 x_1 + k_2 x_2 + c} & d = 0 \\ 1 & d \neq 0 \end{cases} \tag{3}$$

where: k_1, k_2 represents the threat factor of a static obstacle, c represents a self-adjusting constant, $k_1 \in (1, 2)$, $k_2 \in (0.5, 1)$, $c \in (0, 1)$. Then the adaptive adjustment step is

$$l = \begin{cases} f(x_1, x_2) \cdot l_{\max} & d = 0 \\ f(x_1, x_2) \cdot l_{\min} & d \neq 0 \end{cases} \tag{4}$$

where: $l_{\min} \leq l \leq l_{\max}$, $l_{\min} = 0.1m$, $l_{\max} = 0.2m$.

Figures 5 and 6 show the path planning results using the traditional A* algorithm and the adaptive adjustment step algorithm respectively. Table 1 shows the differences in elapsed time, number of nodes, elapsed time reduction rate, and node reduction rate before and after the method was improved.

We can see from these two graphs that the enhanced A* algorithm significantly decreases the number of search nodes. The path before and after the algorithm improvement is studied using the running time, number of nodes, and running time reduction rate as performance indicators to further validate the effect of the adaptive adjustment step length method.

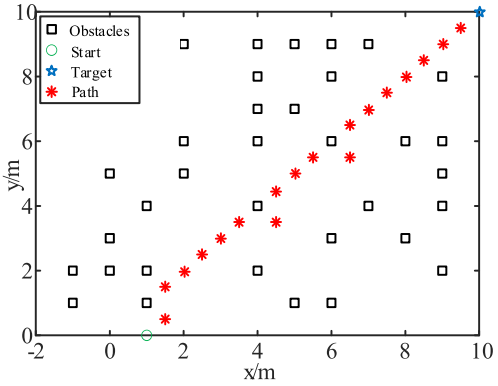


FIGURE 5. Path planning results of the traditional A* algorithm.

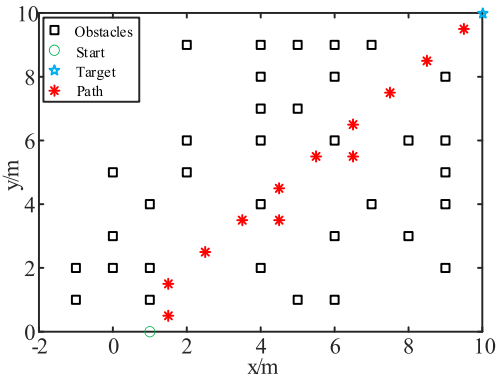


FIGURE 6. Path planning results of adaptive adjustment step algorithm.

TABLE 1. Comparison of performance indicators before and after algorithm optimization.

Parameter names	The elapsed time	Number of nodes	Elapsed time reduction rate	Node reduction rate
Traditional A* algorithm	0.729s	18	—	—
Adaptive regulation step algorithm	0.632s	14	13.31%	33.3%

The performance indices of the traditional A* algorithm and the adaptive step-size adjustment technique are compared in Table 1. The improved algorithm decreases the number of nodes by 33.3% and the running time by 13.31%, resulting in better operational efficiency.

C. ARC OPTIMIZATION

The traditional A* algorithm has many path inflection points, which makes it difficult for the robot to walk, and also poses a huge challenge to the load of the motor. In order to satisfy the nonholonomic constraints of mobile robots, it is necessary to smooth the motion trajectory. The trajectory smoothing process can reduce the frequency and amplitude of motor

start and stop, and thus increase the service life and safety of the robot. Therefore, this paper uses the cubic Bezier curve to optimize the trajectory and compare it with the original trajectory curve.

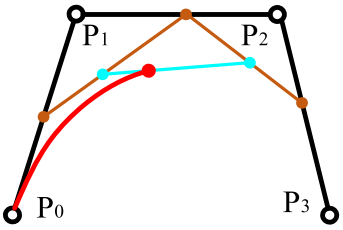


FIGURE 7. Cubic Bezier curves.

The Bezier curve is mainly applied to the smooth processing of two-dimensional plane line segments, as shown in Figure 7. The figure is composed of $P_i(i = 0, 1, 2, 3)$ four nodes and connecting line segments between them, and P_0 is the starting point, P_3 is the end point, and P_i is the control point. Taking $B(t)$ to represent the coordinates at time $t \in [0, 1]$, the cubic Bezier curve formula is:

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3 \tag{5}$$

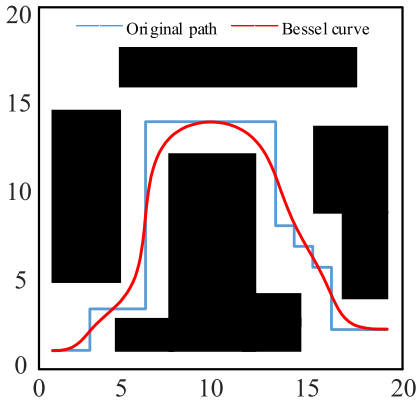


FIGURE 8. Bezier curve path optimization path diagram.

Figure 8 shows the Bezier curve smoothing the entire path, and the peaks at the corners are optimized to ensure the robot travels smoothly during the operation. The total steering angle and path length are significantly lower than the traditional curve, reducing the loss of the motor and avoiding the unbalance of the robot itself. and satisfy the motion constraints of the mobile robot.

IV. IMPROVED DYNAMIC WINDOW APPROACH METHOD

At present, most robots perceive the surrounding environment based on multi-sensor fusion technology, such as depth camera and laser radar, and then use local path planning algorithms to complete tasks such as avoiding obstacles and chasing dynamic targets according to the obtained information. The traditional dynamic window algorithm lacks the guidance of global path planning, and can only plan the paths of obstacles in the environment in real time. However, in a multi-obstacle environment, the robot will fall into narrow channel oscillation due to the lack of guidance from global planning, resulting in a larger global path and being unable to quickly plan the optimal path.

Dynamic Window Approach (DWA) is a velocity-based local planner that transforms the path planning problem into a constrained optimization problem in velocity vector space. The purpose of the dynamic window method is to sample multiple sets of data in a two-dimensional space and simulate the trajectory of the robot at this speed. To finish the local route planning, the ideal trajectory speed is chosen using the designed evaluation function.

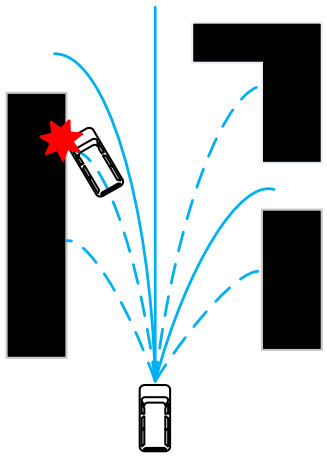


FIGURE 9. Schematic diagram of DWA method.

DWA is shown in Figure 9, model robot, obstacles in grey rectangle represent environment, each curve is forecast to get multiple sets of line trajectory, dotted line means the robot's trajectory and obstacle collision will occur, so the path is not the optimal trajectory, selection of the optimal trajectory only requires evaluating the rest of the track, and finally, the evaluation function's optimal safety trajectory is selected.

A. KINEMATIC MODEL

In order to avoid obstacles in real time, the velocity of the robot must be sampled in space to simulate its trajectory. Generally speaking, the motion state of the robot is measured by the linear velocity and angular velocity. Suppose the velocity of the robot per unit time is (v_t, ω_t), and then select the optimal trajectory from all the trajectory through the evaluation function. Within a unit time Δt interval, the arc-shaped trajectory can be regarded as a linear motion, and the kinematic model is:

$$\begin{cases} x = x + v_x \Delta t \cos\theta_t - v_y \Delta t \sin\theta_t \\ y = y + v_x \Delta t \sin\theta_t - v_y \Delta t \cos\theta_t \\ \theta_t = \theta_t + \omega_t \Delta t \end{cases} \tag{6}$$

Define the pose $\mathbf{q} = [x, y, \theta, \phi]^T$ of the robot in the environment, input its own linear velocity and angular velocity $\mathbf{u} = [v, w]^T$. The schematic diagram and parameters of the model are shown in Figure 10 and Table 2.

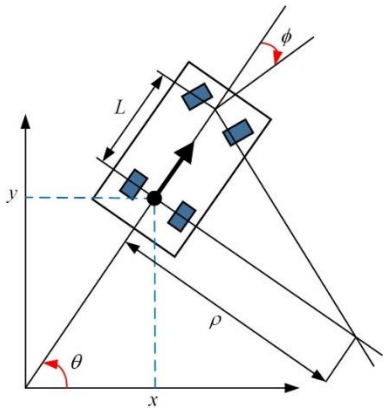


FIGURE 10. Mobile robot kinematics model.

TABLE 2. Parameters and meanings.

Parameter	Meaning
v	Line speed
ω	Front wheel angular velocity
$\mathbf{u}$	Input (including v, ω)
$\mathbf{q}$	Pose
θ	Angle between the longitudinal and x axis
ϕ	Front wheel steering angle
L	Body length

According to the geometric relationship, the angular velocity can be obtained $\dot{\theta}$:

$$\dot{\theta} = \frac{\tan\phi}{L} v \tag{7}$$

so

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ \dfrac{\tan\phi}{L} & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u} \tag{8}$$

In more detail:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ \dfrac{\tan\phi}{L} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{9}$$

B. SPEED SAMPLING

There are infinite groups (v, ω) of robots in the velocity space, and the range of sampling velocity is constrained according to the actual situation.

The speed constraint of the robot under the influence of motor performance is:

$$v_l = \{(v, \omega) | v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \tag{10}$$

The speed range that can be achieved under the acceleration of the robot under the limitation of the motor driving force is:

$$v_d = \{(v, \omega) | v \in [v_c - a_d \Delta t, v_c + a_d \Delta t], \omega \in [\omega_c - \alpha_d \Delta t, \omega_c + \alpha_d \Delta t]\} \tag{11}$$

where: v_c, ω_c indicates the current linear velocity and angular velocity of the robot; a_a, a_d indicates the upper and lower limits of linear acceleration of the robot; α_a, α_d indicates the upper and lower limits of angular acceleration of the robot.

When performing local path planning, the robot must maintain a safe distance to protect its own safety. As a result, the robot must come to a halt before colliding with the obstruction; when the speed is decreased to zero, the speed space is:

$$V_0 = \{(v, \omega) | v \le [2dist(v, \omega) a_d]^{1/2}, \omega \le [2dist(v, \omega)\alpha_d]^{1/2}\} \tag{12}$$

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

C. EVALUATION FUNCTION

In the local path planning of the robot, there are several sampling velocities available in the velocity space, so it is necessary to design an evaluation function to select the optimal trajectory. The parameters considered are azimuth, velocity and distance respectively. The designed evaluation function is:

$$G(v, \omega) = \sigma[\alpha \cdot head(v, \omega) + \beta \cdot stob(v, \omega) + \delta \cdot dyob(v, \omega) + \gamma \cdot velo(v, \omega)] \tag{13}$$

where, $head(v, \omega)$ represents the azimuth evaluation function of the robot, and represents the angular deviation between the end direction of the current simulated trajectory and the global path; $stob(v, \omega)$ represents the vertical distance between the current simulated trajectory and the static obstacle; $dyob(v, \omega)$ represents the vertical distance between the current simulated trajectory and the dynamic obstacle; Evaluation function $velo(v, \omega)$ representing the current simulation speed; σ is smoothing coefficient, and $\alpha, \beta, \delta, \gamma$ are four-term weighting coefficients. Finally, the trajectory with the $G(v, \omega)$ smallest value is taken as the optimal trajectory.

In order to meet the requirements of trajectory smoothness, $head(v, \omega)$, $stob(v, \omega)$, $dyob(v, \omega)$, $velo(v, \omega)$ needs to be normalized and then added. That is, each item is divided by the sum of each item:

$$normal\_head(i) = \frac{head(i)}{\sum_{i=1}^{n} head(i)} \tag{14}$$

$$normal\_stob(i) = \frac{stob(i)}{\sum_{i=1}^{n} stob(i)} \tag{15}$$

$$normal\_dyob(i) = \frac{dyob(i)}{\sum_{i=1}^{n} dyob(i)} \quad (16)$$

$$normal\_velo(i) = \frac{velo(i)}{\sum_{i=1}^{n} velo(i)} \quad (17)$$

where, n is all the sampled trajectory points, i is the current trajectory point to be evaluated. The four weights of the objective function $G(v, \omega)$ are all necessary and finally normalized. By continuously adjusting the weight coefficient and maximizing the objective function $G(v, \omega)$, the robot can avoid obstacles at the fastest speed under the constraints, and at the same time move towards the target position. Although the obstacle avoidance performance of the DWA algorithm depends on the weighting parameter $\alpha, \beta, \delta, \gamma$, the algorithm is still stable even if the value of the weighting parameter changes slightly. We found that $\alpha, \beta, \delta, \gamma$ values of 0.7, 0.7, 0.1 and 0.1 worked well. A higher weight of the target heading parameter makes the robot very close to the obstacle. Choose appropriate parameters according to the environment. In limited barrier situations, a greater target heading weight is preferable, while in a large environment, a lower target heading weight may be preferable.

D. DWA SIMULATION VERIFICATION

The robot will encounter different types of obstacles in the path search, namely static obstacles and dynamic obstacles. In order to verify the robot path planning in response to the effectiveness of the dynamic obstacles, the simulation in MATLAB R2020a validation, grids are built environment setting, starting point and goal, respectively, in the presence of dynamic obstacles in both cases the influence of simulation, and compare the results, verify the algorithm on the merit of trying to avoid dynamic obstacles.

Figure 11 shows the algorithm simulation diagram of DWA. The starting point is (1,1) and the ending point is (9,9). The robot avoids all static obstacles and reaches the goal location to finish the simulation of the static environment when the environment is packed with static obstacles. When a dynamic obstacle appears in the environment and is located on the originally planned path, the robot will change its original trajectory. The robot successfully avoids the dynamic impediment and reaches the target spot, as indicated by the solid blue line in the picture.

When carrying out local path planning, DWA can complete the avoidance of dynamic and static barriers, but it is simple to fall into local optimum and fail to reach the goal location. As shown in Figure 12, when the starting point is set as (1,0) and the end point is set as (10,10), it is easy to choose to go around from the left when using the DWA algorithm. At this time, it just falls into the local minimum point, resulting in path planning failure and failure to reach the target point. This is because when the robot is carrying out local path planning, it only deals with the information of surrounding obstacles each time and lacks the concept of global path planning. As a result, it reaches a dead end of impediments, resulting in path planning failure and failure to reach the target site.

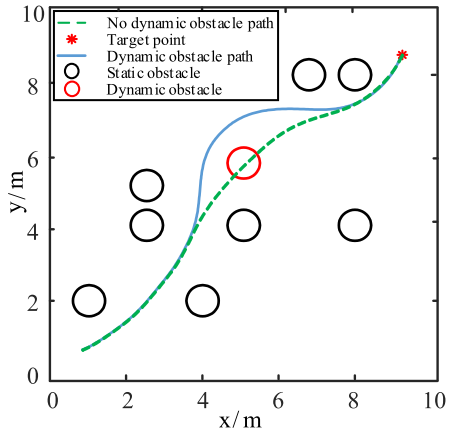


FIGURE 11. DWA algorithm simulation diagram.

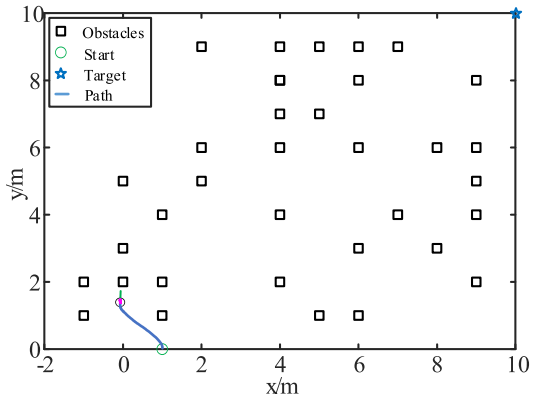


FIGURE 12. DWA algorithm falls into local optimum.

V. HYBRID ALGORITHM

According to the above analysis, robot path planning should consider the interference of dynamic obstacles as well as static obstacles. The global path planning of the A* algorithm only considers static obstacles in the surrounding environment and does not consider the influence of dynamic obstacles, which may lead to collisions between robots and dynamic obstacles. However, the local path planning of the DWA algorithm only considers the obstacles in the surrounding environment without the awareness of global path planning, which results in the robot falling into the local optimal and failing to reach the target point. To solve this problem, the proposed fusion algorithm combining the improved A* algorithm and the DWA algorithm can not only ensure the avoidance of obstacles but also ensure the smoothness and optimality of path planning.

As shown in Figure 13, the improved A * algorithm and DWA algorithm combined with a mixture of path planning system design, mainly includes global path planning and local path planning of two parts, using the improved A * algorithm for global path planning, and then according to the surrounding environment sensors information to update

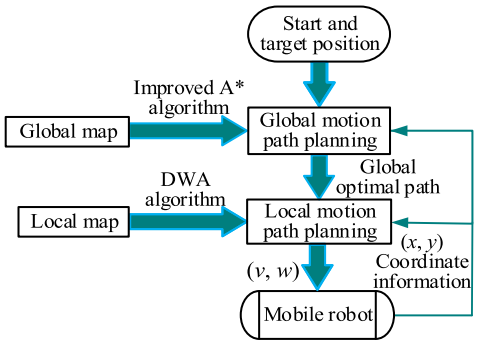


FIGURE 13. Hybrid algorithm planning diagram.

the local maps, combined with planning out the global path generation of target. The DWA method is then utilized to complete local motion path planning, allowing the robot to avoid dynamic impediments, reach the local goal point, update the route continuously, and eventually reach the target location.

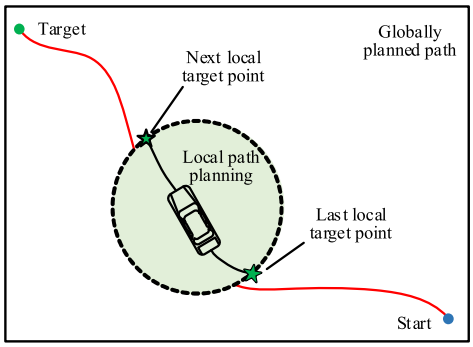


FIGURE 14. Schematic diagram of local target point.

The temporary goal points of each stage of the optimized dynamic window technique are retrieved from the key points of the global path planned by the enhanced A* algorithm. The combination of the improved A* algorithm and the DWA algorithm can solve the defects of their respective algorithms and avoid dynamic obstacles effectively in real time while completing the global path planning. As shown in Figure 14, the local path planning algorithm combined with the global path generates the local target point, and the global target point is finally reached after the continuous update of the last local target point and the next local target point. The fusion algorithm not only ensures the optimal global path, but also ensures good obstacle avoidance and movement effects in local planning.

In order to verify the feasibility and effectiveness of the path planning of the above mixed algorithm, The local optimum state of the DWA algorithm is straightforward to achieve. The simulation conditions are the same, and related simulations are performed. The starting point is still set to (1, 0), and the end point is set (10, 10). The orange path in the figure is the global path planning track by using the traditional A * algorithm. On this basis, the DWA algorithm is integrated, the search results of the mixing algorithm are the red path in the figure. The final result is shown in Figure 15.

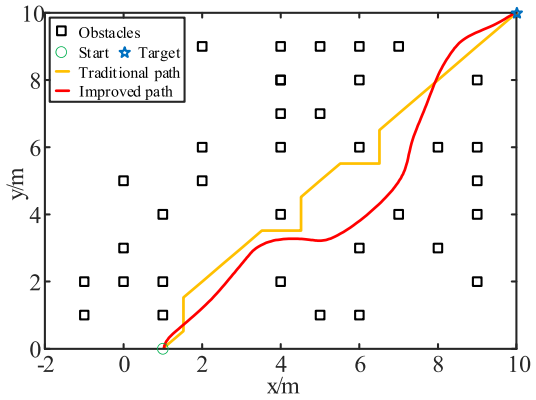


FIGURE 15. Path simulation diagram of hybrid algorithm.

TABLE 3. Performance comparison of each algorithm.

Algorithm type	Number of turning points	Smooth	Avoid dynamic obstacles	Path length /m
Traditional A* algorithm	8	No	No	14.07
Improved A* algorithm	6	Yes	No	11.92
DWA algorithm	-	Yes	Yes	Not arrived
Hybrid algorithm	4	Yes	Yes	13.56

As shown in Figure 15, compared with the traditional A* algorithm, the fusion algorithm in this paper avoids the occurrence of long routes and excessive turning angles. Furthermore, as shown in Table 3, the improved hybrid method decreases the number of turns by 50% and the path length by 3.62% to the traditional A* algorithm, improving planning efficiency.

The hybrid algorithm's trajectory may be seen above. A smooth curve, the combined advantage of the two, not only solves the A* algorithm's inability to avoid dynamic obstacles, but also compensates for the DWA algorithm's inability to fall into the most optimal defects in local path planning, allowing the track to meet the motor and angle to the smoothness of the constraint conditions.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the effectiveness of the fusion algorithm proposed in this paper, the robot operating system (ROS) was used for verification. The experimental environment was a 64-bit Ubuntu 18.04 operating system with 4GB of memory, and the experimental platform was ROS(Melodic). The starting point of path planning was (1,0) and the target point position was (17,15). In Figure 16, the green arrow represents the position and direction of the starting point, and the red arrow represents the position and direction of the target point.

The traditional A* algorithm's path involves several turning sections and large turning degrees, resulting in increased path redundancy and a severe reduction in the motor's operating efficiency and life, which is detrimental to the mobile robot walking.

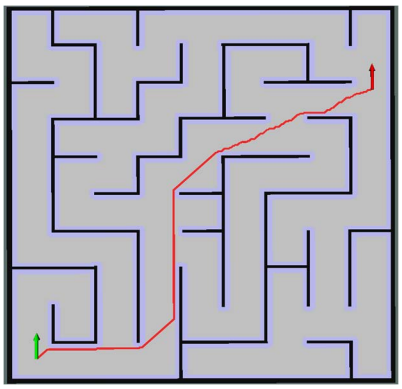


FIGURE 16. Traditional A* path planning under ROS.

The realization results of the fusion algorithm in ROS in this paper are shown in Figure 17. The red line represents global path planning, the green line represents local path planning, and the blue area represents the expansion layer of the obstacle. The path planning process of a mobile robot in the initial, intermediate and final stages is described in the figure respectively. The robot avoids impediments in the general direction of global route planning, as well as local path planning, in order to complete the navigation assignment as rapidly as feasible.

By using the fusion algorithm proposed in this paper, the path smoothness is guaranteed, the redundant points and the turning angle are reduced effectively, and the smoothness and length of the path are optimized.

Considering that the mobile robot may encounter interference from dynamic obstacles when walking, dynamic obstacles are added to the path planned by the fusion algorithm, and the path planned by the mobile robot is shown in Figure 18. The figure describes the path planning process of the mobile robot in the beginning, end and middle to avoid obstacles. The blue circles and bar squares in the figure replace dynamic obstacles. Due to the limited size of the picture, the box selects the local details of the mobile robot walking along the path, and the local zoom is displayed in the lower right corner of the picture.

As can be seen from Figure 18, the mobile robot will move forward along the previous path before encountering dynamic obstacles. When there are dynamic impediments in the way, the mobile robot will use radar location to take emergency obstacle avoidance to escape the dynamic obstacles, allowing the mobile robot to complete its local path planning.

The experiment used the mobile robot which is shown in Figure 19 as the test object. This mobile robot has multiple sensors, such as an inertial measurement unit (IMU), lidar, camera and coded geared motor. It also has four 45-degree mecanum wheels with rollers. The experimental environment is a rectangular area of 8 m × 10 m. The obstacles in the area are three rectangular blocks of 0.3 m × 1 m that are randomly placed in the field. The master and slave connection are configured in the experiment, and the Raspberry Pi on the mobile robot is used as the host, and the Cartographers

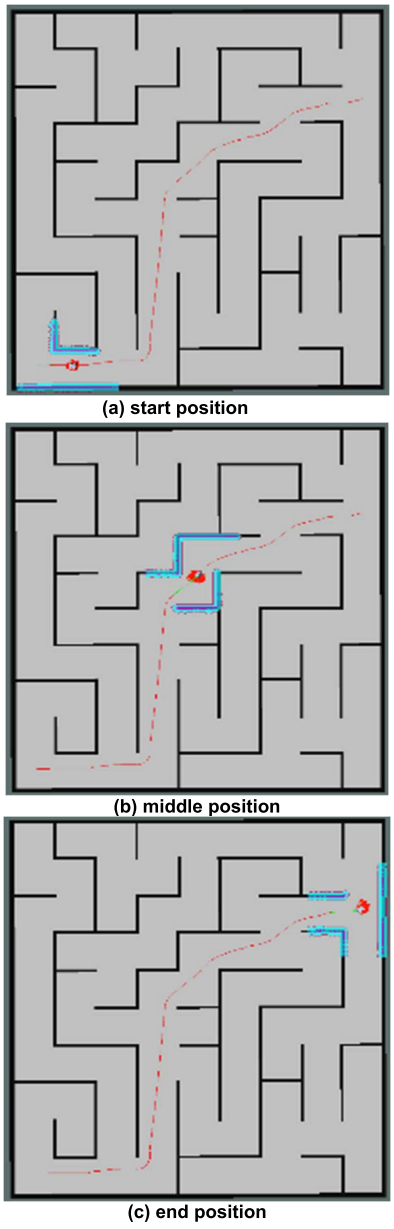


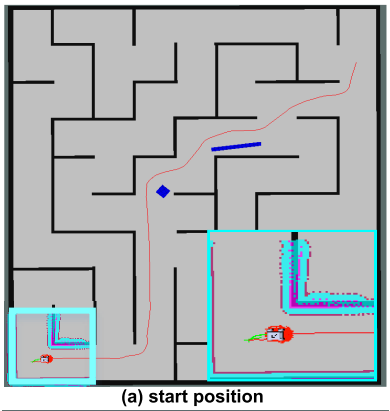
FIGURE 17. Fusion algorithm path planning in ROS.

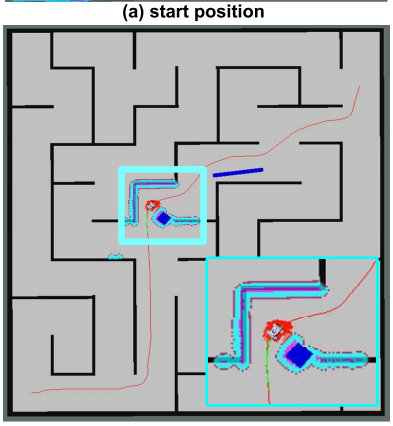
algorithm is used to build a raster map of the experimental scene.

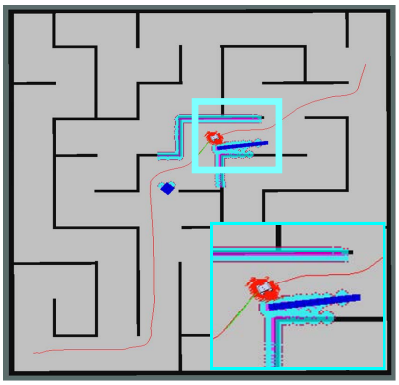
Table 4 lists the selection of some main hardware equipment and parameters in the design process of the mobile robot. The selection of parameters determines the performance of the algorithm to some extent.

The scanning angle of the lidar is selected from 0-360°, which is more conducive to quickly obtaining information about the surrounding environment and obstacles; the minimum scanning distance $d_{\min} = 0.15m$, the maximum scanning distance $d_{\max} = 12m$, the selected scanning distance range are more suitable for the experimental environment of this paper.

The choice of camera will affect the mobile robot's ability to perceive the environment. The better the resolution, frame


(a) start position


(b) position 1


(c) position 2

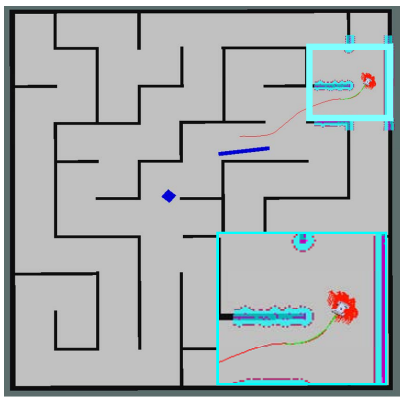

(d) end position

FIGURE 18. Path planning under dynamic obstacles.

rate and detection range of the main parameters, the better. However, considering the experimental cost, the details of the

FIGURE 19. Laboratory equipment and experimental site.

TABLE 4. Hardware equipment and its main parameters.

Hardware equipment	Main parameter or type
Lidar	Scan angle: 0-360° Scan distance:0.15-12m Resolution:320x240(16bits)
Camera	Fps: 30 Detection scope:0.8-6.0m
IMU	MPU9250
Raspberry Pi	Raspberry Pi 3b+

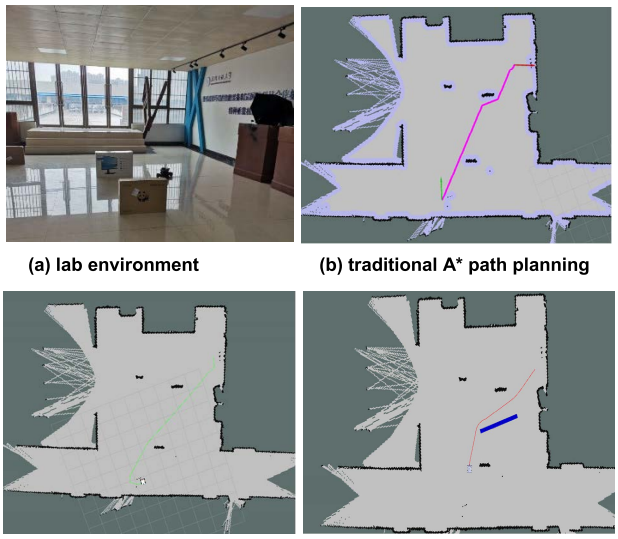
parameters selected in the table have fully met the experimental requirements of this paper. IMU is a device that measures the three-axis angular velocity and acceleration of an object. Angular velocity and acceleration are important parameters for kinematic modeling, and they are also the basic parameters of the algorithm in this paper. The accuracy of the data will greatly affect the performance of the algorithm. The Raspberry Pi is a small single-board computer with the Ubuntu 18.04 (ROS melodic) operating system installed.

Fig. 20(b, c) shows the mobile robot in the face of a complex obstacle, using its own sensor for state estimation, at the same time using the AMCL localization algorithm and mixing to complete the autonomous navigation of the mobile robot path planning algorithm.

Considering that the mobile robot may encounter the interference of dynamic obstacles when walking, dynamic obstacles are added to the path planned by the fusion algorithm. The mobile robot moves forward along the previous path before encountering dynamic obstacles. When there are dynamic obstacles in the path, the mobile robot will rely on radar to locate emergency obstacles and bypass the dynamic obstacles. The experimental results are shown in Figure 20(d).

The experimental results are shown in Table 5. We can infer that the suggested hybrid method can successfully complete path planning based on the smoothness of the experimental site. The initial marching and navigation will undoubtedly influence the mobile robot, but this will have no effect on the ultimate path planning and obstacle avoidance procedure.

Table 5 shows that under the same environment, the hybrid algorithm in this paper reduces the average time consumption by 10.27%, the number of path inflection points by 57.14%,



(a) lab environment (b) traditional A* path planning

(c) fusion algorithm path planning (d) path planning under dynamic obstacles.

FIGURE 20. Experimental environment and algorithm path planning.

TABLE 5. Comparison of traditional and hybrid algorithms.

Navigation results (10 times)	Traditional algorithms	Hybrid algorithms
Experimental site area	80m^2	80m^2
Average elapsed time	55.42s	49.73s
Path turning point	7	3
Average error at the end point	3 cm	2 cm

and the accuracy is higher than 33.33% compared with the traditional algorithm. The results further verify the superiority of the fusion algorithm, which has good applicability and security for real and complex dynamic environments, and can timely and reliably avoid new dynamic obstacles in the path, and has the function of dynamic obstacle avoidance.

VII. CONCLUSION

The traditional A* method is improved in this paper: we employ an adaptive modifying step size algorithm and a cubic Bezier curve to handle the concerns of too many turning points and too big turning angles in the search route, reducing run time and increasing robot motion efficiency.

The global path planning system based on the A* algorithm and the Bezier curve in this article evaluates the effects of weights, optimizes the corners of the produced task path, and smooths the path using the Bezier curve.

Based on the improvement of A*, the hybrid path planning algorithm is proposed in this paper. It integrates the DWA algorithm for real-time obstacle avoidance, which compensates for the poor timeliness of the A* algorithm. The optimal path is planned by combining the global path-related information to realize the optimization of route length, smoothness and safety performance.

By comparing the simulation experiments, the real-time, validity and security of the proposed fusion algorithm of improved A* and DWA are verified. In the future, robot path planning algorithms will be studied in multi-fields and multi-scenes, and the path planning of mobile robots in multi-task complex scenes will be further explored by combining deep learning and machine vision.

ACKNOWLEDGMENT

The authors would like to thank the editors and the anonymous reviewers whose insightful comments have helped to improve the quality of this paper considerably.

REFERENCES

[1] J. Han and Y. Seo, "Mobile robot path planning with surrounding point set and path improvement," *Appl. Soft Comput.*, vol. 57, pp. 35–47, Aug. 2017, doi: 10.1016/j.asoc.2017.03.035.
[2] Y. Wang, X. Liang, B. Li, and X. Yu, "Research and implementation of global path planning for unmanned surface vehicle based on electronic chart," in *Recent Developments in Mechatronics and Intelligent Robotics*, vol. 690, F. Qiao, S. Patnaik, and J. Wang, Eds. Cham, Switzerland: Springer, 2018, pp. 534–539, doi: 10.1007/978-3-319-65978-7_80.
[3] X. Zhao, Z. Wang, C. K. Huang, and Y. W. Zhao, "Path planning for mobile robot based on improved A* algorithm," *Robot*, vol. 40, no. 6, pp. 903–910, 2018, doi: 10.13973/j.cnki.robot.170591.
[4] S. Y. Duan, Q. F. Wang, X. Han, and G. R. Liu, "A* path optimization method with ensuring safe distance," *J. Mech. Eng.*, vol. 56, no. 18, pp. 205–215, 2020.
[5] X. Lai, J. Li, and J. Chambers, "Enhanced center constraint weighted A* algorithm for path planning of petrochemical inspection robot," *J. Intell. Robotic Syst.*, vol. 102, no. 4, p. 78, Aug. 2021, doi: 10.1007/s10846-021-01437-8.
[6] X. Shi, H. Liu, Y. Li, B. Zhu, and J. Liang, "Location planning of field ammunition depot for multi-stage supply based on dijstra algorithm," *J. Phys., Conf. Ser.*, vol. 2068, no. 1, Oct. 2021, Art. no. 012015, doi: 10.1088/1742-6596/2068/1/012015.
[7] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. ICRA Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia*, Apr. 2000, pp. 995–1001, doi: 10.1109/ROBOT.2000.844730.
[8] L. Zhang and Y. Li, "Mobile robot path planning algorithm based on improved a star," *J. Phys., Conf. Ser.*, vol. 1848, no. 1, Apr. 2021, Art. no. 012013, doi: 10.1088/1742-6596/1848/1/012013.
[9] H. Min, X. Xiong, P. Wang, and Y. Yu, "Autonomous driving path planning algorithm based on improved A* algorithm in unstructured environment," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 235, nos. 2–3, pp. 513–526, Feb. 2021, doi: 10.1177/0954407020959741.
[10] X. L. Ma and H. Mei, "Global path planning for mobile robots based on bidirectional hop search algorithm," *Mech. Sci. Technol. Aerosp. Eng.*, vol. 39, no. 10, pp. 1624–1631, 2020, doi: 10.13433/j.cnki.1003-8728.20190342.
[11] W. Wei, P. Dong, and F. Zhang, "The shortest path planning for mobile robots using improved A* algorithm," *J. Comput. Appl.*, vol. 38, no. 5, p. 1523, 2018.
[12] W. R. Du, X. Y. Wang, F. K. Jia, Z. Zheng, and H. Y. Li, "Research on path planning algorithm of unknown environment based on multi-layer bidirectional A*," *Comput. Appl. Softw.*, vol. 36, no. 12, pp. 261–267, 2019.
[13] C. Henkel, A. Bubeck, and W. Xu, "Energy efficient dynamic window approach for local path planning in mobile service robotics**this work was conducted at the university of Auckland, Auckland, New Zealand," *IFAC-PapersOnLine*, vol. 49, no. 15, pp. 32–37, 2016, doi: 10.1016/j.ifacol.2016.07.610.
[14] Y. Zhang, J. Z. Song, and Q. Q. Zhang, "Local path planning for outdoor cleaning robot based on improved dynamic window method," *Robot*, vol. 42, no. 5, pp. 617–625, 2020, doi: 10.13973/j.cnki.robot.190649.
[15] J. H. Zhang, Q. Feng, A. D. Zhao, W. He, and X. Hao, "Local path planning of mobile robot based on self-adaptive dynamic window approach," *J. Phys., Conf. Ser.*, vol. 1905, no. 1, May 2021, Art. no. 012019, doi: 10.1088/1742-6596/1905/1/012019.

[16] J. Cao, "Robot global path planning based on an improved ant colony algorithm," *J. Comput. Commun.*, vol. 4, no. 2, pp. 11–19, 2016, doi: 10.4236/JCC.2016.42002.

[17] H. B. Wang, C. Hao, P. Zhang, M. Q. Zhang, P. H. Yin, and Y. S. Zhang, "Path planning for mobile robot based on A* algorithm and artificial potential field method," *China Mech. Eng.*, vol. 30, no. 20, pp. 2489–2496, 2019.

[18] B. Curto, V. Moreno, and F. J. Blanco, "A general method for C-space evaluation and its application to articulated robots," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 24–31, Feb. 2002., doi: 10.1109/70.988971.

[19] T. T. Mac, C. Copot, D. T. Tran, and R. D. Keyser, "A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization," *Appl. Soft Comput.*, vol. 59, pp. 68–76, Oct. 2017, doi: 10.1016/j.asoc.2017.05.012.

[20] L. Yu, D. Kong, X. Shao, and X. Yan, "A path planning and navigation control system design for driverless electric bus," *IEEE Access*, vol. 6, pp. 53960–53975, 2018, doi: 10.1109/ACCESS.2018.2868339.

[21] J. Liu, J. Yang, H. Liu, X. Tian, and M. Gao, "An improved ant colony algorithm for robot path planning," *Soft Comput.*, vol. 21, no. 19, pp. 5829–5839, Oct. 2017, doi: 10.1007/s00500-016-2161-7.

[22] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 1986–1991, doi: 10.1109/ROBOT.2007.363613.

[23] M. Nieuwenhuisen and S. Behnke, "Layered mission and path planning for MAV navigation with partial environment knowledge," in *Intelligent Autonomous Systems*, vol. 302, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds. Cham, Switzerland: Springer, 2016, pp. 307–319, doi: 10.1007/978-3-319-08338-4_23.

[24] C. Xu, L. Hua, and F. Jiyou, "Research on robot random obstacle avoidance method based on fusion of improved A* algorithm and dynamic window method," Tech. Rep., Mar. 2021, pp. 132–140, vol. 42, doi: 10.19650/j.cnki.cjsi.J2007064.

[25] F. Bayat, S. Najafinia, and M. Aliyari, "Mobile robots path planning: Electrostatic potential field approach," *Expert Syst. Appl.*, vol. 100, pp. 68–78, Jun. 2018, doi: 10.1016/j.eswa.2018.01.050.

YONGGANG LI received the bachelor's degree from the Anhui University of Technology, China, in 2020, where he is currently pursuing the Ph.D. degree with the School of Mechanical Engineering. His research interests include autonomous navigation, path planning, and machine vision.

RENCAI JIN is currently a Senior Engineer and an Off-Campus Tutor for full-time Professional degree graduate students with the Anhui University of Technology. He is mainly responsible for the company's technological innovation, engineering quality, building informatization, metallurgical national team steelmaking engineering construction research institute, national enterprise technology center, national postdoctoral workstation, construction industrialization, and intelligent construction.

XIANGRONG XU (Member, IEEE) is currently a Professor with the School of Mechanical Engineering, Anhui University of Technology, Anhui, China. He completed his postdoctoral research with Purdue University, IN, USA, in 2001, where he worked as a Researcher Associate, from 2001 to 2002. Since 2002, he has been working as a Researcher with Florida State University, Tallahassee, FL, USA. He has over 100 papers published in international journals and conference proceedings. His research interests include robotics, aerial robot, mechanical design, and biomechanics.

YUANDI QIAN is currently pursuing the Ph.D. degree. He is currently a Senior Engineer and a National First-Class Construction Engineer. He is also the Leader of Company's Green Building Materials and Intelligent Construction Technology, the Innovation Leader of the "Special Support Plan" in Anhui, a member of the Organizing Committee of the ICBTE International Academic Conference, a member of the National Technical Standard Innovation Base (Construction Engineering) Prefabricated Building Professional Committee, and an International Standard Registration Expert.

HAIYAN WANG received the master's degree in management science and engineering from the Anhui University of Technology, in 2012. She is currently a Lecturer with Maanshan University. Her research interests include supply chain management and health services and management. In 2020, she was the Principal Investigator of the Top Talent Project of Anhui Province, China.

SHANSHAN XU received the B.S. degree in nursing from Florida State University, Tallahassee, USA, in 2011. She is currently a Teacher with the School of Osaka Medical Engineering, Maanshan University. Her research interests include rehabilitation robot and bio-medical robot.

ZHIXIONG WANG is currently a Professor with the School of Medicine, Osaka University, Japan. He is mainly engaged in research in the field of biomedical engineering. He has presided over and participated in 13 national and provincial research projects, four of which won provincial awards. He has published 39 influential monographs and papers in international and national journals.