Robot Dynamic Path Planning Based on Improved A* and DWA Algorithms

Chenxi Guan
Graduate school of Tangshan
Southwest Jiaotong University
Chengdu, China
940545134@qq.com

Shuying Wang
School of Information Science and Technology
Southwest Jiaotong University
Chengdu, China
w_shuying@126.com

***Abstract*—When the traditional A* algorithm is applied to robot path planning, it has the problems of low efficiency and unable to avoid obstacles dynamically. In order to solve the above problems, a fusion algorithm based on improved A* algorithm and DWA algorithm is proposed. The A* algorithm is improved in three aspects: reducing the search direction of A* algorithm to reduce the search time, adding path information parameters to dynamically adjust the weight of heuristic function, and introducing important node extraction strategy to reduce the number of turns and shorten the path. Finally, the improved A* algorithm is fused with DWA algorithm. The experimental results show that the improved fusion algorithm can realize global optimal path planning and local real-time obstacle avoidance.**

Keywords—path planning, mobile robot, improved a* algorithm, dynamic window method, real-time obstacle avoidance

I. Introduction

Path planning is one of the main functions of robots performing various tasks [1]. Path planning is mainly divided into two types: one is global path planning containing all environmental information, and the other is local path planning containing unknown environment [2]. At present, the mainstream global path planning algorithms include A* algorithm [3], ant colony algorithm [4], Dijkstra algorithm [5], particle swarm optimization algorithm [6], etc. At present, the mainstream local path planning algorithms include dynamic window method (DWA) algorithm [7], artificial potential field algorithm [8], etc.

A* algorithm is the mainstream global path planning algorithm, but the algorithm has long search time, long planned route, and can not solve the problem of dynamic obstacles, which is not conducive to the actual movement of the robot [9]. In reference [10], aiming at the shortcomings of A* algorithm, a two-way search Binary Tree A* algorithm is proposed. The binary tree data structure is added to the A* algorithm for optimization, and the two-way search strategy is adopted to improve the efficiency of the algorithm. Reference [11] proposed an improved A* algorithm, which simplifies path nodes by smoothing method and considers the actual width of obstacles in path planning. Although the above literature improves the A* algorithm, it does not solve the problem of dynamic obstacles.

In order to solve the problems of long search time, many turning points, long path distance and unable to avoid dynamic obstacles, this paper improves the A* algorithm and integrates the improved A* algorithm with DWA algorithm. The fusion algorithm can not only plan a good global path, but also avoid obstacles in the case of local unknown.

II. Traditional A* and DWA Algorithm

A. Basic Principles of Traditional A Algorithm*

A* algorithm is suitable for global path planning. It is a typical heuristic algorithm with good robustness. The principle of A* algorithm is to search the specified direction from the starting point, and calculate the actual value from the starting position to the current position continuously, as well as the estimated cost from the current position to the end. A* algorithm selects the point with the lowest total generation value as the next node until the end of the search, and completes the global path planning. The cost function of A* algorithm is shown as follows:

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

In this formula, n is the current node; $F(n)$ is the total cost of the current node; $G(n)$ is the actual cost from the starting point to the current node; $H(n)$ is the estimated cost of the current node to the destination. There are three main methods to calculate the cost, namely Euclidean distance, Manhattan distance and Chebyshev distance. When comparing path planning, we mainly compare the path planning length of different algorithms, so we use Euclidean distance to calculate the cost between nodes, and the Euclidean distance calculation formula is as follows.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

Where, (x_1, y_1) is the starting point coordinate; (x_2, y_2) is the coordinate of the target point.

The search diagram of A* algorithm is shown in Fig. 1 as follows: the blue square is the starting point. In the process of path planning, the algorithm will expand 8 adjacent nodes of the current node, and the 8 squares around the blue square in the figure are candidate child nodes in the current state.

2022 4th International Conference on Control and Robotics (ICCR) | 978-1-6654-8641-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCR55715.2022.10053929

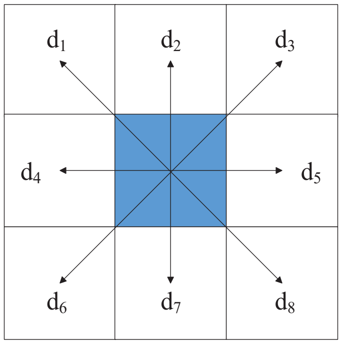


Fig. 1. A* schematic diagram of algorithm node search

A* algorithm is often used in global path planning because of its simple calculation and high efficiency, but the traditional A* algorithm has the following disadvantages:

1. The traditional A* algorithm needs to search nodes in 8 directions nearby, resulting in a long search time.

2. The route planned by the traditional A* algorithm has many turning points and long distance, which is not conducive to the movement of the robot. Many turning points may lead to the rollover of the robot.

3. The traditional A* algorithm is offline when planning the route and will not update the map information, so it will not consider the dynamic obstacles in the future, which makes the A* algorithm unable to avoid in time when encountering sudden obstacles blocking the route.

B. Basic Principles of DWA Algorithm

A* algorithm can not avoid sudden obstacles in real time, and DWA algorithm can effectively make up for this defect. This paper plans to integrate A* algorithm and DWA algorithm, so that it can not only complete global path planning, but also avoid dynamic obstacles in local path planning.

DWA algorithm is a local path planning algorithm, which is suitable for path planning in small map environment. Moreover, due to the reason of algorithm design, DWA algorithm will collect environmental information in real time, and then constantly update the route.

In the process of speed sampling, the robot will be subject to its own maximum and minimum speed limits, and the limits are shown in formula (3) :

$$V_t = \{(v,\omega) | v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \tag{3}$$

Where, $v_{\max}(v_{\min})$ is the maximum (small) linear velocity of the robot; $\omega_{\max}(\omega_{\min})$ is the maximum and minimum angular velocity of the robot.

The mobile robot will also be affected by its own motor. Due to the limited traction force of acceleration and deceleration, a dynamic window will be generated when the robot moves. This dynamic window is the actual speed of the robot, as shown in formula (4) :

$$\begin{aligned} V_r = \{ & (v,\omega) | v \in [v_c - a_m \Delta t, v_c + a_p \Delta t], \\ & \omega \in [\omega_c - \alpha_m \Delta t, \omega_c + \alpha_p \Delta t] \} \end{aligned} \tag{4}$$

Where, $v_c(\omega_c)$ is the linear (angular) velocity at the current moment; $a_m(a_p)$ is the maximum minus (plus) speed of the robot; $\alpha_m(\alpha_p)$ is the maximum Angle minus (plus) velocity of the robot.

It is stipulated that mobile robots cannot collide with obstacles when moving. When the robot is in the state of maximum deceleration, the scope of dynamic window can be further reduced, and the limited scope is shown in formula (5) :

$$\begin{aligned} V_k = \{ & (v,\omega) | v \le \sqrt{2dist(v,\omega)a_m}, \\ & \omega \le \sqrt{2dist(v,\omega)\alpha_m} \} \end{aligned} \tag{5}$$

Where, $dist(v,\omega)$ is the minimum distance from different simulated paths to obstacles; a_m is the maximum deceleration of the robot; α_m is the maximum angular deceleration of the robot.

To sum up, the intersection of the above three constraints is the final dynamic window velocity V , as shown in formula (6) :

$$V \in V_t \cap V_r \cap V_k \tag{6}$$

After obtaining different simulated paths, the DWA algorithm will comprehensively evaluate these paths and take the path with the highest score as the optimal path. Path evaluation function is shown in formula (7) :

$$\begin{aligned} E(v,\omega) = & \alpha Head(v,\omega) + \beta Vel(v,\omega) \\ & + \gamma Dist(v,\omega) \end{aligned} \tag{7}$$

Where, α, β, γ is the weighting coefficient; $Head(v,\omega)$ is the evaluation function of deflection Angle; $Vel(v,\omega)$ is the speed evaluation function; $Dist(v,\omega)$ is the evaluation function of safety factor.

Due to the lack of global planning, DWA algorithm may have evaluation function failure when facing obstacles with close distance. Therefore, DWA algorithm needs to be improved

III. Algorithm to Improve

Firstly, this paper improves the traditional A* algorithm from three aspects to make the A* algorithm have better global path planning ability. Then on this basis, this paper integrates the improved DWA algorithm and the improved A* algorithm to make it have the ability to avoid obstacles dynamically.

A. Improved A Algorithm*

(1) Improvement of search direction

The traditional A* algorithm will search 8 adjacent nodes of the current node every time, but these adjacent nodes do not need to be searched. All of them will waste computing resources and increase time cost. In order to solve this problem, this paper proposes to add search direction judgment when searching nodes, find and delete the three adjacent nodes that are the most opposite to the direction of the target node, and reduce the eight search nodes to five search nodes to improve efficiency. The search principle is as follows:

Step 1: Specify that the right direction of the map is the positive direction of the *X* axis, *T* is the target point, and *B* is the current node. Connect *T* and *B* into a straight line L_0, and the direction is towards the target point.

Step 2: Connect the current node *B* and child node C_1 into a straight line L_1 with the direction towards the child node. Calculate the angle between line L_0 and L_1.

Step 3: Repeat the second step, calculate all the included angles of the corresponding straight lines of the eight sub nodes, discard the three nodes with the largest deviation, and retain only five.

The search principle is shown in Fig. 2. *B* is the starting point, *T* is the target point, green is the discarded node, and gray is the effective search node.

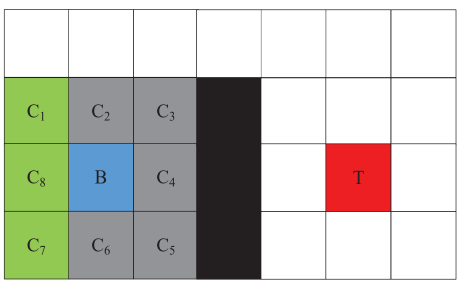


Fig. 2. Improved A* algorithm search graph

(2) Optimization of heuristic function:

The traditional A* algorithm has directionality in search, but it will visit a large number of nodes, which is inseparable from its own algorithm design. When $G(n)$ is greater than $H(n)$, the algorithm finds the optimal path by visiting a large number of nodes, which is very inefficient; if $G(n)$ is less than $H(n)$, the number of nodes accessed is small, resulting in a larger path length. Therefore, this paper sets a dynamic path parameter *K* to adjust the weight of the heuristic function. If the distance from the current position of the robot to the end point is *D* and the total distance from the starting point to the end point is s, then the path parameter *K* can be expressed as formula (8):

$$K = \frac{D}{S} \tag{8}$$

The path parameter *K* is determined by the distance *D* from the current position of the robot to the end point and the distance *S* from the start point to the end point. It is a dynamic parameter. The size of *K* is determined by the position of the robot. The weight of heuristic function $H(n)$ will change according to the change of *K*, so it is closer to the optimal solution. The improved cost function is shown in formula (9):

$$F(n) = G(n) + e^K H(n) \tag{9}$$

(3) Extraction of key points:

The improved A* algorithm can effectively improve the efficiency of the algorithm, but the path planned by the improved A* algorithm still has A large number of redundant nodes and many turns. To solve these problems, this paper designs a path smoothing optimization strategy based on the idea of key points extraction, which deletes redundant nodes and only retains key nodes. The steps are as follows:

Step 1: all nodes on the path generated by the improved A* algorithm are traversed. If A node is on the same line as its front and rear nodes, the node is deleted.

Step 2: After deleting all the redundant nodes in Step 1, set the remaining path nodes as $\{Q_x | x = 1,2,3,...n\}$, and make Q_1 and Q_2 nodes into a line segment path. If there is no intersection between path Q_1Q_2 and obstacle, make Q_1 and Q_3 nodes into a line segment path, and so on, until path Q_1Q_x has intersection with obstacle. At this point, keep path $Q_1Q_{x\text{-}1}$. Meanwhile, only Q_1 and Q_{x-1} nodes are retained, and all intermediate nodes are deleted.

Step 3: Continue Q_{x-1} and repeat step 2 until the last node is traversed.

The path after key points extraction only contains the most critical nodes, which effectively shortens the path length and reduces the number of turns.

B. Improved DWA Algorithm

If the distance between the target point and the starting point is very long, when the traditional dynamic window algorithm encounters an obstacle, the evaluation function may lose its function and approach the obstacle. In this process, the path length will increase and the robot may collide. In order to improve the adaptability of DWA algorithm, the smoothing factor function $Head_{smooth}(v, \omega)$ is added on the basis of dynamic window algorithm. This function needs the path key nodes in any global path planning algorithm, and then allows the robot to prepare in advance at the key position of the global path, change the moving direction in time, and help the robot avoid obstacles in real time.

Although the global path planning algorithm can not avoid obstacles in real time, it can give the global approximate route. The $Head_{smooth}(v, \omega)$ added in this paper needs to use the key turning point in the global path planning to evaluate the advantages and disadvantages of each track by calculating the minimum distance from the forward simulation path to the key node of each sampling speed in the dynamic window. The improved evaluation function is shown in formula (10):

$$E(v,\omega)=\alpha\cdot Head(v,\omega)+\gamma\cdot Dist(v,\omega)$$
$$+\beta\cdot Vel(v,\omega)+\phi Head_{smooth}(v,\omega) \tag{10}$$

Where, α, β, γ, ϕ is the weighting coefficient of each term; $Head(v,\omega)$ is the evaluation function of deflection Angle; $Vel(v,\omega)$ is the speed evaluation function; $Dist(v,\omega)$ is the safety factor evaluation function, $Head_{smooth}(v,\omega)$ is the smoothing factor function of the robot.

C. *Improved Algorithm Flow*

The improved A* algorithm has improved the performance of the algorithm, but the turning of the path is not smooth and abrupt, which does not conform to the motion characteristics of the robot, resulting in collision or rollover, and can not effectively deal with the unknown obstacles in the dynamic environment. If only DWA algorithm is used, there will be a lack of detail orientation and only one orientation at the end point. In this way, it is very easy to produce local optimization, resulting in the increase or failure of path planning distance. Therefore, this paper takes the important node of A* algorithm as the key guiding point of DWA algorithm for local path planning. The flow chart of fusion algorithm is shown in Fig. 3:

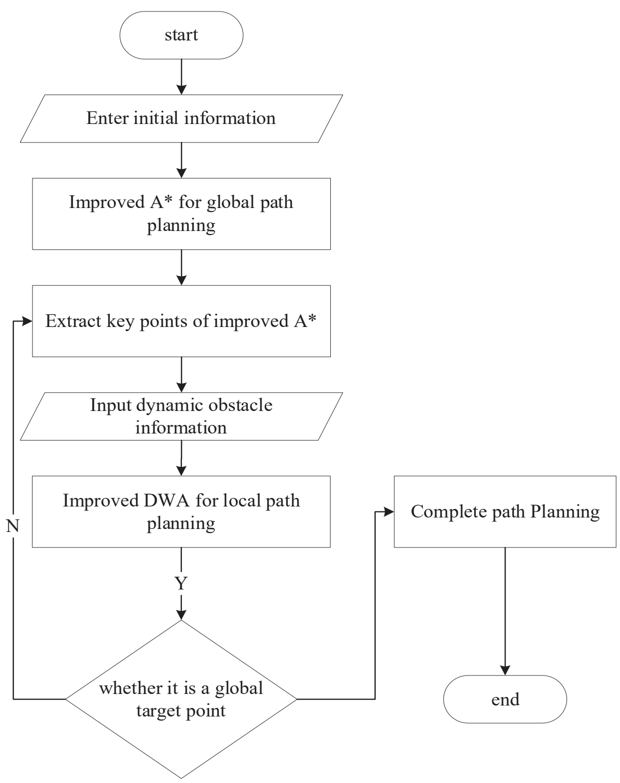


Fig. 3. Flow chart of fusion algorithm

IV. Simulation Verification and Analysis

In order to verify the effectiveness and adaptability of the algorithm, this paper uses a computer with AMD Ryzen 5 4500U processor to conduct simulation experiments, and the simulation software is Matlab 2019a, and uses the grid method to build maps. In the raster map environment, the raster map is composed of different numbers of square grids, each square with a side length of 1m, where the black square represents obstacles; The white squares represent clear areas where the robot can move; △ is the starting point of the robot. ◯ is the end of the robot.

A. *Improved A* Algorithm Simulation Experiment*

Before verifying the effectiveness of the fusion algorithm, this paper needs to verify the excellence of the improved A* algorithm in global path planning. Therefore, this paper uses the traditional A* algorithm and the improved A* algorithm in 15×15、30×30 and 45×45 grid map is simulated.The simulation results of the two algorithms are shown in Fig. 4,Fig. 5 and Fig. 6, the figure on the left is the traditional algorithm diagram, and the figure on the right is the improved algorithm diagram and the performance comparison is shown in Table I,Table II,Table III.

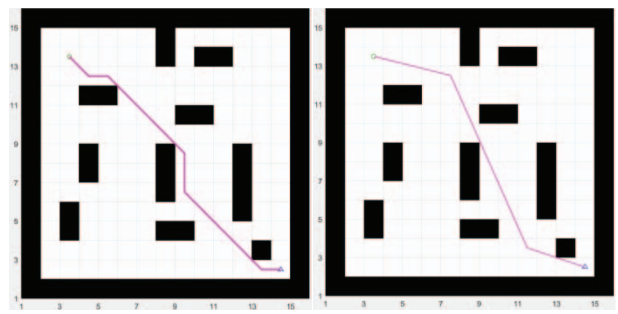


Fig. 4. 15×15 raster map comparison

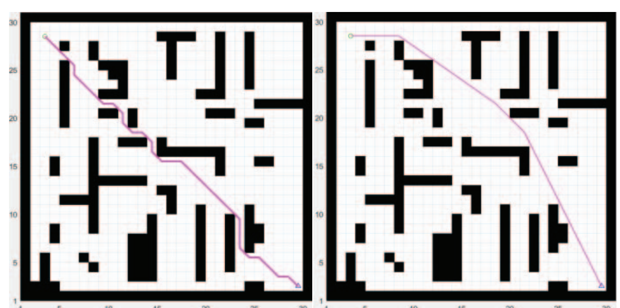


Fig. 5. 30×30 raster map comparison

Fig. 6. 45×45 raster map comparison

TABLE I. 15×15 RASTER MAP COMPARISON

TYPES	15×15 grid map		
	time/s	path length/m	turning point
TRADITIONAL A*	0.034	18.411	5
IMPROVED A*	0.028	17.037	2

TABLE II. 30×30 RASTER MAP COMPARISON

TYPES	15×15 grid map		
	time/s	path length/m	turning point
TRADITIONAL A*	0.041	40.284	18
IMPROVED A*	0.032	37.129	3

TABLE III. 45×45 RASTER MAP COMPARISON

TYPES	45×45 grid map		
	time/s	path length/m	turning point
TRADITIONAL A*	0.051	63.255	16
IMPROVED A*	0.039	57.641	7

According to Fig. 4, Fig. 5, Fig. 6 and Table I, Table II and Table III, the improved A* algorithm has higher efficiency and better path planning effect in three raster map environments with different sizes. In terms of planning consumption time, the improved A* algorithm reduces 17.64%, 21.95% and 23.52%, respectively, compared with the traditional A* algorithm. In terms of path planning length, the improved A* algorithm reduces 7.31%, 7.83% and 8.87% compared with the traditional A* algorithm respectively. In terms of turning points, the improved A* algorithm greatly reduces compared with the traditional algorithm. The experimental results show that the direction of path search is stronger after improving the search direction, introducing dynamic path parameters and key point extraction strategy. The improved A* algorithm improves the algorithm efficiency, reduces the path length, reduces the turning point of the path, and increases the smoothness of the path.

After that, this paper verifies the real-time performance of the A* algorithm by adding obstacles. The traditional A* algorithm and the improved A* algorithm are used to plan the global path respectively, and then the combination of obstacles is randomly added to the planned path. The results are shown in the Fig. 7 below:

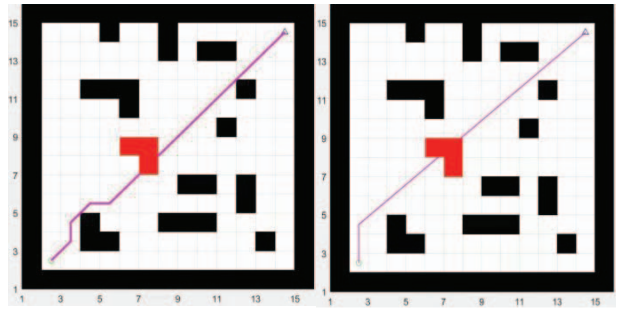


Fig. 7. Traditional and improved A* algorithms encounter dynamic obstacles

It can be seen from the Fig. 7 that the path planned by the improved A* algorithm has fewer turning points and shorter length. However, whether it is the traditional A* algorithm or the improved A* algorithm, the planned route can not avoid sudden obstacles. This is because the A* algorithm is an offline algorithm, which only uses the map information of the initial state

B. *Improved DWA Algorithm Simulation Experiment*

In order to verify the effectiveness of the improved DWA algorithm, this paper uses the traditional DWA algorithm and the improved DWA algorithm for local path planning. The smoothing factor function of the improved algorithm needs to refer to the key points generated by Dijkstra algorithm. DWA algorithm includes robot parameters and algorithm parameters. The robot parameters were set as follows: linear velocity resolution 0.02m/s, angular velocity resolution 0.5°/s, obstacle radius 0.3m, trajectory prediction time 3s, safety distance 0.2m, where α =0.4, β = 0.1, γ =0.5. The algorithm parameters were set as follows: minimum linear velocity 0m/s, maximum linear velocity 1m/s, linear acceleration 0.2m/s^2, angular velocity ±20°/s, angular acceleration 50°/s^2. The experimental results are shown in Fig. 8,the figure on the left is the traditional algorithm diagram, and the figure on the right is the improved algorithm diagram:

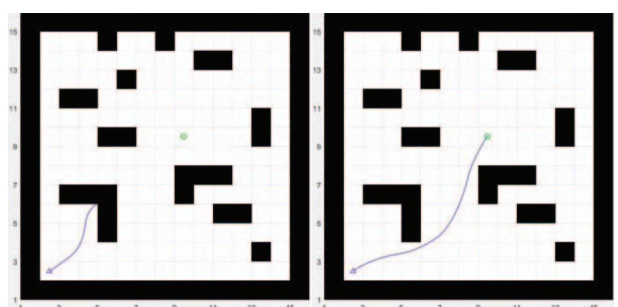


Fig. 8. Comparison of traditional and improved DWA algorithms

It can be seen from the figure that when the traditional DWA algorithm encounters a wide range of obstacle combinations, due to the lack of guidance of key points, it is unable to adjust the moving speed and deflection angle in time, resulting in the robot hitting the obstacles. The improved DWA algorithm adds a smoothing factor to make the robot decelerate and adjust the direction in time before approaching the obstacle, so as to effectively bypass the obstacle and reach the target point.

C. Simulation Experiment of Fusion Algorithm

This paper takes the important nodes of the improved A* algorithm as the key guidance point of local path planning of DWA algorithm. The experimental results are shown in Figure 9 below:

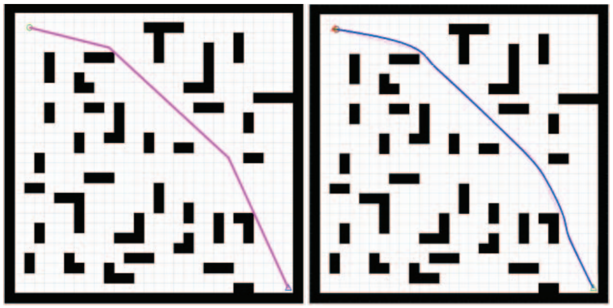
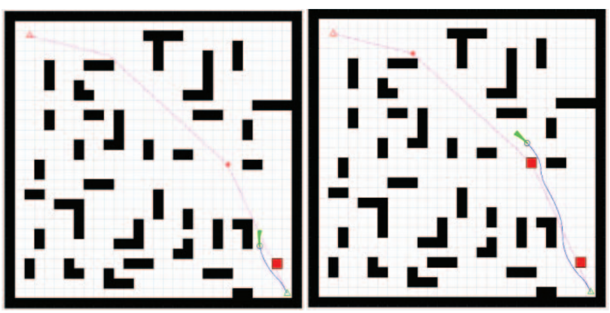


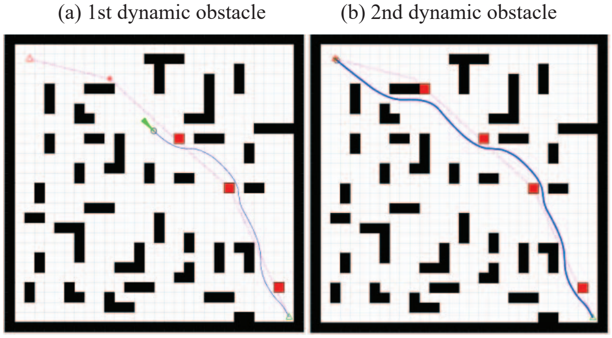
Fig. 9. Path comparison between improved A* algorithm and fusion algorithm

As can be seen from Fig. 9, the path planned by the improved A* algorithm has a large deflection angle at the turning point, and the local path is not smooth and does not conform to the robot motion. The path of the fusion algorithm is smooth and the path curvature changes continuously, which meets the requirements of robot motion stability.

Then, this paper verifies the dynamic obstacle avoidance ability of the fusion algorithm. The robot starts again. In the process of moving, this paper adds four dynamic obstacles to the path in turn. The effect of the robot passing through these dynamic obstacles is shown in Fig. 10. The dotted line represents the path of the improved A* algorithm, and the solid line represents the path of the fusion algorithm:



(a) 1st dynamic obstacle (b) 2nd dynamic obstacle



(c) 3rd dynamic obstacle (d) 4th dynamic obstacle diagram

Fig. 10. Trajectory diagram of avoiding dynamic obstacles

As shown in Fig. 10 (a), the robot carries out local path planning according to the key points of the improved A* algorithm. After the robot encounters the first dynamic obstacle, it uses the improved dynamic window algorithm to avoid the obstacle in real time, turn left in time and pass the first dynamic obstacle. Fig. 10 (b) and Fig. 10 (c) show that the robot passes through the second dynamic obstacle and the third dynamic obstacle in turn. Fig. 10 (d) shows that the robot successfully reaches the target point through all dynamic obstacles.

V. Conclusions

This paper proposes a fusion algorithm based on improved A* algorithm and DWA algorithm. The search direction of A* algorithm is reduced, the heuristic function of the algorithm is optimized, and the important nodes of A* algorithm are extracted. This paper adds smoothing factor function to DWA algorithm to ensure the security of local planning. Finally, the improved A* algorithm is fused with DWA algorithm. The simulation results show that the improved A* algorithm can realize the global optimal path planning, the path planned by the fusion algorithm is smoother, and the dynamic obstacle avoidance can be realized at the same time.

References

[1] H. Wang and W. Chen, "Multi-Robot Path Planning With Due Times," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 4829-4836, 2022.

[2] Q. Jin, C. Tang and W. Cai, "Research on Dynamic Path Planning Based on the Fusion Algorithm of Improved Ant Colony Optimization and Rolling Window Method," in IEEE Access, vol. 10, pp. 28322-28332, 2022.

[3] G. Tang, C. Tang, C. Claramunt, X. Hu and P. Zhou, "Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment," in IEEE Access, vol. 9, pp. 59196-59210, 2021.

[4] C. Wu, S. Zhou and L. Xiao, "Dynamic Path Planning Based on Improved Ant Colony Algorithm in Traffic Congestion," in IEEE Access, vol. 8, pp. 180773-180783, 2020.

[5] M. Luo, X. Hou and J. Yang, "Surface Optimal Path Planning Using an Extended Dijkstra Algorithm," in IEEE Access, vol. 8, pp. 147827-147838, 2020.

[6] L. Wang, L. Liu, J. Qi and W. Peng, "Improved Quantum Particle Swarm Optimization Algorithm for Offline Path Planning in AUVs," in IEEE Access, vol. 8, pp. 143397-143411, 2020.

[7] M. Kobayashi and N. Motoi, "Local Path Planning: Dynamic Window Approach With Virtual Manipulators Considering Dynamic Obstacles," in IEEE Access, vol. 10, pp. 17018-17029, 2022.

[8] P. Wu, F. Gao and K. Li, "Humanlike Decision and Motion Planning for Expressway Lane Changing Based on Artificial Potential Field," in IEEE Access, vol. 10, pp. 4359-4373, 2022.

[9] Z. Liu, H. Liu, Z. Lu and Q. Zeng, "A Dynamic Fusion Pathfinding Algorithm Using Delaunay Triangulation and Improved A-Star for Mobile Robots," in IEEE Access, vol. 9, pp. 20602-20621, 2021.

[10] Z. Xu, X. Liu and Q. Chen, "Application of Improved Astar Algorithm in Global Path Planning of Unmanned Vehicles," 2019 Chinese Automation Congress (CAC), 2019, pp. 2075-2080.

[11] H. Pan, C. Guo and Z. Wang, "Research for path planning based on improved astart algorithm," 2017 4th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS), 2017, pp. 225-230.