ITC 3/52

Information Technology and Control

Vol. 52 / No. 3 / 2023

pp. 594-604

DOI 10.5755/j01.itc.52.3.32791

Robot Path Planning Research Incorporating Improved A* Algorithm and DWA Algorithm

Received 2022/11/17 | Accepted after revision 2022/12/23

HOW TO CITE: Qu, S., Feng, G., Jiang, Y., Han, C., Hu, D., Liang, H. (2023). Robot Path Planning Research Incorporating Improved A* Algorithm and DWA Algorithm. *Information Technology and Control*, 52(3), 594-604. https://doi.org/10.5755/j01.itc.52.3.32791

Robot Path Planning Research Incorporating Improved A* Algorithm and DWA Algorithm

Shiya Qu, Guang Feng, Yuhang Jiang, Chunyu Han, Dingyuan Hu, Hongbin Liang

School of Mechanical Engineering and Automation, University of Science and Technology Liaoning, Anshan114051, China

Corresponding author: lianghb@hit.edu.cn

For the traditional A* algorithm has problems such as long paths, large number of nodes, and the demand for dynamic obstacle cannot be avoided in complex environment. A mobile robot dynamic path avoidance method will be improved to improve the A* algorithm and improve DWA algorithm Two map environments are used for simulation verification. First, the evaluation function and key node selection strategy are optimized for the A* algorithm, and redundant nodes are deleted; then the dynamic obstacle distance evaluation function is added to the DWA algorithm which for the purpose of the obstacle avoidance performance can be enhanced. The results about the improved A* algorithm reduces 12.20% and 58.33% in path length and number of turning points respectively compared with the traditional A* algorithm can be obviously grasped by the simulation experiment; by using the fusion algorithm whose purpose of using arcs instead of the straight lines is to turn more smoothly, and can be closest to the global optimum while avoiding dynamic obstacles to complete the search.

KEYWORDS: Improved evaluation function; key point selection strategy; dynamic obstacles; trajectory optimization.

1. Introduction

In the area of research on multi-robot autonomous navigation, path planning occupies a significant place. For illustration, the Dynamic windowing approach (DWA), artificial potential field method, genetic algorithm, etc. Based on local path planning, greedy algorithm, A* algorithm, Dijkstra algorithm, D* algorithm (a variant of Dijkstra algorithm), etc. [1–5]. Based on global planning. Numerous clever algorithms exist as well, such as the Tenebrae whisker and Ant colony algorithms [6]. A* algorithm is the most widely used because of its simplicity, high efficiency, fast response, and high accuracy. Bai et al. [1] introduced the artificial potential field method into the A* algorithm, the sharp nodes in the path are smoothed by three uniform B spline curves and the tracking control strategy is designed based on the model predictive control theory to achieve accurate tracking of the planned path [7]. Liu et al. [11] used the jump point search method to optimize the A* algorithm, which integrated the DWA algorithm to enhance path smoothness and avoid unknown dynamic obstacles, but the path length is still too long. In Erke et al. [4], to strengthen the method's stability, a variable step size A* algorithm was recommended. Sa et al. [16] introduced the cosine function and added the exponential distance adjustment factor and the information of the node heuristic function of the parent node to the heuristic function and designed a rotation prediction method to optimize the bidirectional smoothing of the path.

The dynamic window algorithm is applied considering the situation where there is an unknown obstacle when the robot is work in [7]. This algorithm for path planning can improve the obstacle avoidance ability of the robot, and the path is smoother [9-10]. However, it is easy to fall into the optimal local solution and cannot reach the specified goal along the global optimum. Niu et al. [14] Improved evaluation function of A* combined with the DWA algorithm, the car could perform dynamic obstacle avoidance. However, the path is still too long. Liu et al. [9] Smoother trajectories and more timely collision avoidance by introducing evaluation factors on orientation change into the DWA algorithm path evaluation function.

For the issues of poor planning efficiency, numerous redundancy points, and turns in the traditional A* and the Dynamic windowing approach algorithms, which can lead to local optimum and lengthy paths. In practice, due to the complexity and variability of the scenario, it is difficult to meet the requirements of real-time and completeness by using global path planning or local path planning alone. Therefore, a better path planning result can be achieved by improving and integrating the two algorithms [10]. This paper proposes to optimize the two algorithms and fuse them so that the obtained path can ensure global optimization and avoid random dynamic obstacles.

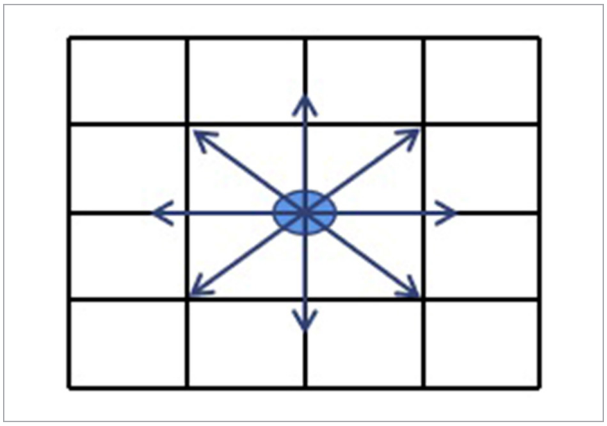
2. Global Planning

2.1. Environment Description

The 2D map environment described in this paper uses the raster method, and the search direction chosen is an 8-neighborhood search, in which the algorithm searches the eight neighboring nodes of the current node each time, as shown in Figure 1.

Figure 1

Node search direction



2.2. Traditional A* Algorithm

The A* algorithm is calculated by evaluating the function selected to reach the endpoint by searching the surrounding fields starting point [9, 12, 13]. The conventional A* algorithm evaluation function formula is:

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

where: n means the current node; $F(n)$ denotes the evaluation function from the starting point to the target point bypass the node; $G(n)$ denotes the actual cost of reaching the node from the starting point; $H(n)$ is the estimated value, which denotes the estimated cost from the node to the target point, called heuristic function too. The distance between any two points is calculated using the Euclidean distance formula because the resulting estimate is the smallest [14], which is given by:

$$H(n)=\sqrt{(x_n-x_t)^2+(y_n-y_t)^2}. \tag{2}$$

2.3. Improving the A* Algorithm

2.3.1. Optimization Heuristic Function

The A* algorithm is a heuristic search algorithm [13] whose performance is primarily directly affected by heuristic functions.

When the estimated cost of the heuristic function< the actual cost, there are many search nodes, and the computation is extensive, but the optimal path can be obtained; when the estimated cost of the heuristic function > the actual cost, there are few search nodes, and the optimal path cannot be obtained; when the estimated cost of the heuristic function= the actual cost, the search performance is optimal. In order to prevent the estimation value is immense and falling into the local optimum, the weight α of should be appropriately adjusted. In summary, this paper designs a heuristic function:

$$\alpha = 1 - \left(\frac{d}{l} + \frac{l}{L} \right) \tag{3}$$

$$F(n) = G(n) + \left[1 - \ln\left(\frac{d}{l} + \frac{l}{L} \right) \right] H(n), \tag{4}$$

where, d is the distance between the current node and the starting place; l is the distance between the current node and the destination; L is the distance between the starting place and the destination.

2.3.2. Key Node Selection Strategy

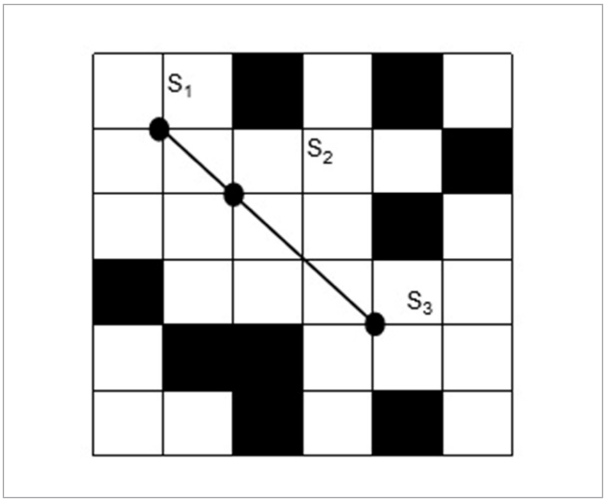
To address the problem, the paths planned by the A* algorithm have more redundant points and turning points, which can lead to lower efficiency of the robot while working. In this paper, an optimized key node selection strategy is designed to remove redundant points and improve the search efficiency of the path [12]. The specific steps of key node selection are as follows:

Let the set of path nodes be $U = \{S_i, 1 \le i \le n\}$, and the position of each node is known. where S_1 is the starting point and S_n is the end point [10]. Set $S_1(x_1, y_1)$, $S_2(x_2, y_2)$, $S_3(x_3, y_3)$, ... $S_n(x_n, y_n)$. From the starting point, solve for the slope $k_{S_1S_2} = \frac{(y_2 - y_1)}{(x_2 - x_1)}$, $k_{S_1S_3} = \frac{(y_3 - y_1)}{(x_3 - x_1)}$ of the line and the line S_1S_3, then determine the magnitude of the slope of the two lines. Discuss the following two cases:

If the slope $k_{S_1S_2}$ is equal to $k_{S_1S_3}$, then it means that S_1, S_2, S_3 three points are on the same line and the S_2 is a redundant common line node, delete the node, as shown in Figure 2.

Figure 2

Key points of the common line

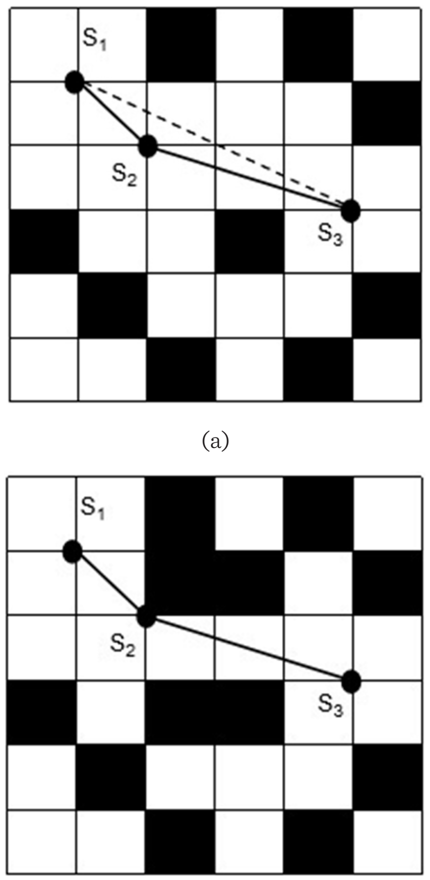


If the slope and are not equal, it means that S_1, S_2, S_3 three points are not on the same line, solve the equation of the line, determine whether there is an obstacle within a predetermined safe distance between the line and the obstacle, if there is no obstacle, it is determined to be a redundant turning point, delete, as shown in Figure 3a; if there is an obstacle, it is a necessary turning point, need to be retained, as shown in Figure 3b.

Next, the above steps are repeated from the beginning, and the set of nodes on the path after eliminating the redundant nodes must be updated continuously until all nodes are finished operating according to the above steps. Eliminating redundant points using the key node selection strategy makes the ideal path identified by the enhanced A* algorithm planning contain only the starting point, necessary nodes, and target points, effectively reducing the robot's path length. The procedure of the improved A* algorithm is given in Algorithm 1.

Figure 3

Key points do not share a line. (a) Point can be removed. (b) point must be retained



(a)

(b)

Algorithm 1

Algorithm: Improving the A*algorithm

```
Initialize
Open list $\longleftarrow$ start node
Closed list $\longleftarrow$ 0
While (current node ~= target node)
current =node in Open list with the smallest f(n);
remove current from Open list;
add current to Closed list;
If
current node== target node;
return success;
for each node that n is adjacent to current do;
If
g(n) > g(current)+ cost of edge from n to current);
g(n) = g(current)+ cost of edge from n to current;
f(n)=g(n) + a*h(n);
n. parent =current;
add n to Closed list;
If
Open list = Null;
return No-Path;
The optimal path from start to goal is obtained.
```

3. Local Planning

The DWA algorithm belongs to the category of robot local path planning. The dynamic window method samples multiple sets of velocities in a velocity vector space consisting of velocities and accelerations [11], taking into account the constraints of velocity and acceleration/deceleration performance. Then, the motion model of the robot is used to simulate the trajectory of these speeds over a while. Then the obtained trajectory is evaluated according to the evaluation index [11]. The trajectory with the highest score is finally selected, and the corresponding velocities and accelerations are used as the travel speed parameters of the robot.

3.1. Robot Motion Model

It is supposed that the linear velocity of the mobile robot and its angular velocity at the time are denoted as $v(t)$, $w(t)$, and the heading angle of the robot is denoted as $\theta(t)$. The sampling period is Δt, then the mathematical expression of the motion model of the robot is [22]:

$$\begin{cases} x(t) = x(t-1) + v(t)\Delta t\cos(\theta(t-1)) \\ y(t) = y(t-1) + v(t)\Delta t\sin(\theta(t-1)) \\ \theta(t) = \theta(t-1) + w(t)\Delta t \end{cases} . \tag{5}$$

3.2. Speed Sampling

In the two-dimensional space of velocities (v, w), there are a large number of velocity groups [14], but the robot is constrained in its motion.

1 own maximum velocity, minimum velocity constraints [7]:

$$V_x = \left\{(v,w) \middle| v \in [v\max_{\min}], w \in [w\max_{\min}]\right\}. \qquad (6)$$

2 Constraints on acceleration and deceleration:

The acceleration of the mobile robot is limited by the output torque of the motor [15], and there exists a maximum acceleration and deceleration, and the actual speed achieved by the robot V_d is

$$V_d = \left\{ (v,w) \middle| \begin{array}{l} v = v_t + a_v \Delta t, w = w_t + a_w \Delta t, \\ a_v \in \left[a_{vd\max}, a_{vi\max}\right], \\ a_w \in \left[a_{wd\max}, a_{wi\max}\right] \end{array} \right\}, \qquad (7)$$

where, v_t is the linear velocity at the moment; is the angular velocity at the moment; $a_{vd\max}$ denotes the maximum deceleration in the linear velocity [16-17]; $a_{vi\max}$ denotes the maximum acceleration in the linear velocity; $a_{wd\max}$ denotes the maximum deceleration in the angular velocity; and $a_{wi\max}$ is the maximum acceleration in the angular velocity.

3 Binding of braking distance.

Braking distance is a certain distance before hitting an obstacle to start decelerating the final speed to 0 to ensure it will not hit the obstacle [18]. The constraint formula is:

$$V_q = \left\{ (v,w) \middle| \begin{array}{l} v \le \sqrt{2dist(v_t, w_t) a_{vd\max}}, \\ w \le \sqrt{2dist(v_t, w_t) a_{wd\max}} \end{array} \right\}, \qquad (8)$$

where, $dist(v_t, w_t)$ denotes the distance between the robot and the nearest obstacle at the speed (v_t, w_t).

The speed of the final DWA algorithm mobile robot is the intersection of the above three constraints, so the DWA algorithm speed satisfies V_W the equation:

$$V_W = V_x \cap V_d \cap V_q. \qquad (9)$$

3.3. Traditional Evaluation Function of DWA Algorithm

After obtaining the velocity group through the above velocity sampling and simulating multiple trajectories according to the robot motion model, the optimal path still needs to be selected based on a comprehensive evaluation function. The evaluation function of the conventional DWA algorithm includes three aspects: the robot's orientation, safety distance, and velocity [13], and the formula of the specific evaluation function is:

$$P(v,w) = \alpha \cdot Head(v,w) + \beta \cdot Dist(v,w) + \gamma \cdot \mathrm{Vel(v,w)}_{,} \qquad (10)$$

where, α, β, γ are the weighting coefficients, $Head(v,w)$ is the azimuth evaluation function $Head(v,w) = 180 - \theta$; $Dist(v,w)$ is the distance evaluation function; $Vel(v,w)$ is the speed evaluation function [19].

3.4. Optimization of the Evaluation Function

For the traditional DWA algorithm, obstacle avoidance will fall into the local optimum, and there will be static obstacles and unknown dynamic obstacles that interfere with each other, resulting in the decline of dynamic obstacle avoidance and other problems [20]. The evaluation function is divided into and denotes the minimum distance between the simulated target point and the known static obstacle and the unknown dynamic obstacle, respectively.

Let the coordinates of the static obstacles are (x_s, y_s) and the coordinates of the predicted trajectory point at the time is (x_t, y_t). The minimum distance between the simulated trajectory point and the static obstacle at the time is D_s. R is the radius of the bottom wheel of the robot, but in practice its not necessarily a true circle, so expand to $1.2R$ on the basis [17]. The formula of the function $Dist\_S(v,w)$ is:

$$D_s = \sqrt{(x_t - x_s)^2 + (y_t - y_s)^2} \qquad (11)$$

$$Dist\_S(v,w) = \begin{cases} \dfrac{1}{D_s}, D_s \ge 1.2R \\ 0, D_s < 1.2R \end{cases}. \qquad (12)$$

Let the coordinates of the unknown dynamic obstacle be (x_d, y_d), and the coordinates of the trajectory point predicted by the algorithm at the time are (x_t, y_t).The closest distance between the simulated trajectory point and dynamic obstacle at the moment is D_d [8]. V_t is the velocity of the robot at the moment of time [25]. L is the motion path of the mobile robot in time Δt , and considering that the robot will appear to change acceleration and deceleration in practical applications, so expand to $1.2L$ on the basis [22]. The formula of the function $Dist\_D(v,w)$ is:

$$L = V_t * \Delta t \tag{13}$$

$$D_d = \sqrt{(x_t - x_d)^2 + (y_t - y_d)^2} \tag{14}$$

$$Dist\_D(v,w) = \begin{cases} \frac{1}{D_d}, D_d \geq 1.2L \\ 0, D_d < 1.2L \end{cases} \tag{15}$$

In summary, the improved DWA evaluation function is [10]:

$$\begin{aligned} P(v,w) &= \alpha \cdot Head(v,w) + \beta \cdot DistS(v,w) \\ &+ \gamma \cdot Vel(v,w) + \eta \cdot Dist\_D(v,w) \end{aligned} \tag{16}$$

where, η is the weighting coefficient the $P(v,w)$ represents the maximum speed of the robot at a guaranteed safe distance from the obstacle [26]. The procedure of the improved dynamic window approach algorithm is given in Algorithm 2.

Algorithm 2

Algorithm: DWA algorithm

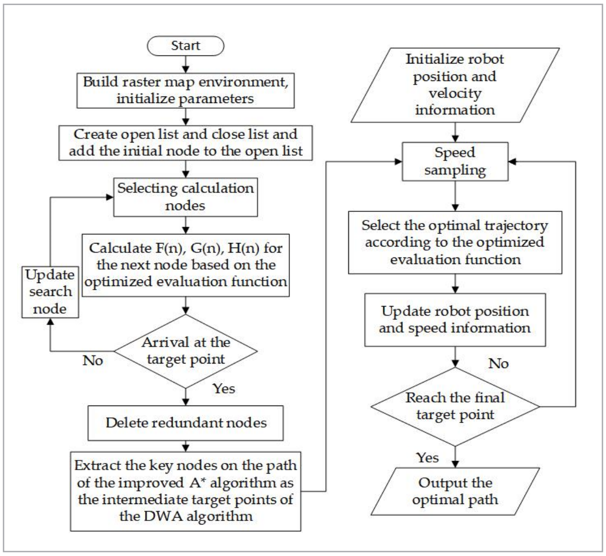
```
Initialization;
Obtain the kinematics model of the robot;
While not reach the target location
{
    Speed sampling of robot
    Simulate motion trajectories;
    Use the evaluation function improved by the global [11] ;
    path information obtained by the improving the A* algorithm to;
    select the optimal trajectory;
    Robot follows the optimal trajectory to move;
}
Get the optimal path;
```

4. Algorithm Fusion

In this paper, the traditional A* algorithm and DWA algorithm are improved, respectively. However, it is still evident that there are twists and turns in the improved A* algorithm path. The improved A* algorithm cannot avoid obstacles in time for the unknown dynamic obstacles in the complex environment [18]. The improved DWA algorithm is still prone to the optimum local situation in a complex environment. According to the above problems, it is proposed to fuse the two improved algorithms, i.e., to extract the key points on the path of the improved A* algorithm as the intermediate target points of the DWA algorithm for path planning. The process is shown in Figure 4.

Figure 4

Algorithm flow chart


Start
Build raster map environment, initialize parameters
Create open list and close list and add the initial node to the open list
Selecting calculation nodes
Calculate F(n), G(n), H(n) for the next node based on the optimized evaluation function
Update search node
Arrival at the target point
No
Yes
Delete redundant nodes
Extract the key nodes on the path of the improved A* algorithm as the intermediate target points of the DWA algorithm
Initialize robot position and velocity information
Speed sampling
Select the optimal trajectory according to the optimized evaluation function
Update robot position and speed information
Reach the final target point
No
Yes
Output the optimal path

5. Simulation Experiments

5.1. Simulation Experimental Environment

To verify the optimisation of the algorithm proposed in this paper.MATLAB2016a experimental platform was used to simulate and verify the traditional A* algorithm, the improved A* algorithm, and the fusion algorithm [19]. The experimental environment was run on a CPU with 8GB memory and a 64-bit WIN10 system, and two different 2D raster map environments were designed for simulation. In the 21×21 simple environment map 1, the starting point S (5,16) and the ending point T (19,3); in the 21×21 maze environment map 2, the starting point S (2,20) and the ending point T (19,3). The coefficients involved in the improved DWA algorithm are =0.2, =0.1, =0.3, and =0.4. The motion parameters of the simulated experimental robot are shown in Table 1.

Table 1

Values of robot motion parameters

Parameters	Value
Initial line speed	0m/s
Max. linear speed	2m/s
Max. linear acceleration	0.4 m/s^2
Initial azimuth angle	-π/2
Angular speed fractional variability	1°/s
Initial angular velocity	0 °/s
Max. angular velocity	40 °/s
Max. angular acceleration	60 °/s^2
Linear speed resolution	0.01m/s
Δt	0.1s
Initial angular velocity	0 °/s

5.2. Improved A* Algorithm Simulation Experiments

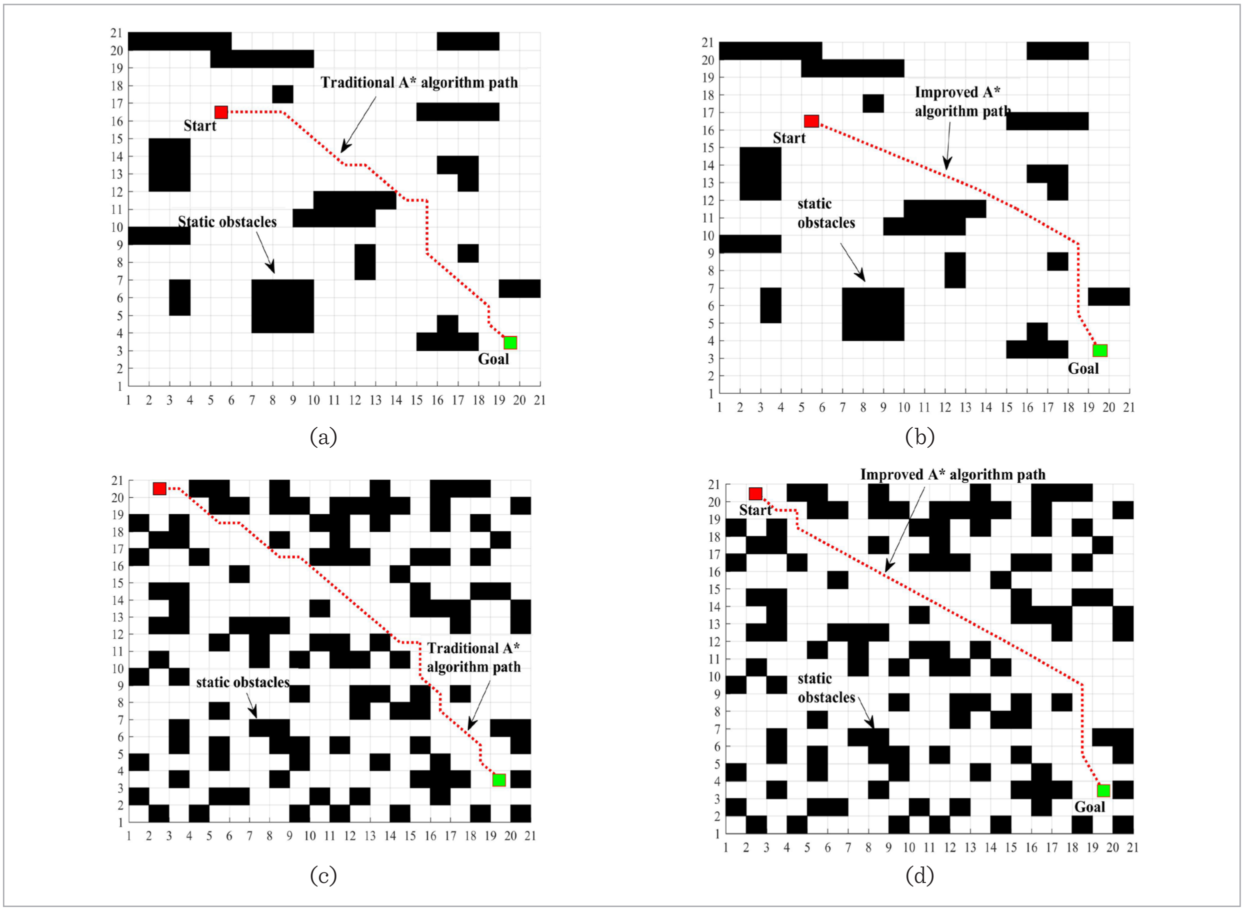
In two raster map contexts, the traditional A* method and the enhanced A* algorithm are simulated, as shown in Figure 5. The simulation data and performance comparison of the algorithms are shown in Table 2.

From Figure 5 and Table 2, we can see that the global paths planned by the improved A* algorithm are better than the traditional A* algorithm, mainly optimizing the number of turning points and path length. In the simple environment map 1, the improved A* algorithm turning points are optimized by 75% on average over the traditional A* algorithm, moreover, the route length of the enhanced A* algorithm is 15.48% shorter than that of the traditional A* algorithm [25].

For the complex maze environment Map 2, the number of turning points and path length of the [2] im-

Figure 5

Comparison of two kinds of map simulation experiments. (a) Map 1 traditional A* algorithm. (b) Map 1 improved A* algorithm. (c) Map 2 traditional A* algorithm. (d) Map 2 improved A* algorithm



(a)

(b)

(c)

(d)

Table 2

Experimental results of two kinds of maps

Map	Algorithm	Turning point number/pc	Turning Point optimization ratio	Path length /m	Path length optimization ratio
Map 1	Traditional A* algorithm	8	–	24.4879	–
	Improving the A*algorithm	2	75%	21.0280	15.48%
Map 2	Traditional A* algorithm	12	–	30.3848	–
	Improving the A*algorithm	5	58.33%	25.1659	17.18%

proved algorithm is optimized on average 58.33% and 17.18%, respectively, compared to the traditional algorithm. In addition, the path planned by the conventional A* algorithm is closer to the obstacle [23], and the improved algorithm path is safer. Therefore, it can be shown that applying the revised approach presented in this work shortens the path and improves its smoothness and safety.

5.3. Simulation Experiment of the Fusion Algorithm

The fusion algorithm is based on the improved A* algorithm [25], adding one unknown dynamic obstacle and two unknown dynamic obstacles in both map environments for experimental comparison, as show in Figures 6-7. The path velocity variation curve with the bit pose is shown in Figure 8.

Figure 6

Map 1 fusion algorithm dynamic obstacle avoidance path simulation results. (a) Obstacles being avoided. (b) Single dynamic obstacle. (c) Second dynamic obstacle being avoided. (d) Dual dynamic obstacles

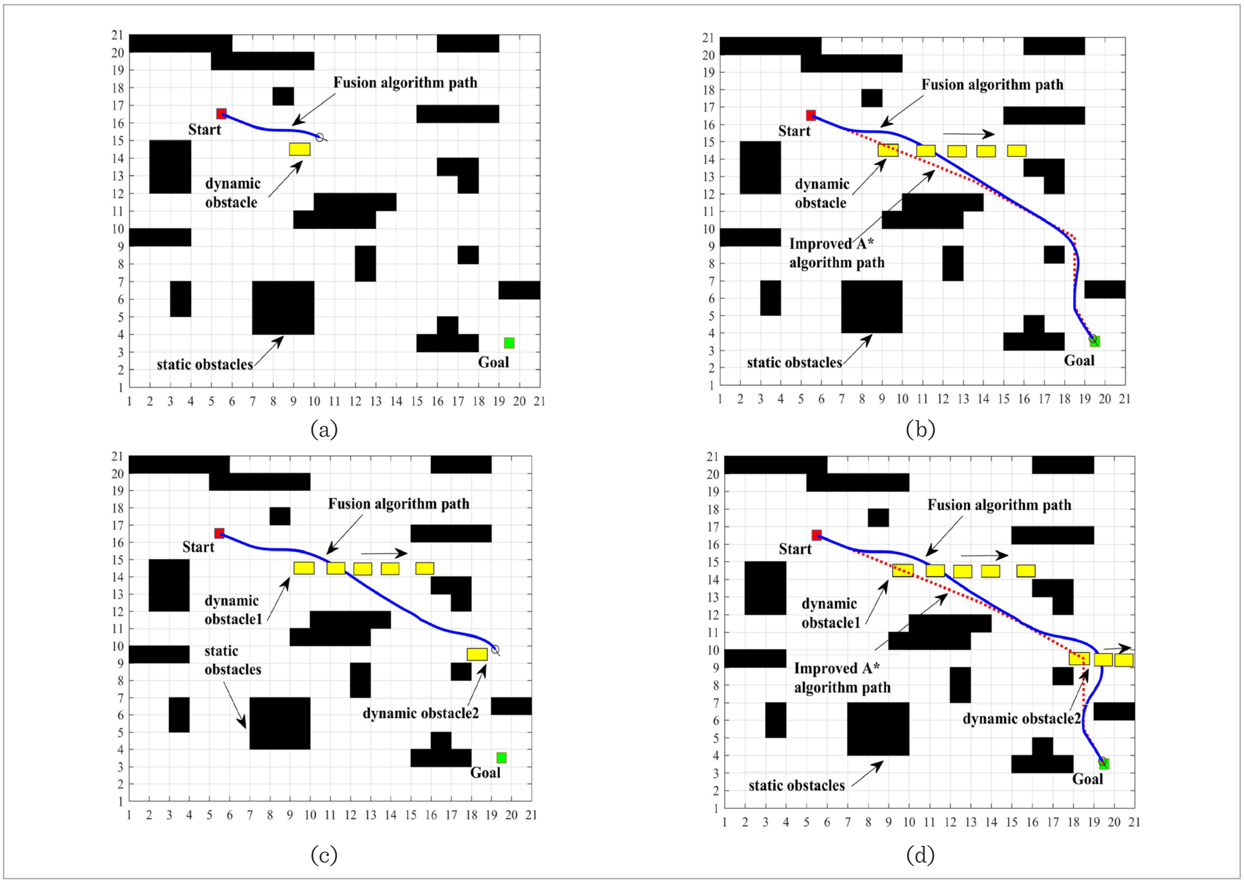


Figure 7

Map 2 fusion algorithm dynamic obstacle avoidance path simulation results.(a) Positive avoidance of a single obstacle.(b) Single dynamic obstacle.(c) Second dynamic obstacle being avoided.(d) Dual dynamic obstacles

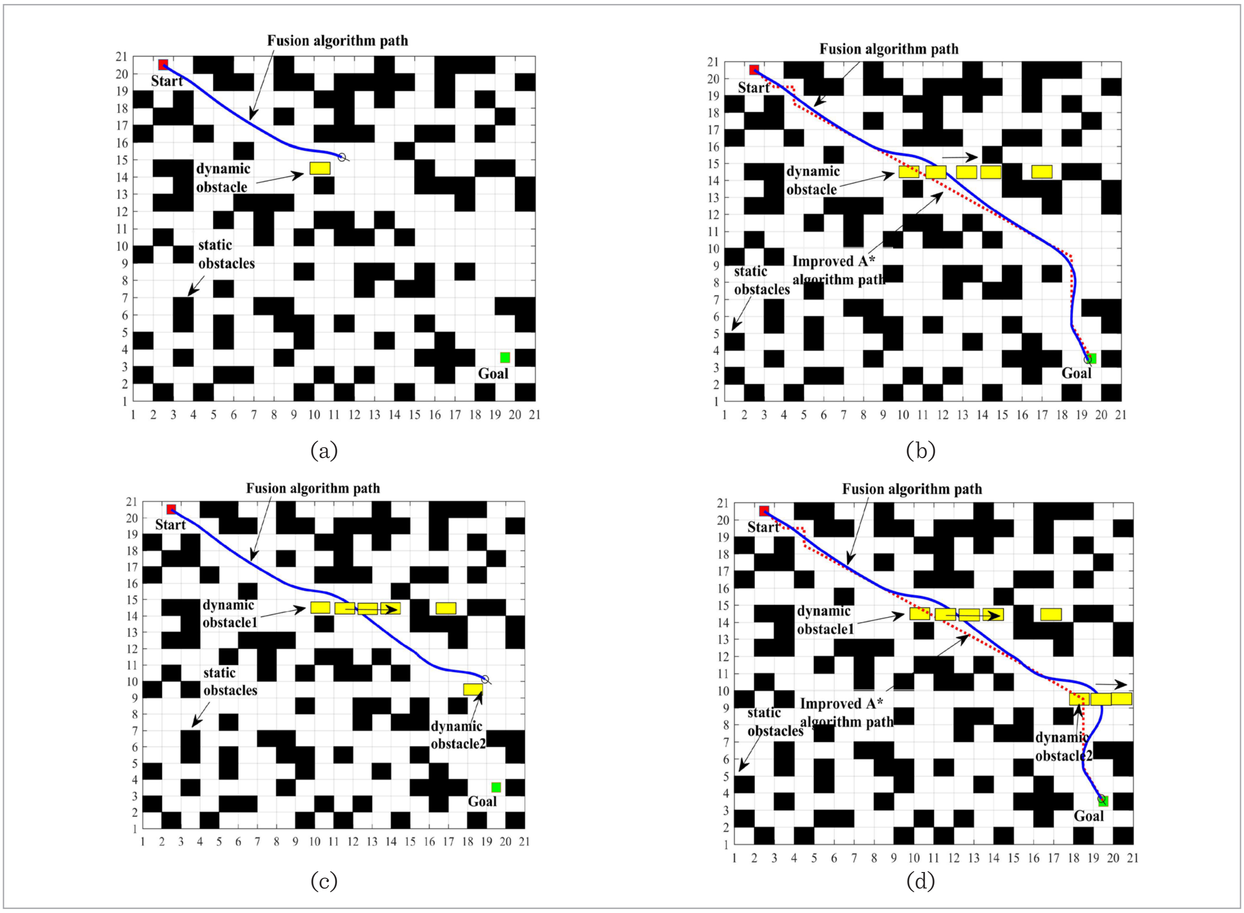
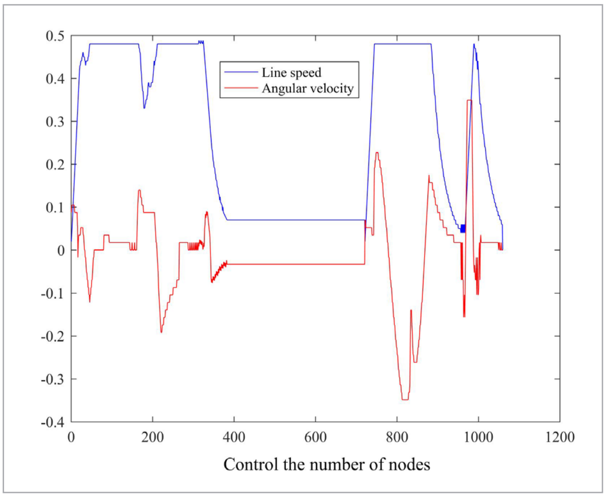


Figure 8

Changes in robot line speed



It proves obvious that in two maps of simple and complex environments with one and two random dynamic obstacles, respectively, the fused algorithms in Figure 6(a)-(c) and Figure 7(a)-(c) are avoiding the obstacles, the obstacles in Figure 6(b)-(d) and Figure 7(c)-(d) move from left to right, and the improved A* algorithm collide with the obstacles, which the fused algorithms can avoid perfectly and successfully reach the target point [23].

As can be seen from Table 3, the path lengths planned by the fusion algorithm [21] are 21.0337m and 21.5279m in the simple environment of Map 1, which are 15.46% and 13.47% less than the traditional A* algorithm path lengths, respectively. The path lengths planned by the fusion algorithm are 26.1942m and 26.6783m in the complex environment of Map 2, which are 13.79% and 12.20% optimized. 26.6783m, optimized by 13.79% and

Table 3

Fusion algorithm path data of maps

Map	Number of random dynamic obstacles	Path length /m	Path length optimization ratio (traditional A* algorithm)	Avoid dynamic obstacles
Map1	one	21.0337	15.46%	Yes
	two	21.5279	13.47%	Yes
Map2	one	26.1942	13.79%	Yes
	two	26.6783	12.20%	Yes

12.20%. As shown in Figure 8, when the robot motion is performed with two dynamic random obstacles avoidance that the number of control nodes ranges from 400-600, there is a significant decrease in the linear and angular velocities of the robot, which will increase the algorithm search time. However, over all the improved fusion of the two algorithms is good at avoiding the random dynamic obstacles and obtaining the global optimal route during path planning. The achievement of a more effective global path planning for real-time obstacle avoidance [24].

6. Conclusion

In this paper, an improved method is proposed for the traditional A* algorithm and DWA algorithm, and the fusion method is designed and the calculation flow of the algorithm is developed. By designing two map environments, the traditional A* algorithm and the improved A* algorithm are simulated experimentally. The improved fusion algorithm is also simulated by adding a different number of dynamic obstacles. The simulated experimental data shows that the improved A* algorithm and the fused algorithm optimize the path length by 15.48%, 15.46% (map1), 17.18%, 13.79% (map2) turning points by 75% (map1), 58.33% (map2) compared to the traditional path length. In addition, the fused improved algorithm can avoid dynamic obstacles completely. Experiments verify the superiority and timeliness of the improved A* algorithm and DWA algorithm, and the obtained path reaches the end point safely while the length is shorter and the number of turning points is reduced, which achieves the expected effect. It can fuse algorithms to optimize robot path length, safety performance and smoothness to meet the optimization of robots in complex dynamic environments, and fuse algorithms to optimize robot path length, safety performance and smoothness to meet the requirements of robot path planning in complex dynamic environments.

Acknowledgement

The authors acknowledge the Department of Science and Technology of Liaoning Province, Natural Foundation Project (Grant: 2015020128).

References

1. Bai, Y., Li, G., Li, N. Motion Planning and Tracking Control of Autonomous Vehicle Based on Improved A* Algorithm. Journal of Advanced Transportation, 2022, 2022, 1-14. https://doi.org/10.1155/2022/1675736
2. Chang, L., Shan, L., Li, J., Dai, Y. The Path Planning of Mobile Robots Based on an Improved A Algorithm. 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC). Banff, AB, Canada, 2019, 257-62. https://doi.org/10.1109/ICNSC.2019.8743249
3. Chen, Y., Wang, P., Lin, Z., Sun, C. Global Path Planning Method by Fusion of A-star Algorithm and Sparrow Search Algorithm. 2022 IEEE 11th Data Driven Control and Learn ing Systems Conference (DDCLS), 2022, 205-9. https://doi.org/10.1109/DDCLS55054.2022.9858435
4. Erke, S., Bin, D., Yiming, N., Qi, Z., Liang, X., Dawei, Z. An Improved A-Star Based Path Planning Algorithm for Autonomous Land Vehicles. International Journal of Advanced Robotic Systems, 2020, 17(5), 172988142096226. https://doi.org/10.1177/1729881420962263
5. Guo, Y.-Q., Zha, X.-M., Shen, Y.-Y., Wang, Y.-N., Chen, G. Research on PID Position Control of a Hydraulic Servo System Based on Kalman Genetic Optimization. Actuators, 2022, 11(6), 162. https://doi.org/10.3390/act11060162
6. Ji, X., Feng, S., Han, Q., Yin, H., Yu, S. Improvement and Fusion of A* Algorithm and Dynamic Window Approach Considering Complex Environmental Information. Arabian Journal for Science and Engineering, 2021, 46(8), 7445-59. https://doi.org/10.1007/s13369-021-05445-6
7. Katoch, S., Chauhan, S. S., Kumar, V. A Review on Genetic Algorithm: Past, Present, and Future. Multimedia Tools and Applications, 2021, 80(5), 8091-126. https://doi.org/10.1007/s11042-020-10139-6

8. Li, C., Zhang, D. A Global Dynamic Path Planning Algorithm Based on Optimized A Algorithm and Improved Dynamic Window Method. 2021 33rd Chinese Control and Decision Conference (CCDC), 2021, 7515-9. https://doi.org/10.1109/CCDC52312.2021.9601816

9. Li, X., Li, Q., Zhang, J. Research on Global Path Planning of Unmanned Vehicles Based on Improved Ant Colony Algorithm in the Complex Road Environment. Measurement and Control, 2022, 55(9-10), 945-59. https://doi.org/10.1177/00202940221118132

10. Li, Y., Jin, R., Xu, X., et al. A Mobile Robot Path Planning Algorithm Based on Improved A* Algorithm and Dynamic Window Approach. IEEE Access, 2022, 10, 57736-47. https://doi.org/10.1109/ACCESS.2022.3179397

11. Liu, L., Yao, J., He, D., et al. Global Dynamic Path Planning Fusion Algorithm Combining Jump-A* Algorithm and Dynamic Window Approach. IEEE Access, 2021, 9, 19632-8. https://doi.org/10.1109/ACCESS.2021.3052865

12. Liu, S., Ma, Y. Research for Bidirectional Path Planning Based on an Improved A* Algorithm. 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA). Dalian, China, 2020; 1036-9. https://doi.org/10.1109/AEECA49918.2020.9213692

13. Liu, Y., Sun, R., Zhang, T., Zhang, X., Li, L., Shi, G. Warehouse-Oriented Optimal Path Planning for Autonomous Mobile Fire-Fighting Robots. Security and Communication Networks, 2020, 2020, 1-13. https://doi.org/10.1155/2020/6371814

14. Niu, C., Li, A., Huang, X., Li, W., Xu, C. Research on Global Dynamic Path Planning Method Based on Improved A* Algorithm. Mathematical Problems in Engineering, 2021, 1-13. https://doi.org/10.1155/2021/4977041

15. Park, S.-O., Lee, M. C., Kim, J. Trajectory Planning with Collision Avoidance for Redundant Robots Using Jacobian and Artificial Potential Field-based Real-time Inverse Kinematics. International Journal of Control, Automation and Systems, 2020, 18(8), 2095-107. https://doi.org/10.1007/s12555-019-0076-7

16. Sa, X., Huaiyu, W., Zhihuan, C. Research of Mobile Robot Path Planning Based on Improved A* Algorithm. 2020 Chinese Automation Congress (CAC). Shanghai, China; IEEE;, 2020; 7619-23. https://doi.org/10.1109/CAC51589.2020.9327676

17. Song, X., Gao, S., Chen, C. B., Cao, K., Huang, J. A New Hybrid Method in Global Dynamic Path Planning of Mobile Robot. International Journal of Computers Communications & Control, 2018, 13(6), 1032-46. https://doi.org/10.15837/ijccc.2018.6.3153

18. Sun, T., Wang, T., Sun, P. Mobile Robot Dynamic Path Planning Based on Improved A* Algorithm. 2021 3rd International Conference on Robotics and Computer Vision (ICRCV), 2021; 24-9. https://doi.org/10.1109/ICRCV52986.2021.9546972

19. Xia, C., Meijun, L., Peng, L., Yaxi, W., Jie, L., Lan, Z. Research on Global Path Planning Based on Improved A* Algorithm. 2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)., 2022; 428-33. https://doi.org/10.1109/AEECA55500.2022.9918827

20. Yuan, X., Yuan, X., Wang, X. Path Planning for Mobile Robot Based on Improved Bat Algorithm. Sensors, 2021, 21(13), 4389. https://doi.org/10.3390/s21134389

21. Zhang, B., Zhang, J. Robot Mapping and Navigation System Based on Multi-Sensor Fusion. 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD), 2021, 632-6. https://doi.org/10.1109/ICAIBD51990.2021.9459053

22. Zhang, J. Hua, Feng, Q., Zhao, A. Di, He, W., Hao, X. Local Path Planning of Mobile Robot Based on Self-Adaptive Dynamic Window Approach. Journal of Physics: Conference Series, 2021, 1905(1), 012019. https://doi.org/10.1088/1742-6596/1905/1/012019

23. Zhang, J., Wu, J., Shen, X., Li, Y. Autonomous Land Vehicle Path Planning Algorithm Based on Improved Heuristic Function of A-Star. International Journal of Advanced Robotic Systems, 2021, 18(5), 172988142110427. https://doi.org/10.1177/17298814211042730

24. Zhang, S., Xu, M., Wang, X. Research on Obstacle Avoidance Algorithm of Multi-UAV Consistent Formation Based on Improved Dynamic Window Approach. 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), 2022, 991-996. https://doi.org/10.1109/IPEC54454.2022.9777606

25. Zhang, Y.-H., Gong, Y.-J., Chen, W.-N., Gu, T.-L., Yuan, H.-Q., Zhang, J. A Dual-Colony Ant Algorithm for the Receiving and Shipping Door Assignments in Cross-Docks. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(7), 2523-39. https://doi.org/10.1109/TITS.2018.2867577

26. Zheng, Y. Multimachine Collaborative Path Planning Method Based on A* Mechanism Connection Depth Neural Network Model. IEEE Access, 2022, 10, 47141-53. https://doi.org/10.1109/ACCESS.2022.3168719