2021 IEEE 4th International Conference on Electronics Technology

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

Han Gao
Institute of Automation
Qilu University of Technology (Shandong Academy of Sciences)
Jinan, China
gaohan925@163.com

Zhengguang Ma
Institute of Automation
Qilu University of Technology (Shandong Academy of Sciences)
Jinan, China
Mazhg@sdas.org

Yongguo Zhao
Institute of Automationon
Qilu University of Technology (Shandong Academy of Sciences)
Jinan, China
sduzyg@sdas.org

2021 IEEE 4th International Conference on Electronics Technology (ICET) | 978-1-7281-7673-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICET51757.2021.9451017

***Abstract*—To solve the path planning problem of mobile robots in complex and dynamic environment, a dynamic path planning method is proposed in this paper. The dynamic path planning method is composed of the improved A* algorithm and the adaptive Dynamic Window Approach (DWA). By expanding the number of neighborhoods that can be explored, introducing a safety cost factor in the evaluation function, and optimizing the path point sequence, the safety and efficiency of the A* algorithm are improved. Based on the DWA algorithm, the weight of the evaluation function is adaptively adjusted to avoid being trapped in to a local optimal solution. To further improve the smoothness of the path and avoid obstacles in real time, the evaluation function with smoothing effect is constructed. The simulation results verify the effectiveness of the improved algorithm.**

Keywords—mobile robot, path planning, improved A* algorithm, adaptive dynamic window, dynamic environment

I. Introduction

Path planning is an important topic in the field of mobile robot research [1, 2]. According to whether the environment information is completely known, path planning can be divided into global path planning and local path planning [3, 4].

The environment modeling methods used in path planning include grid, visual graph, topology approach, etc. The grid method is most commonly used. Its essence is to divide the work environment into a certain number of grids of the same size, the grids are assigned different values to characterize different attributes [5]. The A* algorithm is a heuristic search algorithm, which uses direct-search method to expand the nodes with smallest cost value sequentially. Zhang H, et al. improved the security of the A* algorithm path by improving the environmental modeling method [6]. Lin M, et al. improved the efficiency of the A* algorithm by introducing the parent node, but it increases the probability of falling into the local optimal solution [7]. Shi Hui, et al. introduced two heuristic elements of distance and direction into the A* algorithm, which improves the efficiency of the algorithm [8]. The DWA algorithm is commonly used for local path planning due to it comprehensively considers the speed, direction, and distance of the robot. Seder, et al. proposed a method of predicting the trajectory of the collision point to avoid dynamic obstacles [9]. Cheng C, et al. proposed a hybrid algorithm based on the A* algorithm and DWA algorithm to avoiding falling into the local optimal solution [10]. Zhu Z, et al. realized global dynamic path planning by fusing A* algorithm and DWA algorithm, while this algorithm still cannot meet the robot motion performance [11].

For the global path planning problem, an improved algorithm is proposed to solve the limitations of the traditional A* algorithm. For the local path planning problem, the situation that the robot falls into the local optimal solution is solved by adaptively adjusting the weight of the DWA algorithm evaluation function. Finally, the two algorithms are integrated to realize the purpose of dynamical path planning.

This paper is organized as follows. Section Ⅱ introduces the principle of the A* algorithm and the method of optimizing the algorithm. Section Ⅲ focuses on the traditional DWA algorithm and adaptive adjustment weights. Section Ⅳ describes an optimized fusion algorithm. The validation of the effectiveness of the fusion algorithm is described in Section Ⅴ. Section Ⅵ presents the final conclusion.

II. Improved A* Algorithm

A. Environmental Model Description

This article regards the mobile robot as a point which moves on a plane, the initial position is assumed to be set at (3, 3), the target position is to be set at (28, 20). The black grid represents the obstacle, the center of the grid represents the coordinates of the obstacle, and the blank area represents the barrier-free area. The environment model is shown in Fig. 1.

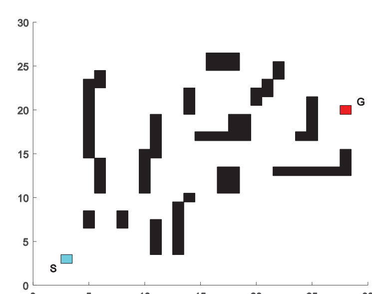


Fig. 1. The grid environment model

B. Traditional A* algorithm

The search direction of the A* algorithm is determined by the heuristic function. Starting from the starting node, the cost value of each node around it is calculated, and the node with the smallest cost is selected as the next path point. The valuation function is shown in (1):

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

$g(n)$ represents the actual cost from the starting node to the current node. $h(n)$ represents the heuristic cost from the current node to the target node. $f(n)$ represents the total costs. The Euclidean distance is used as the heuristic function. The heuristic function is shown in (2). (x_g, y_g) is the coordinates of the target node, (x_n, y_n) is the coordinates of the current node.

$$h(n) = \sqrt{(x_g - x_n)^2 + (y_g - y_n)^2} \tag{2}$$

C. Expanding the Direction of Exploration

In the case of densely distributed obstacles, the robot swing or stagnates due to the inability to turn continuously. In order to improve the path smoothness and reduce the steering capability of robot, this article will expand the number of exploration directions as shown in Fig. 2.

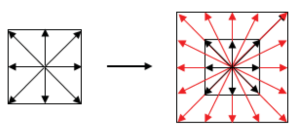
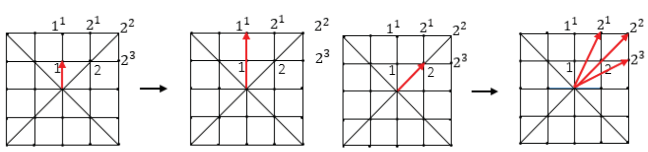


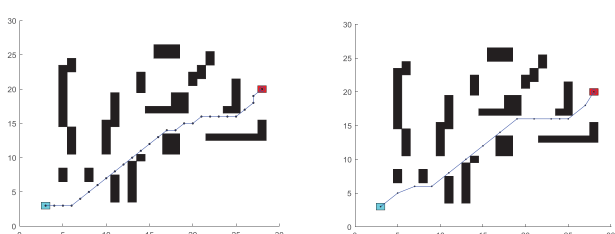
Fig. 2. Expanding exploration direction

Specific steps: If node 1 is free node, the next step is to explore node 1^1, otherwise, stop exploring, as shown in Fig. 3(a). If node 2 is free node, the next step is to explore nodes 2^1, 2^2, and 2^3 in turn, otherwise, stop exploring, as shown in Fig. 3(b).



(a) The vertical direction ((b) The slope direction

Fig. 3. Exploration steps of the improved A* algorithm



(a) The traditional A* algorithm (b) The improved A* algorithm

Fig. 4. Path planning with different algorithm

The improved algorithm is simulated with Fig. 1, the results are shown in Fig. 4 and Table I. It can be seen from Fig. 4 that the improved algorithm solves the problem that the robot makes continuous turns in a short distance. Combining with Table I, the turning points and turning angle of the improved algorithm are decreased by 33.33% and 53.33% respectively.

TABLE I. PERFORMANCE COMPARISON OF DIFFERENT A* ALGORITHM

Algorithm	Turning points		Turning angle		Path length
	number	decline /%	angle(°)	decline/%	
Traditional A*algorithm	9	---	405°	---	32.63
Improved A* algorithm	6	33.33%	189°	53.33%	32.27

D. Improving Valuation Function

In the A* algorithm, only the distance between the start node and the end node is considered, the distance between the free node and the surrounding obstacles is not considered. The safety costs of all nodes is considered to be equal for the robot. However, the safety cost of each point is related to the number of surrounding obstacles. In order to improve the safety of the path, this paper introduces the safety cost factor $s(n)$ into the evaluation function. The safety cost factor is calculated according to the number of obstacles near the node. The required safety cost is positively related to the number of surrounding obstacles. The improved evaluation function is shown in (3):

$$f(n) = a \cdot \mathrm{g(n)} + \mathrm{b} \cdot \mathrm{h(n)} + \mathrm{c} \cdot \mathrm{s(n)} \tag{3}$$

$s(n)$ is the safety cost of the current node. The improved A* algorithm is simulated with Fig. 1, the result is shown in Fig. 5. According to Fig. 5, it can be seen that when passing through a narrow passages, the problem that the robot travels close to one side of the passage is resolved, the path safety is improved.

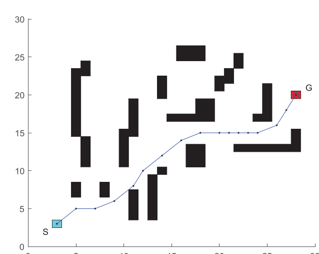


Fig. 5. Improved A* algorithm of valuation function

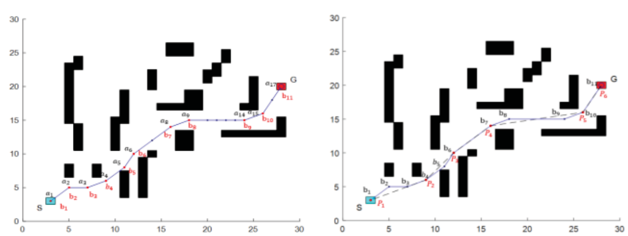
E. Optimizing the Path Point Sequence

There are a large number of redundant points in the path of the improved A* algorithm. This paper extracts key turning points and deletes redundant points to optimize the path. First, the safety threshold is set, and a path segment that two adjacent points in the path point sequence is constituted. The next step is to determine whether the distance between the obstacle and the path segment is greater than the safety threshold, if it is true, the path segment is considered as a safe path. The specific implementation steps are shown as follows:

- Assuming that the initial path point sequence is $A = \{a_i \mid i = 1, 2, ..., n\}$, as shown in Fig. 6(a).

- Extracting key turning points: start from a_2, if a_1, a_2, and a_3 are on the same straight line, then a_2 is a redundant point which need to be deleted, otherwise ,it is a key turning point which need to be reserved, update the path; traverse all path points in turn, and get the path point sequence $B=\{b_i \mid i=1,2,\ldots,s\},(s<n)$.
- Deleting redundant points: start from b_1, connect b_1b_3, judge whether b_1b_3 passes through the obstacle. If b_1b_3 does not pass through the obstacle, continue to connect b_1b_4. Repeat the above operation until $b_1b_i(i=3,4,\ldots,n)$ passes through the obstacle, delete the node between b_1 and b_{i-1}, update the path. Repeat the above operations until all nodes are traversed, and the path point $P=\{p_i \mid i=1,2,\ldots,t\},(t<s)$, as shown in Fig. 6(b)).

The improved algorithm is simulated with Fig. 1, and the results are shown in Fig. 7 and Table II. It can be seen from Table II that the path points are reduced by 64.7%, the number of key turning points is reduced by 55.6% after optimization. Comparing with Fig. 7, the smoothness of the path has been improved. The steering movement of the robot is reduced, and the operating efficiency is effectively improved.



(a) Extracting turning point (b) Deleting redundant points

Fig. 6. The algorithm of optimizing path point

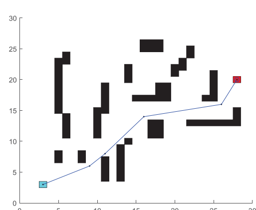


Fig. 7. The A* algorithm for optimizing path points

TABLE II. PERFORMANCE COMPARISON OF DIFFERENT A* ALGORITHM

Algorithm	Path point		Turning points		Path length
	number	decline	number	decline	
Traditional A* algorithm	17	---	9	---	32.7
Improved A* algorithm	6	64.7%	4	55.6%	32.0

III. IMPROVED ADAPTIVE DYNAMIC WINDOW APPROACH (DWA)

Obtaining real-time information about the surrounding environment by using the sensors, the robot can achieve the purpose of avoiding obstacles in real-time. The DWA algorithm mainly includes three parts: motion model, speed sampling and evaluation function.

a) The motion model of robot: the robot's motion trajectory is simulated by the motion model and a given speed, Assuming that the robot moves in a straight line at a constant speed within a given forward simulation time Δt, the pose of the robot at time t is (x_t, y_t, θ_t), then the pose of the robot at time $t+1$ is:

$$\begin{cases} x_{t+1} = x_t + v_x \Delta t \cos\theta_t - v_y \Delta t \sin\theta_t \\ y_{t+1} = y_t + v_x \Delta t \sin\theta_t + v_y \Delta t \cos\theta_t \\ \theta_{t+1} = \theta_t + w_t \Delta t \end{cases} \quad (4)$$

b) Speed sampling: the speed space is composed of the allowable speed (v,w). It is subject to the maximum and minimum speed constraints respectively.

- The robot is restricted by its own hardware, considering the maximum and minimum speed, the speed range of the robot is:

$$V_1 = \{(v,w) \mid v \in [v_{\min}, v_{\max}] \cap w \in [w_{\min}, w_{\max}]\}$$

- The robot is restricted by its own acceleration, the actual speed during the robot forward simulation time Δt is:

$$V_2 = \{(v,w) \mid v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_b \Delta t] \cap \\ w \in [w_c - \dot{w}_b \Delta t, w_c + \dot{w}_b \Delta t]\}$$

v_c, w_c represents the linear speed and angular speed of the robot at the current time respectively. $\dot{v}_b$, $\dot{w}_b$ represents the linear acceleration and angular acceleration of the robot.

Combining the above two constraints, let V represent the speed space of the robot, then V should be expressed as:

$$V = V_1 \cap V_2$$

c) Evaluation function: the linear speed and angular speed are combined to simulate robot trajectories, the optimal trajectory is selected through certain rules. The speed (v,w) corresponding to the optimal trajectory is the speed command to control the robot. The evaluation function of the DWA algorithm is shown in (5):

$$G(v,w) = \sigma(\alpha \cdot head(v,w) + \beta \cdot dist(v,w) \\ + \gamma \cdot velocity(v,w)) \quad (5)$$

$head(v,w)$ represents the angular deviation between the movement direction of the robot and the target point. $dist(v,w)$ represents the distance between the robot and the nearest obstacle. $velocity(v,w)$ represents the linear speed in the speed (v,w). α, β, γ are the weight coefficients of the three evaluation parameters respectively, and σ is the smoothing coefficient.

A. The Adaptive DWA Algorithm

In the traditional DWA algorithm, the weights of the corresponding three evaluation parameters are usually fixed. However, the evaluation function with fixed weights often cannot adapt to the complex obstacle environment [12]. This paper adaptively adjusts the weights of the evaluation parameters according to the real-time distance between the robot and the obstacle.

The first step is to set a safety distance ($dist_{safe}$), and to calculate the shortest distance ($dist_{\min}$) between the robot and the nearest obstacle. The second step is to judge the relationship between the shortest distance and the safety distance. If $dist_{\min} > dist_{safe}$, the robot should speed up and get close to the target firstly (Means: γ, α increases, β decreases). If $dist_{\min} \leq dist_{safe}$, the robot should reduce the speed and avoid obstacles firstly (Means: α, γ increases, β reduces).

The different DWA algorithm were simulated with Fig. 1, and the results are shown in Fig. 8. The DWA algorithm involves a large number of robot parameters and algorithm parameters. The specific parameter settings are shown in Table III and Table IV.

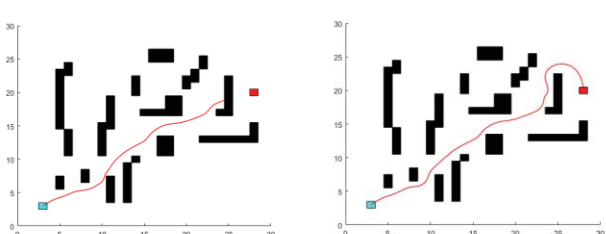
TABLE III. ROBOT KINEMATICS MODEL PARAMETERS

υ(min)	υ(max)	Δυ /Δt	ω(min)	ω(max)	Δω /Δt
0 m/s	1.0 m/s	0.2 m/ s²	-20 °/s	20 °/s	50 °/s²

TABLE IV. DWA ALGORITHM PARAMETERS

Linear speed resolution	Angular speed resolution	Forward simulation time	Analog time resolution	α	β	γ
0.02m/s	2 °/s	2.0 s	0.1 s	0.4	0.1	0.5

Compared with the traditional DWA algorithm, the adaptive DWA algorithm reduces the speed in advance and adjusts the direction of movement when passing through narrow passages. The algorithm successfully avoids concave obstacle traps and obtains a safer path.



(a) Traditional DWA algorithm (b) Adaptive DWA algorithm

Fig. 8. Simulation effects of different algorithms

IV. A FUSION ALGORITHM

This article uses an improved a* algorithm for global path planning, the global path point is used as the sub-target point of the improved DWA algorithm, which guides the robot to track the target point sequence in turn until it reaches the end point.

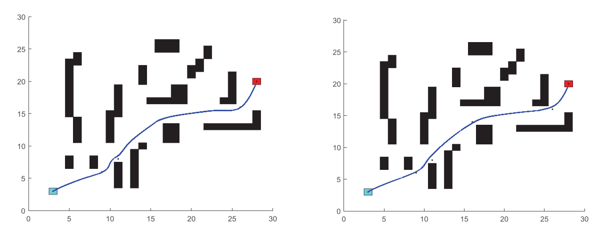
In order to further smooth the path and improve the stability of robot motion, this article adds a smoothing factor to the evaluation function. The sub-target point closest to the current target point of the robot is selected as the smooth point, which can optimize the speed of the robot when it approaches the sub-target point. The evaluation function of the improved fusion algorithm is shown in (6):

$$G(v,w) = \sigma(\alpha \cdot head(v,w) + \beta \cdot dist(v,w) + \gamma \cdot velocity(v,w) + head_{smooth}(v,w)) \tag{6}$$

$head_{smooth}(v,w)$ represents the angular deviation between the smooth point and the movement direction when the robot reaches the end of the trajectory at speed (v,w).

V. SIMULATION AND ANALYSIS

In order to verify the effectiveness of the improved fusion algorithm, the simulation experiments was performed with Fig. 1. The results are shown in Fig. 9 and Table V.



(a) The fusion algorithm (b) The improved algorithm

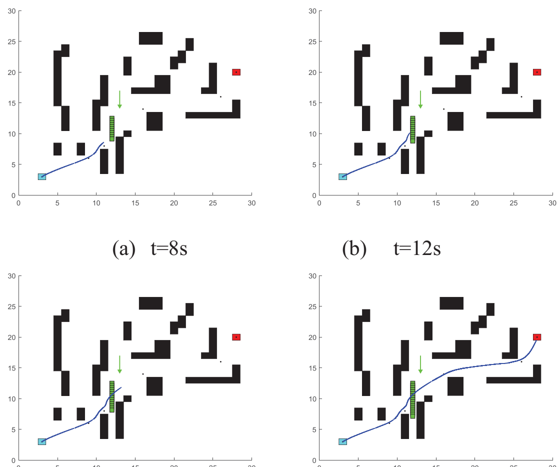
Fig. 9. Simulation effects of different algorithms

TABLE V. PERFORMANCE COMPARISON OF PATH LENGTH WITH DIFFERENT ALGORITHM

Algorithm	Path length
Fusion algorithm	32.68
Smoothed fusion algorithm	31.89

It can be seen from Fig. 8 and Fig. 9 that the fusion algorithm plans a shortest path between the start and end points, which solves the problem of the robot traveling along the periphery of the obstacle. The improved fusion algorithm obtains a smoother path by optimizing the speed of the robot when it approaches the sub-target point. As is shown in Table V, the path length of the improved fusion algorithm is further shortened.

In order to verify the effect of the improved algorithm to avoid dynamic obstacles, dynamic obstacles are added to Fig. 1, and the improved fusion algorithm is applied to simulation experiments. The simulation results are shown in Fig. 10.



(a) t=8s (b) t=12s

(c) t=14s (d) t=32s

Fig. 10. Obstacles approach the mobile robot in a straight line from the front

It can be seen from Fig. 10 that in a complex environment with dynamic obstacle, the improved fusion algorithm avoids obstacles in time and returns to the original trajectory. A smooth and safe path is planned.

VI. CONCLUSION

This paper solves the problem of robot path planning in complex environments by fusing global planning and local planning, and proposes improvements for the shortcomings of global and local path planning respectively. The effectiveness of the improved algorithm is further verified by simulation experiments. Compared with the traditional A* algorithm and DWA algorithm, this algorithm effectively improves the path smoothness and safety. The algorithm in this paper significantly improves the feasibility of the robot's motion state. In the actual robot operating environment, it can satisfy the robot's motion performance to the greatest extent.

ACKNOWLEDGMENT

This work was supported in part by Shandong key research and development plan (Major scientific and technological innovation projects) (2019JZZY010430) and the Pilot Program for the Integration of Science, Education and Industry of the Qilu University of Technology (Shandong Academy of Sciences) under Grant 2020KJC-ZD04.

Corresponding author: Yongguo Zhao, sduzyg@sdas.org

REFERENCES

[1] Z. Da-Qi and Y. Ming-Zhong, "Survey on technology of mobile robot path planning," Control & Decision, 2010.

[2] N. Sariff and N. Buniyamin, "An Overview of Autonomous Mobile Robot Path Planning Algorithms," in Conference on Research & Development, 2006.

[3] L. Zhu, J. Z. Fan, J. Zhao, X. G. Wu, and G. Liu, "Global path planning and local obstacle avoidance of searching robot in mine disasters based on grid method," Zhongnan Daxue Xuebao (Ziran Kexue Ban)/Journal of Central South University (ence and Technology), vol. 42, no. 11, pp. 3421-3428, 2011.

[4] P. Mandal, R. K. Barai, M. Maitra, and S. Roy, "Path planning of autonomous mobile robot: A new approach," in International Conference on Intelligent Systems & Control, 2013.

[5] W. Juan-Juan and C. Kai, "Path-Planning for Robot Based on Grid Algorithm," Agricultural Equipment & Vehicle Engineering, 2009.

[6] Zhang, Hong-Mei, Ming-Long, Yang, and Le, "Safe Path Planning of Mobile Robot Based on Improved A* Algorithm in Complex Terrains," Algorithms, 2018.

[7] M. Lin, K. Yuan, C. Shi, and Y. Wang, "Path planning of mobile robot based on improved A∗ algorithm," in Control & Decision Conference, 2017.

[8] S. Hui, C. Wen, Z. Shu-Long, and Z. Bao-Shan, "Application of an Improved A~* Algorithm in Shortest Route Planning," Geomatics & Spatial Information Technology, 2009.

[9] Seder, Marija, Petrović, and Ivan, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in IEEE International Conference on Robotics & Automation.

[10] C. Cheng, X. Hao, J. Li, Z. Zhang, and G. Sun, "Global Dynamic Path Planning Based on Fusion of Improved A* Algorithm and Dynamic Window Approach," Hsi-An Chiao Tung Ta Hsueh/Journal of Xi'an Jiaotong University, vol. 51, no. 11, pp. 137-143, 2017.

[11] Z. Zhu, J. Xie, and Z. Wang, "Global Dynamic Path Planning Based on Fusion of A* Algorithm and Dynamic Window Approach," in 2019 Chinese Automation Congress (CAC), 2020.

[12] B. K. Oleiwi, H. Roth, and B. I. Kazem, "A Hybrid Approach Based on ACO and Ga for Multi Objective Mobile Robot Path Planning," Applied Mechanics and Materials, vol. 527, pp. 203-212, 2014.