ISSN 2616-5775 Vol. 4, Issue 8: 73-77, DOI: 10.25236/AJCIS.2021.040814

Dynamic Path Planning for Mobile Robots Based on the Improved A-Star Algorithm

Jiaxing Chen[1], Yi Luo[2]

[1]School of Automation, Harbin Institute of Technology, Harbin 150080, China
[2]School of Mechanical and Electrical Engineering, Central South University, Changsha 410083, China

Abstract: *Based on the path planning and obstacle avoidance problems of mobile robots in dynamic environments, a planning algorithm combining global path planning and local path planning is proposed. Firstly, four different path planning algorithms for robots in static environments are compared. On the basis of the A_star algorithm, an improved A_Star algorithm is proposed. The search direction is reduced from eight to five to improve the search efficiency; a path smoothing optimization algorithm is designed to improve the smoothness of the path; and the adaptive function is optimized to speed up the convergence speed of the algorithm. Secondly, the path planning problem of robots in unknown dynamic environments is studied, and an optimization algorithm combining the improved A_Star algorithm with the rolling window algorithm is proposed. The algorithm can plan a global optimal path and obtain map information through the scrolling window of Dynamic Window Approach (DWA) to calculate a suitable obstacle avoidance strategy in real time for the obstacles that appear, so as to plan the optimal path after avoiding the obstacles. Finally, the simulation platform is used to verify whether the fusion algorithm has better path planning performance through comparative experiments in a randomly changing environment.*

Keywords: *robot obstacle avoidance, path planning, fusion algorithm, improved A-Star algorithm, Dynamic Window Approach algorithm*

1. Introduction

With the development of intelligent robotics, mobile robots provide a lot of help to people's lives. Path planning optimization algorithms are very important for the development of future robots. The main implementation of path planning is to find an optimal path from the starting state (position and pose) to the target state (position and state) according to a certain evaluation function (shortest path length and least energy consumption). However, due to the complexity and diversity of the environment, especially in the presence of unpredictable dynamic obstacles, the rational planning of paths in complex environments is a major concern and a hot research problem for mobile robots.

The A_Star algorithm, as a heuristic algorithm, is widely used in logistics and transportation, intelligent warehouse [2], Internet of Things, aerospace, unmanned vehicles and other fields because of its good generality and scalability. The focus of this research is on optimizing and improving the functions based on the A_Star algorithm and integrating it with the rolling window algorithm to study a path optimization algorithm with autonomous obstacle avoidance.

2. Environment modelling

The environmental map in this paper was created by the selected grid method. The environment map is divided into equal binary parameter cells, the position of each cell being represented by two-dimensional coordinates and ordinal numbers. The size of the grid will affect the search speed and the accuracy of the algorithm results. The grid created in this paper has a size of 20×20, and there are 400 nodes on the map. We made the following conventions for the environment map.

(1) Static obstacles in the environment are predetermined, while dynamic obstacles are added randomly after the robot has selected a start and target point. The smallest unit of both static and dynamic obstacles is one grid size.

(2) The distance from each grid of the moving robot to the neighbouring grid is the distance between the centroid of the two grids, with one grid step per move, and the neighbouring grid can be reached by

ISSN 2616-5775 Vol. 4, Issue 8: 73-77, DOI: 10.25236/AJCIS.2021.040814

eight search directions. The movement directions available to the robot in each cell raster are shown in Fig. 2.

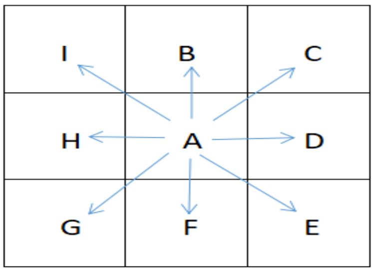


Figure 1: Direction of movement of the robot

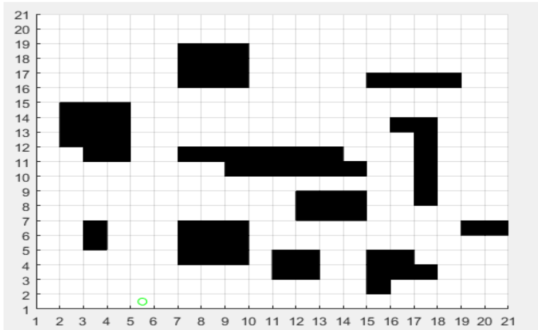


Figure 2: A well-built static obstacle raster

As shown in Fig. 2, the black grids represent static obstacles, the white grids represent the nodes through which the mobile robot can pass, and the horizontal and vertical coordinate values represented by each grid.

3. Design and improvement of the path planning algorithm

3.1. Principle of the A_Star algorithm

The A_Star algorithm is an intelligent heuristic proposed by Hart et al [3], depending on its valuation function *f(n)* as shown in equation (1):

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

Where n represents the number of nodes currently located, $f(n)$ represents the estimated total cost for the robot to reach the target point from the starting point through the corresponding node, $g(n)$ represents the actual surrogate value spent by the robot to reach the corresponding node from the starting point, $h(n)$ represents the estimated cost for the robot to reach the target point from the node, and $h(n)$ represents the actual minimum cost for the robot to reach the target point from the node. Since the Manhattan distance is chosen as the heuristic function since the sub-nodes of the studied map are four-way.

3.2. DWA algorithm principle and optimization

The dynamic window method works by predicting the trajectory of the robot by controlling the linear and angular velocities of the robot during its motion. The search path trajectory is evaluated by means of an evaluation function, from which the locally optimal path is selected.

3.2.1. Velocity sampling

The DWA describes the obstacle avoidance problem as an optimization problem with constraints in velocity space. The range of velocity sampled is then constrained accordingly to the environment, primarily through the robot's velocity, acceleration, and safe distance to the obstacle.

(1) Speed constraints on the robot

Mobile robot constraints on velocity can be divided into constraints on linear velocity and constraints on angular velocity.

$$V_t = \{(v, w)|v_{min} \leq v \leq v_{max}, w_{min} \leq ww_{max}\} \tag{2}$$

v_{min} and v_{max} represent the minimum and maximum linear velocity of the mobile robot. w_{min} and w_{max} represent the minimum and maximum angular velocity of the mobile robot.

(2) Constraints on the acceleration and deceleration of the robot motor

The amount of change in sampling speed per unit of time should be kept within the limits specified for the motor

$$V_s = \{(v, w)|v \in [v_c - a_{dmax}\Delta t, v_c + a_{imax}\Delta t]\ \ w \in [w_c - \alpha_{dmax}\Delta t, w_c + \alpha_{imax}\Delta t]\} \tag{3}$$

ISSN 2616-5775 Vol. 4, Issue 8: 73-77, DOI: 10.25236/AJCIS.2021.040814

a_{dmax} and a_{imax} represent maximum deceleration and maximum acceleration of the linear speed of the robot. α_{dmax} and α_{imax} represent maximum deceleration and maximum acceleration of the angular speed of the robot. v_c and w_c are the current linear and angular velocity of the robot [4].

(3) Constraints on the safe distance of the robot

In order to prevent collisions with obstacles, the robot should maintain a safe distance from the obstacle is defined as

$$V_d = \left\{(v,w)|v \leq \sqrt{2dist(v,w)a_{dmax}}\ ,w \leq \sqrt{2dist(v,w)\alpha_{dmax}}\right\} \tag{4}$$

$dist(v,w)$ represents the distance of the end of the predicted trajectory from the obstacle [4]. Finally, the velocity constraint of the dynamic window algorithm is the intersection of the above three velocity constraints and the dynamic window velocity can be expressed as $V_w = V_t \cap V_s \cap V_d$.

3.2.2. Evaluation function optimization and improvement

The evaluation function of the traditional DWA consists of three main indicators: the azimuth angle pointing to the end point, the current velocity of the robot and the distance between the current position and the obstacle. However, the traditional DWA algorithm suffers from two shortcomings. First, it is easy to fall into the local optimum path. Second, it does not distinguish between known and unknown obstacles, which leads to reduced sensitivity in the dynamic obstacle avoidance process. Therefore, in this paper, the traditional DWA evaluation function is optimized and an improved evaluation function such as Equation (7) is derived, which increases the sensitivity of the robot during dynamic obstacle avoidance. The improved evaluation function is expressed as

$$G(v,w) = \alpha heading(v,w) + \beta vel(v,w) + \sigma path(v,w) + \delta dist(v,w) \tag{5}$$

The $\alpha,\beta,\sigma,\delta$ in the formula are the weighting factor for each sub-function. $heading(v,w)$is the directional angular deviation of the end of the simulated trajectory constantly towards the target point; $vel\ (v,w)$ is used to evaluate the magnitude of the current robot motion speed; $path\ (v,w)$ is used to evaluate the distance between the end of the simulated trajectory and the globally planned path; $dist\ (v,w)$ is used to evaluate the minimum distance from the end of the simulated trajectory to the obstacle.

3.3. Improving the A_Star algorithm

3.3.1. Optimizing the evaluation function of the A_Star algorithm

The evaluation function of the A_Star algorithm mainly consists of the actual cost function $g(n)$ and the heuristic function $h(n)$, where the dominant role is played by the heuristic function $h(n)$. When there are fewer obstacles, the weight of the heuristic function $h(n)$ is reduced appropriately to reduce the search range, improve the speed of the algorithm and increase efficiency. When there are more obstacles, increase the weight of the heuristic function $h(n)$ appropriately, so as to increase the search range, avoid the emergence of local optimum, deadlock problem. We change the evaluation function of the classical A_Star algorithm is improved from$f(n) = g(n) + h(n)\ to\ f(n) = g(n) + (1 + r/R) * h(n)$. Where r is the distance from the current position of the robot to the target point and R is the distance from the start point to the target point.

3.3.2. Optimizing the bend rate of the path

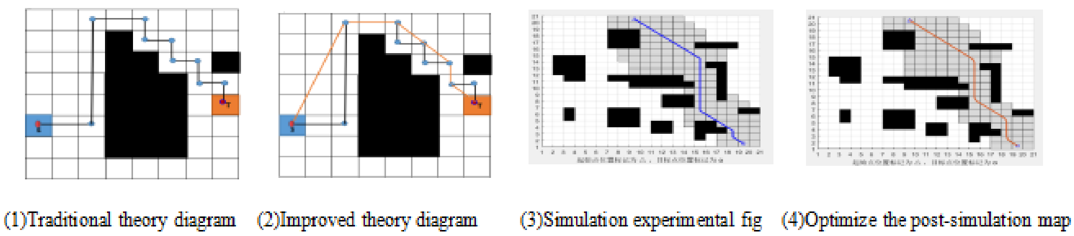


Figure 3: Comparison diagram before and after the optimization

First, all nodes are traversed. The redundant nodes in the middle of each path are deleted, but the starting point and the inflection points are retained. Subsequently, the starting point is connected to each node searched. The distance of all paths from obstacles is metricized. When the metric distance is less than the safe distance threshold, the path is discarded. On the contrary, the path is retained. Finally, the

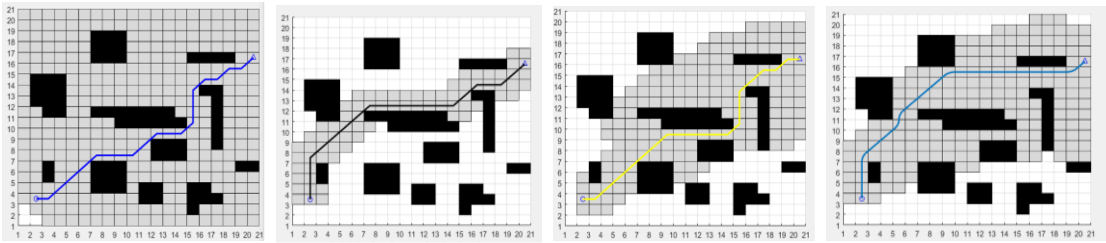
ISSN 2616-5775 Vol. 4, Issue 8: 73-77, DOI: 10.25236/AJCIS.2021.040814

nodes under the retained paths are drawn in arcs at the bends. Nodes at non-bends are drawn with straight line connections and form the optimal path.

4. Simulation experiments

4.1. Simulation experiments of the improved A-Star algorithm

The simulation experiments were verified in the MATLAB 2016b environment. In order to verify the adaptability and effectiveness of the algorithm, a raster map was randomly created. Each raster was set as a square with 1 cm side length. A white raster represented the obstacle-free area and a black raster represented the obstacle area. △ represents the starting point of the robot and ○ represents the target point. In this paper, the improved A-Star algorithm, A-Star algorithm, Dijkstra algorithm and BFS algorithm are compared and simulated.



(a) Dijkstra (b) BFS (c) A_Star (d) Improved A_Star

Figure 4: Comparison diagram of the simulation experiment

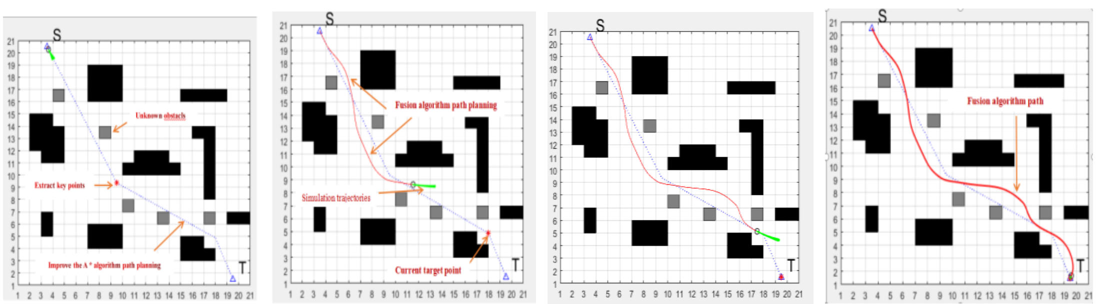
Table 1: The broken path length of the four path planning algorithms compares the calculation time with the number of traversal nodes

Algorithm	The fold path length/m	computing time/m	The number of nodes
Dijkstra	25.14	0.064	324
BFS	25.77	0.008	74
A_Star	25.10	0.028	161
Improved A_Star	24.90	0.046	189

When the robot moves from one node to another, it first needs to turn towards its original position, turn in the direction of the next node and then continue on, so too much curvature in the path can cause a drain on the robot's energy and time. Under the same starting and ending point conditions, it can be seen from the paths of the four comparison graphs that the improved A_Star algorithm has the smallest rate of path curve curvature. As can be seen from Table I, the improved A_Star algorithm plans the shortest paths. Therefore, the improved A_Star algorithm outperforms the other three path planning algorithms in terms of optimal path planning. The improved A_Star algorithm also has a slight increase in computational time due to the increased number of nodes that need to be searched.

4.2. Simulation experiments of the fusion algorithm

The fusion algorithm simulation experiment focuses on manually adding a few random points of unknown obstacles to the A_Star_plus algorithm, with the aim of verifying the adaptability and effectiveness of the fusion algorithm for global path planning in the face of unknown obstacles. The simulation coefficient of the evaluation function of the fusion algorithm is α = 0.3, β=0.2, σ=0.3, δ=0.2.



(a) Global path planning (b) DWA obstacle avoidance (c) Avoid the unknown obstacles (d) Arrive at target point

Figure 5: Graph of the simulation results of the fusion algorithm

ISSN 2616-5775 Vol. 4, Issue 8: 73-77, DOI: 10.25236/AJCIS.2021.040814

The blue dashed line in Fig.5 (a) shows the global path planned by the modified A-Star algorithm before the unknown obstacle is added; the * on the path represents the extraction of key points as intermediate guide points for the DWA algorithm; the grey squares in the figure are used to represent the randomly added unknown obstacle points in the global path. The short green curve at the end of the path planning curve in Fig.5 (b) represents the simulated trajectory; the part of the arc that does not match the blue dashed line represents the fusion algorithm is bypassing the unknown obstacles; Fig.5(c) represents the fusion algorithm has bypassed all the unknown obstacles and is moving towards the final target point; Fig.5 (d) the orange curve represents the fusion algorithm has completed the path planning from the start point to the target point.

5. Conclusion

The experimental results show that the improved A-Star algorithm effectively reduces the length of the planned path under the same conditions, and the planned path maintains a safe distance from the obstacles throughout. This reduces the energy consumption and ensures the safety of the robot's travel movement. The path planned by the fusion algorithm not only achieves the optimal choice of the optimal path, but also avoids the unknown obstacles in the environment. The experimental results show that the improved A-Star algorithm effectively reduces the length of the planned path under the same conditions, and the planned path maintains a safe distance from the obstacles throughout. This reduces the energy consumption and ensures the safety of the robot's travel movement. The path planned by the fusion algorithm not only realizes the optimal choice of the optimal path, but also avoids unknown obstacles in the environment, which can be well applied to the complicated environment in real life.

References

[1] Li Longshu, Yu Huan. Improved ant colony algorithm for robot path planning in complex environments [J]. Small Microcomputer Systems, 2017, 38(9): 2067-2071.
[2] Wang Y. Research on path planning of multiple mobile robots in intelligent warehouse system [D]. Harbin Institute of Technology, 2010.
[3] HART P E, NILSSON N J, RAPHAEL B. A formal basis for the heuristic determination of minimum cost paths [J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.
[4] Liu Jianjuan, Xue Liqi, Zhang Huijuan, Liu Zhongpu. Fusing improved A~ and DWA algorithms for dynamic path planning of robots [J]. Computer Engineering and Applications, 2021, 57(15): 73-81.*