2021 20th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

Research on Path Planning Algorithm of Intelligent Wheeled Robot Based on ROS

Wang Ming-yuan
College of Information &Control Engineering
Jilin Institute of Chemical Technology
Jilin, China

Zhao Meng-yao
College of Information &Control Engineering
Jilin Institute of Chemical Technology
Jilin, China

Zhu Jian-jun
College of Information &Control Engineering
Jilin Institute of Chemical Technology
Jilin, China
e-mail: zjj099@163.com

Abstract—**Aiming at some limitations of the A-star algorithm, the algorithm is improved and optimized, robot path planning experiment is carried out. In the traditional A-star algorithm, there are too many turning points of the path and the path is not smooth enough, the car deviates from the path when navigating due to the arc at the corner is too large. This paper combines the conjugate gradient and Gaussian process principle to optimize the A-star algorithm and eliminate the invalid path information to jump points, to implement a path with less angle and smoother. The results show that the hybrid algorithm has a shorter and smoother path than the A-star algorithm.**

Keywords-component; A-star algorithm; path planning; conjugate gradient; Gaussian process; jumping point

I. INTRODUCTION

With the rapid development of technology, intelligent devices are gradually entering people's daily lives. Smart wheeled mobile robots are a major branch of smart applications and have a wide application in daily life, such as industrial applications, medical catering and other industries. The autonomous navigation of robots is the most important part of intelligent wheeled mobile robots [1], it requires mobile robot to plan a path based on some optimization criteria or indicators (such as the smallest energy consumption, the shortest completion time, the smoothest path, the shortest walking path, etc.) The mobile robot is required to plan a safe and collision-free shortest trajectory from the start point to the target point [2]. Due to the expansion of the working range of mobile robots, the requirements of path planning technology are also increasing. Currently, the most widely used path planning algorithms include the A-star algorithm, ant colony algorithm, DWA (Dynamic Window Approach) algorithm, artificial potential field method, Dijkstra algorithm, etc. [3-6] Path planning is one of the key factors in autonomous robot navigation. The advantage of the A-star algorithm is that it can find the best path, but in a complex environment, the A-star algorithm takes a longer time to find the path and the path is not reach the optimal path. The path planned by the A-star algorithm is an angular turn at the corner and the arc-like turn required by the car is not considered.

Literature [7] proposed an improved ant colony algorithm, this algorithm uses gradient descent to optimize the path of the ant colony algorithm, which improves the convergence speed and accuracy of the ant colony algorithm. Literature [8] adopts the A-Star algorithm based on changing the step size to reduce the calculation time of the algorithm and improve the stability and robustness of the algorithm performance. Literature [9] uses a three-domain search A-star algorithm combined with artificial potential fields to optimize the path planning problem of mobile robots, which shortens the length of the path, reduces the search time and the number of nodes. Literature [10] uses the hybrid-A star algorithm, which is optimized by the conjugate gradient method and is suitable for Ackerman vehicles. Because kinematics is added to plan the vehicle's backward route, the overall planned route may be longer than the route planned by the A-star algorithm, but this algorithm improves the smoothness and safety of the route. Literature [11] proposed a path planning algorithm based on jumping point search combined with the Bezier curve to achieve path optimization.

II. AN IMPROVED HYBRID ALGORITHM BASED ON A-STAR ALGORITHM

The A-star algorithm estimates the optimal path based on the step length and cost. The JPS (Jump Point Search) algorithm speeds up the operating efficiency of the A-star algorithm by reducing the number of nodes traversed, reducing the length of the path and the number of corners turned. However, the JPS algorithm also has problems such as the path is not smooth enough and the corners are too large. The algorithm is optimized by combining the advantages of the JPS algorithm, first, perform conjugate gradient processing on the A-star algorithm to obtain a piece of new path information. Then the path information is pruned, only part of the optimal path information is retained, and then the Gaussian process is performed on the path to obtaining an optimal and smooth path.

The principle of the jumping point is to improve the heuristic function of the A-star algorithm, the expression is as (1).

$$h(n)=\lambda L - k \cos a \tag{1}$$

2473-3636/21/$31.00 ©2021 IEEE
DOI 10.1109/DCABES52998.2021.00025

L is the Manhattan distance from the start point to the target point, the expression is shown in (2), λ and k are the weights of distance and direction, cos$a$ is the direction information of the angle between the parent node and the child node, and the child node and the target point. When the two directions are the same, the value of cos$a$ is + 1, and the direction cos$a$ is the smallest, and vice versa, the value of cos$a$ is -1, the direction has the largest cost.

$$L = \beta \ (\text{abs}(n.x - \text{goal}.x) + \text{abs}(n.y - \text{goal}.y)) \quad (2)$$

β is the weight, n.x and n.y are the abscissa and ordinate of the current position node, goal.x and goal.y are the abscissa and ordinate of the target node respectively.

The Gaussian process is a commonly used optimization method, which is widely used in image processing and curve processing. The Gaussian process is often used in machine learning, where the core of the Gaussian process is the kernel function. The final results obtained by applying different kernel functions will also be different. The measurement method of each kernel function is different and the processing properties of the Gaussian process are also different. The more commonly used kernel function is the Gaussian kernel function, which is also called the radial basis function (RBF), the expression is shown in (3).

$$K(x,x') = \delta^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right) \quad (3)$$

Where x and x' are the abscissa of the sampling points, and δ and l are the hyperparameters of the Gaussian kernel. Figure 1 uses sin$x$ as a sample function, where the green line is the prediction curve, the red points are the sampling points and the light blue is the variance confidence interval. When l=0.5, δ=0.5, the curve has more corners and more twists and turns compared to the other three pictures. When l=0.5, δ=1.0, the curve is smoother than that in Figure a, but it is still not smooth enough. When l=1.0, δ=0.5, the curve becomes smoother but the smoothness is higher. When l=1.0, δ=1.0, the smoothness and gentleness of the curve are relatively the best. From the four graphs of a, b, c, d, it can be concluded that the l increasing curve is smoother and the confidence interval between the sampling points is smaller. Otherwise, the curve is more tortuous. The value of δ can control the size of the confidence interval. When the value of δ increases, the confidence interval increases, and vice versa.

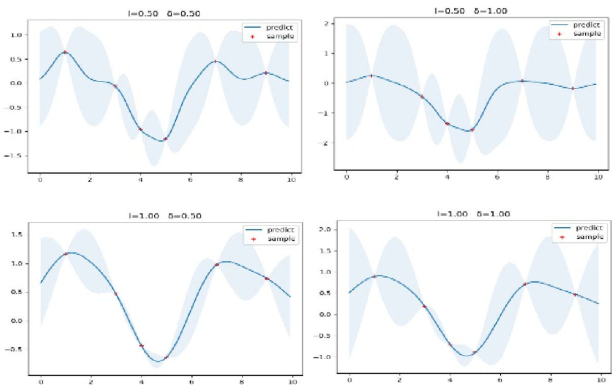

Figure 1. Gaussian process curve

Find the optimal parameters δ and l by maximizing the edge likelihood, the formula is shown in (4).

$$p(y|\delta,l) = \frac{1}{(\sqrt{2\pi})^n\sqrt{K_{yy}}} e^{-\frac{1}{2}y^T K_{yy}^{-1} y} \quad (4)$$

After logarithmic processing, formula (5) is obtained.

$$\log p(y|\delta,l) = -\frac{1}{2}y^T K_{yy}^{-1} y - \frac{1}{2}\log|K_{yy}| - \frac{n}{2}\log(2\pi) \quad (5)$$

Maximizing the logarithmic marginal likelihood refers to the maximum marginalization on the function value y, where K_{yy} is the covariance matrix. After derivation, the optimal hyperparameters are l=1.2, δ=0.8, and the test results are shown in Figure 2 show.

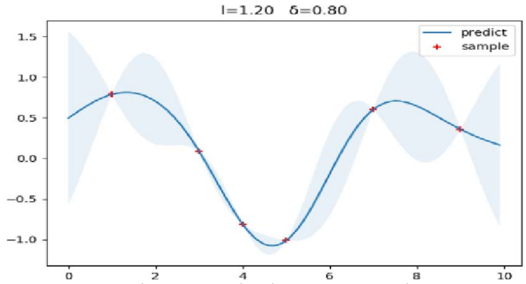

Figure 2. Optimal parameter graph

The flowchart of the improved hybrid algorithm is shown in Figure 3. The starting point and the target point are set, the path is planned by the conjugate gradient A-star algorithm, and the path is skipped. Gaussian smoothing is performed after the path that eliminates interference is obtained, and the optimal smooth path is obtained.

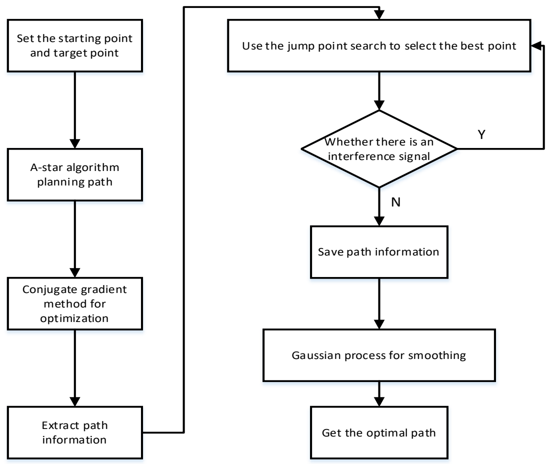

Figure 3. Hybrid algorithm flow

III. SIMULATION EXPERIMENT COMPARISON

A. *Without obstacle simulation*

Optimize the path without obstacles and test it on the maps of 3x3 and 9x14 specifications. The red line is the unprocessed path and the blue line is the optimized path, the test results are shown in Figure 4 and Table 1.

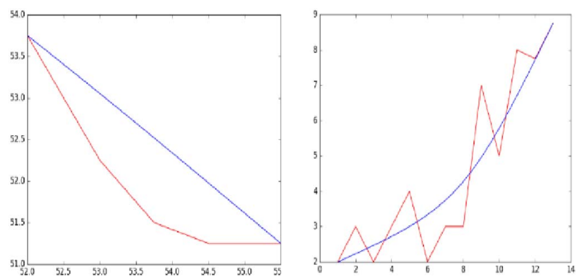

Figure 4. Without obstacle simulation

TABLE 1. Simulation data

Map Specifications	Path Length (cm)	
	unprocessed path	*optimized path*
3x3	5.201	4.301
9x14	26.733	19.065

The data comparison shows that the processed path is shorter than the unprocessed path and the path is smoother.

B. *With obstacle simulation*

This simulation experiment is set up in Gazebo as shown in Figure 5, the planned paths are displayed in Rviz. Figure 6 shows the A-star, Conjugate gradient, Jump point search and Gaussian smoothed path from top to bottom. The path after the conjugate gradient processing is smoother than the path of the A-star, but the overall path is not smooth enough. After the jumping point processing, the invalid path information is eliminated and the path between the two turning points is straighter. Then the Gaussian process optimization is performed on the path after the jump point, and a smooth optimal path is obtained.

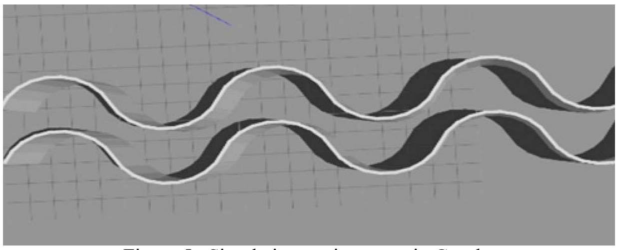

Figure 5. Simulation environment in Gazebo

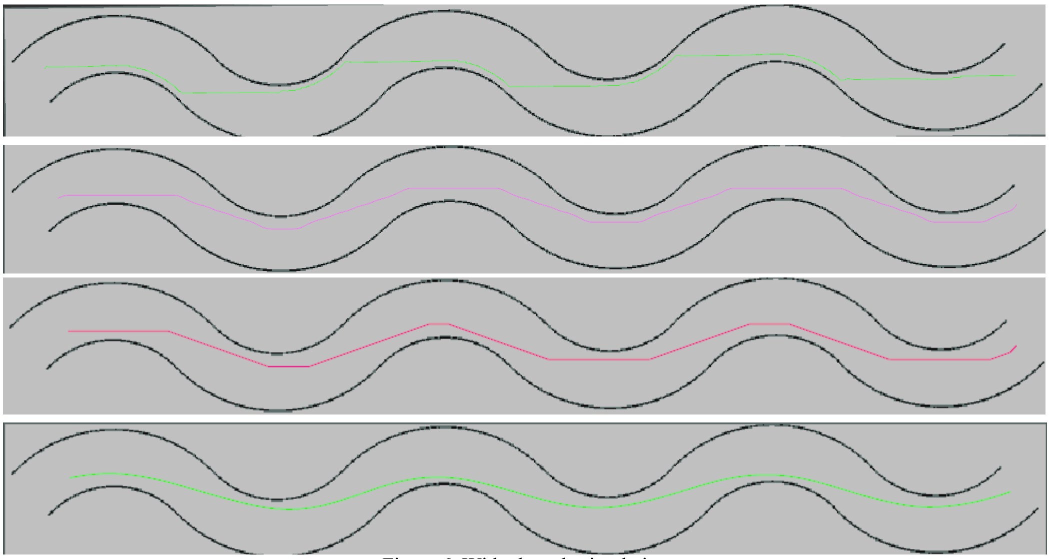

Figure 6. With obstacle simulation

TABLE 2. Simulation data

Simulation Map	Race Track			
	A-star	*Conjugate gradient*	*jump point*	*Gaussian smoothed*
Path length	81.254	77.782	75.554	70.251
Number of nodes	7322	5502	573	573

By comparing the path length with the number of points traversed, the A-star algorithm path has more nodes and longer paths. The number of nodes in the path after conjugate gradient processing decreases and the path length is shorter than the path of the A-star algorithm. The jumping point is the selection of path nodes on the path after gradient processing. The number of nodes traversed by the path decreases significantly, and the path length is slightly shortened. The length of the path smoothed by the Gaussian process is significantly shortened, and the number of nodes is equal to the number of nodes processed by the hop point. The optimized hybrid algorithm preliminarily solves the non-optimal path of the A-star algorithm in a complex environment, the path is not smooth enough, the corners are too much.

IV. NAVIGATION SIMULATION EXPERIMENT

In order to verify whether the path planned by the hybrid algorithm meets the motion requirements of the mobile robot in a complex environment, a path navigation simulation experiment is carried out for the mobile robot, the robot runs along the planned path from the starting point to the specified target point. Figure 7 shows the simulation results of robot navigation, where the green line is the route planned by the hybrid algorithm and the red line is the historical trajectory of robot operation. It can be seen from the figure that the route of robot operation is consistent with the route planned by the algorithm, which verifies the feasibility of the algorithm.

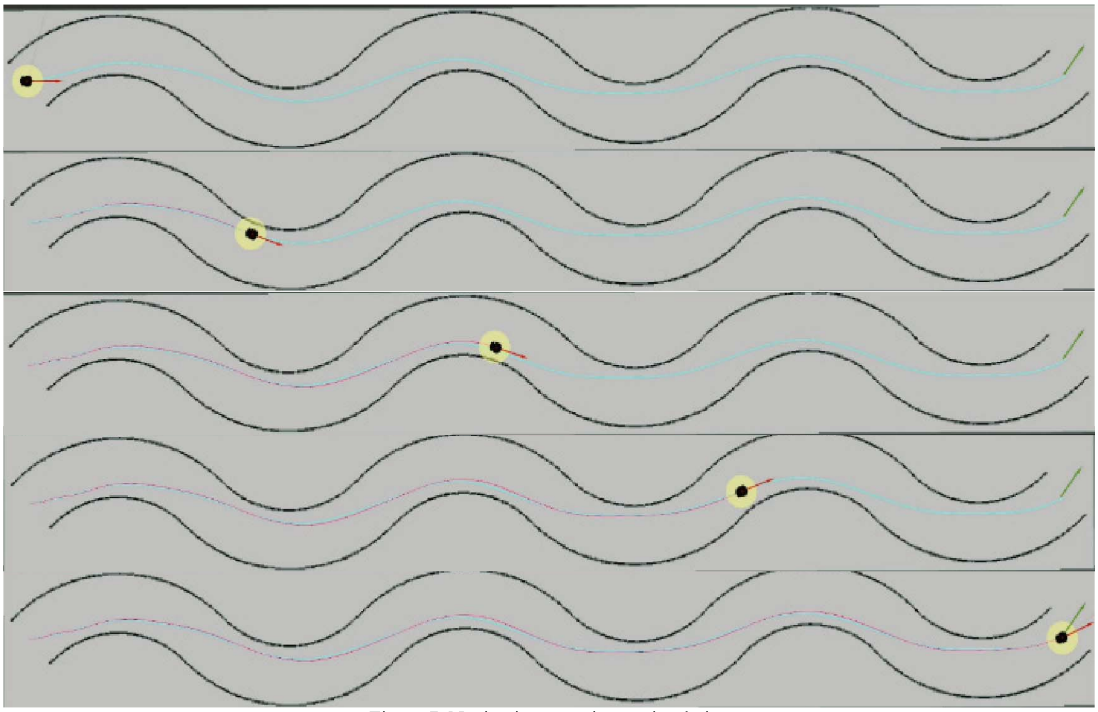

Figure 7. Navigation experiment simulation

V. Conclusion

Experiments show that the path planned by the improved hybrid algorithm in this paper is better than the path planned by the A-star algorithm. It reduces the number of corners and the arc of the mobile robot in a complex environment, the path is smoother. In the robot navigation simulation experiments, the robot's forward trajectory is the same as the planned route, which provides an effective algorithm for the mobile robot to quickly and stably run to the target point.

References

[1] Zhang Xin, Zhang Yu, Su Xiaoming.Design and Implementation of Autonomous Positioning and Navigation System for Mobile Robots, Machine Tool and Hydraulics, vol 48, 2020, pp.88-91.

[2] Zhang Jinyue, Hou Zhicheng, Zhang Gong, et al.Improved A* Algorithm Based on Traffic Constraints and Multivariate Heuristic Function. Modular Machine Tool and Automatic Processing Technology, 2021(01):53-56.

[3] Dai Xiaolin, Long Shuai, Zhang Zhiwen, Gong Dawei. Mobile Robot Path Planning Based on Ant Colony Algorithm With A* Heuristic Method. Frontiers in neurorobotics, vol. 13, 2019, pp.15.

[4] Qin Yuannian, Liang Zhonghua.New progress in research and application of ant colony algorithm.Computer Engineering and Science, vol.41, 2019, pp.173-184.

[5] Jiang Weinan. Research and Application of AGV Path Planning and Autonomous Obstacle Avoidance[D]. Changchun: Changchun University of Technology, 2020.

[6] Dong Xuanliang, Zhao Guiqing.Robot navigation path planning based on artificial potential field-guided ant colony algorithm.Machinery Design and Manufacturing, 2021(06):169-173.

[7] Qin Kun, Guan Zequn, Li Deren, et al.Research on the best path analysis method based on raster data.Remote Sensing for Land and Resources, 2002(02):38-41.

[8] Shang Erke, Dai Bin, Nie Yiming, Zhu Qi, et al. An improved A-star based path planning algorithm for autonomous land vehicles. International Journal of Advanced Robotic Systems, vol.17, 2020, pp.1729881420564952.

[9] Chen Jiqing, Tan Chenzhi, Mo Rongxian, et al. Research on path planning of three-neighbor search A* algorithm combined with artificial potential field. International Journal of Advanced Robotic Systems, vol.18, 2021, pp.17298814211026449D.

[10] Dušan Nemec, Michal Gregor, Emília Bubeníková, et al. Improving the Hybrid A* method for a non-holonomic wheeled robot. International Journal of Advanced Robotic Systems, vol.16, 2019, pp.1729881419826857.

[11] Zhang Ben, Zhu Denglin. A new method on motion planning for mobile robots using jump point search and Bezier curves. International Journal of Advanced Robotic Systems, vol.18, 2021, pp.32.