Automation and Machine Learning (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/autml.2025.060106
ISSN 2516-5003 Vol. 6 Num. 1

Collaborative Design of Path Planning and Motion Control for Mobile Robots in Complex Environments

Ming Qian

R&D Center, Potevio Rail Transit Technology (Shanghai) Co., Ltd., Shanghai, China
qianming@potevio-rail.com

Keywords: Path Planning; Motion Control; RRT* Algorithm; Dynamic Obstacle Avoidance; Adaptive Path Adjustment

Abstract: Given how quickly mobile robot technology is developing, utilizing current path planning techniques in complex environments faces challenges in dynamic obstacle avoidance, path optimization and real-time response. To deal with these problems, this paper proposes a collaborative design method for mobile robot path planning and motion control for complex environments. This paper introduces an improved version of the RRT* (Rapidly Expanding Random Tree) algorithm, combined with adaptive path adjustment and dynamic obstacle avoidance strategies, to help increase the robot's movements accuracy and planning of paths efficiency in challenging situations. The RRT* algorithm can efficiently create a workable route between the beginning and destination points while minimizing the path length and considering motion constraints. The adaptive path adjustment method updates the path in real time during the robot operation, especially under the influence of dynamic obstacles, to guarantee the path's stability and consistency. The motion control system works in conjunction with the path planning algorithm to ensure that the robot accurately tracks the optimized path and maintains safe driving. Through detailed simulation experiments, this paper compares the performance of this method with traditional algorithms (A* algorithm and basic RRT algorithm), focusing on evaluating key indicators such as execution time, path length, and obstacle avoidance performance. The experimental results show that in terms of collaborative effect, the improved RRT scores 8.5 in a static environment, higher than A*'s 7.8, indicating that its path planning and motion control collaborative optimization effect is better.

1. Introduction

With the development of automation, intelligence and integration of production workshops, the participation and coupling degree of mobile robots in the workshop production process are getting higher and higher. According to the current level of robot intelligence and workshop production needs, robots do not have the ability to completely replace humans. Workshop production is completed by machine tools, robots and natural persons, making the workshop environment complex and changing dynamically in real time. Studying the path planning problem of robots in complex and dynamic workshop environments is the basis for their deep participation in the production process in the workshop. This study mainly focuses on comparing the effectiveness of various path planning algorithms in both static and dynamic settings, and explores the impact of the co-design of motion management and scheduling of paths on the overall performance of the robot. We select two common path planning algorithms: the improved RRT algorithm and the A algorithm, and conduct experiments in static environments, dynamic environments, and adaptive path adjustment scenarios, respectively, to evaluate the performance differences of different algorithms and their feasibility in practical applications.

This paper first presents the investigation's background and goals, and explains the importance of co-design of motion management and scheduling of paths and the selection of related algorithms. Then, the experimental design and setup are described in detail, including the experimental environment, algorithm configuration and evaluation indicators. Subsequently, combined with experimental data, Analysis is done on how well certain algorithms work in various settings, and their path planning accuracy, motion control effect and system performance are compared. Finally, the bottleneck problems in the research are proposed, and future research directions are prospected, emphasizing the key role of improving the real-time performance and computational efficiency of the algorithm in further optimizing the robot path planning and control.

2. Related Work

As path planning technologies continue to advance in the artificial intelligence and automation fields, more and more researchers have combined different optimization techniques and suggested a number of approaches to increase course the planning field's precision as well as productivity. García et al. combined the optimization ability of the A* algorithm with the search ability of the co-evolutionary algorithm to generate collision-free paths in real time and apply them to edge computing devices. According to experimental data, this approach outperforms previous algorithms in complex circumstances; however, additional optimization of the calculation time is still necessary[1]. To accomplish path planning without collisions in intricate settings, Gu et al. improved the network structure based on M-DQN (Munchausen deep Q-learning network) and divided the entire network construction into two parts: the beneficial mechanism and the value mechanism. An artificial interest field is suggested as a means to configure the function of reward in order to push the creature's trajectory beyond the obstruction in order to address the issue of the velocity being too near to the edge[2]. Khlif et al. enhanced the exploration strategy and suggested a novel Q learning-based path planning technique for mobile robots. The suggested approach performs better through simulated when thinking of the expense operate, path duration, and time to execution [3]. Mohanraj et al. enhanced the algorithm by considering the improvement field and real-time constraints so as to effectively use the algorithm in robot navigation. For larger grid sizes, the proposed hybrid algorithm has less computational time and path cost than the traditional algorithm. The A* algorithm performs better in environments with smaller grid sizes[4]. Tran and Lin suggested a two-step process for welding robotics navigation plans that makes use of multi-sensor contact. Experimental results show that the proposed research has great potential in realizing welding robot automation in manufacturing[5]. Shen et al. proposed an optimized RRT* path planning strategy based on operability. When sampling the search space, the path length and operability index are constrained to find the minimum cost path, in order to prevent settings singularity and allow the end contact to travel the working space on a quicker path[6]. In their analysis and evaluation of the path planning algorithms frequently employed by fruit and vegetable picking robots over the previous ten years, Zeeshan et al. compared how well each algorithm performed in terms of avoiding obstacles, using the least amount of computing power, and reaching the target fruit in the shortest amount of time[7]. Szczepanski proposed a new local path planning algorithm, Safety Artificial Potential Field (SAPF). The algorithm replaces the traditional repulsive potential with a universal obstacle potential (including repulsive potential, vortex potential, or their superposition) while keeping the target attractive potential unchanged, and provides an analytical tuning method for SAPF[8]. Fang et al. analyzed the geometric features of the impeller, extracted the weld curve and the blade edge curve to achieve path planning. A new structured light sensor combined with an improved ICP (Iterative Closest Point) algorithm was used to achieve automatic alignment and positioning of the impeller and automatically generate the welding path. Simulation and experiments verified the effectiveness of this method [9]. Eshtehardian and Khodaygan proposed a method that combines RRT* with B-splines to smooth the path generated by RRT*. This method optimizes the path through computer-aided geometric design curves and introduces collision avoidance correction [10]. Mier et al. introduced Fields2Cover, an open source library for path planning with agricultural vehicle coverage. Fields2Cover provides a modular and extensible framework that supports a variety of vehicles and is suitable for a variety of applications such as agriculture [11]. The bottlenecks of existing research mainly focus on optimizing computing time, real-time adaptability in dynamic environments, and improving path planning accuracy in complex scenarios [12].

3. Method

3.1 Environmental Modeling

The goal of environment modeling is to abstract the obstacles and free space in the real world into a structure that is convenient for path planning algorithms to process. The following are common ways to model the environment:

Adaptive path adjustment: Considering the real-time dynamic changes in the environment and obstacles, the value of each grid $\Delta x \times \triangle y$ is 0 (free space) or 1 (obstacle). The representation method is:

$$M(i,j) = \begin{cases} 1 \\ 0 \end{cases} \tag{1}$$

In the formula, 1 represents an obstacle and 0 represents free space. Among them, i, j is the coordinate of the grid.

Topological map: By representing the key positions and connection paths in the environment, a graph model can be constructed, in which nodes represent key positions and edges represent accessible paths. Assuming the node set is V and the edge set is E, the graph G can be represented as:

$$G = (V, E) \tag{2}$$

Among them, V is the key node in the environment, and E is the connection between these nodes.

Hybrid map: A method that combines raster and topological maps. Usually, topological maps are used for rough path planning in the global scope, and raster maps are used for fine obstacle avoidance and path adjustment in the local scope.

3.2 Goals and Constraints

Creating a path that satisfies the robot's motion limitations from its starting location to the finish point is the aim of path scheduling:

Speed and acceleration constraints: Assuming that the greatest momentum and linearity of the robot are v_{max} and $a_{\max}$, respectively, then path planning needs to take these constraints into account to ensure that the path can be completed safely within the agreed time. Motion constraints can be expressed in the following form

$$\|v(t) \leqq v_{max}\|, \|a(t) \leqq a_{max}\| \tag{3}$$

Among them, $v(t)$ and $a(t)$ are the velocity and acceleration at time t, respectively.

Target position setting: The target position $P_{target} = (x_{target}, y_{target})$ is the final goal of path planning. The task of the path planning algorithm is to generate an effective path from the starting point $P_{start} = (x_{start}, y_{start})$ to the target point P_{target}.

3.3 Path Planning Algorithm

The path planning algorithm's goal is to use environmental data to create a workable route from the starting point to the destination. The Rapidly Expanding Random Tree (RRT) algorithm is introduced as follows:

Random expansion: By growing from an existing node to a randomly selected location in the empty area, RRT builds a path continuously. Given a starting point and a target point, a random point is sampled each time, and then the tree is expanded:

$$x_{new} = x_{nearest} + \triangle t.\frac{x_{rand} - x_{nearsst}}{\|x_{rand} - x_{nearest}\|} \tag{4}$$

Among them, $x_{nearest}$ is the node closest to the random point x_{rand} in the tree, and $\triangle t.$ is the step size of the expansion.

Path optimization: The paths generated by RRT are usually tortuous, so post-processing is required to optimize the path and reduce unnecessary turns. Path optimization can use a smoothing algorithm.

The core goal of the path smoothing algorithm is to post-process the path generated by RRT to eliminate excessive turns and make the path smoother and more natural.

B-spline curve fitting: B-spline curve is a commonly used smooth curve form that can accurately describe the path through control points. The path points generated by RRT are set as a set of control points $\mathrm{P_0, P_1, \cdots, P_n}$, and the B-spline curve is used to fit these control points to obtain a smooth path. The mathematical expression of the B-spline curve is:

$$\mathrm{S(t) = \sum_{i=0}^{n} P_i B_{i,k}(t)} \tag{5}$$

Among them, $\mathrm{B_{i,k}(t)}$ is the i-th basis function, k is the order of the spline, and $\mathrm{t \epsilon [0,1]}$ can smooth the path and reduce turns by adjusting the control points.

3.4 Adaptive Path Adjustment

Dynamic obstacle avoidance: During the movement of the robot, it may encounter dynamic obstacles and real-time path updates are required. Assuming the robot's present location is $\mathrm{x(t)}$, the target position is x_{target}, and the obstacle position is $x_{obs}(t)$. If the distance between the obstacle and the robot path is less than the set safety distance d_{safe}, the path needs to be replanned:

$$\|x(t) - x_{obs}(t)\| < d_{safe} \tag{6}$$

Real-time path replanning: By continuously sensing the environment and updating the path in real time, the robot can avoid obstacles and find new feasible paths in a dynamic environment. For example, using rapidly expanding random trees (RRT) for dynamic replanning to continuously optimize the current path:

$$\min_{u} \int_0^T \left\|X(t) - X_{target}\right\|^2 dt \tag{7}$$

Among them, $X(t)$ is the robot's present location, X_{target} is the target position, u is the control input, and T is the time range.

Through these methods, path planning can effectively cope with dynamic changes in complex environments and ensure that the autonomous vehicle can swiftly and safely arrive at the objective.

4. Results and Discussion

4.1 Experimental Equipment and Environment

Robot platform: A mobile robot with autonomous navigation capabilities is selected, equipped with necessary sensors (such as LiDAR, RGB-D camera, IMU, etc.).

Experimental environment: A simulated or actual environment with static and dynamic obstacles is constructed, including scenarios of different complexity (such as narrow passages, obstacle movement, etc.).

Experimental software: Simulation platforms such as Robot Operating System (ROS) and Gazebo are used, and combined with customized path planning and control algorithms.

4.2 Experimental Settings

Table 1 lists the experimental settings, which covers the application of different algorithms under different environmental conditions. These experimental settings are designed to comprehensively assess the effectiveness of various path planning methods in various environments, and then provide a basis for selecting appropriate path planning algorithms.

Table 1. Experimental settings

Experiment No.	Algorithm	Remarks
1	Improved RRT*	Static environment
2	A*	Static environment
3	Improved RRT*	Dynamic environment
4	A*	Dynamic environment
5	Improved RRT*	Adaptive path adjustment
6	A*	Adaptive path adjustment

4.3 Experimental Indicators

Motion control indicators include position accuracy, dynamic obstacle avoidance effect, and smoothness. Position accuracy reflects the actual deviation of the robot along the planned path. Dynamic obstacle avoidance effect evaluates the robot's ability to avoid dynamic obstacles through the number of collisions and obstacle avoidance reaction time. Smoothness measures the smoothness of the changes in speed and acceleration during the robot's movement to avoid sharp changes. This experiment compares and analyzes the motion control performance of the improved RRT and A* algorithms in different environments, including static environments, dynamic environments, and adaptive path adjustment.

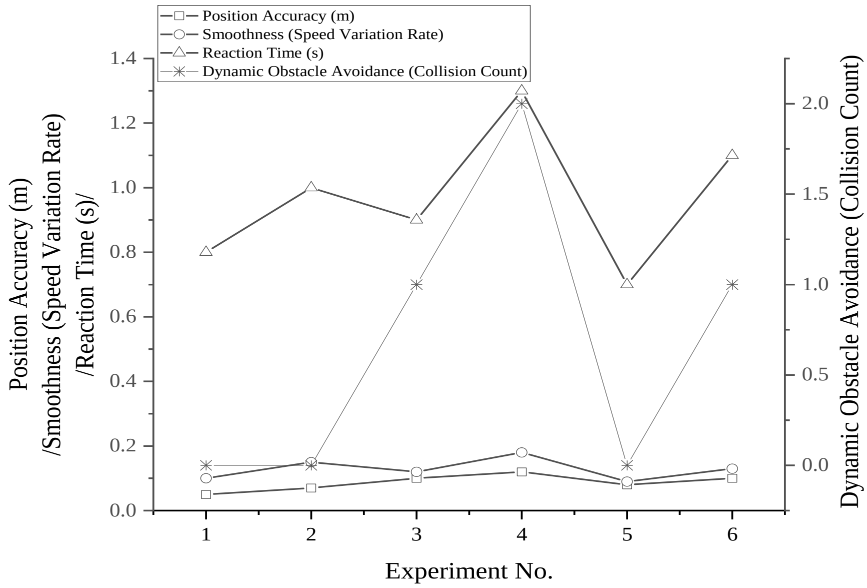

Figure 1. Position accuracy, dynamic obstacle avoidance effect and smoothness

From the experimental data in Figure 1, it can be seen that the improved RRT* algorithm has better performance indicators of position accuracy, number of collisions, reaction time and smoothness than the A* algorithm in the static, dynamic and adaptive path adjustment environments. In the static environment, the position accuracy, number of collisions, reaction time (s) and smoothness of the improved RRT* algorithm are 0.05m, 0, 0.8s and 0.1 respectively. In contrast, the values of the four evaluation indicators of the A* algorithm are 0.07m, 0, 1s and 0.15, respectively.

System performance indicators include computing resource consumption and real-time performance. Computing resource consumption is evaluated by CPU, memory usage, and algorithm computational efficiency. Real-time performance measures the response time of path update and obstacle avoidance. This experiment compares and analyzes the computing resource consumption and real-time performance of the improved RRT and algorithms in different environments.

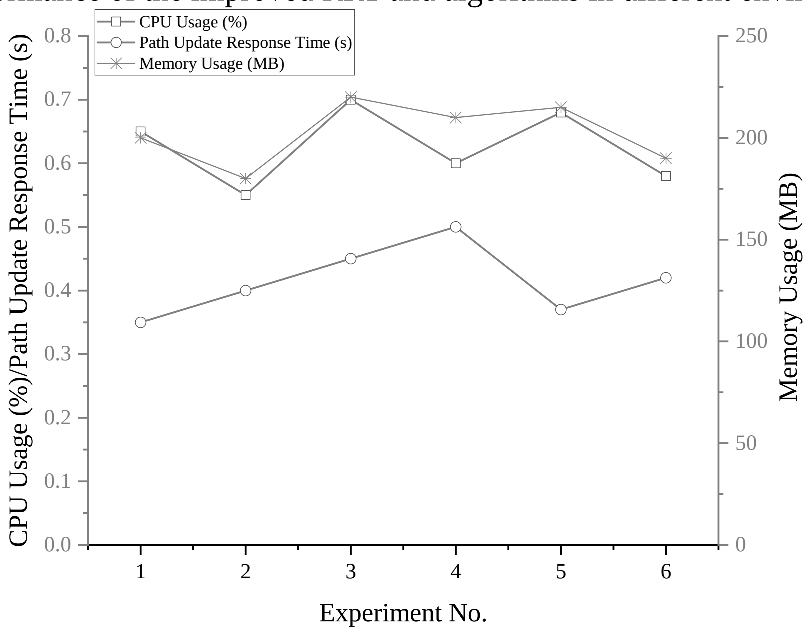

Figure 2. Computational resource consumption and real-time performance

Figure 2's present-day performance statistics and operational resource usage show that, with respect of operational resource use, the CPU usage of the improved RRT in a static environment is 65%, and the memory usage is 200MB, while the CPU usage of A* is 55%, and the memory usage is 180MB, indicating that the computational load of the improved RRT is slightly higher. As the complexity of the environment increases, in a dynamic environment, the CPU usage of the improved RRT rises to 70%, and the memory usage is 220MB, while the CPU usage of A* is 60% and the memory usage is 210MB. In the adaptive path adjustment mode, the resource usage of the improved RRT decreases slightly, with a CPU usage of 68% and a memory usage of 215MB, compared to a CPU usage of 58% and a memory usage of 190MB for A*. Overall, the improved RRT slightly increases the computing resource consumption compared to A**, but its ability to optimize path planning and motion control is also improved. In terms of real-time performance, the path update response time of the improved RRT in a static environment is 0.35 seconds, which is better than the 0.4 seconds of A*.

The collaborative effect analysis is to compare the effects of the collaborative design of path planning and motion control and evaluate their performance in actual operation. This experiment compares and analyzes the differences between the improved RRT and the A algorithm in terms of collaborative effect, execution efficiency and actual operation performance.

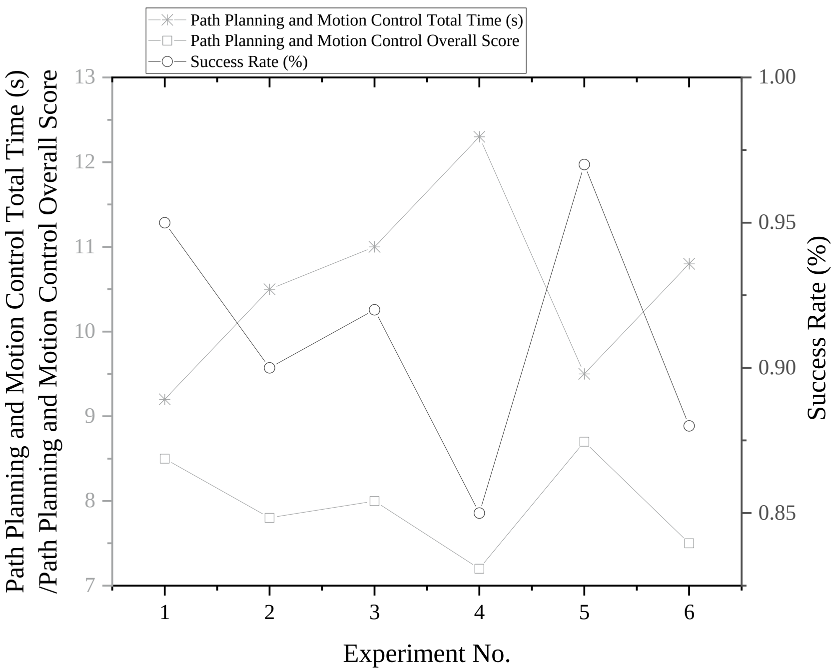

Figure 3. Analysis of synergy effect

In terms of synergy effect, the improved RRT scores 8.5 in a static environment, higher than A*'s 7.8, indicating that its path planning and motion control synergy optimization effect is better. In a dynamic environment, due to the increase in obstacle changes, the improved RRT's score drops slightly to 8.0, but it is still better than A*'s 7.2. In dynamic environments, the success rate of the improved RRT drops slightly to 92%, but is still higher than 85% of A*, indicating that it is more adaptable in complex environments. In the adaptive path adjustment mode, the success rate of the improved RRT reaches 97%, while that of A* is 88%, further proving the robustness and reliability of the improved RRT* in complex scenarios, as shown in Figure 3.

Abnormal situation handling records and analyzes abnormal situations that occur during the experiment, such as path planning failure, collision, control error, etc., and proposes improvement plans.

Table 2. Abnormal situation handling

Experiment ID	Abnormal Situation	Abnormal Occurrence Frequency (times/experiment)	Improvement Plan
1	Path planning failure	2 times	Introduce more sensor information to enhance real-time path re-planning ability
2	Control error	1 time	Optimize motion control algorithm and add precise motion feedback mechanism
3	Collision	3 times	Increase sensing and avoidance mechanism for dynamic obstacles
4	Path planning failure	2 times	Combine topological map and grid map for more precise path planning
5	Control error	1 time	Add path smoothing algorithm to avoid sharp speed changes
6	Collision	2 times	Enhance algorithm real-time performance and optimize dynamic obstacle avoidance strategy

In the experiments in Table 2, a variety of abnormal situations occurred, mainly including path planning failure, control errors, and collisions. In Experiment 1, path planning failure occurred twice, and the improvement plan is to introduce more sensor information to enhance the real-time path replanning capability. In Experiment 2, there is one control error. In order to improve the motion control accuracy, an optimized control algorithm is proposed and a precise motion feedback mechanism is added. In Experiment 3, collisions occurred three times. To address this problem, a dynamic obstacle perception and avoidance mechanism is added to improve obstacle avoidance capabilities. In Experiment 4, path planning failed twice. It is proposed to combine topological maps and grid maps for more sophisticated path planning to optimize path generation.

5. Conclusion

As industrial manufacturing develops towards automation and intelligence, robots have been widely used in many fields. However, since the working scenes of robots are no longer limited to traditional static environments, but tend to be more open and dynamic environments, robots urgently need the ability to autonomously navigate and perform tasks in complex environments! Efficient and accurate path planning is the core link for robots to perform tasks! Therefore, the continuous upgrading of challenges such as dynamic environments and multiple obstacles has put forward higher requirements for robot path planning algorithms. This paper studies the path planning through the RRT* algorithm, and eliminates the excessive turns of the path generated by RRT through B-spline curve fitting, making the path smoother and more natural. The experimental results show that the improved RRT* algorithm shows high flexibility and path adjustment ability in dynamic environments and adaptive path adjustment scenarios, and it is superior to the A* algorithm in terms of position accuracy and obstacle avoidance reaction time. For different application scenarios, the appropriate path planning algorithm should be selected according to actual needs. However, some bottlenecks were also exposed in the study, especially in the real-time performance and computing resource consumption of the algorithm in complex environments. Although the improved RRT* algorithm has advantages in path planning accuracy, its high computing resource consumption and slow path update response time limit its application in tasks with high real-time requirements. The coordinated design of path planning and motion control is crucial to improving the overall performance of robot autonomous navigation. Future research can further optimize the algorithm and improve its real-time and adaptability to cope with more complex environments and task requirements.

References

[1] García E, Villar J R, Tan Q, et al. An efficient multi-robot path planning solution using A and coevolutionary algorithms[J]. Integrated Computer-Aided Engineering, 2023, 30(1): 41-52.*

[2] Gu Y, Zhu Z, Lv J, et al. DM-DQN: Dueling Munchausen deep Q network for robot path planning[J]. Complex & intelligent systems, 2023, 9(4): 4287-4300.

[3] Khlif N, Nahla K, Safya B. Reinforcement learning with modified exploration strategy for mobile robot path planning[J]. Robotica, 2023, 41(9): 2688-2702.

[4] Mohanraj T, Dinesh T, Guruchandhramavli B, et al. Mobile robot path planning and obstacle avoidance using hybrid algorithm[J]. International Journal of Information Technology, 2023, 15(8): 4481-4490.

[5] Tran C C, Lin C Y. An intelligent path planning of welding robot based on multisensor interaction[J]. IEEE sensors journal, 2023, 23(8): 8591-8604.

[6] Shen H, Xie W F, Tang J, et al. Adaptive manipulability-based path planning strategy for industrial robot manipulators[J]. IEEE/ASME Transactions on Mechatronics, 2023, 28(3): 1742-1753.

[7] Zeeshan S, Aized T. Performance analysis of path planning algorithms for fruit harvesting robot[J]. Journal of Biosystems Engineering, 2023, 48(2): 178-197.

[8] Szczepanski R. Safe artificial potential field-novel local path planning algorithm maintaining safe distance from obstacles[J]. IEEE Robotics and Automation Letters, 2023, 8(8): 4823-4830.

[9] Fang W, Ding L, Tian X, et al. A robot welding path planning and automatic programming method for open impeller[J]. The International Journal of Advanced Manufacturing Technology, 2023, 124(5): 1639-1650.

[10] Eshtehardian S A, Khodaygan S. A continuous RRT-based path planning method for non-holonomic mobile robots using B-spline curves[J]. Journal of ambient intelligence and humanized computing, 2023, 14(7): 8693-8702.*

[11] Mier G, Valente J, de Bruin S. Fields2Cover: An open-source coverage path planning library for unmanned agricultural vehicles[J]. IEEE Robotics and Automation Letters, 2023, 8(4): 2166-2172.

[12] Gang Wang, Hongyuan Wen and Jun Zhou. Indoor Positioning Technology and Control of Mobile Home Service Robot[J]. Scientific Programming, 2022, 20