Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/32/20230216

Research on performance comparison of patrol path planning techniques for mobile robots in nuclear power plants

Shuying Liu

Maynooth International Engineering College, Fuzhou University, Fuzhou City, Fujian Province, 350000, China

shuying.liu.2021@mumail.ie

Abstract. Path planning, as a basic problem of mobile robots, is important in the application of industrial patrol robots. This paper takes the nuclear power plant as an example and solves the problem of multi-target patrol path of patrol robot. Firstly, this paper processes multi-target points applying Euler's formula to obtain a reasonable order of patrol target points. Then three path planning algorithms, A* algorithm based on graph search, RRT* algorithm based on sampling and Q-learning algorithm based on reinforcement learning, are applied on path planning, and combined with Minimal-Jerk algorithm for optimizing trajectories. The performance of the results is finally compared using two evaluation metrics. The acceleration variance and route analysis in planar images are used as evaluation metrics in this paper. It is considered that the smaller the acceleration variance, the more smoothly the robot can move. The gentler the route is, the more effective the robot movement is.

Keywords: plant patrol, path planning, trajectory algorithms, performance comparison.

1. Introduction

Industrial patrol is closely related to factory production safety [1]. Mobile robots equipped with various technologies are a good solution for problems such as poor instrumentation patrols, difficult manual inspection operations and lack of effective solutions for high-risk sites. For mobile robots, good patrol route planning is fundamental [2,3]. This paper takes a mobile robot in a nuclear power plant as an example and applies many kinds of path planning algorithms to get the patrol path. The path is then made more suitable for robot movement by a trajectory optimization algorithm. Finally, acceleration variance and path curve analysis are used as evaluation metrics to estimate path stability through acceleration variance, and path smoothness is analyzed through path curve.

For path planning, graph search-based algorithms are the most common path planning algorithms. This kind of algorithm is based on iterative or improved iterative logic, but is prone to exponential explosion problems for high dimensional cases. In practical scenarios, it is certainly best to find the optimal path, but more often than not, it is sufficient to find a sub-optimal or feasible better path. Hence the emergence of sample-based planning algorithms, which are centred on random sampling. Alternatively, reinforcement learning-based path planning algorithms refer to the learning of the system from the environment to the behavioral mapping. It aims to maximize the value of the reward signal function [4]. Reinforcement learning methods may yield results that are better adapted to the real environment than path planning based on a fixed model of graph searching and sampling.

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/32/20230216

However, the resulting paths produced by the three path planning algorithms described above have more inflection points and are not suitable for the robot to follow the path directly. This can lead to more problems with decelerating and turning, making the acceleration of the motion non-linear and demanding on the robot's motor. With a known series of trajectory points, taking into account dynamics constraints and environmental constraints, the use of curves from computer graphics techniques to generate a feasible smooth path that allows for continuous acceleration can result in an optimized route that is robot friendly [5].

Therefore, this paper chooses to select the typical algorithm that works well among the three path planning algorithms, with the A* algorithm representing graph search-based path planning, the RRT* algorithm representing sampling-based path planning and the Q-Learning algorithm representing reinforcement learning-based path planning. After obtaining results by the above three algorithms, this paper applies the Minimal Jerk algorithm to complete trajectory optimization using derivative constraints, continuity constraints, and obstacle constraints [6].

2. Methodology

2.1 Environmental characterization of nuclear power plants

This paper takes the nuclear power plant as an application scenario to design the patrol path of a mobile robot. A section of the 3D design map from the nuclear power plant is intercepted and simulated, as shown in Figure 1 (a). Based on the specific design data, it is transformed into a plane raster map. At the same time, the patrol path has to pass through multiple objective points based on the analysis of the fire-prone points of the specific nuclear power plant equipment. The final plane map is used as the basic environment, as in Figure 1 (b).

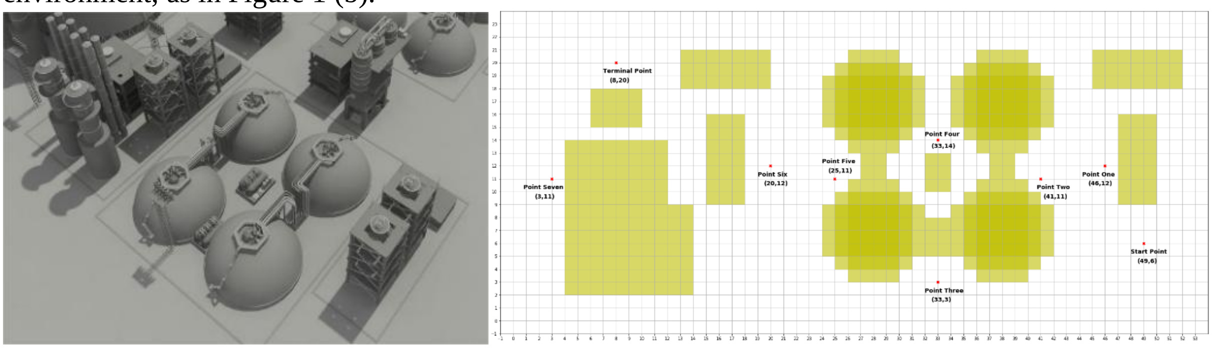


Figure 1. Nuclear power plant scene map.

2.2. Multi-objective path planning for mobile robot patrols

2.2.1. Processing of multi-objective points for path planning. This paper derives relatively better patrol paths by calculating the Euclidean distance between multiple objective points. The equation for calculating the Euclidean distance is shown in Equation 1.

$$\sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{1}$$

The procedure for obtaining the patrol order of multiple objectives is as follows. First, add 9 patrol points to an array. Start with the "start point" first, and at the same time delete the ‘start point’ that has been reached in the array. Calculate the Euclidean distance between the ‘start point’ and the rest of the points in the array [7]. Move to the nearest point and use it as the new start point. Then, delete the current point in the array that has been reached. Finally, the search for the nearest point is repeated until the patrol point ‘end point’ is reached. After experimentation, the final patrol order is: start point - first point - second point - fourth point - fifth point - sixth point - third point - seventh point - end point.

DOI: 10.54254/2755-2721/32/20230216

2.2.2. Search-based planning. The A* algorithm, based on graph traversal search, is very commonly used in path planning tasks. With the research of scholars, it has good performance and accuracy. Therefore, this paper uses the A* algorithm as a representative search-based method to complete the initial planning of patrol paths.

The A* algorithm which is a heuristic algorithm uses a priority queue [8]. Elements in the priority queue are given priority, with the highest priority element being first deleted. The heuristic search is planned through the valuation function of the location. The valuation function is represented as shown in Equation 2.

$$f(n) = g(n) + h(n) \tag{2}$$

where n represents node n, f(n) is the valuation function of node n, g(n)is the actual cost from the initial node to node n in the real state space, and h(n) is the estimated cost of the best path from n to the target node, which can guide the search algorithm towards the end point, mainly using the Euclidean distance or Manhattan Distance.

2.2.3 Sampling-based planning. The RRT algorithm, as a typical sampling-based algorithm, is often compared and discussed with the A* algorithm [9]. However, this paper argues that the A* algorithm has undergone many evolutions, just using the original RRT algorithm for comparison is not the right standard. Therefore, this paper uses an optimized RRT algorithm, the RRT* algorithm, as one of the comparison algorithms.

The path obtained by RRT* algorithm is asymptotically optimal, i.e., the path planned becomes more optimal and less costly as time increases. RRT* is based on the original RRT algorithm, with optimizations for reselecting the parent node and reconnecting.

2.2.4. Reinforcement Learning-based Planning. Q-Learning in Reinforcement Learning is the most basic algorithm, but applied to path planning, it usually can only be based on the grid to achieve four directions movement to obtain a path with the highest Q value. In order to better compare with A* and RRT* algorithms, this paper preliminarily optimizes the Q-learning algorithm so that it can move along eight directions and obtain a reasonable Q value. This simple optimized Q-Learning algorithm is used in this paper as a representative of reinforcement learning-based planning.

Q-learning algorithm aim to learn expectation of the benefit of the state action value function [10]. Q-learning is solved using a value iteration approach, the core iterative formula of which is shown in Equation 3.

$$Q_{k+1}(s,a) = r(s,a) + \gamma \cdot \max_{a^\wedge \in A}\{Q_k(s^\wedge, a^\wedge)\} \tag{3}$$

where $Q_{k+1}(s,a)$is the $(k+1)$th iteration function, s and a denote the current state and the action performed and they belong to state space S and action space A, respectively, $r(s,a)$ denotes the immediate reward after the execution of action a in state s, $s^\wedge$and $a^\wedge$ denote the next state and action, and γ denotes the discount factor.

2.3. Trajectory optimization and performance comparison metrics

2.3.1. Trajectory optimization. Minimum-jerk algorithm directly constrains the position, velocity, and acceleration of the head and tail by a total of 6 equations, so the optimization parameters must provide more than 6 degrees of freedom. 5th order polynomials have 6 coefficients, so the minimum order of a polynomial that meets the requirements is 5, so a 5th order polynomial can be chosen to represent each segment of the trajectory. By constructing an objective function that satisfies the derivative constraint, continuity constraint and obstacle constraint simultaneously, continuity optimization is ultimately accomplished in position, velocity, and acceleration. This paper uses the minimal-jerk algorithm for trajectory optimization, which minimizes the total jerk by solving for the coefficients of each segment of the trajectory while satisfying the constraints.

The trajectory is constrained using the path points obtained from path planning. Continuity constraint enables smooth transitions between adjacent trajectories. Obstacle constraint allows trajectories to be smooth without colliding into obstacles and boundaries. For paths with collisions, the midpoints will be taken for the points at the two ends of the collision and added to the initial path points, and the process will be repeated until no collisions occur.

2.3.2. Performance comparison metrics. The magnitude of acceleration is an important parameter for measuring the degree of change in the motion of an object. The variance of acceleration can be used to indicate whether the change in velocity is smooth or not, i.e., the smaller the variance of acceleration, the smaller the degree of change in velocity. Combined with the mobile robot for analysis, the mobile robot adjusts the acceleration of the movement by adjusting the rotational speed of the motor, and frequent changes in acceleration during the design process will increase the load of the motor, and at the same time led to a large difference between the simulation results and the actual performance. Therefore, this paper considers that the smaller the acceleration change, the better the trajectory optimization effect.

From the obtained map, the path curve can be clearly seen. A smooth path curve is the goal of trajectory optimization. The Minimal-jerk algorithm will increase the path points on the original path when satisfying the obstacle constraints, which leads to oscillations of the path at the corners. Therefore, this paper argues that the smoother the path in the resulting graph, the better the trajectory optimization.

Ultimately, this paper determines to use acceleration variance and path curve analysis as evaluation metrics, estimating path stability through acceleration variance and analyzing path smoothness through path curves.

3. Experiment results and analysis

3.1. Multi-objective path planning for mobile robot patrols

3.1.1. Experimental procedure for three types of path planning. The result of the determined patrol sequence applying the A* algorithm to is shown in Figure 2 (a). As the A* algorithm is optimized based on a raster map, each step has 8 directions, so it is clear from the figure that the path is strictly along the raster line.

Applying the RRT* algorithm and setting the number of iterations to 20,000, the good initial path is found in the determined patrol sequence, and the obtained path results are shown in Figure 2 (b). Because the RRT* algorithm is based on a random tree that iterates through the map after random sampling to find the best path, the figure shows the original path is smoother than the raster-based path planning algorithm.

Setting the number of iterations to 20,000, the good initial path in the determined patrol sequence applying the Q-Learning algorithm is shown in Figure 2 (c). In this paper, the Q-value is calculated by deducting 1 point for "up, down, left and right" actions and 2 points for "up right, down right, up left and down left" actions according to the length of the movement path. This method optimizes movement in only four directions.

DOI: 10.54254/2755-2721/32/20230216

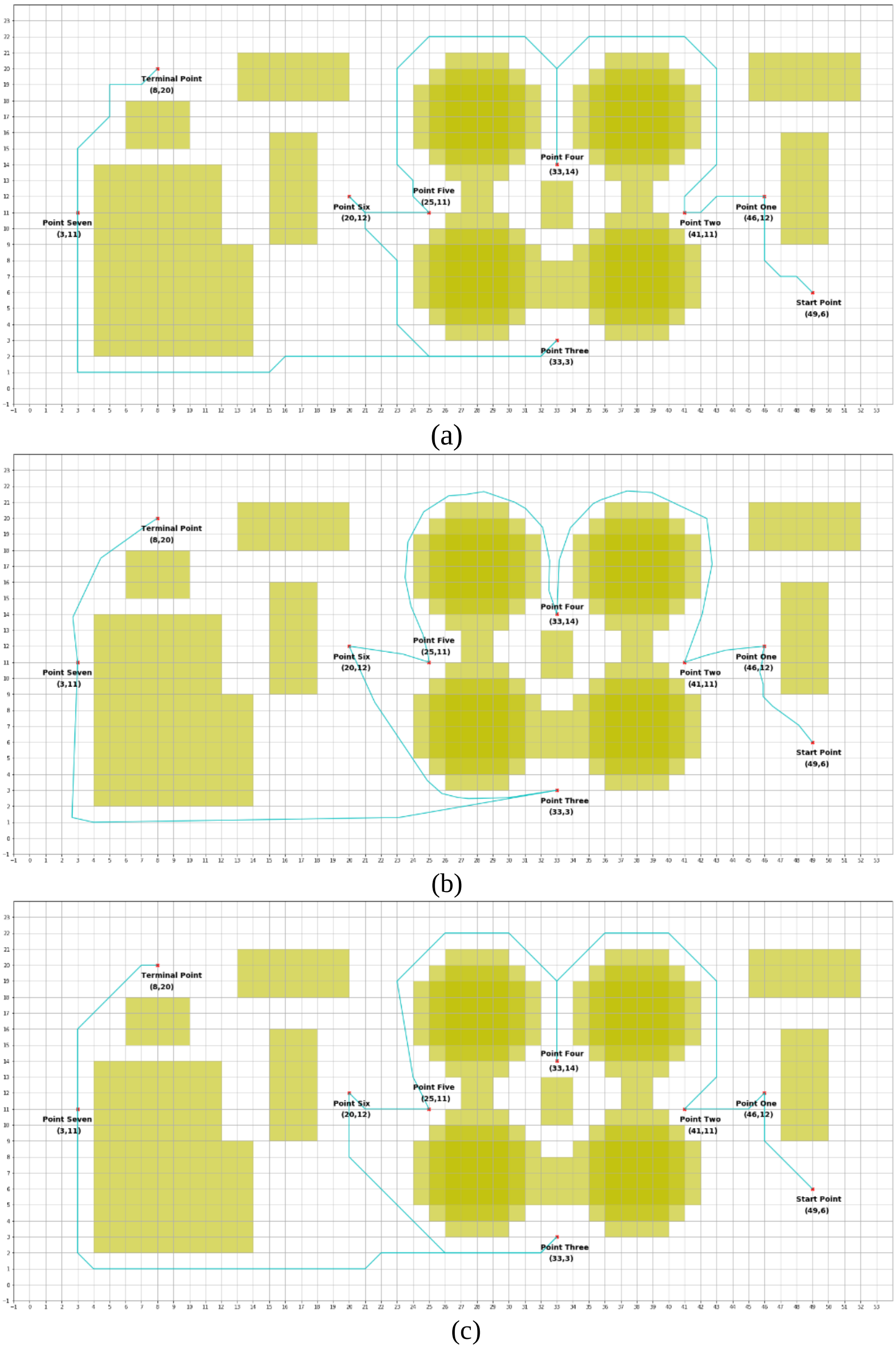
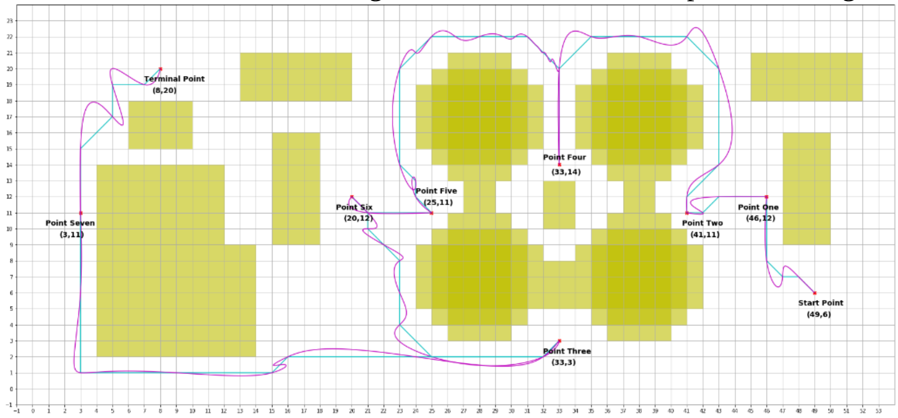


Figure 2. Original path maps applying three representative algorithms.

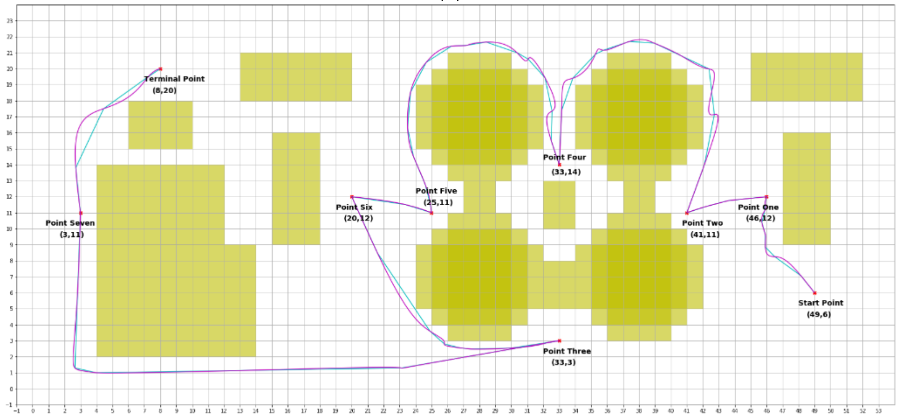
Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/32/20230216

3.2. Trajectory optimization and performance comparison

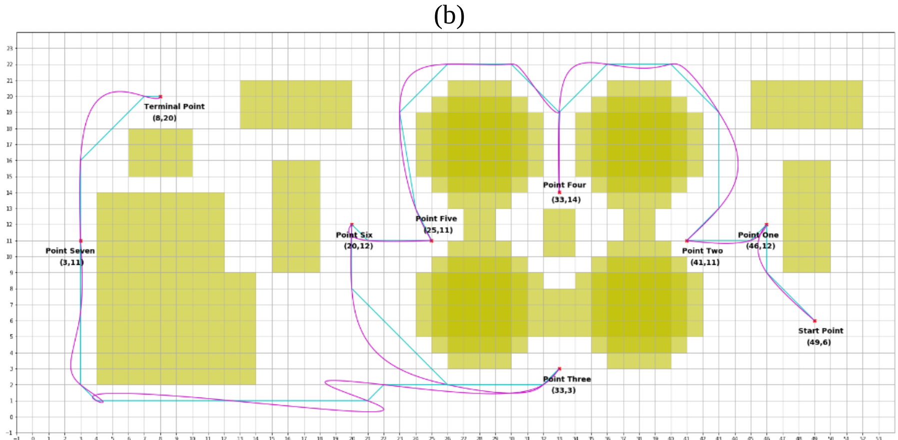
3.2.1. Trajectory optimization. Applying the Minimal-jerk algorithm, trajectory optimization is performed for each patrol segment of the results obtained from the three path planning algorithms separately, and the results obtained after integration of the routes are implicated in Figure 3.



(a)



(b)



(c)

Figure 3. Path maps after trajectory optimization with the application of three representative algorithms.

Most of the trajectory optimization makes the turning points of the path smooth, ensuring that the speed at the start and end points is reduced to zero for operations such as flame recognition and turning. However, in the vicinity of obstacles with circular edges, there is a part of the optimization that allows the curve to have a back-and-forth route in order to pass through all the path points and avoid the obstacles. For this problem, Dubin's curve can be added to the application of the Minimal jerk algorithm. Dubin's curve finds the shortest smooth path connecting the points in a given curvature range using arcs and line segments, given two points in the plane and the direction of motion.

Therefore, combining the metrics of the route in the image, the Q-learning algorithm is optimal, followed by the A* algorithm. the RRT* algorithm is less effective after trajectory optimization as there is already some optimization of the route itself.

3.2.2. Performance comparison. Data visualization of the time and acceleration obtained after trajectory optimization for each section of the path was also carried out, as shown in Table 1.

Table 1. Data visualization after trajectory optimization.

	Segment	Time	Acc_max	Acc_min	Acc_var
A star	1	8	1.195646795	-1.099370938	
	2	6	3.473192473	-3.950846006	
	3	16	1.132717089	-1.944708846	
	4	40	1.0414029	-0.82221811	
	5	4	6.444610778	-4.441829464	
	6	18	2.804554038	-2.971793668	
	7	18	2.902686338	-3.122272144	
	8	10	2.114110073	-1.928609996	
	Total	120	6.444610778	-4.441829464	1.335669913
RRT star	1	16	0.322229195	-0.941182813	
	2	8	1.026808454	-1.344704875	
	3	24	1.026808454	-1.344704875	
	4	30	0.730899365	-0.904737421	
	5	4	2.57232128	-2.103306515	
	6	16	1.776005243	-0.928693255	
	7	14	4.982078066	-5.970267201	
	8	8	1.402541805	-0.734994606	
	Total	120	4.982078066	-5.970267201	1.432004361
Q-learning	1	4	5.937767322	-3.36799225	
	2	4	3.322899681	-3.910393499	
	3	12	2.207471676	-2.759411464	
	4	16	1.846009635	-1.603679727	
	5	4	6.444610778	-4.441829464	
	6	8	3.344967509	-2.853384159	
	7	16	7.844555797	-9.923111617	
	8	6	2.202703002	-3.505841076	
	Total	70	7.844555797	-9.923111617	5.980346245

In Table 1, Time indicates the time at which it will move after optimisation. Acc_max indicates the maximum value of acceleration in the patrol path. Acc_max indicates the minimum value of acceleration in the patrol path. In the Total row, the first is the calculated global time, and the second and third are the maximum and minimum acceleration values along the entire patrol path. Acc_var indicates the variance of the acceleration over the full patrol path.

The data implicate after trajectory optimization the variance of acceleration for the A* algorithm is approximately 1.34, the variance of acceleration for the RRT* algorithm is approximately 1.43 and the variance of acceleration for the Q-learning algorithm is approximately 5.98. Therefore, based on the metrics, this paper concludes that the optimized A* algorithm and RRT* algorithm have lower motor requirements for the robot.

Combining the optimized trajectories shows that the A* algorithm and the RRT* algorithm add more path points into the optimization in order to avoid obstacles near circular obstacles. This resulted in an increase in the robot's movement time, accompanied by an overall decline in acceleration. Therefore, this could be the reason for the small difference in their acceleration.

4. Conclusions

In summary, for the A* algorithm, the algorithm has a short running time and also results in a better path. After trajectory optimization there are some oscillating routes around circular obstacles, but the overall acceleration variance is small and less demanding on the robot motor.

For the RRT* algorithm, the algorithm has a long run time and may be more demanding on the robot drive board. Because the algorithm itself is not limited by the raster map, the path obtained is more closely matched to the path after trajectory optimization. However, the optimization time is limited in order to go through all the path points, so the robot has a slower travel time and is less demanding on the robot motors.

For the Q-learning algorithm, the algorithm runs in few time, while the result is a better path. Because of the algorithm's reward system, the robot's movement path is more realistic in order to get a higher score, so the algorithm obtains fewer path points. This means more freedom in the path, making the optimization of the trajectory simple. However, in the middle of each path, the robot will be faster. As a result, the acceleration of the robot can fluctuate considerably while ensuring that the starting and ending points are reduced to zero. This is reflected in the large variance of the acceleration obtained, which is more demanding on the robot's motor.

The addition of Dubin's curve to the Minimal-jerk algorithm will be considered to improve the oscillation of the path. In order to get good results, many path points are generated during initial planning. These path points will be compulsory in the subsequent optimisation. Delete some of the path points involved in the optimisation may result in a good optimised path and faster running speed.

References

[1] Alhassan A B, Zhang X, et al. Power transmission line inspection robots: A review, trends and challenges for future research [J]. International Journal of Electrical Power & Energy Systems, 2020, 118: 105862.

[2] Sánchez-Ibáñez JR, Pérez-del-Pulgar CJ, García-Cerezo A. Path Planning for Autonomous Mobile Robots: A Review. Sensors. 2021; 21(23):7898.

[3] Mohanty, P.K., Singh, A.K., Kumar, A., Mahto, M.K., Kundu, S. (2022). Path Planning Techniques for Mobile Robots: A Review. In: Abraham, A., et al. Proceedings of the 13th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2021). SoCPaR 2021. Lecture Notes in Networks and Systems, vol 417. Springer, Cham.

[4] Chen, H., Ji, Y. & Niu, L. Reinforcement learning path planning algorithm based on obstacle area expansion strategy. Intel Serv Robotics 13, 289–297 (2020).

[5] F. Bullo and W. T. Cerven, "On trajectory optimization for polynomial systems via series expansions," Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187), Sydney, NSW, Australia, 2000, pp. 772-777 vol.1, doi: 10.1109/CDC.2000.912862.

[6] Richter, C., Bry, A., Roy, N. (2016). Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments. In: Inaba, M., Corke, P. (eds) Robotics Research. Springer Tracts in Advanced Robotics, vol 114. Springer, Cham.

[7] Y. Yuan, J. Liu, W. Chi, G. Chen and L. Sun, "A Gaussian Mixture Model Based Fast Motion Planning Method Through Online Environmental Feature Learning," in IEEE Transactions on Industrial Electronics, vol. 70, no. 4, pp. 3955-3965, April 2023, doi: 10.1109/TIE.2022.3177758.
[8] Karaman S, Frazzoli E, Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research. 2011;30(7):846-894. Doi:10.1177/0278364911406761.
[9] J. Chen, M. Li, Y. Su, W. Li, Y. Lin. "Direction constraints adaptive extended bidirectional A* algorithm based on random two-dimensional map environments", Robotics and Autonomous Systems, 2023Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research. 2011;30(7):846-894. doi:10.1177/0278364911406761.
[10] C. Jin, Y. Lu, R. Liu and J. Sun, "Robot Path Planning Using Q- Learning Algorithm," 2021 3rd International Symposium on Robotics & Intelligent Manufacturing Technology (ISRIMT), Changzhou, China, 2021, pp. 202-206, doi: 10.1109/ISRIMT53730.2021.9596694.