

# The path planning of hybrid algorithm for patrol robot in complex environment

Zhiyong Wu<sup>1</sup> and Weiwei Hu<sup>2\*</sup>

<sup>1</sup>Nanjing University of Posts and Telecommunications, NanJing, 210046, China  
wzykk09@163.com

<sup>2</sup>Nanjing University of Posts and Telecommunications, NanJing, 210046, China  
hww1986116@126.com\*

**Abstract.** This paper proposes a improved hybrid obstacle avoidance algorithm to plan path in complex environment. This algorithm is combined with two improved algorithms which are improved artificial potential field (IAPF) and improved A-star. Before making path planning for the patrol robots(P-RBs), we need a suitable map for navigation, we expanded the edges of the obstacles in the map. This processing can ensure that the planned path is safe and collision-free, and the planning makes this planning algorithm universal by adjusting the expansion area. The expansion area can be adjusted according to the radius of the robot and also has a good planning effect. The IAPF method has a small amount of calculation, easy to implement, and has good adaptability to dynamic environment obstacle avoidance, but it does not have certain adaptability to complex environment. The algorithm gets stuck in local minima and U-shaped obstacles. In order to solve this problem, we introduced IA\* to complete the planning of the remaining paths. The path planned by this hybrid path planning algorithm can plan a suitable path no matter what kind of complex environment. In addition, considering the limitation of four-wheel robot steering, we use quasi-uniform B-spline to smooth the path, shorten the length of the path and reduce the curvature of the path. Finally, through the path planning of the hybrid algorithm, a safe path can be gained.

**Keywords:** hybrid algorithm, Safe avoidance, complex environment, patrol robot.

## 1 Introduction

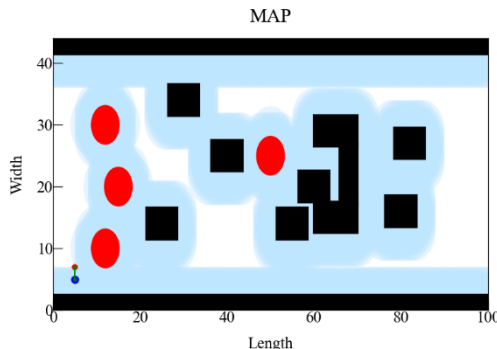
Intelligent patrol robots have great acceptance, but the control and navigation of these devices are very difficult, and the lack of the ability to handle fixed obstacles and avoid obstacles is a basic requirement for these systems. [1] Using patrol robots to carry out inspection with measuring equipment can not only reduce the danger to inspection personnel, but also improve the efficiency of inspection. [2] patrol robot is a robot system with significant research results in sensor perception and autonomous obstacle avoidance in a changing environment. [3] Robots can perform autonomous task inspection without human operation, and can also replace manual exploration and complete some detection tasks in some dangerous environments, which not only improves work

efficiency, but also reduces labor costs. Path optimization algorithms are divided into two groups, namely, global path planning (GPP) in unknown environment and local path planning in known environment [4]. Local path planning, on the other hand, uses sensory information to create local maps to avoid dynamic obstacles in complex environments. The main goal of LPP is to find a collision-free path by avoiding dynamic obstacles instead of finding the best path [5]. Researchers have proposed many path planning methods based on GPP, LPP, or hybrid algorithms to determine the optimal path. [6]

The a-Star algorithm is the most useful method to find the shortest path, and it has better results in static environments. It also has some disadvantages, such as many redundant nodes and slow path planning in wide environments. Dijkstra's algorithm also considers the path weight and finds the shortest path in the directed graph. Similar to A-Star algorithm, Dijkstra algorithm also has shortcomings when dealing with large environments [7]. Lee's algorithm is effectively used for route planning of autonomous vehicles. The algorithm is optimal, easy to use, and computationally inexpensive. However, Lee's algorithm is generally slower than the A-Star algorithm [8]. The D\* algorithm is equivalent to the dynamic A-Star algorithm for finding collision-free paths in environments with moving obstacles. The D\* algorithm performs better than the A-Star algorithm in complex environments. However, the main drawback of this algorithm is the high memory cost [9]. RRT algorithm has high search efficiency but usually plans a long path[10]. APF has several advantages, including ease of use, easy implementation, good dynamic performance, and the ability to generate a safe path to a goal in real time [11]. However, in complex environments, it often falls into local minima and U-shaped obstacles, leading to path planning failure [12]. In the actual motion process, the path planned by the patrol robots must comply with the kinematic and dynamic constraints, so the path planned by the robot should be as smooth as possible. The commonly used path smoothing methods are: polynomial, Bessel curve, B-spline curve and other fitting methods [13,14,15].

The remainder of this paper are organized as follows. Section 2 presents the processing of obstacles map, Section 3 introduces the evolution of hybrid algorithm, the improvement of the APF algorithm, the improvement of the A\* algorithm, IAPF- IA\* in complex environments, and the planning smoothing using quasi-uniform B-splines, Section 4 reports the simulation results. Section 5 concludes all the work.

## 2 Optimization of the obstacle map



**Fig. 1.** Obstacle dilation treatment

We gain a 100x50 grid map by lidar SLAM mapping, and the coordinates of the obstacles on the map are recorded on the map. The obstacles consist of rectangle and circle, and the blue area is the area that has been dilatation hardening and is equivalent to the obstacle area. The path generated by the P-RB in the grid map may intersect with a corner of the obstacle or the path passes through the obstacle. Although the path has been successfully planned, it does not take into account the actual movement of the robot, which makes the planned path unsafe. In order to solve the above possible situations, we adds the dilation layer to the obstacles in the raster map. In order to prevent the planned path from intersecting with the obstacle, the expansion area is set on the edge of the obstacle which is a little bit larger than the moving width of the robot. As shown in Fig.1, the introduction of the expansion layer can ensure that the P-RB will not collide with obstacles in the actual movement process, and improve the safety of robot movement. Moreover, because there is no need to search the nodes of obstacles when the A\* algorithm starts planning, the number of search nodes of algorithm is effectively reduced, and the search time is reduced. Let's assume the obstacle has a map coordinate  $(ox, oy)$  (the obstacle has a shape), and then we have an extended boundary point  $(x_i, y_i)$  that satisfies the following equation.

$$\{(x_i, y_i) \mid (x_i, y_i) \in \sqrt{(ox - x_i)^2 + (oy - y_i)^2} \leq robot_{radius}, (x_i, y_i) \notin (ox_i, oy_i)\} \quad (1)$$

In addition, in path planning, the grid map we actually use is different from the actual map. On the basis of the actual grid map, we first need to expand it, and regard the obstacles within a certain range as obstacles to ensure that the P-RB does not collide with the actual obstacles. In the APF or A\*, we only need to scan the obstacle boundary and do not need to consider the interior of the obstacle, which can reduce the amount of calculation of the repulsion force of the obstacle to the P-RB.

### 3 hybrid algorithm (IAPF- IA\*)

For the more complex static environment, it is necessary to not only find the appropriate path, but also ensure the safety of the car body. Therefore, we introduce a hybrid algorithm that is to IAPF- IA\* algorithm for navigation. At the beginning, IAPF method was used for navigation. In the IAPF, the repulsive force of obstacle to the P-RB and the repulsive force of target point to the P-RB are calculated to infer the next coordinate point of the robot. When the P-RB falls into U-shape obstacle, IA\* algorithm was used to find whether there was a path to the target point. while the path exists, the remaining planning path will be the planning path.

### 3.1 APF method and its improvement

For the traditional APF, the attraction force on the P-RB is proportional to the distance from the target point. When the P-RB reaches the target point, the attraction force becomes zero, so the attraction force on the vector space is as follow:

$$\overset{\mathbf{1}}{F}_{att} = k_{att} \overset{\mathbf{1}}{d}(x_{current}, x_{gobal}) \quad (2)$$

$\overset{\mathbf{1}}{F}$  is the force on the robot in the direction of the target point;  $\overset{\mathbf{1}}{d}$  is the distance vector between the current position of the P-RB and the target point, whose direction is consistent with the direction of the gravitational vector,  $x_{current}$  is the current position coordinate of the P-RBS, and  $x_{gobal}$  is the position coordinate of the target point.

Suppose  $n$  obstacles, or obstacle boundary points, are detected on an existing map. The basic principle of repulsion force in APF is the interaction force between charges. We regard the P-RB and the obstacle as the same kind of charge, and the repulsion force between them is ignored when it is outside a certain range. When the distance between them is less than a certain range, it shows mutual repulsion, and the closer they are to the obstacle, the greater the repulsion. The repulsive force of the P-RB by the  $i$  obstacle is as follow:

$$\overset{\mathbf{r}}{F}_{rep_i} = \begin{cases} \frac{1}{2} k_{rep} \overset{\mathbf{r}}{e}(x_{obs_i}, x_{current}) \left( \frac{1}{\rho_i} - \frac{1}{\rho_0} \right)^2, & \rho < \rho_0 \\ 0 & , \rho \geq \rho_0 \end{cases} \quad (3)$$

Then the sum of the repulsive force vectors received by the P-RB at the spatial position is as follow:

$$\overset{\mathbf{r}}{F}_{rep} = \sum_{i=1}^N \overset{\mathbf{r}}{F}_{rep_i} \quad (4)$$

Where  $\overset{\mathbf{1}}{F}_{rep}$  is the repulsive force vector whose direction is directed by the obstacle to the P-RBS,  $k_{rep}$  is the repulsive force coefficient,  $\overset{\mathbf{1}}{e}(x_{obs_1}, x)$  is the unit vector whose direction is consistent with the repulsive force,  $\rho_0$  is the radius of repulsion of the obstacle and  $\rho_i$  is the distance from the  $i$  obstacle to the current position of the P-RBS, Its size is as follows:

$$\rho_i = |x_{current} - x_{obs_i}| \quad (5)$$

The resultant force on the inspection robot is as follows:

$$\dot{\mathbf{F}} = \dot{\mathbf{F}}_{att} + \dot{\mathbf{F}}_{rep} \quad (6)$$

The traditional APF has disadvantages such as unreachable targets, local minima, and U-shaped obstacles that cannot be crossed. The P-RB simply rely on the traction of the gravitational repulsive force. Once the gravitational force is equal to the repulsive force, the P-RB will maintain balance. The calculation of the traditional APF function will make the P-RB trapped in the most local minimum and U-shaped obstacles unable to escape, which leads to the situation in Figs. 2 and Figs. 3. It can be seen from the image that the traditional APF cannot escape the trap and find other possible paths to the target point. Comparing Figs. 2 and Figs. 3, it can be seen that if the radius of the P-RB is considered, once it cannot pass the obstacle, the next step of path planning cannot be carried out. This makes the APF method unable to apply to more complex environments, and considering the shape of the car, it is necessary to re-calculate the gravitational repulsion function.

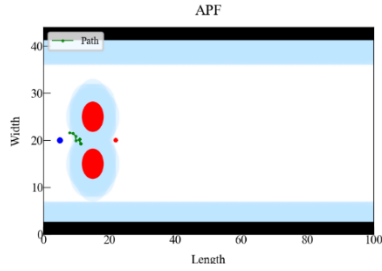


Fig. 2. Local minimum

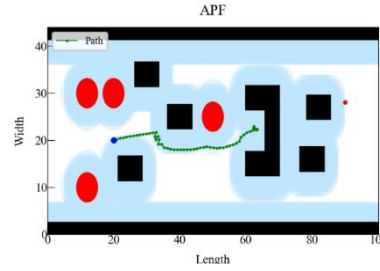


Fig. 3. U-shaped obstacles

For the P-RB in Fig. 4 with Ackerman steering, its front wheel Angle is assumed to be  $\delta(\delta \in (-\theta_1, \theta_1))$ , and the initial direction of the vehicle is stipulated to remain parallel to the X-axis at the initial time. When the next coordinate point direction Angle of the P-RB is as follows.

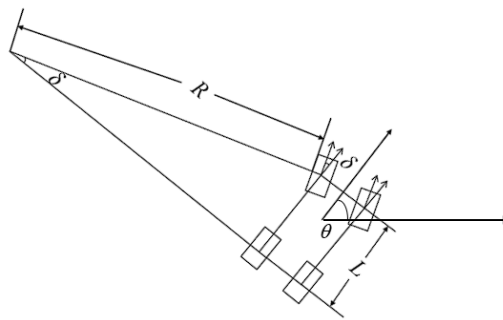
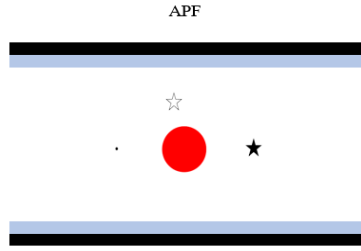


Fig. 4. Steering structure of P-RBs

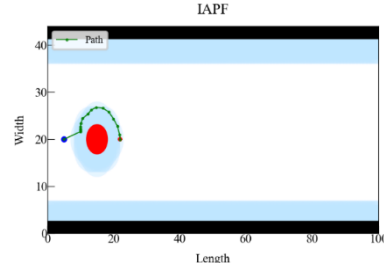
$$\theta = \frac{\vec{F}}{|\vec{F}|} \quad (7)$$

In addition to turning in place, in the normal driving process, when  $\theta > \theta_1$  or  $\theta < -\theta_1$ , we set  $\theta = \theta_1$  or  $\theta = -\theta_1$  to achieve the effect of turning angle constraint.

In other side, in order to cope with the local minimum problem, we add the way of virtual target points to guide[16].



**Fig. 5.** Join the virtual target point



**Fig. 6.** Virtual point navigation

One of the reasons for this is that when the gravitational force and the repulsive force are in the same direction, the vehicle cannot continue to move forward because of obstacles. Therefore, we add a direction correction. When such a situation occurs, we set a virtual target point on the left side of the P-RB to guide the P-RB to escape from the scene in Fig. 5 The result of the improvement is shown in Fig. 6. In other case, we can reduce coefficient of attraction or increase the repulsion factor .

Through introducing a distance factor, the problem of unreachable targets can be improved. The principle is shown in the following equation:

$$\vec{F}_{1rep} = \begin{cases} \frac{1}{2} k_{rep} \vec{r} e^{-k_d \rho} & , \rho < \rho_0 \\ \left( \frac{1}{k_d \rho_i} - \frac{1}{\rho_0} \right)^2 & , \rho < \rho_0 \\ 0 & , \rho \geq \rho_0 \end{cases} \quad (8)$$

$$k_d = \begin{cases} \sin \varphi + m_d & \varphi \in (0, \theta) \\ 1 + m_d & \text{others} \end{cases} \quad (9)$$

In real,  $k_d$  is the improved distance adjustment factor between the vehicle and the obstacle, and  $m_d$  is a constant. In the APF,  $m_d$  is equivalent to 0. In order to make obstacle avoidance planning more efficient and secure,  $m_d$  is taken to be 0.6.

Then the sum repulsion function is as follows:

$$\overset{\cdot}{F}_{1rep} = \sum \overset{\cdot}{F}_{rep_i} + \overset{\cdot}{F}_{1rep} \quad (10)$$

For U-shaped obstacles or other complex environments, we make use of the IA\* algorithm to realize path planning, as shown below.

### 3.2 Principle of IA\* algorithm

The IA\* is improved on the basis of A\*. The IA\* algorithm is a heuristic search algorithm, which is a global path planning algorithm. Because IA\* algorithm needs to search the map to find the appropriate path, it will cost more memory compared with the APF algorithm, but the search efficiency is high and the shortest path is obtained. In the grid map, the P-RB starts path planning with the grid it is currently in as the starting node. Eight Adjacent grid around this grid are searched and they are considered as eight candidate nodes. Then according to the evaluation function of the IA\* algorithm, as shown in Equation 12, the node with the minimum cost is selected as the current node, and the search continues for the eight nodes around this node. The loop executes this process repeatedly until the target point is found and the path planning is completed. The movement of the vehicle basically conforms to the curve model of Fig. 10, in order to make the A\* algorithm more suitable for the steering design of the P-RB, we reduce the searching directions from the original eight to six.

The heuristic function is defined as follows:

$$f(n) = g(n) + h(n) \quad (11)$$

The best cost function is selected based on repeated experiments as follows:

$$h(n) = |x_{current} - x_{goal}| + |y_{current} - y_{goal}| \quad (12)$$

Where  $n$  represents the current node where the current P-RB is located, and  $g(n)$  is the cost value, which refers to the actual movement cost from the starting point to the current node, which is selected according to the six directions of movement.  $h(n)$  is the estimated value, or cost function, which is the expected cost from the current node to the next node.

### 3.3 Quasi-uniform B-splines curve fitting

In path planning, we only get a series of discrete points, and do not directly get a smooth path curve. There are still some points with large curvature in the set of points produced by original path planning. Quasi-uniform B-spline fitting is adopted in this paper. When the Bezier curve uses more control points, it will obviously change the original curve shape and deviate from the original local planning path, and quasi-uniform B-spline has obvious advantages in optimizing the inflection point of the path. The curve fitted by quasi-uniform B-spline is more in line with the requirements of path planning.

Assuming  $P_0, P_1, P_2, \dots, P_n$ , a total of  $n+1$  control points, these control points are used to define the direction and limit of the curve, then the  $k$  order B-spline curve with  $n+1$  control points is defined as follows.

$$p(u) = [P_0, P_1, \dots, P_n] \begin{bmatrix} B_{0,k}(u) \\ B_{1,k}(u) \\ \dots \\ B_{n,k}(u) \end{bmatrix} = \sum P_i B_i(u) \quad (13)$$

Where  $B_{i,k}(u)$  is the  $i$  to  $k$  order B-spline basis function corresponding to the control point  $P_i$ ,  $k \geq 1$ ;  $u$  is the independent basis function,  $B_{i,k}(u)$ 's value is as follows.

$$B_{i,k}(u) = \begin{cases} \begin{cases} 1, & u_i \leq u \leq u_{i+1}, k=1 \\ 0, & \text{others} \end{cases} \\ \frac{u-u_i}{u_{i+k-1}-u_i} B_{i,k-1}(u) + \\ \frac{u_{i+k}-u}{u_{i+k}-u_{i+1}} B_{i+1,k-1}(u), & k \geq 2 \end{cases} \quad (14)$$

If the denominator is zero and numerator is also zero, convention that the whole term is zero. If the numerator is not zero, the denominator is one by convention.  $u_i$  is a non-decreasing sequence of continuously changing values called node vector, whose first and last values are generally defined as zero and one, and the sequence is as follows.

$$[u_0, u_1, \dots, u_k, u_{k+1}, \dots, u_n, u_{n+1}, \dots, u_{n+k}] \quad (15)$$



## 4 Experiment

Step 1: establishing the obstacle map, and expand the map obstacle edge.

Step 2: Initialize parameters, starting point, end point, radius of P-RB, IAPF parameters: attraction coefficient, repulsion coefficient, repulsion radius, step size. IA\* parameters: step size, radius of repulsion.

Step 3: According to model of IAPF method, calculate the resultant force of obstacles to infer the next coordinate point during the inspection process.

Step 4: Determine whether the P-RB has reached the target point, if it has, complete the path planning, if it has reached the target point and changes with time, the current coordinate point does not change, and start the IA\* algorithm.

Step 5: Put the starting point into the open list, traverse the open list, and calculate the heuristic function of the points in the open list. Find the node with the smallest distance from the parent and treat it as the current node to process. If the node is unreachable or on a closed list, ignore it. Otherwise, do the following.

Step 6: If the node is not in the list, add it to the list. Find the point in the list with the lowest cost from the parent node, and set it as the parent node.

Step 7: Move the current parent node to the closed list. Close all the elements in the list no longer need attention. The program terminates when one of the following conditions is met, in one case, the destination is added to the open list (the path is now found), in other case, the end cannot be found, and the open list is empty (there is no path). If the endpoint is found, the shortest path is that from the endpoint, each node follows its parent node until it reaches the start.

Step 8: Flip the path and store the path coordinates into the waypoints collection.

Step 9: The waypoints are processed with a path smoother to obtain the final planning path, and the path planning is over.

Step10: Model predictive control is used to track the path.

The experimental environment uses CPU AMD Ryzen 5 5600H with Radeon Graphics/3.30 GHz, and the simulation environment is carried out on PyCharm.

### 4.1 Computation time comparison

**Table 1.** Algorithm planning time

Algorithm	Time(s)
A*	1.35
IA*	1.29
IAPF	2.41
RRT	5.72
D*	1.45

We repeat the experiment three times for some Several algorithms. Through the comparison in table 1, we can conclude that IA\* can find the path to the target point faster and the path is relatively smooth in Fig. 10. The RRT has a small computation, but its path planning time is longer in Fig. 7. In the D\*, The path Angle is too large to prevent the P-RB from turning in Fig. 8. The algorithm with the expansion area can keep a certain distance from the obstacle in Fig. 9, Fig. 10, Fig. 11, Fig. 12. And IAPF-IA\* can realize path planning in complex environments in Fig. 12.

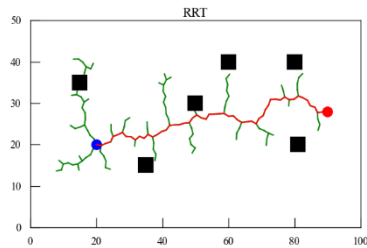


Fig. 7. RRT Planning

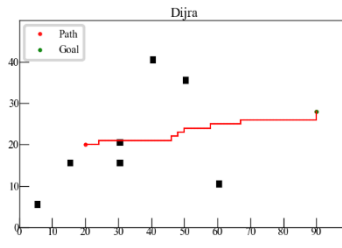


Fig. 8. D\* Planning

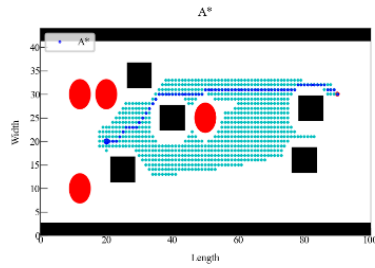


Fig. 9. A\* Planning

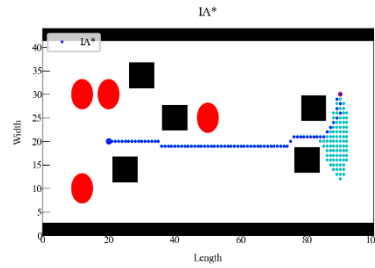


Fig. 10. IA\* Planning

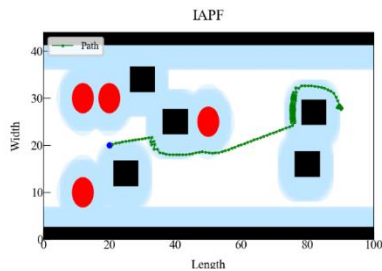


Fig. 11. IAPF Planning

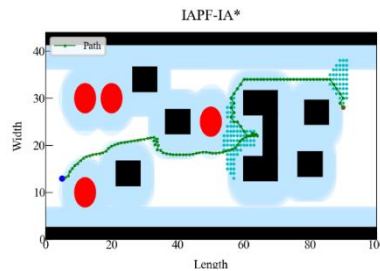


Fig. 12. IAPF-IA\* Planning

## 4.2 Curve fitting performance

We compare and analyze three kinds of curves, they are using broken line to link (Green line), Bessel curve fitting (Blue line) and B-spline fitting (Red line).

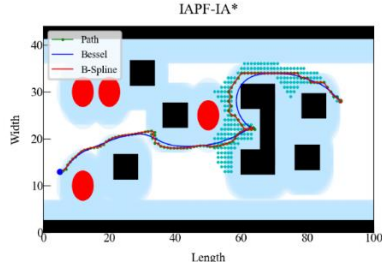


Fig. 13. Curve Fitting

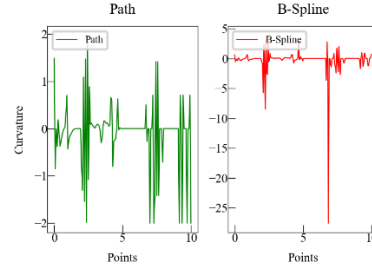


Fig. 14. curvature change

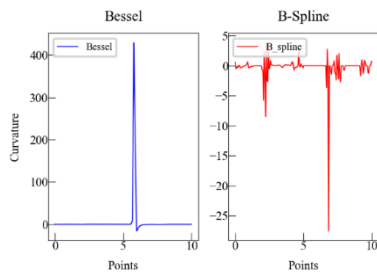


Fig. 15. curvature change

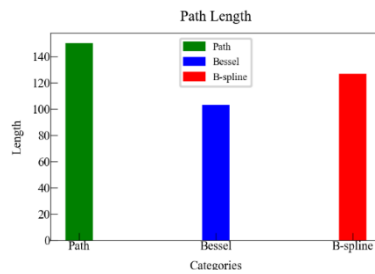


Fig. 16. Path length comparison

Through the experimental analysis, the B-spline has better local characteristics under the condition of ensuring the safety of the car body from Figure 13 and Figure 15. Although the path of the Bessel curve is smoother, but the smooth path deviates from the original trajectory and there is the danger of collision with obstacles in Figure 13. The B-spline can effectively smooth the curve in Figure 15 while shorting the path in Figure 16, and ensure the safety of the car body.

## 5 Conclusion

Through the analysis of navigation structure of patrol robot, firstly, we deal with the grid map of navigation to ensure that the vehicle does not collide with obstacles. By improving the repulsion function of the traditional APF, and putting forward improved method for the APF to fall into a local minimum, and the motion of the APF is constrained by considering the minimum turning radius of the P-RB. For the U-shaped complex environment, though we have gained IAPF, it cannot escape the U-trap or complex environment, so we introduce an improved IA\* to help the P-RB find the best path. Through comparative experiments, we verify the performance of the hybrid algorithm in the complex environment, the time is short, and can ensure the safety of the P-RB. Finally, we compare several curve fitting methods and verify that the performance of quasi-uniform B-spline is more suitable for P-RB and get a smooth path in a complex environment.

**Acknowledgement.** This work was supported by the National Natural Science Foundation of China under Grant 12002172.

## References

1. Zafer Duraklı, Vasif Nabiyev.: A new approach based on Bezier curves to solve path planning problems for patrol robots, *Journal of Computational Science*, Volume,58(2022)
2. MinKi Choi, Woojin Chung, Yongkwan Kwon, Hyungchul Kim.: Safe and high speed navigation of a patrol robot in occluded dynamic obstacles. *IFAC Proceedings Volumes*, Volume 41, Issue 2, Pages 1685-1690(2008)
3. Zhihuan Chen, Huaiyu Wu, Yang Chen, Lei Cheng, Binqiao Zhang.: Patrol robot path planning in nuclear power plant using an interval multi-objective particle swarm optimization algorithm. *Applied Soft Computing*, Volume 116 (2022)
4. Qian Zhou, Yang Lian, Jiayang Wu, Mengyue Zhu, Haiyong Wang, Jinli Cao.: An optimized Q-Learning algorithm for patrol robot local path planning. *Knowledge-Based Systems*, Volume 286(2024)
5. Yang Jiang, Yifan Yu, Ruohuai Sun, Bin Zhao.: Research on dynamic path planning method of electric patrol robots based on fuzzy neural network. *Energy Reports*, Volume 9, Supplement 8, Pages 483-491(2023)
6. Li X , Hu X , Wang Z ,et al.: Path Planning Based on Combaiaon of Improved A-STAR Algorithm and DWA Algorithm. [C]//2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)(2020)
7. Khaled Alawadi, Rim Anabtawi, Ibrahim Ismail, Muna Alshehhi.: From local to global: Uniting neighborhood planning units for more efficient walks, *Transportation Research Interdisciplinary Perspectives*, Volume 23(2024)
8. Donghun Yu, Myung-II Roh, Method for anti-collision path planning using velocity obstacle and A\* algorithms for maritime autonomous surface ship, *International Journal of Naval Architecture and Ocean Engineering*, Volume 16, (2024)
9. Miyombo Ernest Miyombo, Yong-kuo Li.: Optimal path planning in a real-world radioactive environment: A comparative study of A-star and Dijkstra algorithms. *Nuclear Engineering and Design*, Volume 420(2024)
10. R. Mashayekhi, M. Y. I. Idris, M. H. Anisi, I. Ahmedy and I. Ali.: Informed RRT\*-Connect: An Asymptotically Optimal Single-Query Path Planning Method. in *IEEE Access*, vol. 8, pp. 19842-19852(2020)
11. H. An, J. Hu and P. Lou.: Obstacle Avoidance Path Planning Based on Improved APF and RRT. 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Changsha, China, pp. 1028-1032 (2021)
12. Y. Li, Y. -J. Hao, H. -Y. Li and M. Li, "Path Planning of Unmanned Ships Using Modified APF Algorithm," 2023 IEEE 11th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, pp. 112-119(2023)
13. Jolly K G , Kumar R S , Vijayakumar R .A Bezier curve based path planning in a multi-agent robot soccer system without violating the acceleration limits[J].*Robotics & Autonomous Systems*, 57(1):23-33(2009)
14. Vahide Bulut.: Path planning of patrol robots in dynamic environment based on analytic geometry and cubic Bézier curve with three shape parameters. *Expert Systems with Applications*, Volume 233(2023)
15. Qingquan C , Jing N , Xunhe Y .: The Path Planning Study of Autonomous Greeting Robot based on Ant Colony Algorithm[J]( 2020)
16. Lee M C , Park M G.: APF based path planning for patrol robots using a virtual obstacle concept. (2003)