Proceeding of the 2015 IEEE
International Conference on Information and Automation
Lijiang, China, August 2015

Dynamic Path Planning of a Mobile Robot with Improved Q-Learning algorithm

Siding Li, Xin Xu, Lei Zuo
College of Mechatronics and Automation, National University of Defense Technology, Changsha 410073, P. R. China
lsd0323@163.com

***Abstract*—Path planning of a mobile robot under dynamic environment is a difficult part of robot navigation. In this paper, a new path planning method based on improved Q-learning (IQL) algorithm and some heuristic searching strategies is proposed for mobile robot in dynamic environment. A new exploration strategy which combines ε-greedy exploration with Boltzmann exploration is used in IQL. In addition, the heuristic searching strategies are provided to reduce the search space and limit the variation range of orientation angle. From simulations, the better performance of the proposed method was certified in terms of time taken and optimal path comparison with classical Q-learning (CQL) and other planning methods. Meanwhile, the reduction in orientation angle and path length has significance in the robotics literature of the energy consumption.**

***Index Terms*—Mobile robot; Path planning; Q-learning; Heuristic searching strategies**

I. INTRODUCTION

Nowadays, path planning is one of the important tasks in mobile robot navigation. Path planning means to find out an optimal or near-optimal collision-free path with respect to some criteria from an initial point to a goal point in an environment with some obstacles [1]. Several approaches have been proposed to solve the mobile robot path planning problem. According to the information of the environment, the path planning is divided into two categories, global path planning and local path planning. If the environment is a known static terrain, it is global path planning and the path must be found before execution. Global path planning methods include cell decomposition [2], visibility graphs [3], Voronoi diagrams [4] and artificial potential field [5]. Global path planning methods are computationally expensive when the environment is complex. If the environment is unknown or partly unknown terrain, it is local path planning. There are many local path planning methods, such as path planning methods based on fuzzy logic [6], neural networks (NN) [7], evolutionary algorithms [8], simulated annealing method [9], particle swarm optimization (PSO) [10], ant colony algorithms (ASO) [11] and reinforcement learning (RL).

Among those existing methods to solve the path planning problem, RL has its own unique characteristic, which is an alternative learning mechanism and relies on the principle of reward and punishment. Q-learning is one of the most classical RL algorithms and has successfully applied in many domains, such as the control of mobile robot, vehicle dispatch and multi-agents intelligent decision. It is also used to solve the mobile robot path planning problems. Goswami *et al* [12] proposed an extended Q-learning (EQL), in which the Q-table only stores the best action at a state. It can't be used for planning when the next state due to the best action is an obstacle. Amit *et al* [13] proposed an improved Q-learning (IQL) based on the work in [12]. In IQL, the Q-value isn't updated when the state is locked, instead of repeatedly updating them like the classical Q-learning (CQL). A heuristic Q-learning algorithm in mobile robot navigation is proposed in [14], which is only used in simple static environment. Pratyusha *et al* [15] proposed a meme algorithm which used differential evolution and Q-learning to solve the multi-robots path planning. Mariano and Martínez proposed a new method which integrated of cell-mapping and Q-learning techniques for motion planning of car-like robots [16]. A neural Q-learning method [17] was proposed to solve the path planning problem in an unrecognized environment, in which the learning process was divided into two stages, the Q-learning stage and the neural network training stage. However, the time complexity increased quickly with the increasing of the environment area. Andre *et al.* [18] proposed an option-based hierarchical learning approach in which basic behaviors are applied to accomplish the robot motion planning task. Although Q-learning algorithm has been applied to solve the mobile robot path planning problem for a long time, it is remaining a difficult problem of mobile robot path planning in dynamic environment with Q-learning algorithm.

In this paper, a new path planning method based on improved Q-learning (IQL) algorithm and some heuristic searching strategies is proposed for mobile robot path planning problem in dynamic environment. In IQL, a new exploration strategy is proposed to select action at states, which combines ε-greedy exploration with Boltzmann exploration. Different from the traditional ε-greedy strategy, the new strategy is capable of taking the tradeoff between the exploration and exploitation appropriately. In addition, random action which is executed in the exploration process is required to satisfy the heuristic searching strategies. After introducing the heuristic searching strategies, the search space of IQL is smaller than CQL. As a result, the proposed method is effective in avoiding local optimum and reducing the time taken comparison with CQL and A* algorithm. Moreover, the orientation angle of the robot is decreased comparison with CQL. The efficiency of the proposed method has been proven by some experiments.

The rest of this paper is organized as follows. Markov decision process model of the problem is described in Section II. The new path planning method is given in Section III. The experimental results and analysis are presented in Section IV. The conclusions and future work are listed in Section V.

978-1-4673-9104-7/15/$31.00 ©2015 IEEE

II. Markov Decision Process model of path planning problem

Solving problems with reinforcement learning algorithms are usually based on a Markov Decision Process (MDP) model. In this section, we will give a brief overview of MDPs and the MDP model of mobile robot path planning problem.

A. Brief overview of MDPs

A MDP is defined as a 4-tuple(S, A, P, R), where S is a finite set of states and A is all actions of robot. P is the transition probability and R is a reward function. The expected reward R for a state-action pair (s, a), as:

$$R(s,a)=\sum_{s'\in S} p(s,a,s')R(s,a,s') \tag{1}$$

where s is the current state, s' is the next state after robot taking action a in state s.

$Q(s, a)$ is the state-action pairs and their corresponding reward value. The exact Q values of all state-action pairs can be found by solving the linear system of the Bellman equations:

$$Q(s,a)=R(s,a)+\gamma\sum_{s'\in S} p(s,a,s')\sum_{a'\in A} Q(s',a') \tag{2}$$

For every MDP, there exists an optimal deterministic policy π^*, which maximizes the total expected reward from the initial state to the goal state.

B. MDP of path planning problem

Path planning of mobile robot could also be modeled as a MDP model. The four elements are shown as follows.

1) Environment Model: A grid-based model to represent the environment space has been used in many path planning methods, as shown in Fig.1. In this paper, assume that there are some static and indeterminate obstacles in the environment. Furthermore, moving obstacle with constant orientation angle and speed also exists in the environment. As shown in Fig.1, the black area and red rectangle with arrow denote the static and moving obstacle, respectively.

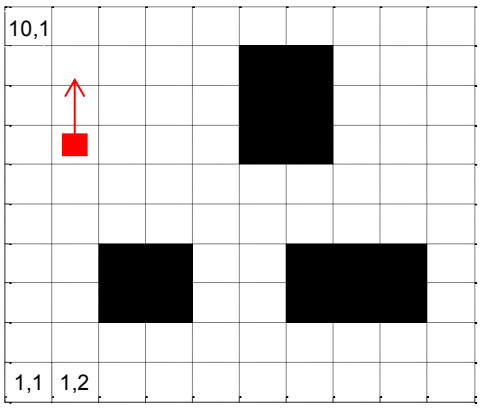


Fig.1. Grid-based environment model.

In this article, to simplify the problem, assume that the robot moves with fixed speed and the robot could be regarded as a point every time in the experiment.

2) State: In this paper, each grid in the environment model represents a state. The value of each state is the position of the grid in the matrix of the environment model. As shown in Fig.1, the (1, 1), (1, 2) and (10, 1) are the states.

3) Action: To decrease the distance of the path and the orientation angle of the robot, 8 actions are adopted in this paper. As shown in Fig.2, action 1, 2 … 8 denote the up, right-up … left-up, respectively.

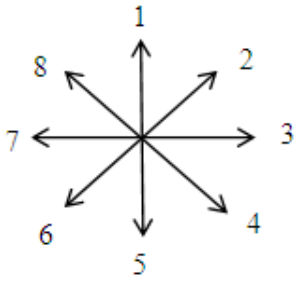


Fig.2 The 8 actions.

4) Reward function: Reward and punishment are the important factors that affect the efficiency of RL methods. In Q-learning algorithm, the agent executes an action reaching to next state and receives the reward from the environment. The purpose of the agent is to maximize the accumulated expected rewards for a sequence states from the initial state to the goal state. Therefore, the design of reward function affects the learning effective directly. In this paper, the reward function is designed as follows:

$$r=\alpha*(d_s-d_{s_0})-\beta/d_o \tag{3}$$

where d_s and d_{s0} denote the distance of current state and previous state to goal state, respectively, d_o denotes the distance between current state and the nearest obstacle. α and β are related parameters.

III. Proposed method

Q-Learning algorithm was proposed by Watkins and Dayan [19] and has been applied in many domains. In this section, we propose a path planning method based on IQL and some heuristic searching strategies. The details of the method are introduced as follows.

A. A new exploration strategy

The performance of Q-learning algorithm is affected by the exploration strategy. There are many exploration strategies, such as ε-greedy and Boltzmann exploration strategies. In ε-greedy exploration, parameter ε represents exploration probability which is introduced to balance the ratio between randomly and greedy action selection. The ε-greedy strategy is shown as follows:

1. Generate p randomly, $p\in(0,1)$;
2. **If** $p < \varepsilon$;
3. a = rand(A);
4. **Else**
5. a = arg max($Q(s,a_i)$);
6. **End if**.

In Boltzmann exploration, the probability of random strategy is larger than greedy strategy at the beginning of the learning process. With the number of iterations increasing, the probability of greedy strategy is reduced. The agent selects an action a_i with a probability p is shown as follows:

$$p(s,a_i)=\frac{\exp(\frac{Q(s,a_i)}{T})}{\sum_{a_k\in A}\exp(\frac{Q(s,a_k)}{T})} \tag{4}$$

$$T_k=\lambda^k T_0 \tag{5}$$

where T is the temperature parameter, it decreases with the number of iterations. λ is the temperature discount rate.

In this paper, we propose a new exploration strategy which combines ε-greedy exploration with Boltzmann exploration. As a result, it has the advantages of avoiding local optimum and accelerating the expected Q values converges to the actual Q values. The strategy is shown as follows:

1. Generate p randomly, $p\in(0,1)$;
2. **If** $p<\varepsilon$;
3. a = rand(A);
4. **Else**
5. a = Boltzmann(A);
6. **End if**.

B. Heuristic Searching Strategies

To accelerate the learning process, in this paper, we introduce some heuristic searching strategies. Random action is required to satisfy the heuristic searching strategies. Assume that (x_c, y_c) and (x_g, y_g) denote the current state and goal state, respectively. The details of the heuristic searching strategies rules are listed as follows:

a) If $x_c < x_g$ and $y_c < y_g$, a=rand(1,2,3); else if $x_c > x_g$ and $y_c > y_g$, a=rand(5,6,7).
b) If $x_c < x_g$ and $y_c > y_g$, a=rand(3,4,5); else if $x_c < x_g$ and $y_c > y_g$, a=rand(7,8,1).
c) If $x_c = x_g$ or $y_c = y_g$, a=rand(1,5) or a=rand(3,7);
d) Else $a = a$-1 / a / a+1;

Strategies a)-c) are developed to increase the possibility of the robot reaches to the goal state, while strategy d) is used to decrease the orientation angle of the path. It limits the variation range of orientation angle between 0~90°.

C. Improved Q-learning Algorithm

In mobile robot path planning problem, Q-learning algorithm is looking for the optimal action policies from the initial state to the goal state under the mapping from environment state to action. In the process of interacting with the environment, the robot attempts to find an action to generate the maximum cumulative reward which is obtained by the approximation of the optimal state-action function $Q(s, a)$ in a state s and the Q-value is updated according to the equation as follows:

$$Q(s,a)\leftarrow Q(s,a)+\alpha(r+\gamma\max_{a'}Q(s',a')-Q(s,a)) \tag{6}$$

where r is the immediate reward obtained by executing action a under the current state, s is the current state and s' is the next state after s execute the action a. $\gamma(0\le\gamma\le 1)$ is the discounted rate and $\alpha(0<\alpha\le 1)$ is the learning step.

In IQL algorithm, the new exploration strategy mentioned before is used to select optimal action. In addition, random action is required to satisfy heuristic searching strategies. Assume that there are n states and m actions in the path planning problem. It requires an array of $n\times m$ size to store the Q values called Q-table. The IQL algorithm is given as follows:

Algorithm 1 Improved Q-learning algorithm **(**IQL)

1. Input : Information of environment G;
2. Initial state s_i and goal state s_g;
3. α , γ , ε, ξ and max iterations.
4. Output: optimal Q-table.
5. **Initialization**
6. Set the $Q_{n\times m}=\{0\}$;
7. $s = s_i$;
8. **While** ($\|Q_t - Q_{t-1}\| > \xi$ && Num<Max iterations)
{
9. **Repeat**
{
10. **If** $Q(s, :)$ is empty
11. a = Random (A) satisfied HSS;
12. **Else**
13. Generate η randomly, $\eta\in(0,1)$;
14. **If** ($p<\varepsilon$)
15. a = Random (A) satisfied HSS;
16. **Else**
17. a = Boltzmann(A);
18. **End if**;
19. **End if**;
20. s' = Exact (s, a);
21. **If** (s' = obstacle or beyond the environment)
22. $$Q(s,a)=Q(s,a)-P; \tag{7}$$
23. **Break;**
24. **Else if** ($s'=s_g$)
25. $$Q(s,a)=Q(s,a)+P; \tag{8}$$
26. **Break;**
27. **Else**
28. $$Q_t(s,a)=\begin{cases}(1-\alpha)Q_t(s,a)+\alpha\left[R(s,a)+\gamma\max_{a\in A}Q(s',a)\right]; & s=s_t, a=a_t\\ Q_{t-1}(s,a)\ ; & others\end{cases} \tag{9}$$
};
29. Update Q-table.
};
30. **Return** Q-table.
31. End.

where s_i is the initial state, s_g is the goal state. P is the punishment parameter, which is affected by the reward function. HSS is the abbreviation of heuristic searching strategies.

D. Path-Planning algorithm

The IQL algorithm presented in the previous subsection stores the Q-values of each state for all actions in the Q-table. After learning process, the optimal Q-table can be used for path planning. During the planning process, the robot in current state s_c, identifies the next best state s_n, where the Q-value is bigger than the Q-value of other adjacent states of s_c. In addition, if there are more than two next states which have the largest Q-value among the adjacent states of s_c, the robot would select the action which has the minimum orientation angle with the previous action.

In this subsection, a path planning algorithm is provided to generate a collision-free optimal path for the robot from initial point to goal point after the optimal Q-table is obtained by the IQL. In the algorithm, s_i is the initial state and s_g is the goal state. s_0 is the previous state. s is the current state and s' is the next state of s after executing the action a. The path planning algorithm is shown as follows:

Algorithm 2 Path planning algorithm

1. Input : Information of environment G;
2. Initial state s_i and goal state s_g;
3. Output: optimal sequence of actions.
4. **Learning process**
5. Obtain the optimal Q-table by IQL.
6. **Planning process**
7. $s = s_i$; $s_0 = s$; R;
8. **while** ($s' \sim = s_g$)
   {
9. **If** (No moving obstacles in R)
10. **If** ($s \sim =$obstacle)
11. **If** ($Q_n > Q_r, \forall r$)
12. $s' = \arg\max Q(s_r, a)$;
13. $s = s'$;
14. **Else if** ($Q_{n1} = \ldots = Q_{ni} > Q_r, \forall r$)
15. $s' = \arg\min(abs(a_c - a_r))$;
16. $s = s'$;
17. **End.**
18. **Else**
19. $Q(s) = -P$;
20. $s = s_0$;
21. **End.**
22. **Else** (An obstacle in R)
23. s_o = Infer(s_{o1}, s_{o2});
24. Back to 9.
25. **End.**
    }
26. **Return** Optimal sequence of actions.

where R is the observable range of robot , Q_n is the biggest Q-value of next state in the adjacent states of s. The line 14 means that there are more than 2 biggest Q-value of next states. a_c is the action of robot in current state, while a_r is action in the previous state. P is the punishment parameter. In the line 23 of the path planning algorithm, robot infers the next state of the moving obstacle according to the two sequential states of it. Therefore, the moving obstacle becomes a static obstacle with variable situation for the robot.

IV. Simulation and Analysis

In this section, some experiments were carried out to demonstrate the feasibility and effectiveness of the proposed method applied for mobile robot path planning problem. Firstly, we get the optimal Q-table using IQL in a simple and static environment. After that, the optimal Q-table is applied for the path planning problem in different dynamic environments. At last, B-spline method is introduced to smooth the path planned by the proposed method.

In this paper, we consider an environment of 20×20 grids, where each grid represents a state. The black areas denote the static obstacles as shown in Fig.3. S and G denote the initial state and goal state in all the world maps of the following experiments, respectively.

A. Path planning in a static environment

In this subsection, the IQL algorithm is applied to obtain the optimal Q-table in the simple environment. After that, the path planned by the optimal Q-table is compared with CQL and A∗ algorithm as shown in Fig.3.

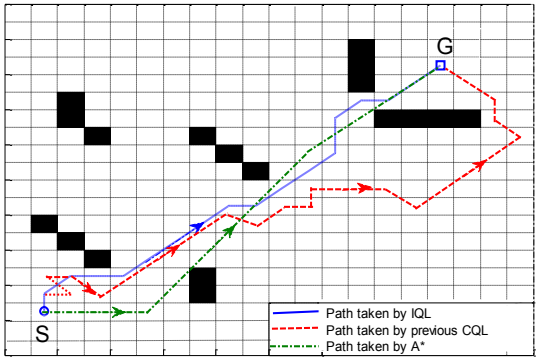


Fig.3 Path planning in static environment with different methods.

From Fig.3 we can see that the IQL is more effective than CQL in both distance and orientation angle of the path. Because of the new exploration strategy and the heuristic searching strategies, the number of iterations required to learn the world map of Fig. 3 by the IQL is 787, while the EQL is 6502 iterations and the CQL is 8276 iterations, respectively. In addition, the path planned by A* algorithm doesn't take the safety range with the obstacle into account.

B. Path planning in an uncertain environment

Generally, it is difficult to get the whole information of the environment. Some unknown and uncertain obstacles will be met by the robot in the path planned advanced. In this experiment, the robot walks along the path planned by A* algorithm, IQL and CQL in Fig.3 in the environment with three new additional obstacles. The result is shown in Fig.4.

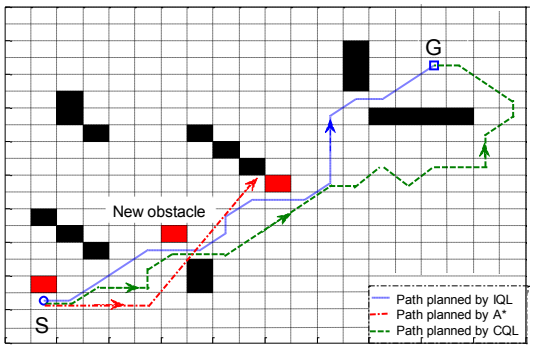


Fig.4 Path planning in the environment with uncertain obstacles.

As we can see from Fig.4, the robot avoids colliding with the 3 additional new obstacles by the planning method based on IQL and CQL. Moreover, the path planned by IQL is optimal in path length and orientation angle comparison with CQL. However, the path planned by A* algorithm as the red line in Fig.4 is failed to avoid the new obstacles in this case.

C. Path planning in the environment with moving obstacle

Although there are many effective methods for mobile robot path planning problem, it is remaining difficult to solve the problem in the environment with moving obstacles. In this subsection, we suppose an environment with a moving obstacle. Assume that the orientation and speed of the obstacle are fixed. The results by proposed method are shown in Fig.5

The robot infers the trajectory of the moving obstacle in its observable range after observing some sequential states of the obstacle. The collision point of the robot with moving obstacle could be forecast. In Fig.5 (a), the robot infers that it will collide with the obstacle after executing the optimal action in current state, and selects the action which opposite with the direction of the moving obstacle until the distance between robot and obstacle reaches a safe range. The final result is shown in Fig.5 (b). The result without avoid the moving obstacle in this dynamic environment is shown in Fig.5 (c).

D. Path planning with random initial point

In this subsection, the initial point of the path planning problem is random. The result using optimal Q-table with random initial point (11, 8) is shown in Fig.6:

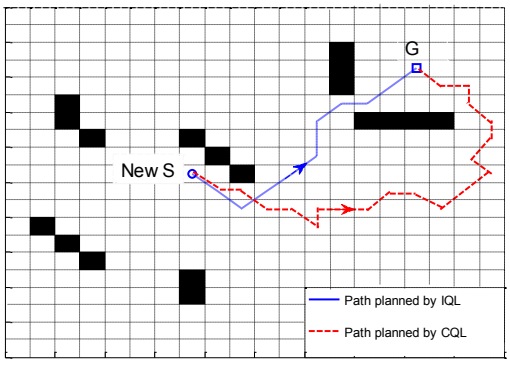
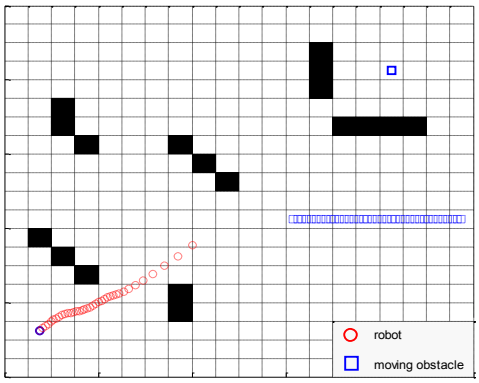
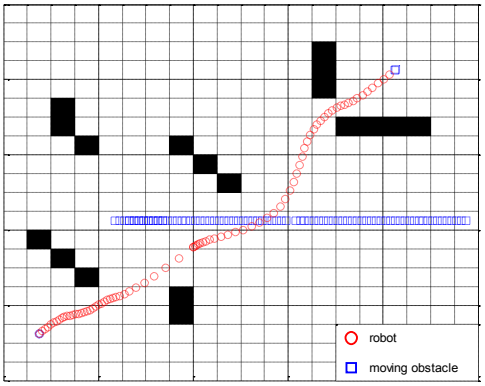


Fig.6 Path planning with random initial point (11, 8).

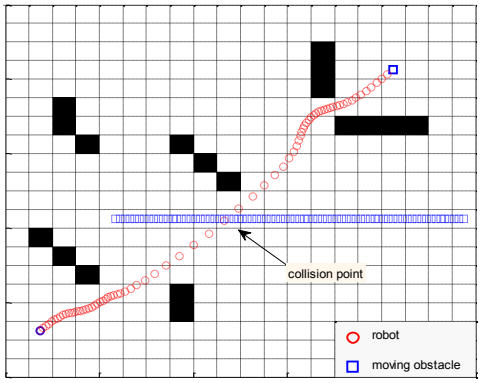
Because the Q-table stores the Q-values of all actions at each state, it is easy to solve this problem for IQL and CQL without a new learning process, while for other methods, such as GA and A*, need to re-plan the path from the new initial point. As a result, the proposed method saves much time comparison with other methods. As shown in Fig.6, the robot gets to the goal point from the random initial point successfully by IQL and CQL, and the path planned by IQL is optimal in path length and orientation angle comparison with CQL.



(a) Meet the obstacle.



(b) Avoid the obstacle.



(c) The result without avoid the obstacle.

Fig.5 Path planning in the environment with moving obstacle.

From experiments *A* to *D* we can see that the proposed method for path planning problem only needs to obtain the optimal Q-table by IQL in a simple static environment, and then the optimal Q-table could be used to solve various problems as mentioned before.

Comparisons of time taken in different experiments with different methods are listed in Table.I. "--" means the method failed to plan the path.

TABLE I
COMPARISON OF TIME TAKEN IN DIFFERENT METHODS

Different environment	Planning time taken in seconds			
	IQL	CQL	EQL	A*
Fig.3	3.227	38.54	24.82	32.03
Fig.4	3.225	34.78	--	--
Fig.5	4.773	56.23	--	--
Fig.6	3.112	40.60	19.67	--

It is shown in Table.I that time taken of proposed method is faster than other methods in the same environment. In addition, the EQL and A* are failed for path planning problem in some experiments without a new planning process as shown in Table.I. Meanwhile, from Fig.3 - Fig.6 we can see that the proposed method could find the path with all orientation angles on more than 90° .

In conclusion, the results demonstrated that the proposed method for path planning has many advantages than other methods, such as A*, CQL and EQL in [12].

Spline curve is a series of smooth curve through some given points. It has been widely used in practice by taking the balance between the computational efficiency and the smoothness of path curves. Inspired by [8], the B-spline is used to smooth the path planned by the proposed method in Fig.3. The path after smoothing is more likely to be applied in the real world of unmanned vehicles.

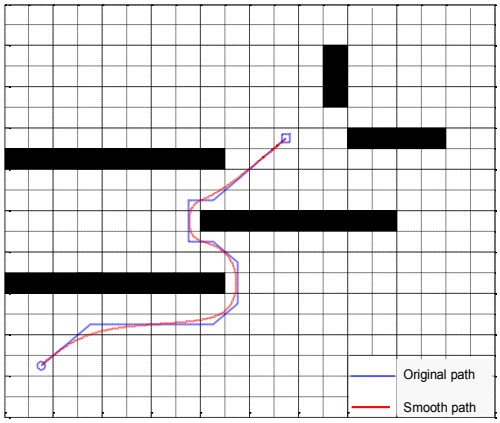


Fig.7 The path smoothed by B-spline.

The result in Fig.7 shows that the B-spline is effective to smooth the path planned by proposed planning method without colliding with the obstacles.

V. CONCLUSION

In this paper, we have proposed a new path planning method based on IQL and some heuristic searching strategies for mobile robot path planning problem in dynamic environment. In IQL, a new exploration strategy is provided to substitute the traditional greedy strategy, which has the advantages of avoiding local optimum and accelerating the convergence rate. In addition, the heuristic searching strategies are capable of reducing the number of iterations in learning process and limiting the variation range of robot orientation angle. As a result, experiments simulated in different environments confirm the superior performance of the proposed method comparison with CQL and other path planning methods in time taken. At last, we use the B-spline to smooth the path planned by the proposed method. The path after smoothing is more likely applied in real world unmanned vehicle navigation. Future work need to research the path planning problem in an unknown dynamic environment and the multi-robots path planning problem.

REFERENCE

[1] Z. Bien and J. Lee, "A minimum-time trajectory planning method for two robots," *IEEE Trans. Robot. Autom.,* vol.8, no.3, pp.443-450, Jun.1992.

[2] C. Cai and S. Ferrari, "Information-driven sensor path planning by approximate cell decomposition," *IEEE Trans. Syst., Man Cybern.,* vol. 39, no. 3, pp. 672–689, Jun. 2009.

[3] C.A.E. Poulos, M.G. Inp, "Path planning for a mobile robot," *IEEE Trans. Syst. Man. Cybern.,* vol.22, no.2, pp318–322, 1992.

[4] O. Takahashi and R. J. Schilling, "Motion planning in a plane using generalized voronoi diagrams," *IEEE Trans. Robot. Autom*., vol. 5, no. 2, pp. 143-150, Apr. 1989.

[5] Anugrah and Keum-Shik, "A path planning algorithm using vector Potential unctions in triangular regions," *IEEE Trans. Syst. Man Cybern.,* vol.43, no 4, Jul. 2013.

[6] Bing Sun *et.al*, "A novel fuzzy control algorithm for three-dimensional AUV path planning based on sonar model," *J. Intell. Fuzzy Syst.*, vol. 26, pp. 2913-2926, 2014.

[7] Hong Qu and Simon X.Yang, "Real-time robot path planning based on a modified pulse-coupled neural network model," *IEEE Trans. Neural Netw.,* vol. 20, no.11, pp. 1724-1739, 2009.

[8] Ching-Chih Tsai, Hsu-Chih Huang andCheng-Kai Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *IEEE Trans. Indus. Elec.*, vol. 58, no. 10, Oct. 2011.

[9] Hui Miao, Yu-Chu Tian, "Dynamic robot path planning using an enhanced simulated annealing approach," *Applied Mathematics and Computation* 222 pp.420–437, 2013.

[10] Yong Zhang, Dunwei Gong and Jianhua Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neuro computing*, pp.172–185, 2013.

[11] Imen Chari *et.al,* "An efficient hybrid ACO-GA algorithm for solving the global path planning problem of mobile robots," *Int. J. Adv. Robot syst.,* Mar 2014.

[12] Goswami, P. K. Das, A. Konar, and R. Janarthanan, "Extended Q-learning algorithm for path-planning of a mobile robot," in *Proc. 8th Int. Conf. SEAL,* pp. 379–383, Dec. 2010.

[13] Amit Konar,Senior, Indrani Goswami Chakraborty, "A deterministic improved Q-Learning for path planning of a mobile robot," *IEEE Trans Syst, Man Cybern.,* vol. 43, no. 5, Sep.2013.

[14] Widyawardana Adiprawita, Adang Suwandi Ahmad, "Reinforcement learning with heuristic to solve POMDP Problem in mobile robot path planning," 2011 I*nt. Conf. Electrical Engineering and Informatics.,* 17-19 Jul 2011, Bandung, Indonesia.

[15] Pratyusha Rakshit, Amit Konar, "Realization of an adaptive memetic algorithm using differential evolution and Q-Learning: A case study in multi-robot path planning," *IEEE Trans. Syst., Man Cybern.,* vol. 43, no. 4, Jul. 2013.

[16] Mariano Gómez Plaza, Tomás Martínez-Marín, "Integration of Cell-Mapping and Reinforcement-Learning techniques for motion planning of car-like robots". *IEEE Trans Instr. Measurement.,* vol. 58, no. 9, Sep. 2009.

[17] Velappa Ganapathy, Soh Chin Yun, "Neural Q-Learning controller for mobile robot," *IEEE Int. Conf. Advanced Intelligent Mechatronics*. Singapore, Jul, 2009.

[18] Andrea, Rosa R and Lozano-Martínez, "Hierarchical reinforcement learning approach for motion planning in mobile robotics". *IEEE Latin American Robotics Symposium*. 2013

[19] C. Watkins and P. Dayan, "Q-learning," Mach. Learn., vol. 8, no. 3, pp. 279-292, May 1992.