Journal of Computers Vol. 33 No. 3, June 2022, pp. 187-194
doi: 10.53106/199115992022063303015

Research on Path Planning Strategy of Rescue Robot Based on Reinforcement Learning

Ying-Ming Shi[1], Zhiyuan Zhang[2*]

[1] School of Automation Engineering, Tangshan Polytechnic College,
Tangshan City 063600, Hebei Province, China
yingming5230@163.com
[2] School of Electronic and Information Engineering, Key Laboratory of Communication and Information Systems,
Beijing Municipal Commission of Education Beijing Jiaotong University, Beijing, China
zhangzhiyuan@bjtu.edu.cn

Received 16 April 2022; Revised 16 May 2022; Accepted 16 June 2022

Abstract. How rescue robots reach their destinations quickly and efficiently has become a hot research topic in recent years. Aiming at the complex unstructured environment faced by rescue robots, this paper proposes an artificial potential field algorithm based on reinforcement learning. Firstly, use the traditional artificial potential field method to perform basic path planning for the robot. Secondly, in order to solve the local minimum problem in planning and improve the robot's adaptive ability, the reinforcement learning algorithm is run by fixing preset parameters on the simulation platform. After intensive training, the robot continuously improves the decision-making ability of crossing typical concave obstacles. Finally, through simulation experiments, it is concluded that the rescue robot can combine the artificial potential field method and reinforcement learning to improve the ability to adapt to the environment, and can reach the destination with the optimal route.

Keywords: rescue robot, potential field algorithm, reinforcement learning, optimal route

1 Introduction

In recent years, natural disasters such as earthquakes and tsunamis, as well as accidents such as chemical, nuclear radiation, and poisoning have occurred frequently, and the complexity, severity and diversity of various disasters are also increasing. For rescue, 72 hours after the disaster is the golden rescue time. Therefore, how to quickly enter the disaster scene and rapid rescue has always been a hot spot in the field of rescue research. Mobile rescue robots undertake more and more important rescue tasks in the rescue process of the above complex and dangerous situations [1]. Path planning is the calculation process for the robot to complete a reasonable route from the starting point to the target point, and it is the key for the robot to achieve autonomous movement. The core of path planning is to complete the design of the path planning algorithm [2]. The artificial potential field method is a classic and efficient algorithm in the traditional path planning algorithm. The algorithm completes the path planning by configuring the information of the known environment. It has good real-time performance, strong adaptability to dynamic environments, and smooth planned paths. At the same time, the artificial potential field method also has obvious shortcomings, that is, the robot is easy to fall into a local minimum during the navigation process using the artificial potential field method, and the manifestation is that the robot stagnates or oscillates nearby after encountering an obstacle [3].

In order to solve the local minimum problem of the artificial potential field method, many scholars have proposed corresponding improved algorithms. Reference [4] proposes an algorithm for virtual obstacles, which can better solve the local minimum problem. Li Qing [5] and others proposed that when the robot meets the local minimum point, the robot can get out of trouble by changing the repulsion angle at the local minimum point and setting the virtual minimum local area, and at the same time, the genetic algorithm is used to optimize the repulsion change angle and the virtual local area. Li Wei [6] proposed a fast search random tree algorithm to complete the convergence of the planning space during the path planning process, and then used the artificial potential field method to accelerate the convergence speed to improve the speed of obtaining the best route.

In this paper, the traditional artificial potential field method is used to plan the path of the robot to obtain an efficient and smooth path. There are generally no dynamic obstacles in the rescue environment of rescue robots,

[*] Corresponding Author

so the focus of this paper is the robot's ability to avoid obstacles to static obstacles. When the robot encounters a concave obstacle in the route planned by the artificial potential field method and generates a local minimum value, it helps the robot to choose a coping strategy through reinforcement learning. After getting out of trouble, it returns to the artificial potential field method algorithm.

2 Artificial Potential Field Method for Path Planning

The artificial potential field method assumes that the gravitational effect of the gravitational potential field of the target point in the environment where the robot is located is affected by the repulsive force of the obstacle repulsion potential field, as shown in Fig. 1.

2.1 Schematic Diagram of Artificial Potential Field Method

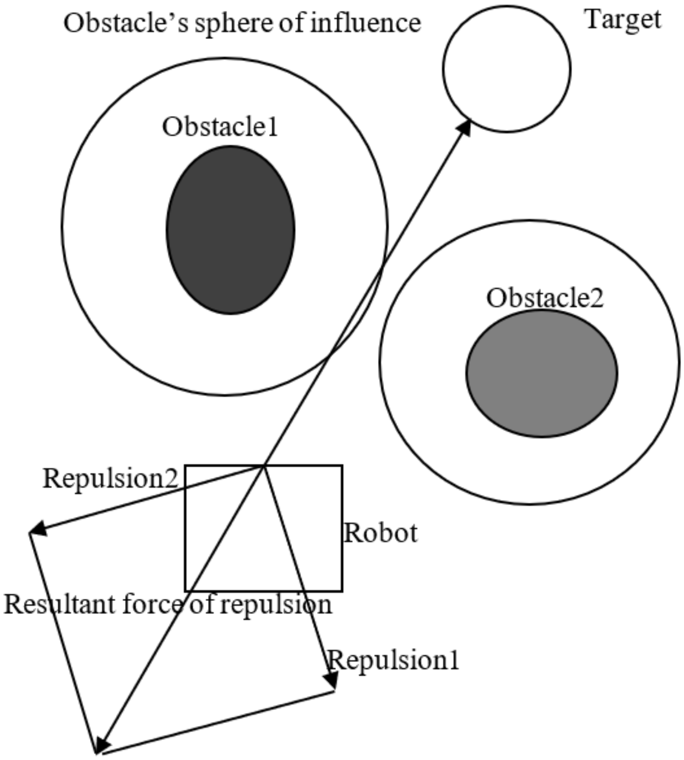

Fig. 1. Artificial potential field method for path planning

2.2 Expression of Potential Field Function

Suppose the robot target point is a mass point, the current position of the robot is $q = [x, y]^T$, The target position is $q_{goal} = [x_{goal}, y_{goal}]^T$, so the gravitational field function U_{at} is [7]:

$$U_{\text{at}} = \frac{1}{2}\lambda_{\text{a}} d^2\left(q - q_{goal}\right) \tag{1}$$

Among them, λ_a is the gravitational coefficient, and $d(q - q_{goal})$ is the Euclidean distance from the robot's current position to the target point. The repulsion field function U_{re} generated by the obstacle X_{ob} in the process of traveling is:

$$U_{re} = \begin{cases} \frac{1}{2}\lambda_r \left(\frac{1}{d_{\min}} - \frac{1}{d_{ob}^*} \right)^2 & d_{\min} \le d_{ob}^* \\ 0 & d_{\min} > d_{ob}^* \end{cases} \tag{2}$$

$$d_{\min} = \min_{q^{'} \in X_{ob}} d\left(q, q^{'}\right) \tag{3}$$

Among them, λ_r is the repulsion coefficient, d_{min} is the shortest distance between the robot and the obstacle, q' is the point in the obstacle X_{ob} that is closest to the robot's current position, d_{0b}^* is the repulsion range of the obstacle, the formula means, when the robot After the distance from the obstacle exceeds the repulsion range, the robot will not be affected by the repulsion of the obstacle. The resultant force of the robot in the potential field guides the robot to approach the target point, and the negative gradient of the potential field is the force, namely:

$$F = -\nabla U \tag{4}$$

$$F_{at} = \lambda_a d\left(q - q_{goal}\right) \tag{5}$$

$$F_{re} = \begin{cases} \lambda_r \left(\frac{1}{d_{\min}} - \frac{1}{d_{ob}^*} \right) \frac{1}{d_{\min}^2} \frac{\partial d_{\min}}{\nabla q} & d_{\min} \le d_{ob}^* \\ 0 & d_{\min} > d_{ob}^* \end{cases} \tag{6}$$

Therefore, the resultant force on the robot is F_{total}, and the expression is:

$$F_{\text{total}} = F_{at} + F_{re} \tag{7}$$

The main reason for the local minimum is the shape of the obstacle. The general model of this obstacle is that the point on the connecting line of two points on it is not inside the obstacle, that is, it is a concave obstacle [8]. Therefore, for concave obstacles of various shapes and sizes that may exist in the environment, the robot is easy to fall into the local minimum value, and the lack of dynamic environment adaptability problem, the reinforcement learning idea is integrated into it, so based on reinforcement learning, it helps the robot in Make decisions when encountering various types of recessed obstacles.

2.3 Algorithmic Transition Condition

The conditions for the robot to transfer from the artificial potential field method to reinforcement learning are [9]:

$$\text{Condition 1:} \quad \left| f_{at} + \sum_{j=1}^{n} f_{re,j} \right| < \varepsilon \tag{8}$$

$$\text{Condition 2:} \quad \left| q - q_A \right| < \beta s_A \tag{9}$$

When condition 1 or condition 2 is satisfied, the robot is considered to be at the minimum point. Among them, ε is a small positive number, indicating that the virtual resultant force of the robot is close to 0. β is a positive number, and $\beta \in (0，1)$, q_A is a certain state during the movement of the robot, s_A is the total distance of the robot from q_A to q, when condition 2 is satisfied, the robot has moved a long distance, and The displacement is small, and when condition 2 occurs, the robot oscillates near the local extreme point. The robot can then switch to a reinforcement learning model to make decisions.

3 Reinforcement Learning Model Building

As an important branch of artificial intelligence algorithms, reinforcement learning has been widely used in the autonomous decision-making process of robots. Reinforcement learning is a machine learning method with a reward mechanism. When facing obstacles, the robot does not need to be guided by a priori data with labels, but through continuous trial and error, it continuously accumulates the decisions made to obtain the maximum reward, so as to obtain the maximum reward. Continuously optimize the robot selection strategy. The ability of robot environment adaptability and self-learning is improved.

By constantly exploring and trying to maximize the accumulated reward, learn the optimal action to take in different states. When using the artificial potential field method, the robot encounters the problem of a local minimum value after the path planning. The reinforcement learning method is used to make decisions, that is, when the minimum value is encountered, the rotation angle and the minimum range for escape are selected. Different concave objects have different choices. Through reinforcement learning, the robot's avoidance ability when encountering concave obstacles is continuously improved. Design action spaces, action-value functions, and reinforcement learning for search angle and search range policies.

3.1 Reinforcement Learning Modeling

In reinforcement learning, the problem to be solved is often described and modeled by a Markov Decision Process (MDP) [10]. For a certain state, if this state only depends on the current state and has nothing to do with the past historical state, the Markov process formula is as follows:

$$P\left(S_{t+1} \mid S_t\right) = P\left(S_t \mid S_1, S_2, \ldots, S_t\right) \tag{10}$$

In the formula, S_t represents the state at time t. The Markov decision process includes five elements: state set S, action set A, state transition probability $P_{SS'}^{a}$, reward function $R_{SS'}^{a}$, discount factor Υ, so they are carried out separately. design.

(1) State set S: The state set experienced by the robot. Contains the current speed of the robot, the size information of the obstacle itself, etc.

(2) Action set A: The set of actions that the robot can take. Contains the random rotation angle θ of the robot, and the movement step r of the robot during the principle local minima.

(3) State transition probability $P_{SS'}^{a}$: the probability of the robot transitioning from state S to S' after taking action a.

(4) Reward function $R_{SS'}^{a}$: After the robot takes action a, the reward obtained by transitioning from state S to S'.

(5) Discount factor Υ: It is used to control the importance of the current reward and the future reward. It is also a factor introduced mathematically to facilitate the solution of the Markov decision process, within the range of $\gamma \in (0, 1)$.

Therefore, the robot obtains the state S_t at time t, and according to the strategy π, performs the corresponding action a_t and then reaches the new state S_{t+1}, and then obtains the reward r_t. The ultimate goal is to obtain the optimal strategy π^*, so the Sum of awards received:

$$R_t^{\infty} = \sum_{i=t}^{\infty} \eta r_i \tag{11}$$

where η is the discount coefficient, η ∈ (0, 1), and the state value function can be expressed as the expected value of the reward:

$$V(S_t) = E\left[r_{t+1} + \eta V(S_{t+1})\right] \tag{12}$$

Among them, E is the mathematical expectation. Therefore, if the optimal strategy π^* determines the execution of the highest reward strategy, the Bellman equation is used to represent the highest reward value function:

$$V^*(S_t) = \max_{a_t \in A}\left(r_t + \eta\int_{S_{t+1}} P^{a_t}_{S_t S_{t+1}} V^*(S_{t+1})\, ds_{t+1}\right) \tag{13}$$

3.2 Algorithm Training Process

In order to improve the computing efficiency and avoid the randomness in the process of robot exploration and learning, the real action space model of the robot is approximated, and an effective action space model is selected according to the environmental information around the robot, so as to reduce the consumption of computing resources and improve the learning rate. Finally, a strategy that is closer to the optimal is found. The Q-learning algorithm based on approximate action model strategy selection is adopted, and its design process is shown in Fig. 2:

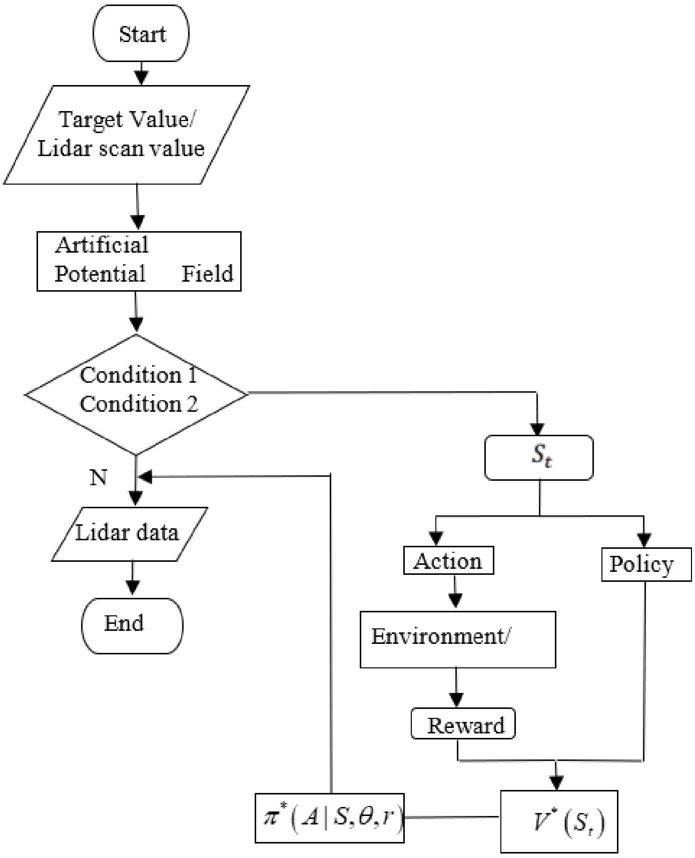

Fig. 2. Training program design process

3.3 Simulation Training Results and Analysis

Usually, the rescue robot moves on a two-dimensional plane, and the obstacles are generally static obstacles, which are unstructured environments, and the global map information is unknown. The robot is modeled by lidar scanning. The simulation environment platform used in this paper is Stage [11], and the main speed commands used by the rescue robot studied in this paper include linear speed and angular speed. In order to speed up the collection of data in the simulation environment, this paper sets up multiple types of U-slot static obstacles to enhance the robot's ability to avoid obstacles, as shown in Fig. 3 and Fig. 4. The parameters of the experiment are set as follows: the learning rate $a = 1$, the discount factor $\eta = 0.5$, the maximum number of episodes is 5000, and the maximum number of execution steps is 500. And take the angle θ as ± 15°, ± 30°, ± 45°, ± 60° and ± 75° respectively. At this time, in order to ensure that the phenomenon of returning to the local minimum does not occur again, the local escape radius r should be larger, where r is taken as a fixed value of 0.

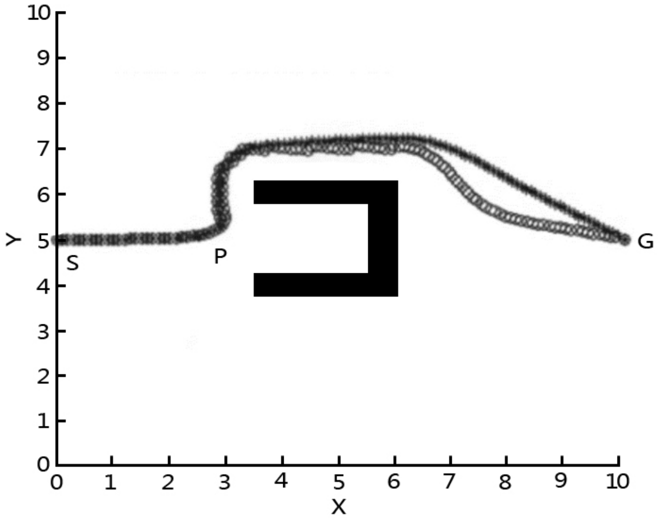

Fig. 3. The first U-groove environment

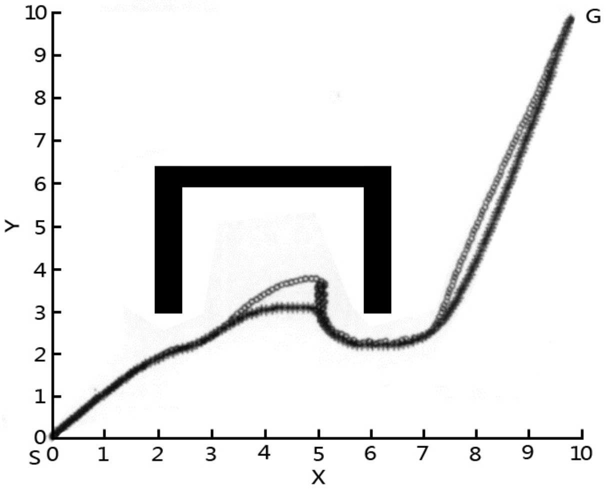

Fig. 4. The second U-groove environment

The hardware platform used for training in this section is NVIDIA GTX1080Ti, and the CPU is Intel i5 8th generation series. It takes about 48 hours to train for 4000 iterations.

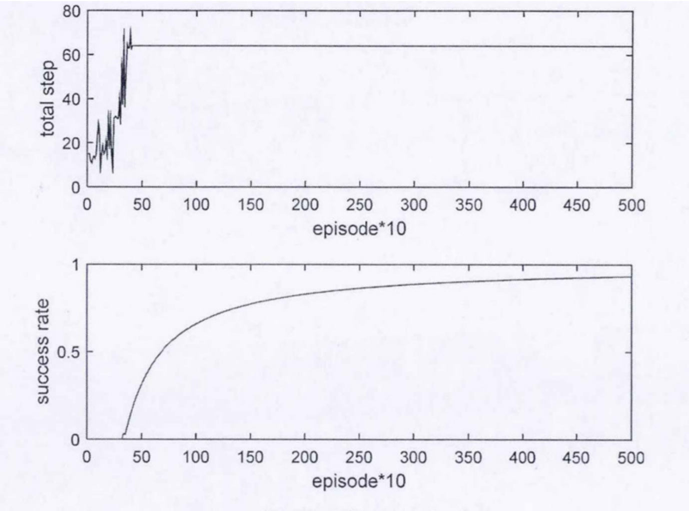

Fig. 5. Convergence result of the first U-groove environment

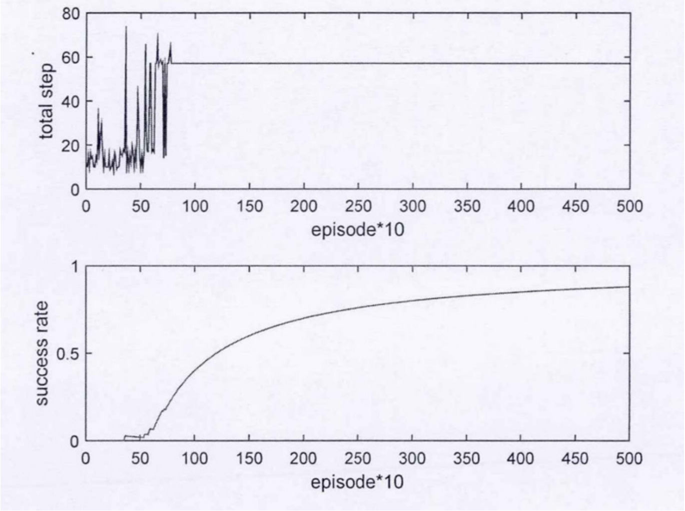

Fig. 6. Convergence result of the first U-groove environment

The path planning simulation results show that for the first U-shaped groove, it is obvious that the trained mobile robot can reach the target point from the starting point without collision, and find a nearly optimal path with a path length of 42.4m and a total of 53 steps. The optimal path length obtained by the second U-slot robot is 44m, and a total of 55 steps are performed. In the early stage of training, the environmental information is unknown, so the training success rate is low. As shown in Fig. 5 and Fig. 6.

4 Conclusion

In this paper, aiming at the unstructured environment faced by rescue robots, an improved artificial potential field method based on reinforcement learning is proposed. The problems of stagnation and oscillation are solved by changing the repulsion angle and setting the virtual local minimum area. The simulation study of two sizes of U-shaped grooves is mainly used to illustrate that the algorithm proposed in this paper has the following characteristics: (1) The traditional artificial potential is maintained. Some of the advantages of the field method are that the algorithm is simple and the path safety and smoothness are good; (2) it overcomes the shortcomings of the traditional artificial potential field method with local minimum; (3) the self-learning and self-adaptation of reinforcement learning.

5 Acknowledgement

This work has been supported by the Fundamental Research Funds for the Central Universities (grant number: 2022JBMC014 and Beijing Nova Program (Z211100002121120) from Beijing Municipal Science & Technology Commission.

References

[1]S. Al-Milli, L.D. Seneviratne, K. Althoefer, Track-terrain modelling and traversability prediction for tracked vehicles on soft terrain, Journal of Terramechanics 47(3)(2010) 151-160.

[2]A. Pandey, R.K. Sonkar, K.K. Pandey, D.R. Parhi, Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller, in: Proc. IEEE 8th International Conference on intelligent Systems and Control, 2014.

[3]J. Wang, X.-B. Wu, Z.-L. Xu, A path planning algorithm avoiding a class of local minima in artificial potential field, Computer Simulation 24(11)(2007) 151-154.

[4]M.C. Lee, M.G. Park, Artificial potential field based path planning for mobile robots using a virtual obstacle concept, in: Proc. 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2003.

[5]Q. Li, L.-J. Wang, B. Chen, Z. Zhou, Y.-X. Yin, An improved artificial potential field method with parameters optimization based on genetic algorithms, Chinese Journal of Engineering 34(2)(2012) 202-206.

[6]W. Li, S.-J. Jin, Optimal path convergence method based on artificial potential field method and informed sampling, Journal of Computer Applications 41(10)(2021) 2912-2918.

[7]J. Estremera, J.A. Cobano, P.G. Santos, Continuous free-crab gaits for hexapod robots on a natural terrain with forbidden zones: An application to humanitarian demining, Robotics and Autonomous Systems 58(5)(2010) 700-711.

[8]M. Chen, M.-J. Li, Y.-W. Li, Z.-Q. Lai, Mobile robot navigation based on deep reinforcement learning and artificial potential field method, Journal of Yunnan University (Natural Sciences Edition) 43(6)(2021) 1125-1133.

[9]T. Weerakoon, K. Ishii, A.A.F. Nassiraei, An Artificial Potential Field Based Mobile Robot Navigation Method To Prevent From Deadlock, Journal of Artificial Intelligence and Soft Computing Research 5(3)(2015) 189-203.

[10]M. Botvinick, S. Ritter, J.-X. Wang, Z. Kurth-Nelson, C. Blundell, D. Hassabis, Reinforcement Learning, Fast and Slow, Trends in Cognitive Sciences 23(5)(2019) 408-422.

[11]R. Vaughan, Massively multi-robot simulation in stage, Swarm Intelligence 2(2-4)(2008) 189-208.