Path Planning for Mobile Robot's Continuous Action Space Based on Deep Reinforcement Learning

Tingxing Yan, Yong Zhang*, Bin Wang
School of Electrical Engineering, University of Jinan, China, 250022
e-mail: ytxing_job@qq.com, cse_wangb@ujn.edu.cn, * cse_zhongy@ujn.edu.cn

***Abstract*—In this paper, a path planning method based on deep reinforcement learning in the unknown environment of mobile robots is proposed, in order to meet the path planning of the kinematic model and constraint conditions of the mobile robot under continuous action space. The path planning method plans a path from the starting point to the target point to avoid obstacles, but these planned paths do not meet the kinematic model of the mobile robot and cannot be directly applied to the actual mobile robot control. The proposed approach based on depth reinforcement learning path planning meets the motion model and constraints of mobile robots. The optimal strategy is found in the continuous action space, and the optimal path is obtained through the evaluation criteria. This path is obtained by using the mobile robot motion model, so the movement configuration of the mobile robot can be solved directly. The experimental results show that a mobile robot motion model can be used to plan a collision free optimal path in the unknown environment, and this path is also the actual running track of the mobile robot.**

Keywords-Deep reinforcement learning; mobile robot; unknown environment; continuous action space; path planning

I. INTRODUCTION

Path planning is one of the key technologies in the field of mobile robot research. Path planning means that a robot automatically searches a collision free, optimal or sub optimal path from the initial state to the target state in its environment. At the same time, performance indicators such as distance, time or energy consumption are optimized, and distance is the most commonly used criterion.

Artificial potential field method is a kind of virtual force method, which abstracts the environment into an artificial force field. The motion of a robot in the environment is equivalent to the motion in artificial gravity field [1]. Obstacles impose repulsion on robots, and the target points impose gravity on robots, and their resultant force determines the direction and speed of subsequent motion of robots. The artificial potential field method is simple and easy to calculate, but if the obstacle is near the target point, the mobile robot will approach the obstacle at the same time near the target. According to the gravitational field and the repulsive field function, it is difficult for a mobile robot to reach the target point and will produce a "local optimal solution" problem.

Neural network algorithm refers to the human brain's image thinking, and it is a nonlinear fitting process, which can process and store information in parallel [2]. Artificial neural network has strong generalization ability, self-organization and self-adaptation ability. A network system consisting of large numbers of neurons can fit multiple functions and achieve various behaviors. The self-organizing neural network is trained to improve the cognition of the robot to the environment, and the output of the network is the motion information of the robot. It has strong learning ability and generalization ability, it can deal with complex nonlinear relation, but the convergence speed of neural network algorithm is slow, and it has the problem of sample dependence.

The genetic algorithm simulates the process of natural evolution, which belongs to a kind of evolutionary algorithm. The solution of search strategy and optimization problem is inspired by the theory of natural evolution [3]. Starting from a random population representing the potential solution set of the problem, the population is evolved through cross, selection, mutation and other operations. Finally, the optimal solution set of the problem is obtained. The specific operation is to generate the primary population randomly, a better and better solution to the problem has evolved from generation to generation according to the principle of "survival of the fittest". In each generation, according to the fitness function of the individual to determine which individual can generate the offspring, then imitates the natural genetics to combine and mutate, and evolves the new population, and finally obtains the optimal solution of the problem. Genetic algorithm is easy to integrate with other algorithms, and the search is performed by taking groups as unit, and multiple individuals are compared at the same time. But genetic algorithm cannot make use of the feedback information of the network, the search speed is slow, and the real-time performance needs to be improved.

Fuzzy control is a kind of control method that simulates people's thinking, reasoning and judgment [4]. It expresses human experience and common sense in the form of natural language, which not only eliminates the establishment of mathematical models, but also facilitates the direct conversion of expert knowledge into control signals. This method does not require high accuracy of sensor information and high information about the environment around the robot and its pose information, which makes the behavior of robot shows good consistency, continuity and stability. This method is easy to combine with other methods, and it is suitable for path planning in time-varying unknown environment, so it is widely applied. But because the design of membership functions and the formulation of fuzzy rules

This work was supported partly by National Natural Science Foundation of China (No.61603150), and Doctoral Foundation of University of Jinan (No. XBS1605).

978-1-5386-6136-9/18/$31.00 ©2018 IEEE

mainly depend on expert experience and trial and error, it is difficult to sum up the fuzzy rules, and it is difficult to adjust online once the rules are determined. In addition, too many fuzzy rules are prone to redundancy, which is also a problem to solve. Therefore, how to get the optimal membership functions and fuzzy rules and adjust the fuzzy rules online is an important problem to be solved.

Fuzzy logic, artificial potential field method, genetic algorithm and neural network are all successful and effective robot path planning methods, but these methods usually need to assume complete environment configuration information. However, in a large number of practical applications, robot needs to be able to adapt to uncertain environments. The reinforcement learning method interacts with the environment through the robot and tries to make the action choice to make the maximum cumulative return [5]. It does not need to give any teacher signals under any state, which is completely different from the supervised learning method such as neural network. Learning and optimizing control parameters in the interaction between the mobile robot and the environment, and it has broad application prospects in complex optimization decision problems with less prior information, which not only can effectively compensate for the shortcomings of the previous methods, but also improve the adaptability and self-learning ability of mobile robots in unknown environments.

The path planning method described above does not take into account the kinematic model and constraints of the mobile robot, and the action strategy in the current environment has been obtained, which cannot be applied directly to the actual mobile robot control. The path planning of continuous action space mentioned in this paper starts from the current situation of mobile robot, we get the best path through evaluation criteria under the trial and error mechanism of reinforcement learning by using mobile robot motion model. Because this path is obtained by mobile robot motion model, it can directly solve the mobile robot's action configuration, and the obtained parameters can directly control the actual mobile robot's movement according to the planned path.

II. MOBILE ROBOT MODEL

A. Kinematic Model of Mobile Robot

This paper takes the pioneer3 mobile robot as the research object. The model is shown in figure 1. It is composed of the rear two coaxial drive wheels and the front two steering wheels, the rear wheel drive speed is v. The pose of mobile robot is represented by three-dimensional state vector $q(x, y, \theta)$. Where (x, y) denotes the position coordinates of the reference point of the mobile robot (ie, the midpoint of the robot's rear axle, where it is used as a reference point) in the system coordinate system, and θ represents the angle between the fixed coordinate system of the robot and the fixed coordinate system of the space, that is, the direction angle of the mobile robot. φ is the steering angle, which represents the angle between the mobile robot steering wheel and the horizontal axis of the fixed coordinate system of the mobile robot. The wheelbase of vehicle driving wheel (rear wheel) and steering wheel (front wheel) is L_1.

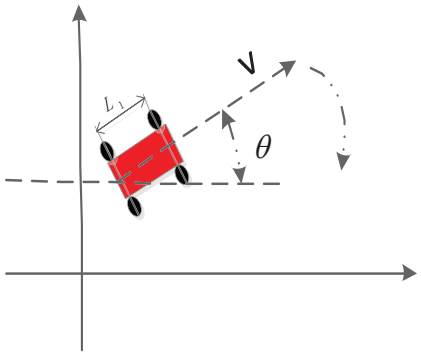

Figure 1. Mobile robot schematic

The kinematics model of the system depicts the mathematical relationship between the location and speed of the system. The kinematics model is as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \left(\frac{vtan\varphi}{L_1}\right) \end{bmatrix} \quad (1)$$

B. Simulation Model of Mobile Robot

Using the python pyglet to create a 500*500 callback map environment as shown in Figure 2. The four inner corner coordinates are (100,100), (400,100), (100,400), (400,400). The robot approximates a rectangle 60 in length and 30 in width. The starting position of the robot (450, 300, -π/2). The speed of the mobile robot is v=50, and the steering angle is continuous, it is randomly selected from the uniformly distributed (-π/4, π/4) interval.

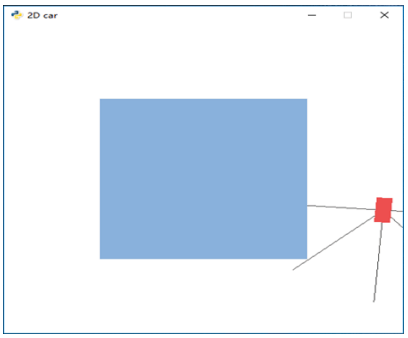

Figure 2. mobile robot simulation environment map

The robot pose can be obtained from the equation (2) Discretization equation, where T is the sampling time. The pose in the simulation process can be obtained in real time, and the robot's shape can be visually displayed in the simulation environment according to the current posture.

$$\begin{cases} x_{k+1} = x_k + T * v * \cos\theta_k \\ y_{k+1} = y_k + T * v * \sin\theta_k \\ \theta_{k+1} = \theta_k + T * \frac{v * tan\varphi_k}{L_1} \end{cases} \quad (2)$$

III. PATH PLANNING FOR CONTINUOUS ACTION SPACE BASED ON DEEP REINFORCEMENT LEARNING

Reinforcement learning is to search for reasonable behavior a under the state of s, and use environmental feedback r to make corrections. Here, the state s refers to the position $(\mathrm{x}, \mathrm{y}, \theta)$ of the mobile robot as the action a had done; the action a refers to the steering angle φ of the execution of the mobile robot selected under the environment state s, and the feedback r refers to the reward of the mobile robot to act a under the state s. In the reinforcement learning problem, the agent can change the state s through the action a, and use feedback r to change the behavior a. The action a and the state s are combined to determine the corresponding feedback r, the interaction process is shown in Figure 3.

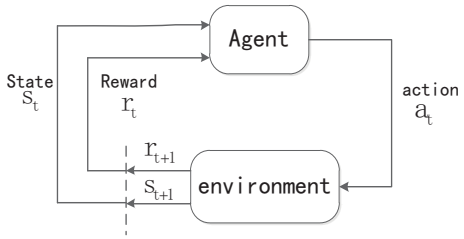

Figure 3. the interaction between the agent and the environment

A. The Basic Idea of Q Learning

When the loop number is t, the state is in the s_t, and the action is at a_t, we can return the expected $\mathrm{E}(R_t|s_t, a_t)$ of the R_t, as a function of s_t and a_t, instead of Q_t:

$$Q_t(s_t, a_t) = E(R_t|s_t, a_t) \quad (3)$$

Q_t is abbreviated as Q, which is a commonly used concept in reinforcement learning and represents the future long-term decay returns of actions in the current state. The one-step Bellman formula for the Q-value strategy can be written directly:

$$\mathrm{Q}(\mathrm{s}, \mathrm{a}) = \mathrm{r}(\mathrm{s}, \mathrm{a}) + \gamma \mathrm{arg}_{a'} \max \mathrm{Q}(\mathrm{s}', \mathrm{a}') \quad (4)$$

B. Deep Q Value Network DQN

DQN is actually a Q learning combination of neural networks. The basic idea of DQN is to learn Q value by neural network, so that Bellman formula is established [6]. The loss L is introduced to measure the degree of DQN network fitting to Bellman formula.

$$\mathrm{L} = |\mathrm{Q}(\mathrm{s}, \mathrm{a}) - \mathrm{r}(\mathrm{s}, \mathrm{a}) - \gamma \max Q'(s', a')|^2 \quad (5)$$

The learning optimization of DQN is equivalent to solving a problem of the most value. Search for the best network parameter set $\{\theta_n\}$, so that the expression map $\mathrm{Q}: (\mathrm{s}, \mathrm{a}) \rightarrow \mathrm{R}$ has the smallest loss L. This is the minimum value problem $\{\theta_n\}$, the problem is solved by the backward propagation of $\nabla \mathrm{L}$ on the Q network.

C. The Principle of DDPG Algorithm

DDPG is developed on the basis of DQN, and the most obvious difference is that the choice of the strategy is no longer obtained by solving $\mathrm{arg}_a \max \mathrm{Q}(\mathrm{s}, \mathrm{a})$, but generated through the learning of the network [7-8]. DDPG adopted action evaluation Actor-Critic architecture system. Using the learning optimization action of μ network, Q network was used to optimize the evaluation aided actions. The DDPG network training flow chart is shown in Figure 4

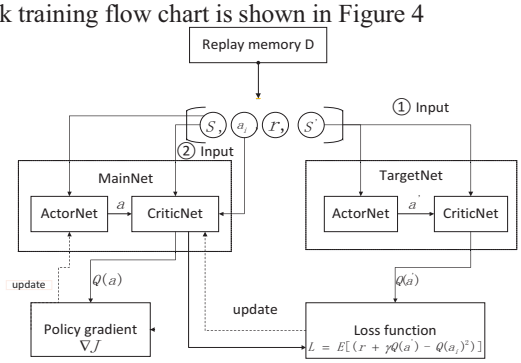

Figure 4. DDPG network training flow chart

In the generation of the strategy, the DDPG neural network can generate the solution independently, instead of selecting the maximum value in the known enumeration results of DQN. The DDPG compensates for the insufficiency of the DQN that can only select limited discrete actions and can solve the training and lifting problem of the continuous strategy generation [9-10]. When dealing with the degree of freedom of the object, it is only the input layer of the linear widening μ and Q network, rather than an exponential increase in the action corresponding in the input layer of the Q network.

D. Path Planning of Mobile Robot Based on DDPG

In the reinforcement learning training, the mobile robot has constant speed. Through the steering angle control direction, the action space A is the continuous action space of $\varphi \in (-\pi/4, \pi/4)$, the position of the robot $(\mathrm{x}, \mathrm{y}, \theta)$ constitutes the state space S of the mobile robot. Since the action space is continuously infinite, the state space is also infinite. When the distance from the reference point of the mobile robot to each boundary less than 60, the mobile robot has a collision, at this time, the given environment feedback $\mathrm{r} = -1$. The environmental feedback $\mathrm{r} = 0$ when no collision occurs.

Set 1,000 episodes of training, the maximum number of steps is 1000 for one episode, the learning rate of the actor network is 0.0001, the learning rate of the criminal network is 0.0001, and the cumulative reward discount factor is 0.9.

Set the memory size to 10000 and record all the steps you have gone through, these can be learned over and over again. The Actor network is updated every 900 steps, and the Critic network is updated every 800 steps.

The deep neural networks with parameters $\theta^{\mu}$and θ^{Q} are used to represent the deterministic strategy $a = \pi(s|\theta^{\mu})$ and the action value function $\mathrm{Q}(\mathrm{s}, a|\theta^{Q})$. The total return expression for the objective function with a discount is:

$$\mathrm{J}(\theta^{\mu}) = E_{\theta^{\mu}}[r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots] \tag{6}$$

The end-to-end optimization of the objective function by the stochastic gradient method is aimed at increasing the total return J. The gradient of the objective function with respect to θ^{μ} is equivalent to the expected gradient of the Q-value function with respect to θ^{μ}:

$$\frac{\partial \mathrm{J}(\theta^{\mu})}{\partial \theta^{\mu}} = E_s[\frac{\partial Q(s,a|\theta^{Q})}{\partial \theta^{\mu}}] \tag{7}$$

According to the deterministic strategy $\mathrm{a} = \pi(\mathrm{s}|\theta^{\mu})$:

$$\frac{\partial \mathrm{J}(\theta^{\mu})}{\partial \theta^{\mu}} = E_s[\frac{\partial Q(s,a|\theta^{Q})}{\partial a}\frac{\partial \pi(s|\theta^{Q})}{\partial \theta^{\mu}}] \tag{8}$$

Update the parameters of the actor network in the direction of increasing Q-value.

The Critic network is updated by updating the value network in the DQN. The formula is:

$$\frac{\partial \mathrm{L}(\theta^{Q})}{\partial \theta^{Q}} = E_{s,a,r,s'\sim D}[(TargetQ - Q(s,a|\theta^{Q}))\frac{\partial Q(s,a|\theta^{Q})}{\partial \theta^{Q}}] \tag{9}$$

$$TargetQ = r + \gamma Q'(s', \pi(s'|\theta^{u'})|\theta^{Q'}) \tag{10}$$

θ^{μ} and θ^{Q} denote the parameters of the target strategy network and the target value network respectively and update the value network with gradient descent.

IV. Simulation Result

As is show in the figures below, figure 5 is the curve of the collision-free steps in the episode, figure 6 is the movement trajectory of the mobile robot after the training is completed, figure 7 shows the curve of the steering angle executed during the movement of the mobile robot in degrees, figure 8 shows the curve of the direction angle transformation during the movement in radians.

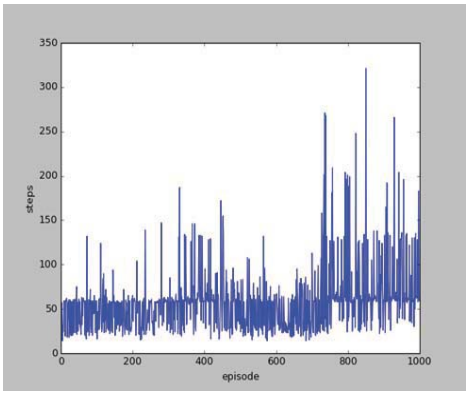

Figure 5. The curve of the collision-free steps in the episode

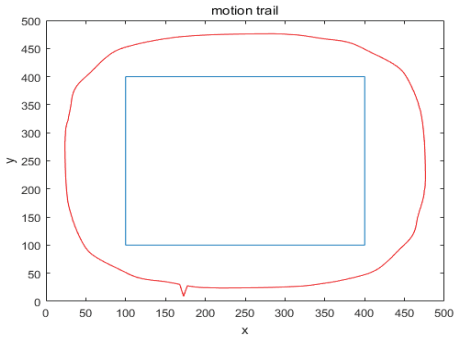

Figure 6. is the movement trajectory of the mobile robot

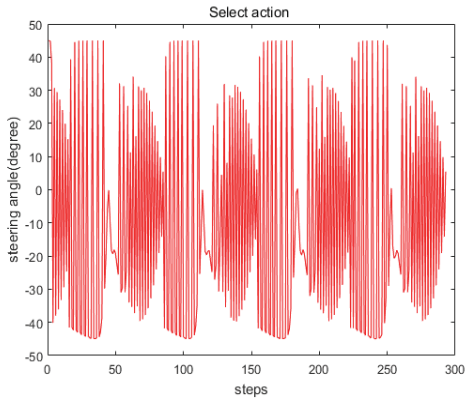

Figure 7. The curve of the steering angle executed during the movement

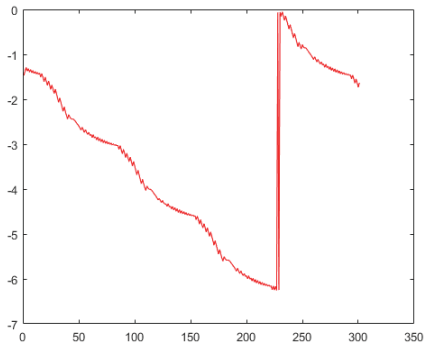

Figure 8. The curve of the direction angle transformation during the movement

It can be concluded from figure that 5 the mobile robot continuously accumulates the knowledge to save the memory through continuous training, the number of steps without collision during one episode gradually increases, and the mobile robot can reach a farther position during training. Figure 6 shows the experience of the mobile robot were called from the memory after the training is completed, it is no longer explored, and can generate collision-free trajectories around the map. Figure 7 shows the steering angle performed during the movement of the mobile robot. It can be seen that the action space of the mobile robot is a continuous interval of $(-\pi/4, \pi/4)$, Performing a

continuous right turn at four right-angle bends for collision-free cornering. Figure 8 shows the change of the direction angle during the movement of the mobile robot. The robot starts from the direction $-\pi/2$ according to the unit transformation. The transition in the graph is due to the conversion of the direction angle from -2π to 0. The final direction angle reached $-\pi/2$ again, indicating that the mobile robot has moved around the map. To sum up, the unknown environment path planning method of mobile robot based on DDPG can realize the path planning of continuous action space. The obtained parameters can be directly applied to the mobile robot control, that is, the planned path is track of movement.

V. CONCLUSION

This paper takes the pioneer3 mobile robot as the research object, the kinematics model of mobile robot is set up, then a method of path planning based on deep reinforcement learning is designed, trained by DDPG algorithm, and the action strategy of mobile robot is optimized by the dual network architecture of Actor-Critic, solve the training problem of a large number of continuous actions of the mobile robots. Through the simulation of the fourth part of the text, it can be seen that the path planning method based on DDPG can solve the optimal path in the unknown environment of the mobile robot with less prior information. Using the kinematic model of the mobile robot to train in the continuous action space, the training is carried out so that the movement configuration of the mobile robot can be directly solved, and the obtained parameters can be directly applied to the mobile robot control, that is, the planned path is track of movement.

REFERENCES

[1] Triharminto H H, Wahyunggoro O, Adji T B, et al. An Integrated Artificial Potential Field Path Planning with Kinematic Control for Nonholonomic Mobile Robot[J]. 2016, 6(4)

[2] Jiang-Wei L I, Lun-Hui X U. Research on Path Planning Based on Simulated Annealing Algorithm and Neural Network Algorithm[J]. Automation & Instrumentation, 2017.

[3] Cai W, Zhu Q, Hu J. Path Planning Based on Biphasic Ant Colony Algorithm and Fuzzy Control in Dynamic Environment[C]// International Conference on Intelligent Human-Machine Systems and Cybernetics. IEEE, 2010:333-336.

[4] Sutton, Richard S, Barto, et al. Introduction to Reinforcement Learning[J]. Machine Learning, 2005, 16(1):285-286.

[5] Song B, Wang Z, Sheng L. A new genetic algorithm approach to smooth path planning for mobile robots[J]. Assembly Automation, 2016, 36(2):138-145.

[6] Osband I, Blundell C, Pritzel A, et al. Deep Exploration via Bootstrapped DQN[J]. 2016.

[7] Duan Y, Chen X, Houthooft R, et al. Benchmarking Deep Reinforcement Learning for Continuous Control[J]. 2016:1329-1338.

[8] Li J, Monroe W, Ritter A, et al. Deep Reinforcement Learning for Dialogue Generation[J]. 2016.

[9] Khaksar W, Vivekananthen S, Saharia K S M, et al. A review on mobile robots motion path planning in unknown environments[C]// IEEE International Symposium on Robotics and Intelligent Sensors. IEEE, 2016:295-300.

[10] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013.