Behavior Control Algorithm for Mobile Robot Based on Q-Learning

Shiqiang Yang[1], Congxiao Li[2]

[1,2] Faculty of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an, P.R.China. 710048,

Email: [1]yangsq@126.com, [2]congcong_0901@126.com

Abstract. In order to adapt to navigation in unknown environment, the mobile robot must have intelligent abilities, such as environment cognition, behavior decision and learning. The navigation control algorithm is researched based on Q learning method in this paper. Firstly, the corresponding environment state space is divided. The action sets mapping with states are set. And the reward function is designed which combines discrete reward returns and continuous reward. The feasibility of this algorithm is verified by computer simulation.

Keywords: Q-learning, navigation, behavior control, simulation

1. Introduction

With the development of intelligent control technology, the mobile robot autonomous navigation control algorithm in unknown environment has become one of the interesting researches in the field of artificial intelligence[1,2]. The mobile robot must have more intelligent abilities, such as environment cognition, behavior decision and learning, to adapt to navigation in unknown environment[3]. Because of the uncertainty of the environment information and the lack of prior knowledge, the robot navigation in unknown environment becomes more complex and difficult. Reinforcement learning is a kind of mapping study between environment states to actions[4]. It hopes to take the actions which can get the largest cumulative award by learning. So, it is a kind of autonomous learning to adapt to the environment that takes environment feedback as the input. It is a good idea that makes a learning agent achieves goal by the interaction with the environment. An agent must perceive the environment and take correct actions to complete the assignment. At the same time, the agent must get one or more environment information about the target state in learning.

The reactive robot navigation control method is researched based on Q learning by the interaction between robot and the environment to adapt unknown different environment. The mobile robot navigation control strategy is represented in this paper.

2. Q Learning Algorithm

Reinforcement learning algorithm is a kind machine learning algorithms between supervised learning

and unsupervised learning. It is a kind of online learning based on reward and punishment mechanism. It learns environment based on deterministic or uncertain returns without model. It adjusts the parameters of the environment through the feedback of learning environment to the system. The mathematical model of reinforcement learning algorithm is based on the Markov chain and dynamic programming. It is a kind of strategy selection optimization method based on trial and error behavior.

Q-learning algorithm is a kind of typical model of reinforcement learning method. Learning agent has experienced ability to select the optimal action in the action sequence in Markov environment. The state estimation value function at different times is achieved while learning to optimize an iterative calculation function, so that the most optimal strategy is obtained. Q-learning is a kind of incremental online learning process. The map of "state - action" is used to iteration Q-learning. It should be ensure that the convergence of each iteration learning process.

The value of $Q(s,a)$ should be initialized in Q-learning process first. An action a should be choose obeying certain strategy, such as epsilon-greedy or the Boltzmann strategy, according to the learning agent initial state s to get the next state s'. And instantaneous reward r is achieved. The value of Q function is updated following the renewal rules. When agent access to the end of the target state or meet the correct conditions to end, the Q algorithm complete one cycle learning. Agent would continue the iteration cycle until the end of learning.

The optimal function values are approximated by optimizing value of the iterative function $Q(s,a)$ in Q-learning process. The renewal rules in Q-learning process are as formula (1) and formula (2).

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + a\Delta Q(s_t,a_t) \tag{1}$$

$$Q(s_t,a_t) = (1-a)Q(s_t,a_t) + a[r_{t+1} + \gamma \max_{a\in A} Q(s_{t+1},a)] \tag{2}$$

Where a represents learning step. r_{t+1} represents the instantaneous rewards while the action a_t is performed in the process from the state s_t to state s_{t+1}. γ represents the discount rate of the $\max_{a\in A} Q(s_{t+1},a)$.

A lot of episodes are contained in Q-learning process. And they all repeat the following steps[5].

Step 1: observe the current state s_t;

Step 2: choose and perform an action a_t;

Step 3: observe the next state s_{t+1};

Step 4: receive a transient signal of rewards and punishments r_{t+1};

Step 5: update the value of Q function following to the formula (2);

Step 6: turn to the next moment, $t \leftarrow t+1$.

3. The Design of Q-Learning System for Mobile Robot Navigation

It needs two main functions, to avoid obstacles and to move to target, while the robot reactively

navigating in an unknown environment. The design of the Q-learning system from the robot model, environment state, action space, and reward in return to make it intelligent. For the convenience of description and to simplify the algorithm, the obstacles which site the 180° range only ahead the robot are considered. Obstacles position is divided into three directions: the left, right and ahead.

3.1 The division of the environment state

Q-learning is based on the discrete Markov decision model. So discretizing the continuous environment space into limited environment unit is needed for mobile robot navigation by Q-learning. It discretizes the environment from two aspects, the distance from robot to the obstacle and the angle from the robot to the target.

The environment is divided into three states according to the distance from robot to the obstacle:

$$s_{d-ro}=\begin{cases} sf & d_{R-O}>d_{sf} \\ m & d_{\min}\le d_{R-O}\le d_{sf} \\ d_g & d_{R-O}<d_{\min} \end{cases} \tag{3}$$

Where d_{R-O} represents the distance between the robot and the obstacles. And d_{R-O} is always less than the maximum distance which robot could detect. d_{sf} is set to the minimum safe distance without obstacle avoidance. $d_{\min}$ is set to robot safety threshold.

Define the distance on the left side as $s_{LR\_O}=d_l$, which is the minimum distance of the left side to the barrier measured by the sensor. Define the distance of the obstacles ahead as $s_{FR\_O}=d_m$, which is the minimum distance to obstacles ahead. Define the distance of the right obstacles as $s_{RR\_O}=d_r$, which is the minimum distance of the right side to the barrier.

The environment state is divided into the following eight states by the perspective of the angle from robot to target:

$$s_{a\_rg}=\{\mathrm{NVB},\mathrm{NB},\mathrm{NS},\mathrm{NVS},\mathrm{PVS},\mathrm{PS},\mathrm{PB},\mathrm{PVB}\} \tag{4}$$

Where NVB represents the target lies on the left rear of the robot within a range of 180°-135° counter clockwise. NB represents that the target locates on the robot left rear within the range of 135°-90°. NS represents that target lies on the left rear within the range of 45°~0°. $\{\mathrm{PVS},\mathrm{PS},\mathrm{PB},\mathrm{PVB}\}$ represent the case that the target lies on the robot right, it is just the reverse to the above. So the mobile robot state can be expressed as $s_{a\_rg}\in\{NVB,NB,NS,NVS,PVS,PS,PB,PVB\}$ .The definition of the eight states can be shown as the following.

$$s_{a-rg}=\begin{cases} NVB & -\pi\le \arg<-3\pi/4 \\ NB & -3\pi/4\le \arg<-\pi/2 \\ NS & -\pi/4\le \arg<-\pi/4 \\ NVS & -\pi/4\le \arg<0 \\ PVS & 0<\arg<\pi/4 \\ PS & \pi/4\le \arg<\pi/2 \\ PB & \pi/2\le \arg<3\pi/4 \\ PVB & 3\pi/4\le \arg<\pi \end{cases} \tag{5}$$

Where a_{rg} represents the angle between the forward direction and the direction from current point to the target point.

In summary, the environment state set S by discretization can be expressed as:

$$S=s_{a\_rg}\cup s_{d\_ro} \tag{6}$$

3.2 The definition of robot action

Autonomous mobile robot motion can be decomposed into translational motion and rotation motion. So the robot action set should be defined according to the rotation motion and translation motion. The concrete are defined as follows.

$$A = \{a_1, a_2, a_3, a_4, a_5\} \tag{7}$$

Where a_1 represents that the robot contrarotates for 45° and the mobile robot moves step is Step long, but Step=0. This means it just makes a pure rotation without shift. a_2 represents rotating for 15° counterclockwise and the mobile robot moves Step, but Step=0. a_3 represents the robot rotate for 0° and the mobile robot moves Step, Step= STEP(STEP for setting the robot moving step length unit). This means the robot make a pure shift without rotation. a_4 represents rotating for clockwise 15° and Step=0. a_5 represents rotating for clockwise 45° and Step=0.

3.3 The setting to reward function

Reward in return is immediate return to the robot for taking an action in a state. It can not evaluate the action is good or not in global, but reward function setting is a key factor to Q-learning system to get the corresponding strategy. It' s the unique useful information can be used in the strategy learning. Reward return function setting is the key to Q-learning system and learning strategies. A simple discrete efficient reward function is proposed after the above environment states division by the distance to obstacle and angle to target.

$$r_{a\_rg} = \begin{cases} -2 & s_{a\_rg} = NVB \quad or \quad s_{a\_rg} = PVB \\ -1 & s_{a\_rg} = NB \quad or \quad s_{a\_rg} = PB \\ 1 & s_{a\_rg} = NS \quad or \quad s_{a-rg} = PS \\ 4 & s_{a\_rg} = NVS \quad or \quad s_{a\_rg} = PVS \end{cases} \tag{8}$$

$$r_{d\_ro} = \begin{cases} 5 & s_{d\_ro} = sf \\ -1 & s_{d\_ro} = m \\ -5 & s_{d\_ro} = dg \end{cases} \tag{9}$$

Where r_{a_rg} represents the rewards in state s_{a_rg} . And r_{d_ro} represents the rewards in state s_{d_ro} .

Continuous reward in return for each action is set according the distance from current position to the target to guide robot towards the goal quickly. The specific design of the reward pay back system is shown as follows:

$$r_{d\_rg} = R_0 * Setp * \cos(a_t) / d_t \tag{10}$$

Where R_0 is the initial parameter values which depends on the value of the reward in return shown above. And a_t represents the angle from the moving direction to target point at the time t . The d_t represents the distance from the robot location to the target at the time t . The value of reward return r_{d_rg} is increasing while robots getting closer and closer to target.

4. Reactive Navigation Algorithm Based on Q-Learning and Simulation

The learning algorithm is introduced into the autonomous mobile robot reactive navigation control. Q-learning steps are as follow:

Step 1: Define state space and action space in the simulation environment. The reward return follows formula (8) and (9) above. Point (10, 15) sees as initial starting location and 0° as the attitude angle. The point (42, 30) sees as the target point. Initialize the parameters in Q-learning: Step = 1, $Q(\mathrm{s},\mathrm{a})=0$, $e(\mathrm{s},\mathrm{a})=0$ to all map state-action;

Step 2: Observe the current state s_t ;

Step 3: Select an action a_t according to the Boltzmann distribution action selection mechanism, and then execution;

Step 4: Observe the new state s_{t+1} and get reward $r(s_t,a_t)$;

Step 5: Calculate the value: $\delta \leftarrow r(s_t,a_t)+\max_{a\in A} Q_{t-1}(s_{t+1},a)-Q(s_t,a_t)$, $e_t(s_t,a_t)\leftarrow 1$;

Step 6: Calculate the Q-function: $Q_t(s,a)\leftarrow Q_{t-1}(s,a)+a\delta e_t(s,a)$;

Step 7: Judge whether the distance from robot to obstacle is less than the safe distance in the new state s_{t+1} . If the distance is less than the safe distance, turns back to the starting point and restart from the second step. Whether to reach the target point should be judged if it is greater than the safe distance. If it is true, then end the iteration, or turn to the second step to continue.

Two different simulation environments are set up to verify the feasibility of this above algorithm. The position (x, y) is represented for vertical and horizontal coordinates in simulation. The robot is simplified to a point and the obstacles are simplified to solid circles. Many random arrangement circular obstacles are in the environment. Robot starting point lies in the lower left corner in the simulation environment and the target is located in upper right, and both of them are shown as “ ☆ ”. The location of robot is shown as “※” in the simulation figure, and the line of “※” shows robot path from the initial starting point to the target.

As shown in Fig. 1 is the first circle learning process based on the Q learning, 30 obstacles are arranged random in the simulation. It can be seen that robot is always move to the target with much repetition. The path is extremely unreasonable and the robot fails to reach the target at last.

The results after 237 learning cycles are shown in Fig. 2. A feasible path is found from the chaotic state shown in figure 1, and robot arrives at the target safely. This path is better than the first learning circle apparently.

The simulation environment is changed in Fig. 3 and Fig. 4. The number of obstacles is 40 with random arrangement.

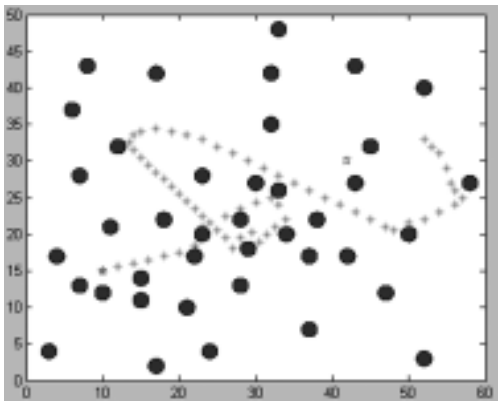

Fig.1 Simulation after 1st cycles learning

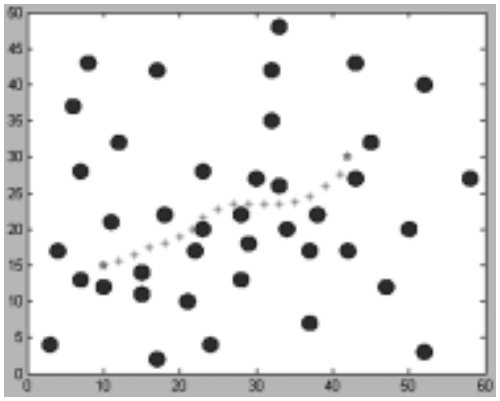

Fig.2 Simulation after 237 cycles learning

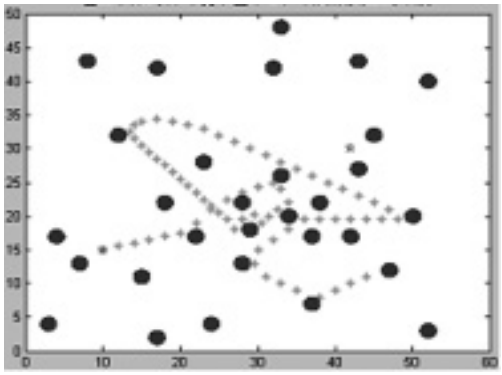

Fig.3 Simulation after 1st cycle learning

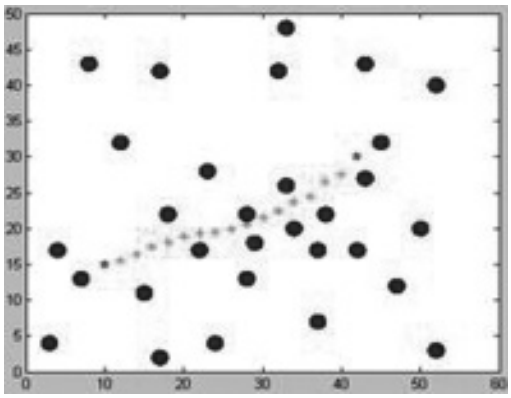

Fig.4 Simulation after 237 cycles learning

It can be seen that the path is in chaos after 1st cycle learning. The robot arrive at the target after 237 cycles learning. The path is feasible and safe.

According to the different unknown environments simulation, it can be seen that the robot behavior control algorithm based on Q-learning can adapt to the changes of the environment. Robot can get a feasible path by the online learning. Reactive navigation based on Q-learning has the effective path optimization.

5. Summary

The mobile robot navigation method is researched based on Q-learning method in unknown environment. The method to divide the environment state, to set the action sets and to set reward function are discussed. The Boltzmann selection mechanism is adopted for the learning process to action selection. The simulation results show that the reactive autonomic navigation approach based on the Q-learning can adapt to complicated unknown environment, this method is effective and feasible.

References

[1] Guo Rui, Wu Min, Peng Jun,etc. A new Q learning algorithm for multi-agent systems. Acta automatica sinica, Vol.33, No.4, p. 367-372, 2007

[2] LIAN Chuanqiang, XU Xin , WU Jun, LI Zhaobin. Q-CF multi-Agent reinforcement learning for resource allocation problems. CAAI Transactions on Intelligent Systems, Vol.6, No. 2, p.95-100, 2011

[3] FANGMin，LI Hao. Heuristically Accelerated State Backtracking Q－Learning Based on Cost Analysis. PR & A, Vol.26, No.9, p. 838-844, 2013

[4] HU Jun, ZHU Qing-bao. Path planning of robot for unknown environment based on prior knowledge rolling Q-learning. Control and Decision, Vol.25, No.9, p.1364-1368,2010

[5] J. C. H. Watkins Christopher, Dayan Peter. Q-learning. Machine Learning, Vol.8, No.3, p.279-292,1992

Author Brief and Sponsors

Shiqiang Yang is PhD. and associate professor in mechnical engineer department of Xi'an University of Technology. And his research interest is on mobile robot control. This work was financially supported by the National Natural Science Foundation of China (Grant No.51475365), the Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No.2013JK1000).