Intellectual Method of guiding Mobile Robot Navigation using Reinforcement Learning Algorithm

G.Nirmala
Assistant Professor,
Department of Computer Science and Engineering
Kamaraj College of Engineering and Technology
Virudhunagar

Dr.S.Geetha
Professor
School of Computing Science and Engineering
VIT-University, Chennai Campus, Chennai

Dr.S.Selvakumar
Professor
Department of Computer Science and Engineering
G.K.M. College of Engineering and Technology
Chennai

***Abstract*—one of the interesting parts in mobile robot is to navigate independently. It is a difficult task, which requiring a complete showing of the environment and intelligent algorithm. This paper presents an Intellectual navigation method for an autonomous mobile robot which requires only a learning signal such as a feedback indicating the quantity of the applied action. The Q-learning algorithm of reinforcement learning is used for the mobile robot navigation by discrete states and actions in the environment. The markov decision process is used to improve the performance of the robot navigation. The effectiveness of this optimization method is verified by simulation.**

Keywords—mobile robot, Intellectual algorithm, markov decision process, Qlearning

I. INTRODUCTION

Navigation is the important aspect of the movement of autonomous mobile robot. It is considered as a task of determining a collision-free path that enables the robot to travel through an obstacle course, starting from source position and ending to a goal position in a environment where there are one or more obstacles, by respecting the constraints movement of the robot and without human intervention. The process of finding such optimized path is also known as path planning problem of mobile robot [1]. Obstacle avoidance is one of the key assignments of a mobile robot. It is a major task that must have all the robots, because it permits the robot to move in an unknown environment [1] [2]. A control strategy with a learning capacity can be carried out by using the reinforcement learning; which the robot receives only a feedback. This reinforcement makes it possible the navigator to adjust its strategy in order to improve their performances. It is deliberated as an automatic variation of the robot performance in its navigation environment [3]. The reinforcement learning is a method of optimal control, when the agent starts from an ineffective solution which gradually improves according to the knowledge gained to solve a successive decision problem [4]. To use reinforcement learning, several approaches are possible. The first consists in manually discrete the problem for obtaining states and actions spaces; which could be used directly by algorithms using Q tables [4]. It is however necessary to pay attention to the choice of discretization, so that they allow a correct learning by providing states and actions which contain a intelligible rewards. The second method consists in working at continuous spaces of states and actions by using value functions [5]. Indeed, to use the reinforcement learning, it is necessary to estimate correctly the value function. The results obtained show substantial improvements of the robot behaviors and the speed of learning

The present paper is organized as follows: Section 2 describes related work of the mobile robot path planning. Section 3 gives the necessary background of reinforcement learning and we discuss the Q-learning algorithm for a searching goal task. In section 4 describes results and discussion and section 5 concludes this paper.

II. MOBILE ROBOT PATH PLANNING

R. S. Sutton and A. G. Barto proposed One of the basic representative architecture for RL is Q-learning, which estimate the discounted future rewards for taking actions from given states based on Qlearning learning. Yishay Mansour proposed the various learning rates for constructing optimal policy using Qlearning strategy. Torvald Ersson and Xiaoming proposed the path planning of robot navigation in an unknown environment with grid model for path selection

III. REINFORCEMENT LEARNING APPROACH

In reinforcement learning, an agent acquires to optimize collaboration with a dynamic environment through trial and error. The agent receives a scalar value or reward with every action it executes. The goal of the agent is to learn a strategy for selecting actions such that the expected sum of discounted rewards is maximized [4]. In the standard reinforcement learning model, an agent is associated to its environment via perception and action. At any given time step t, the agent perceives the state s_t of the environment and selects an action a_t . The environment returns by giving the agent reinforcement signal and changing into state S_{t+1}. The agent chooses the actions that tend to increase the long run sum of values of the

978-1-4799-1854-6/15/$31.00 ©2015 IEEE

reinforcement signal r. It can learn to do this overtime by trial and error, directed by a wide variety of algorithms [4][9]. template is used to format your paper and style the text.

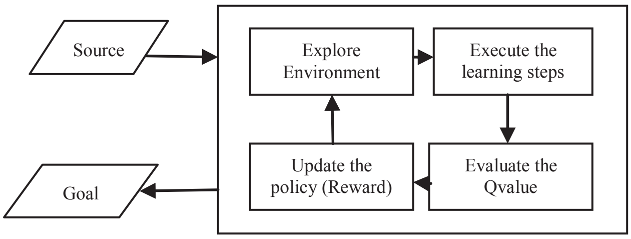


Fig. 1. Policy improvement with reinforcement learning

The agent goal is to find an optimal policy, π: {S, *A*} which maps states to actions that maximize some long run measure of reinforcement. In the general case of the reinforcement learning problem, the agent's actions determine not only its immediate rewards, but also the next state of the environment. As a result, when taking actions, the agent has to take the future into account. Generally the value function is defined in a problem of the form of a Markovian decision-process

$$V_{\pi(s)} = E_\pi(\Sigma\ \gamma^k r_{t+k} \mid s_t = s) \tag{1}$$

where $\gamma\ \varepsilon(0,1)$ is a factor to adjust the importance of future returns. The most algorithms of reinforcement learning use a quality function noted Q-function, representing the value of each pair state-action to obtain an optimal behavior. It gives for each state, the future return if the agent pursues this policy π :

$$Q^{\pi}(s, a) = E_{\pi}(R_t \mid s_t = s, a_t = a) \tag{2}$$

The optimal quality is:

$$Q^*(s, a) \max Q^{\pi}(s, a) \tag{3}$$

A. Q-learning

The proposed idea of Q-learning is to learn a Q-function that maps the current state s_t and action a_t to a utility value, that predicts the total future discounted reward that will be received from current action a . In that it learns the optimal policy function incrementally as it interacts with the environment after each transition (s_t, a_t, r_t, s_{t+1}). This update is done by observation of the instantaneous transitions and their rewards associated by the following equation:

$$Q(s_{t+1}, a_{t+1}) = Q(s_t, a_t) + \eta(r_{t+1} + \gamma \text{ Max } (Q(s_{t+1}, a_{t+1})) \tag{4}$$

Where $Q(s_t, a_t)$ is the value function of state-action pair at the moment t, η is the learning rate, γ is a discount factor, and $r_{t+1}$is the reward received while taking action at a_t state s_t.

B. Learning behavior using Q-learning algorithm

At each time step the robot must define the state in which it is, and starting from this state, it must make a decision on the action to be carried out. According to the result obtained during the execution of this action, it either is punished, to decrease the probability of execution of the same action in the future, or reward, to support this performance in the similar situations. For a goal finding task by a mobile robot, the around space is divided into sectors according to the angle between the direction of the robot Action at State st . Agent Environment state s_{t+1} reinforcement r_t . The delivered actions are: turn left, turn right, bottom and top. These actions are chosen by the exploration-exploitation policy (PEE) in order to explore the state spaces. During the learning phase, the robot receives the following values as reinforcement signals.

C. Q-Learning Algorithm

```
1. Initialize randomly Q( $s_t$, $a_t$)
2. t ← 0, observe the state $s_t$
3. For each state, compute Qvalue
4. For each action, choose a conclusion with the Exploration and Exploitation Policy
5. Compute the action A($s_t$) and correspondence quantity Q( $s_t$ ,$a_t$)
6. Apply the action ($s_t$ ). Observe the new state $s'_t$ .
7. Receive the reward $r_t$ .
8. Compute a new evaluation of the state value.
9. Update parameters Q value using this evaluation.
10. t←t + 1,Go to 4.
```

In the learning phase, in order to improve the used navigation of mobile robot, the initial positions are selected randomly, where each episode starts with a random position and finishes when the robot reached the target or strikes the limits of its environment. For a random position of the mobile robot selects the optimized path of the mobile robot using Q-learning algorithm is to improvement of the robot performance. The maximization of the average values of the received reinforcements. In all cases, the mobile robot moves toward the goal for initial position by executing continuous actions. The learning is faster than the previous using Q-learning algorithm.

Several implementations of the Q-learning algorithm were applied by varying the number of states and actions suggested to obtain an acceptable performance. The increasing of the state action pairs makes it possible to improve the behavior of the robot, but requires a more significant memory capacity and time learning. The use of the Q-learning algorithm requires the storage of the Q-functions for all pairs (state - action). We can use tables. But in the case of continuous spaces of states and actions like the mobile robot navigation task; the number of situations is infinite and the representation of the Q-function by tables is difficult. The universal value function offer promising solutions for approximating the Q-values.

IV. RESULTS AND DISCUSSION

A. Mobile Robot trajectory after the learning process with maximized reward

This paper was obtained by simulation of learning methodology for Reinforcement learning for mobile robot navigation using V-Rep Pro Edu simulator.

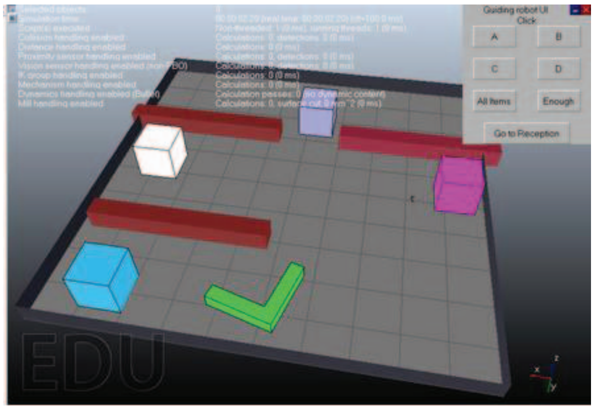


Fig. 2. Simulated grid environment

In order to improve the mobile robot performances, we use in this work, these controllers are characterized by the introduction of without prior knowledge so that the initial performance is acceptable.

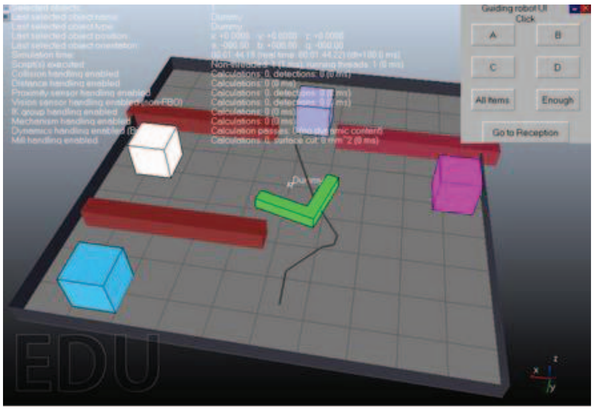
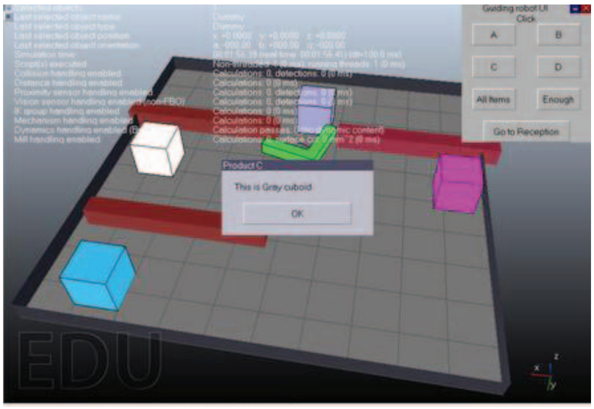


Fig. 3. Goal seeking using Q-learning algorithm

B. Q-values and Learning Rate

The behavior of its Q-values with the learning parameter from 0.1 to 0.9. “Goal” denotes the action command produced by the Move to Goal behavior. As the learning rate has the effect on the performance, reinforcement learning increases with respect to the Q-values


Product C

This is Gray cuboid

OK

Fig. 4. The mobile robot reaches the goal (Gray Cuboid)

The mobile robot selects the optimized path between the source and goal (Gray cuboid) in grid environment (10 X 10). Mobile robot avoids the obstacles in the environment. During the learning phase the mobile robot learn the environment. After a learning phase, the robot obtains the best path to reach the target (any one of the target position (cuboid A, B, C, D or mobile robot can guide with all targets). The selection can be based on the maximized the total rewards. The robot can avoid the obstacles and moves in the direction of the target. If there is a near obstacle, it chooses the turn right action.

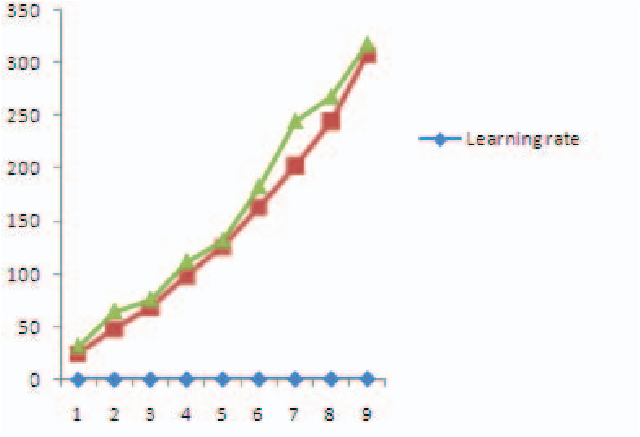


Fig. 5. Learning rate with Q-values algorithm

In order to generalize the robot navigation for all possible situations, the training is made with a random initial position of the robot and target in each episode. Figure shows the robot paths obtained after a learning task. As depicted, the robot moves toward the target from its initial position (the robot can reach the target in all cases) by executing discrete actions. Like a learning indicator, figure shows the average return per trial performance of the controller during the learning process. It is observed that the behavior improves during the learning time.

V. CONCLUSION

In this paper, we presented an Intellectual method for the mobile robot navigation using reinforcement learning algorithm. This technique is based on the optimization of the value function in order to maximize the reward function. The Q-learning algorithm is a powerful tool to obtain an optimal performance which requires only one feedback indicating the quality of the applied action. This feedback signal makes the navigator able to adjust his strategy in order to improve its performances. This algorithm combines the advantages of the two techniques and regarded on the one hand as a method of a markov decision process, and Q-learning algorithm to continuous state and action spaces. Environment with obstacle, the mobile robot produced to increase the performance of 63% to the mobile robot obstacle learning and the learning time performance with 66.8% increased. The optimization function parameters and number of episodes will improve the performance of the proposed method.

References

[1] Wentao Yu1, Jun Peng1, Xiaoyong Zhang1 and Kuo-chi Lin2, A Cooperative Path Planning Algorithm for a Multiple Mobile Robot System in a Dynamic Environment" International Journal of Advanced Robotic Systems, pp. 225- 236, 2014

[2] Shuzhi Sam Ge, Frank L. Lewis, “Autonomous Mobile Robots, Sensing, Control, Decision, Making and Applications”, CRC, Taylor and Francis Group, 2006

[3] L. Khriji and al,” Mobile Robot Navigation Based on Q-learning Technique”, International journal of advanced Robotic System, Vol. 8, No. 1, pp 45-51, 2011.

[4] E. Sacks, “Path planning for planar articulated robots using configuration spaces and compliant motion,” IEEE Transactions on Robotics and Automation”, vol. 19, no. 3, 2003.

[5] L. M. Zamstein, A. A. Arroyo, E. M. Schwartz, S. Keen, B. C. Sutton, and G. Gandhi , "Koolio : Path Planning using Reinforcement Learning on a Real Robot Platform " FCRAR, 19th Florida Conference on Recent Advances in Robotics, 2006.

[6] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. The MIT Press, Cambridge, MA, 1998.

[7] P. Y. Glorennec, and L. Jouffe, "Fuzzy Q-learning," in Proc. 1997, FUZZ-IEEE’97, 6th IEEE International Conference on Fuzzy Systems, pp. 659-662, 1997.

[8] C. Ye, N. H.C. Yung and D. Wang, "A Fuzzy Controller with Supervised Learning Assisted Reinforcement Learning Algorithm for Obstacle Avoidance" IEEE Trans. Syst., Man, and Cybern. B, vol. 33, no.1, pp.17-27, 2003.

[9] Maaref H., and Barret CSensor Based Navigation of an Autonomous Mobile Robot in an Indoor Environment”, Control Engineering Practice, Vol. 8, pp. 757-768, 2000.

[10] A. Oustaloup. A. Poty, and P. Melchior, “Dynamic path planning by fractional potential,” in Second IEEE International Conference on Computational Cybernetics, 2004.

[11] E. Prestes. Jr., P. Engel, M. Trevisan, and M. Idiart, “Exploration method using harmonic functions,” Journal of Robotics and Autonomous Systems, vol. 40, no. 1, pp. 25–42, 2002.

[12] S. T. Hagen and B. Krose, "Q-Learning for Systems with Continuous State and Action Spaces,"in Proc. BENELEARN,10th Belgian-Dutch Conference in Machine Learning, 2000.

[13] S. Singh, T. Jaakkola, M. L. Littman, and et al, “Convergence results for single-step on-policy reinforcement-learning algorithms,” Machine Learning, vol. 39, pp. 287–308, 2000

[14] Yadav Ramashare, Anurag Upadhyay, and Jainendra Shukla"Outlook of Mobile Robot Navigation in Dynamic Environment", International Journal of Current Engineering and Technology Vol.3, No.2 pp.414-416, 2013.