Path Finding Algorithms for Autonomous Robots Based on Reinforcement Learning

Tran Xuan Sang, TranQuoc Kiet, Nguyen ThiUyen

Normally, the problem of path finding is solved by classical algorithms, such as the Dijkstra, Bellman-Ford,Johnsonalgorithm. However, the classical path finding algorithms cannot handle the problem of dynamic environment in the realworld.This paperintroducesmodelforautonomous robot path finding based onreinforcement learning algorithms.We describe the basic concept, principle and the method of reinforcement learning algorithms, and do some simulation experiments in order to evaluate the effectivenessamong different reinforcement learning algorithms.Simulation results show that Q-learning algorithm does not work well when the environment has changed constantly and there are too many obstacles.Both SARSA algorithms and Q-learning algorithms are relatively slow in convergence.Meanwhile, Dyna-Q algorithm determines the path of the robot relatively fast and the stability rates are high.

***Index Terms*— Path Finding, Reinforcement Learning Algorithms, Q-learning, Autonomous Robots**

I. INTRODUCTION

Nowadays, robotics has been rapid development. "Robots are able to work on repetitive tasks tirelessly and continuously and in many businesses they are welcomed as valuable team members because they do the work that humans don't want to do" Professor Willcocks says. However, the mobile robot willencounter a variety of obstacles in explore environment. In the real world, the environment has always changed. In order to adapt to unknown environments, mobile robot should has learning ability. In the field of machine learning, there are three types of learning:supervised learning, unsupervisedlearning and reinforcement learning. Supervised learning is based on the training data set which has input variables (x) and an output variable (y) and then an algorithm is used to learn the mapping function from the input to the output. Unsupervised learning is based on training data set whichhas only input data (x) and no corresponding output variables.Unsupervised learning can model the underlying structure or distribution in the data in order to learn more about the data.

Recently, reinforcement learning has been widely used in intelligent control, robotics, and decision analysis and other fields. Reinforcement learning (RL) is a real-time, online learning method. Interacting with the environment, robot selects and performs actions and affect environment. At the same time, according to the reinforcement signal which is environment given to, robot constantly adjust their actions and find the optimal movement strategy through trial and error method, and the system behavior obtain the maximum value of the accumulated reward from the environment [1]. Reinforcement learning is based on reward value which receives when interacting with a complex, uncertain environment. The agent will try to maximize the total amount of reward. Reinforcement learning is best suited for autonomous robot path finding because it allows robot dealing with dynamic environment.Beyond the agent and the environment, one can identify four main sub-elements of a reinforcement learning system: a policy, a reward function, a value function, and, optionally, a model of the environment. A policy defines the learning agent's way of behaving at a given time, a reward function defines the goal in a reinforcement learning problem and indicates what is good in an immediate sense, a value function specifies what is good in the long run, and, the model of the environment is something that mimics the behavior of the environment. Fig 1 represents the general model of Reinforcement Learning[2].

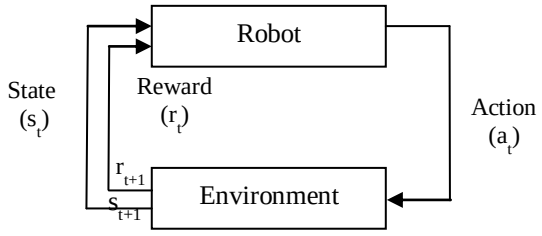

Fig 1: Reinforcement LearningModel

In this paper, somepath finding algorithms for Autonomous Robotsbased on reinforcement learningareintroduced. The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 describes themodel of autonomous robot path finding based onreinforcement learning. Section 4 provides the experimental results of different reinforcement learning algorithms. Finally, section 5 summarizes the work of this paper.

II. RELATED WORKS

There are several researches in path finding for autonomous robots.O.Hachourproposed algorithm allows a mobile robot to navigate through static obstacles, and finding

Manuscript received April, 2017.
***Tran Xuan Sang**, Computer Science Department-Vinh University (e-mail: sangtx@vinhuni.edu.vn). Vinh city, Viet nam.*
***Tran QuocKiet**, Center of Continuing Education, Tan chau district –TayNinh– Viet nam.(email: kiet093tn@gmail.com), TayNinh, Viet nam.*
***Nguyen ThiUyen**, Computer Engineering Department-Vinh University (e-mail: uyennt@vinhuni.edu.vn). Vinh city, Viet nam.*

the path in order to reach the target without collision [3].PuShiet.al proposeda dynamic path planning scheme based on genetic algorithm for navigation and obstacle avoidance of mobile robot under unknown environment [4].Lee et.al proposed a high-level robot fuzzy navigation method in unknown environment, with the ultrasonic sensor to provide environmental information, and the navigation based on fuzzy control to calculate the information[5].Babuet.al developedan autonomous robot that shows the use of Q-learning for navigation in a complete unknown environment [6]. Woo et.al proposed a reinforcement learning approach involving a shortest path finding algorithm. This paper presented how to cope with slow-convergence problem in learning process and how to partly solve non-Markov problem [7].Yun et.al proposed a method of Genetic Algorithm (GA) for mobile robot to move, identify the obstacles in the environment, learn the environment and reach the desired goal in an unknown and unrecognized environment [8]. Reinforcement learning methodsare widely used in various robotic systems. El-Fakdi& Carreras (2013) proposed a two-stepgradient-based reinforcement learning control system for solving the actionselection problem of an autonomous underwater vehicle in a visual-based cabletracking task. A neural network reinforcement learning method was studied forvisual control of a robot manipulator (Miljković et al., 2013). A direct mappingfrom the image space to the actuator command was developed by using two differentRL algorithms, Q-learning and SARSA, combined with neural networks.

III. REINFORCEMENT LEARNING-BASED PATH FINDING

In this section, we explain how to model the environment and how to apply Reinforcement Learning algorithm for robot path finding problem.

3.1. Environment´s Model

The robot should have set of *states* and *actions*. In case of robot path finding problem, location of robotcan be consider as*state*; and robot's movement from one location to another location is called *action.*

The environment can be modeled by a graph as follow. Vertex presents the locations of robot. The direct connections among these locations are presented by edge of graph.Each edges of graph are assigned reward values. For initial step, the reward values are set as below:

$$\begin{cases} a_{ij} = 100 \; if \; \text{vertex}_j \; is \; destination \\ a_{ij} = 0 \; if \; \text{vertex}_j \; is \; not \; destination \end{cases} \quad (1)$$

An autonomous robot can be placed in any node and try to find the optimal route to reach the destination.

Figure 1 show an example of environment´s model.Robot will start from location P3 and then find the route to reach to location P6.

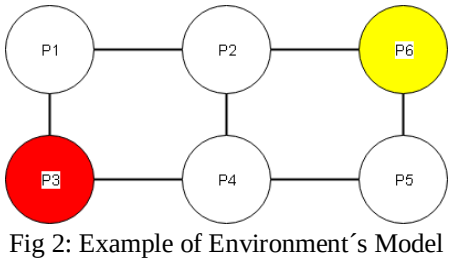

Fig 2: Example of Environment´s Model

In above example, there are six locations and each location has direct connectionspresented by edges.

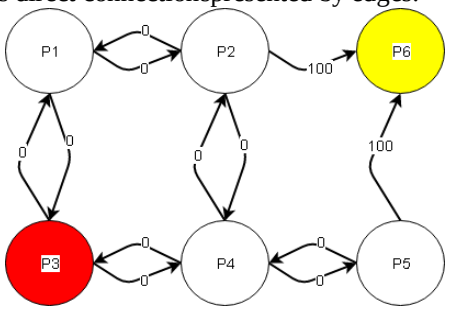

Fig 3: Graph with reward values

Fig 2 presents the graph with reward values. The edges which connect directly to destination P6 will have reward value of 100.

In this model, if there is no directconnection between two nodes, it means there is an obstacle between two locations corresponding to the nodes. For example, there is an obstacle between nodes P3 and P2etc.

3.2.Q-learning Algorithm

The Q-Learning is a reinforcement learning method which is used to find optimize solutions in Markov decision process problems. At each step of time, a robot observes the currentstates$_t$ then chooses an action u_t t to perform. As the robotmoves to states$_{t+1}$, the robot receives a reinforcement valuer(s_t , u_t). The goal of the training is to find the sequential order of actions which maximizes the sum of the future reinforcement values, thus leading to the shortest path from departure to destination.

The transition rule of Q-learning is shown as below:

*Q(state, action) = R(state, action) + gamma * Max[Q(next state, all actions)]* (2)

The gamma parameter has a range of 0 to 1 (0 <= gamma < 1), and ensures the convergence of the sum. If gamma is closer to zero, the robot will tend to consider only immediate rewards. If gamma is closer to one, the robot will consider future rewards with greater weight, willing to delay the reward.

The Q-Learning algorithm goes as follows [9]:

1. Set the gamma parameter, and environment rewards in matrix R.
2. Initialize matrix Q to zero.
3. For each episode:
Select a random initial state.
Do While the goal state hasn't been reached.
+ Select one among all possible actions for the current state.
+ Using this possible action, consider going to the next state.
+ Get maximum Q value for this next state based on all possible actions.
*+ Compute: Q(state, action) = R(state, action) + gamma * Max[Q(next state, all actions)]*
+ Set the next state as the current state.
End Do
End For

3.3. SARSAAlgorithm

The Sarsa algorithm was introduced by Rummery and Niranjan [10]. It is a very similar to Q-learning. The difference is that Sarsa is an on-policy algorithm, which

means that it updates the Q-function of the policy that is executing and therefore changes the policy as it runs.The Q-value depends on the current state of the agent "S1", the action the agent chooses "A1", the reward "R" the agent gets for choosing this action, the state "S2" that the agent will now be in after taking that action, and finally the next action "A2" which the agent will choose in its new state.

The SARSA algorithm goes as follows [Michael, 2015]:

1. Initialize $Q_o(s,a)$
2. Repeat (for each episode)
Initialise S_o
Choose a_o *from* S_o *using policy derived from Q*
Repeat (for each step of episode):
Take action a_t, *observe* r_{t+1}, s_{t+1}
Choose a_t *from* s_o *using policy derived from Q*
$Q_{t+1}(s_t,a_t) = (1-\eta)\ Q_t\ (s_t,a_t\ ) + \eta(\ r_{t+1} + \gamma Q_t(s_{t+1},a_{t+1}))$
set t = t + 1
until s_{t+1} *is terminal*

3.4. Dyna-Q Algorithm

Dyna-Q algorithmexploits a middle ground, yielding strategies that are both more effective than model-free learning and more computationally efficient than the certainty-equivalence approach.The Dyna-Qalgorithm goes as follows [11]:

Initialize $Q_o(s,a)$
Do forever:
(a)$s \leftarrow current\ (nonterminal) state$
(b) $a \leftarrow greedy(s,Q)$
(c) Execute action a: observe resultant state, s' and reward r
(d) $Q(s,a) \leftarrow Q(s,a)+\alpha[r+\gamma max_{a'}\ Q(s',a') - Q(s,a)]$
(e)Model(s,a) $\leftarrow s', r$ *(assuming deterministic environment)*
(f) Repeat N times:
s← random previously onserved state
a←random action previously taken in s
s', r ← Model(s,a)
$Q(s,a) \leftarrow Q(s,a)+ \alpha[r+ \gamma max_{a'}\ Q(s',a') - Q(s,a)]$

IV. Experiment Results

We set the some simulation scenarios in Matlabto compare the effectiveness of Q-learning,SARSA and Dyna-Q algorithm applying in robot path finding problem.

The simulation environment is a network of cells, each cell is considered as state or obstacle.

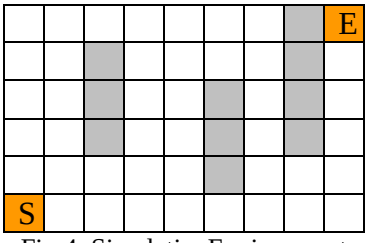

Fig 4: SimulationEnvironment

It consists of 9 x 6 = 54 cells in which are10 obstacles and the number of states of the environment is 44. The task of robot is to movewith a shortest path from cell [1,1] to cell [9,6], as shown in Figure 4.

We set the learning parameters as $\alpha = 0.1, \gamma = 0.95$, $\varepsilon = 0.1$ and applied three algorithms Q-learning,SARSA and Dyna-Q to evaluate the maximum and minimum number of movement of robot to reach to the target for certain episodes. Table 1,2,3 show the results of the simulation.

Episodes	100	200	300	400	500	600	700	800	900	1000
Maximum movement	794	134	94	79	53	46	34	38	52	34
Minimum movement	14	11	11	11	11	11	11	11	11	11

Table 1: Movement number in Q-leaning algorithm

Episodes	100	200	300	400	500	600	700	800	900	1000
Maximum movement	622	118	83	56	44	46	33	28	30	20
Minimum movement	17	11	10	10	10	10	10	10	10	10

Table 2: Movement number in SARSA algorithm

Episodes	100	200	300	400	500	600	700	800	900	1000
Maximum movement	531	21	18	17	17	18	19	18	18	18
Minimum movement	11	11	11	11	11	11	11	11	11	11

Table 3: Movement number in Dyna-Qalgorithm

From simulation results, we found that in the same environment, Q-Leaning and SARSA algorithm cannot find the optimum pathfor robotwithin 100 episodes and the robot performs quite high number of moves to reach the target. But the Dyna-Q algorithmcan find the optimal path for the robot in the first 100 episodes, and the robots made the number of movements to reach the target lower than the Q-learning andSARSA algorithm.The number of shortest path for each algorithm is presented in table 4. The Dyna-Q algorithm can find the number of shortest path higher than the Q-learning andSARSA algorithm.

Algorithm	Q-learning	Sarsa	Dyna-Q
Number of shortest path	465	453	623
Percentage	31%	30%	41,5%

Table 3:Number of Shortest Path

We found that a Dyna-Q algorithm gets to convergence within 300 first episodes and it is relatively stable shown as yellow line in Fig 5.

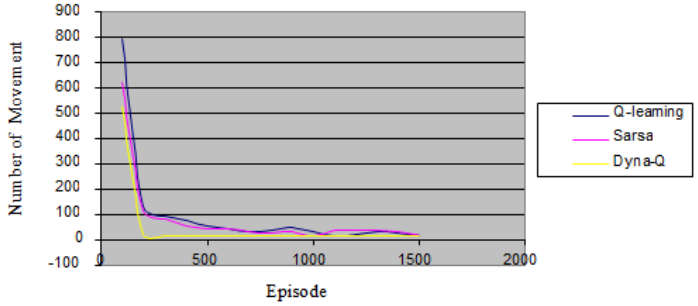

Fig 5: Convergence of Algorithm

We also change the learning rate α to evaluate its relation to reward value for each episode.

Episode	Reward value		
	α = 0.01	α = 0.05	α = 0.1
100	41650	22600	15962
200	24224	8688	4622
300	18149	5476	2661
400	14665	3776	1862
500	12386	2922	1497
600	10775	2378	1254
700	9442	1989	1196
800	8444	1701	1173
900	7661	1592	1172
1000	6887	1461	1181
1100	6300	1288	1138
1200	5736	1257	1155
1300	5439	1206	1170
1400	5019	1205	1143
1500	4737	1252	1172

Table 4: Relation between learning rate and reward value

Fig 6 represents relation between learning rate and reward valuegraphically.

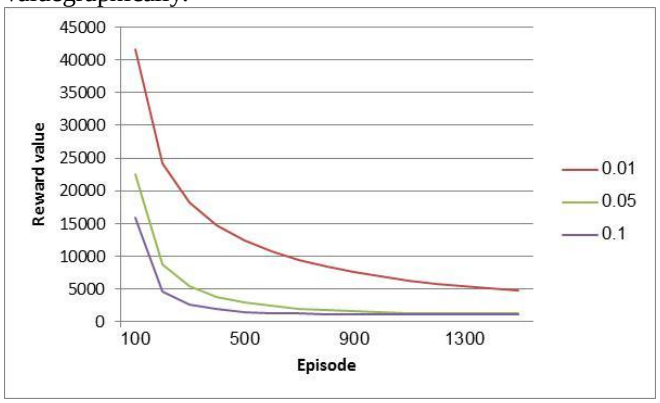

Fig 6: Relation between learning rate and reward value

With a learning rate of 0.1, the robot improves its reward value faster but it is unstable to an optimal value. While with a learning rate of 0.01, the robot approaches to convergence value slow but it isstable.

From the results of the experiment, we find that if the learning rate is large, the robot is more sensitive to the changes of the environment. Therefore we need to use high learning rate in the early episodes of interaction between the robot and the environment, and then reduce the learning rate to advance to the optimal strategystably.

V. CONCLUSION

In this paper, we describe the basic concept, principle and the method of reinforcement learning algorithms, and do some simulation experiments in order to evaluate the effectiveness among different reinforcement learning algorithms.We observe three popular reinforcement learning algorithm (Q-learning, SARSA, Dyna-Q) and apply in problem of path finding for autonomous robot.

Simulation results show that Q-learning algorithm does not work wellwhen the environment haschanged constantly and there are too many obstacles. Both SARSA algorithms and Q-learning algorithms are relatively slow in convergence but it has high accuracy in finding shortest path. Meanwhile, Dyna-Q algorithm determines the path of the robot relatively fast and the stability rates are high. However, the path that Dyna-Q algorithm finds is not exactlyoptimal.

In fact, there aren't algorithms which can be used for all environments. The selection of reinforcement algorithms for problem of path finding is depending on the needs of the user and the environment in which they are applied.

In the future works, we will do experiments for path finding with real autonomous robot to evaluate the effectiveness of different reinforcement algorithms in practical situations.

REFERENCES

[1] Qian Zhang, Ming Li, Xuesong Wang. Yong Zhang, "Learning in Robot Path Optimization", Journal of software, vol. 7, no. 3, March 2012
[2] Sutton, R. S. and Barto, A. G., "Reinforcement Learning: An introduction". *The MIT Press.*1998.
[3] O.Hachour, "Path planning of Autonomous Mobile robot",*International journal of systems applications, engineering & development*, issue 4, Volume 2, 2008.
[4] Pu Shi and Yujie Cui, "Dynamic path planning for mobile robot based on genetic algorithm in unknown environment," *Chinese Control and Decision Conference*, Xuzhou, 2010.
[5] T.H. Lee, H.K. Lam, F.H.F.Leung, P.K.S.Tam, "Apraetieal Fuzzy Logie Controller for the Path Tracking of Wheeled Mobile Robots", *IEEEControl System Magazine*,2003.
[6] V. M. Babu, U. V. Krishna and S. K. Shahensha, "An autonomous path finding robot using Q-learning," *10th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, 2016.
[7] Woo Young Kwon, Sanghoon Lee and Il Hong Suh, "A reinforcement learning approach involving a shortest path finding algorithm," *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.
[8] Yun, Soh Chin, S. Parasuraman, and VelappaGanapathy. "Dynamic path planning algorithm in mobile robot navigation." *Industrial Electronics and Applications (ISIEA*, 2011.
[9] John McCullock, "Q-learning tutorial",*Technical report* 2012.
[10] G. A. Rummery and M. Niranjan,"On-line q-learning using connectionist systems",*Technical report*, 1994.
[11] Richard S. Sutton. "Planning by incremental dynamic programming". In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 353-357. 1991.

Tran Xuan Sang, He received his BS, MS and PhD degrees in 2003, 2006 and 2013, respectively.He now works as lecturer at IT faculty–Vinh University.His research interests include expert system, intelligent computing, GIS modeling and simulation.
Tran QuocKiet, He received BS, MSdegrees in 2010, 2017, respectively. He now works as lecturer at center ofContinuing Education, Tan chau district –TayNinh– Viet nam. His research interests include expert system, intelligent computing.
NguyenThiUyen, She received her BS, MSdegrees in 2009, 2013, respectively.She now works as lecturer at IT faculty–Vinh University. Her research interests include expert system and intelligent computing.