2014 IEEE International Conference on Systems, Man, and Cybernetics
October 5-8, 2014, San Diego, CA, USA

A Reinforcement Learning Based Robotic Navigation System

Bashan Zuo[1], *Student Member, IEEE*, Jiaxin Chen[1], Larry Wang[2] and Ying Wang[1], *Member, IEEE*

1. Department of Electrical and Mechatronics Engineering, Southern Polytechnic State University, Marietta, GA, 30060, USA

2. Department of Mathematics, Southern Polytechnic State University, Marietta, GA, 30060, USA

Email: ywang8@spsu.edu

***Abstract*— It is a challenging task for an autonomous robot to navigate in an unknown environment. Machine learning could be useful to support the robot to adapt to the environment and learn the correct navigation skills quickly. In this paper, a reinforcement learning (Q-learning) based approach is proposed to help a robot to move out of an unknown maze. The definitions of the world states, actions and rewards of the algorithm are presented and some experiments are completed to validate the approach. The experimental results show that the proposed approach does have a good performance on mobile robot navigation.**

***Keywords—Q-learning; Mobile Robots; Navigation*.**

I. INTRODUCTION

Robotic navigation means to require a mobile robot to find a collision-free path from its current location to the goal location in an unknown and even dynamic environment. Robotic navigation could be applied in a variety of fields, such as Mars exploration, home service robots and robotic workers in manufacturing factories. Most of these environments are unknown or partly unknown for a robot. Hence, the robot has to collect information from the environment by using sensors and make navigation decisions based on the sensory data and its knowledge base.

Based on robots' awareness of the environment, robotic navigation techniques can be divided into two categories: navigating in a known environment and navigating in an unknown environment. The former means that the robot knows the global environment in advance, including the positions of all obstacles. Through some path planning technologies, a global path could be found from the current position to the goal position. Although path planning in such circumstances is relatively simple, the robot is not easy to obtain the global map which is needed in the whole planning process. The latter can be divided into the behavior-based approach and the learning-based approach. The behavior-based approach is also called local motion planning. It means that the robot can detect environment information and determine a path within the sensor coverage only. When the local environment changes, (i.e. the robot moves to a new position), the robot needs to re-plan the path. However, this approach is based on some fixed behavior rules, so the robot cannot deal with a complex environment with many states. The learning-based approach is proposed to address this deficiency. Especially, the robot can learn the navigation strategy by itself and improve its performance continuously to ensure the robot to find local optima.

In this paper, a Q-learning based approach is proposed to help a robot navigate in a maze autonomously. Due to its simplicity and good real time performance, Q-learning could be quite effective to make a robot learn the environment quickly and navigate inside it successfully.

II. RELATED WORK

A lot of research has been done in robot navigation using machine learning. Vargas, Petrilli-Barcelo, and Bayro-Corrochano proposed to employ extended Kalman Filtering (EKF) to implement Simultaneous Localization and Mapping (SLAM) in a robot navigation task [1]. In particular, the visual landmarks are detected using the Viola and Jones approach and are used to improve robot localization using EKF. Procopio, Mulligan and Grudic introduced how to employ the ensemble selection approach to select and combine models from an existing model library so that a robot could learn the navigation strategy in a dynamic environment

978-1-4799-3840-7/14/$31.00 ©2014 IEEE

[2]. Some important questions such as the composition of the model and how to combine selected models' outputs are addressed in this paper, and some experimental results are used to validate their approach. In order to make a robot learn the positions of obstacles and how to avoid them, Singh and Maulik proposed an artificial immunity based approach [3]. They found that the efficiency of the robot was dramatically increased if the artificial immunity approach was included to detect and avoid obstacles. Silver et al. suggested active learning from demonstration to implement robust and reliable robot navigation systems [4]. They proposed two approaches for active learning. It was demonstrated that generalization performance was improved and expert interaction was reduced. In order to meet challenges of robot navigation in a dynamic environment, da Costa et al. presented a joint approach combining Q-learning and knowledge-based systems [5]. In particular, a robot is first put in a simulated environment to learn the optimal navigation policy using Q-learning. Once the optimal navigation policy is learned, the policy is coded into a symbolic knowledge base that uses first-order logic as the knowledge representation formalism. The knowledge base will help the robot find a collision-free path in the navigation process. Farooq et al. proposed a neural-network based navigation approach for low-cost autonomous vehicles in an unknown environment [6]. Through collecting information from the sensors as the inputs, a neural-network is developed to make navigation decisions and send commands to the vehicle. Duane et al. presented a robot navigation approach using reinforcement learning in a dynamic environment [7]. In particular, the optimal navigation policy obtained by Q-learning is modified by information from a greedy heuristic. The performance of the proposed algorithm is compared to the well-known Greedy search algorithm. Besides neural-networks, fuzzy algorithms are also employed commonly in robot navigation. For example, Vaščák proposed an adaptive Fuzzy Cognitive Maps (FCM) for navigation and obstacle avoidance of robotic vehicles [8]. Variations of Hebbian learning as well as least mean square methods are employed and some experiments are designed to demonstrate the approach.

III. Problem Formulation

In this paper, a robot is placed in the center (initial position) of a spiral maze, as shown in Fig.1, and the robot is required to employ the sonar sensors to move out of this spiral maze without hitting any obstacles. In the beginning, the robot has no knowledge about the global map of the environment and also does not know how to select a good action to avoid obstacles.

Within the sensory range of the robot, the sonar sensors are employed to collect information of the unknown environment. Then the robot computer will send a motion command to tell the robot how to move.

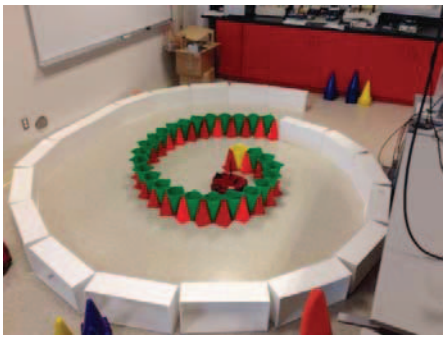


Fig.1. The spiral maze for robot navigation.

IV. Q-learning Based Robot Navigation

A Q-learning based approach is proposed in this paper to help the robot select actions and navigate out of the environment smoothly. As a kind of reinforcement learning algorithm, Q-learning has the following advantages: 1) Low requirements of computational resources, 2) Good real-time performance, 3) it does not need training samples. The control architecture of the robot system is presented in Fig. 2.

A. Control Architecture and Q-learning algorithm

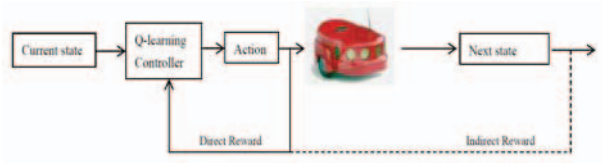


Fig.2. The control architecture of the robot system.

In Fig.2, the next action of the robot is determined by a Q-learning controller which is based on the Q-learning algorithm [9]. Once an action is executed in the current state, the robot will receive a direct reward that tells the robot about whether a right action has been selected under that state. In

addition, after the action is executed, the robot will observe a new state (the next state) and will receive an indirect reward based on the evaluation of the new state. The direct reward and indirect reward will be employed to update the Q-table (the knowledge base). The details of the Q-learning algorithm can be found in [9] and are presented again as follows:

- For each state $s_i \in (s_1, s_2, \cdots, s_n)$ and action $a_j \in (a_1, a_2, \cdots, a_m)$, initialize the table entry $Q(s_i, a_j)$ to zero. Initialize τ to 0.9. Initialize the discount factor $0 < \beta \leq 1$ and the learning rate $0 < \eta \leq 1$.
- Observe the current state *s*
- Do repeatedly the following:
  - Probabilistically select an action a_k with probability $P(a_k) = \frac{e^{Q(s,a_k)/\tau}}{\sum_{l=1}^{m} e^{Q(s,a_l)/\tau}}$, and execute it
  - Receive the immediate reward *r*
  - Observe the new state s'
  - Update the table entry for $Q(s,a_k)$ as follows:
  - $Q(s,a_k) = (1-\eta)Q(s,a_k) + \eta(r + \beta \max_{a} Q[s', a'])$
  - $s \leftarrow s'$, $\tau \leftarrow \tau * 0.9$

B. Definition of Environment States and Actions

In order to determine the world states in the Q-learning algorithm, two front-facing sonar sensors and four lateral sonar sensors are used to detect the distances between the robot and the closest obstacles in three specific directions. As shown in Figure 3, the three specific directions have been defined as three different color regions.

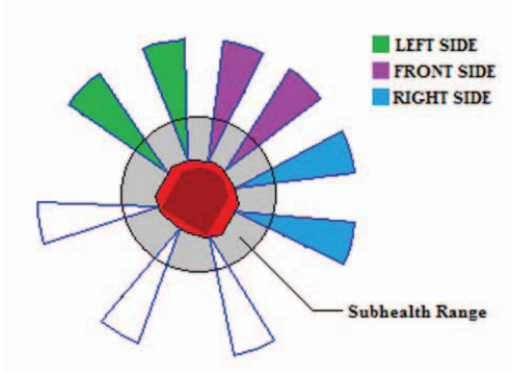


Fig.3. Schematic diagram of sonar direction.

Based on the sonar measurements, a 3-element vector $s = [left\ \ right\ \ front]^T$ is created to represent the current world state. In particular, the three elements of the vector represent the left, right and front distance and their relative amplitudes. Each element is an integer number ranging from "1" to "3". For example, as shown in Fig. 4, because three distances returned by the sonar sensors are 320*mm*, 750*mm,* and 600*mm,* a world state vector of [1 3 2] will be created, which indicates a small left distance, a big right distance and a middle front distance.

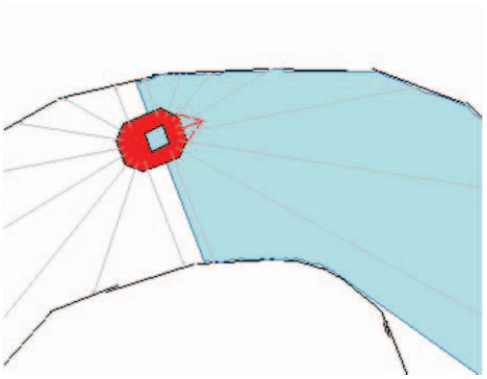


Fig.4. A sonar measurement example: left distance 300mm, right distance 750mm, and front distance 600mm.

Basically, once the robot detects the current world state, it then has to select one of three actions, as shown below.

- *Action F: Move forward for 100mm*
- *Action L: Turn left for 15°*
- *Action R: Turn right for 15°*

In addition, in order to prevent the robot from hitting an obstacle, a protection action is applied in the training process: if one of three distances is smaller than 170mm, the robot will move backward for 100*mm*.

C. Reward

For each action in a specific state, a reward value is defined. As shown in Fig. 4, robot is currently in the state of "132", so the best action in this state should be *Action R* (turning right for 15°). As a result, if the robot does turn right under state, it will receive a relatively big reward value.

In this research, in order to distinguish the world states more effectively, a new concept is introduced. Unlike the

traditional Q-learning algorithm, the world states here are divided into two categories: health states and sub-health states. A health state indicates the obstacles are still very far away from the robot, and they will not threaten its movement. Hence, after an action is executed, if a health state is observed by the robot, a positive reward will be received. The reward table for health states is presented in Fig. 5.

$$R = \begin{array}{c} state \setminus action \\ 123 \\ 321 \\ 132 \\ 312 \\ 213 \\ 231 \end{array} \begin{array}{c} \begin{array}{ccc} L & R & F \end{array} \\ \begin{bmatrix} 0 & 1 & 5 \\ 5 & 1 & 0 \\ 0 & 5 & 1 \\ 5 & 0 & 1 \\ 1 & 0 & 5 \\ 1 & 5 & 0 \end{bmatrix} \end{array}$$

Fig.5. The reward table in the health states.

On the other hand, a sub-health state indicates that at least one of obstacles is very close to the robot and the robot needs to move away from it immediately. Thus, a robot in a sub-health state will always receive a negative reward (or punishment). The reward table for the sub-health states is presented in Fig. 6.

$$R = \begin{array}{c} state \setminus action \\ 123 \\ 321 \\ 132 \\ 312 \\ 213 \\ 231 \end{array} \begin{array}{c} \begin{array}{ccc} L & R & F \end{array} \\ \begin{bmatrix} -3 & -1 & 0 \\ 0 & -1 & -3 \\ -3 & 0 & -1 \\ 0 & -3 & -1 \\ -1 & -3 & 0 \\ -1 & 0 & -3 \end{bmatrix} \end{array}$$

Fig.6. The reward table in the sub-health states.

While the robot is moving, any obstacle within 300*mm* will trigger a sub-health state. Specifically, the distance of 300mm could be adjusted according to the maze width or other parameters.

V. Experimental Results

In order to evaluate the proposed approach in Section IV, an experiment was completed, which is shown in Fig. 7. The plastic cones are used as the inner wall of the maze while the rectangular boxes are used as its outer wall. An AmigoBot™ manufactured by Adept Mobilerobots is placed in the center of the maze in the beginning.

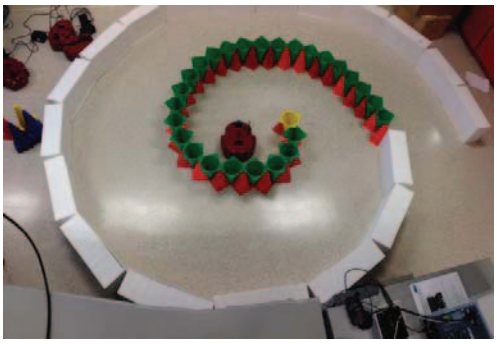


Fig.7. The spiral maze for field tests.

Fig. 8 shows the trajectory of the robot in an experimental test. As shown in Fig. 8, in the early stage, because the robot had a little knowledge about how to select correct actions, some wrong actions were executed, causing the robot to move back and forth for several times. From the second stage, the robot obviously learned some navigation skills, so the trajectory became much smoother.

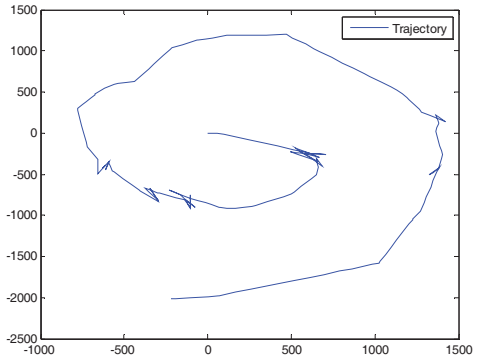


Fig.8. The trajectory of the robot in a physical experiment.

The action switching history is shown in Fig. 9. Especially, The discrete values of “0”, “1”, “2’ and “3” in the y-axis correspond to the actions of left turn, right turn, going forward and moving backward. Based on Fig. 9, after 150 steps, the action of moving backward has rarely appeared. It means that the robot learned the correct action selection strategy and seldom used the action of moving backward to avoid the wall.

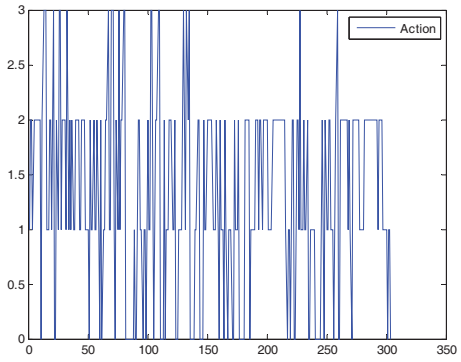


Fig.9. The action switching history in the real experiment.

Fig. 10 shows the history of all the Q values in the entire training process. Due to the space limitation of the maze, only 300 learning steps can be obtained.

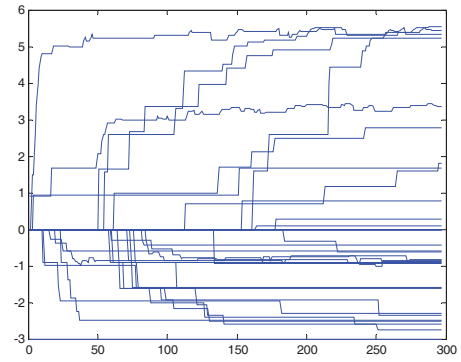
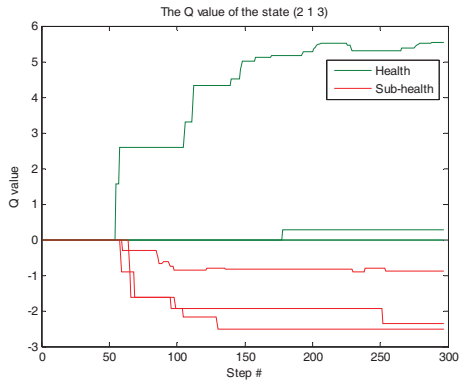
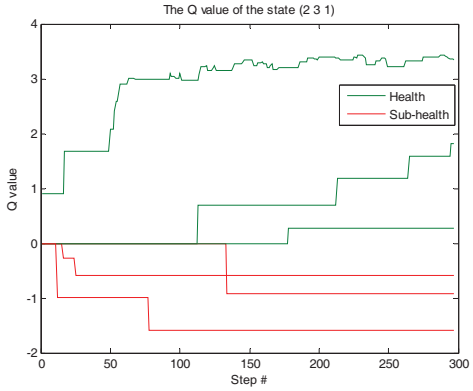


Fig.10. The history of all the Q values in the real experiment.

Finally, the history of the Q values in two specific states is presented in Fig. 11 below.



(a)



(b)

Fig.11. The history of the Q values under the specific states of [2 3 1] and [2 1 3].

From Fig. 10 and Fig. 11, it's very clear that the Q values converge after 300 steps. It means that the robot learned the correct action selection strategy in this unknown maze.

VI. Conclusions

In this paper, a navigation technology based on the Q-learning algorithm was proposed. The definitions of the states, actions and rewards of the algorithm were presented. Especially, a new concept called health and sub-health states were suggested to distinguish the world states more delicately. Based on the proposed approach, an autonomous mobile robot was required to navigate in an unknown maze and move out of it as soon as possible. The experimental results showed the proposed navigation technique was effective and successful to help a robot navigate in an unknown environment and avoid obstacles.

References

[1] H. Casarrubias-Vargas, A. Petrilli-Barcelo, and E. Bayro-Corrochano, "EKF-SLAM and machine learning techniques for visual robot navigation," *2010 International Conference on Pattern Recognition*, Istanbul, August, 2010.

[2] M. J. Procopio, J. Mulligan, and G. Grudic, "Learning in Dynamic Environments with Ensemble Selection for Autonomous Outdoor Robot Navigation," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008.

[3] C. T. Singh and U. Maulik, "A framework for an artificial immunity and speech based navigation for mobile robots," *IEEE World Congress on*

Computational Intelligence, Hong Kong, June 2008.

[4] D. Silver, J. A. Bagnell, and A. Stentz, "Active learning from demonstration for robust autonomous navigation," *IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, May 2012.

[5] A. L. da Costa, R.G. Cerqueira, T.T. Ribeiro, "Omnidirectional mobile robots navigation: a joint approach combining reinforcement learning and knowledge-based Systems," *2013 ISSNIP Biosignals and Biorobotics Conference (BRC)*, Rio de Janerio, February 2013.

[6] U. Farooq, M. Amar, E. ul Haq, M. U. Asad, H. M. Atiq, "Microcontroller based neural network controlled low cost autonomous vehicle," *2010 Second International Conference on Machine Learning and Computing*, Bangalore, February 2010.

[7] E. D. S. Costa and M. M. Gouvea Jr., "Autonomous navigation in dynamic environments with reinforcement learning and heuristic," *2010 Ninth International Conference on Machine Learning and Applications*, Washington, DC, December 2010.

[8] J. Vaščák, "Approaches in adaptation of fuzzy cognitive maps for navigation purposes," *8th IEEE International Symposium on Applied Machine Intelligence and Informatics*, Herlany, January 2010.

[9] Y. Wang, H. Lang and C.W. de Silva, "A hybrid visual servo controller for robust manipulation using mobile robots," *IEEE/ASME Transactions on Mechatronics*, Vol. 15, No. 5, pp. 757-769, 2010.