Q ROBOT : A LOW COST CRAWING ROBOT FOR EDUCATION AND RESEARCH IN REAL-WORLD REINFORCEMENT LEARNING

[1]*Ya-Tang Yang* [1*] *Feng-Yu Wang,* [1] Jhih-Ci Li[1] *and Jyun-Wei Shih*[1]

[1] Department of Electrical Engineering,
National Tsing Hua University, Hsinchu, Taiwan,
*E-mail: ytyang@ee.nthu.edu.tw

ABSTRACT

Current robot platforms available for research are too expensive for educational use in reinforcement learning. We develop a low cost Q-Robot with two degrees of freedom(DoF) with remote control(RC) servos and 3D printed parts. The robot can learn to craw from scratch within less than 20 min using Q learning algorithm in reinforcement learning. We validate the platform with reinforcement learning experiments and provide baseline results on a set of benchmark tasks. All the training and learning task is on-board. The optimal solution is a periodic orbit in the state space consisting of only 36 states with the discrete servo angle as the state variable.

Keywords: *reinforcement learning, locomotion control, robotic arm, robotics, Q learning.*

1. INTRODUCTION

The field of reinforcement learning (RL), derived from the root of “dynamical programming” has advanced significantly in recent years, with numerous success stories in solving challenging control problems[1-3]. This is largely due to the availability of simulators that allow for rapid testing of algorithmic performance, which are inexpensive, fast, and can be run in parallel. However, simulators often make unrealistic assumptions about the world. In another context, animals are known to learn various locomotion movement in their lifespan to adapt to their environments. For example, soaring birds often rely on ascending thermal plumes in the atmosphere as they search for prey or migrate across large distances. RL has been demonstrated in training a glider to fly and navigate in the field[4-5]. RL has also been demonstrated to optimize the parameters of central pattern generator model using policy gradient method in robotic fish swimming[6]. Some insects are known to be able to learn to walk within a short period of time when they are born[7]. The researchers in Aalto University, Finland, develop and validate RealAnt, a physical version of the popular *Ant benchmark* available in OpenAI Gym. RealAnt robot platform from Ote Robotics is designed for real-world reinforcement learning research and development[8].

The implementation of reinforcement learning has its root on the dynamic programming and suffers from the famous “ curve of dimensionality” according to Richard Bellman as the size of the state space is getting larger[1]. To circumvent this, approximation scheme such as a deep neural network is used to provide an approximation scheme for the Q table with much smaller number of parameters[2-3]. In this work, we propose that the lookup table representation of the reward function should not be overlooked whenever the problem is “easy” in the sense that the state space is of moderate size but an exact model is not available or too complex to be used[2]. The exact model here refers to the crawling dynamics approximated as the stochastic system transition probabilities(Markov chain). Specifically, we demonstrated a Q-learning algorithm[9] with a crawling robot with two servo motors with only 36 states and 4 actions with Arduino UNO and provide the detailed profiling of the learning process.

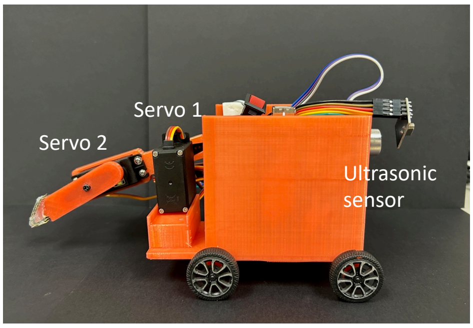

Fig. 1. Mechanical design of the Q robot.

2. Q ROBOT

Our Q robot is developed with similar motivation as Real Ant but it has one two degree of freedom but the power consumption is 3 Watt for the servo motor operation and the power for computation alone is only ~ 10 mW from the micro controller. Because aggressive exploratory actions taken by RL algorithms can easily damage the components of a robot[7]. Plastic gears in remote control(RC) servos or naively designed 3D printed parts can easily break during random exploration and learning. In Q robot, they can be replaced easily at low cost.

2.1. Mechanical Design

To be brief, Q robot weights ~ 340 g(Fig. 1 and Fig. 2). The arm consists of two high torque servos. The main body consists of a 3D printed housing of dimensional 12.5 cm(l) x 7cm(w) x 8.5 cm(h). The crawling robot uses one arm consisting of two servo motor (MG996R, 180 deg). The ultrasonic sensor(SR04) is mounted on the robot. The carpet of length ~ 1 m is preferred and is made of synthetic rubber. The tip of the arm is made of sand paper with specific roughness. The choice of the combination of the tip and the carpet is critical factor of successful training and learning. Also, a dummy weight of the order ~ 100 g is placed inside the main body of Q robot to shift the center of the mass in the correct position to provide stability of the robot during crawling.

There are several important time scales. The most important one is the delay τ_{roll} for the robot to roll over once the command for the servo is sent out.

2.2 Electronics components

The microcontroller of the robot is Arduino UNO, which has the maximal SRAM memory of 2kbyte. In particular, the most demanding part of the Q learning algorithm uses a look-up table of 144 floating numbers. Two lithium ion batteries with capacity of 1200 mAh and 7.4V is sufficient to power up the Q robot. The voltage is regulated to 5 V by a voltage regular LM7805. A ultrasonic sensor (SR05) is mounted on the main body to provide the distance measurement from a reference wall. The maximal distance is ~ 3 meter and the resolution in distance is 1 cm. The measured distance from the ultrasonic sensor between the wall and the robot is used as cumulative reward. When Q robot is up and running, the data is transmitted via a wireless communication module nrf 24. An external laptop computer equipped with another micronctoller board (Arduino UNO) with another nrf 24 is used to receive the data with a python code to convert the sent data into CSV file format. The usage of the external computer is limited to data communicaton and the external computer is not involved in the learning process such as computing the updated value of Q or finding the optimal action.

2.3 Description of the implementation of Q learning algorithm

Briefly, Q-learning is a reinforcement learning algorithm that tries to find optimal actions by learning a state-action value function Q[s,a]. The underlying idea is to use system transition probabilities or the Markov chain to model the dynamics of the crawling robot. The state-action value function, or simply, is a look-up table having rows as states, actions as columns, and values as entries. Thus if the value function is known, then the optimal policy is simply to select the action having the highest value for the current state.

$$a' = \text{argmax}_{a'} \, Q(s, a) \qquad (1)$$

Four possible actions, denoted by a_i (i=0,1,2,3) are defined as

Action 0 (a0) : increment of angle of servo 1 by 4 degree.
Action 1 (a1) : increment of angle of servo 1 by -4 degree.
Action 2 (a2) : increment of angle of servo 2 by 14 degree.
Action 3 (a3) : increment of angle of servo 2 by -14 degree.

The optimal solution of the Markov decision problem is a sequence of actions that move the agent forward at an efficient rate. The main loop consists of getting the distance as reward from ultrasonic sensor, getting optimal action from Q by searching for the maximal value, and updating the setpoint of the servos. The update rule for Q[s][a] is on policy version of Q learning rule[5].

$$Q(s,a) \leftarrow Q(s,a)+\eta(r+\gamma \max_a Q(s',a')-Q(s,a)) \qquad (2)$$

η is the learning rate (η =0.1) and r is the immediate reward. γ is the discount factor (γ =0.75).

To explore, *ε-greedy* search where with probability ε, we choose one action uniformly randomly among all possible actions, namely, explore, and with probability $1-\varepsilon$, we choose the best action, namely, exploit. In the code, we let probability ε to decrease exponentially over time. Typically, we use $\varepsilon = \exp(-t/\tau)$ with τ = 1 minute. The typical learning time of ~10 minutes corresponds to 3000 updates with η =0.1.

For Q robot of the typical size described here, τ_{roll} = 0.2 sec. In general, this roll over time depends on the mass and size of the robot if we decide to use robot of different scale. There is another time scale for the servo to reach its set angle. The time constant is 4.5 ms * delta theta. Delta theta = (maximal angle – minimal angle)/ N. To give a specific number, 18 ms is need for 4 deg. In our work, we use N=6 and θ_{1max} = 102 deg, θ_{1min}=82 deg, θ_{2max}=160 deg, θ_{2min}=90 deg. These values correspond to the boundary of the gridworld state space. The action that leads the servo angle out of this range is nullified.

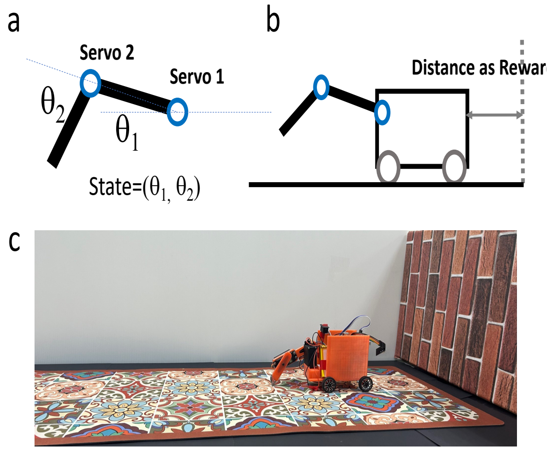

Fig. 2. Schematics of learning experiment. a. Definition of state variable b. Distance as measured from ultrasonic sensor is used as the cumulative reward in Q learning algorithm c. Snapshot of Q robot learning to walk on a carpet

4. RESULTS AND DISCUSSION

The Q robot is able to learn to crawl in typically less than 20 minutes. In a typical run of 10 trials, we are able to record the details of the learning process. A sample of "optimal solution" is shown in Fig. 3. Fig 3b and Fig. 3c are an expanded view of the transition from learning to the learned optimal solution of crawling. The distance as the cumulative reward has different slope in the "learned" state from the "learning-in-progress" state. The movement speed in the learned state is ~ 0.5 cm/sec estimated by fitting the distance versus time data by a linear curve. The learned optimal solution in Fig. 3c shows a periodic pattern occasionally disrupted by noise. Note that because the number of action is only four, we are able to directly visualize such a pattern. Actions are represented as integers from 0 to 3. When the corresponding trajectory plotted in state space, the optimal solution is a sequence of six actions, i.e., a3 -> a1->a1->a2 -> a0-> a0. Such a sequence will result in a closed orbit in the state space schematically shown in Fig. 4. Previous simulation work on a two-arm link crawling robot based on genetic algorithm has been done and reveals the optimal solution is periodic[10,11].

Note that the state space is 2D rectangular gridworld. At each cell, all the actions correspond to movement in up, down, left, and right. This is reminiscent of the gridworld examples in the classic textbook of RL by Sutton and Barto[3]. We do observe some interesting cases that when a large noise spike from the ultrasonic sensor disrupt the learning, the robot will return back to "learning-in-progress". Amazingly, the robot is able to recover and find the optimal solution again after a period of time.

Most research on robotics is conducted on industrial robots that are very expensive, costing thousands of dollars. This is not very affordable to all researchers, let alone educational use[8]. Our development is much along the line of the RealAnt robot[8]. We would like to make a comparison with the RealAnt. The fully assembled one of the RealAnt costs around ~$410 in materials and power supply needed is 12 V 5A. (The estimated the computation power is around ~10 Watt.) In contrast, our Q robot costs around USD$100. When all the components is available, our robot can be assembled in less than 1 hr with no calibration on ultrasonic sensor. Also, we do not rely on camera-based image process method for pose estimation and therefore the power consumption is primarily due to the servos that produce the movement.

We have previously used a similarly constructed version of Q robot in educational settings. In particular, we have solicited five volunteers and provide them with a simple instruction. All of them can do hands-on experimentation for a representative experiments, e.g., to test the capability of Q robot's walking on different carpets.

We can also comment on the computation loading of an enhanced look-up table representation of Q learning with four servo motors (mechanical degrees of freedom) and such a code only uses up to ~20 kB of flash memory with 6^4 state and 8 actions to store Q value look-up table in Tweensy 4.0 microcontroller and can be potentially implemented in the Q robot with four servo motors (two arms).

4. CONCLUSION

Reinforcement learning of a crawling robot with two servo motors is successfully demonstrated. Explicit and detailed profiling of the learning process can be obtained. While our initial experiment focuses on proof-of-concept experimentation in educational settings and the prototype robot is relatively simple. We are investigating whether or not such a straightforward look up table representation is applicable if we increase of number of mechanical degrees of freedom. We expect such implementation will open up new possibilities in robots of few DoFs such as multi-legged microbots or inworm crawling robots with limited computational resource[12-17].

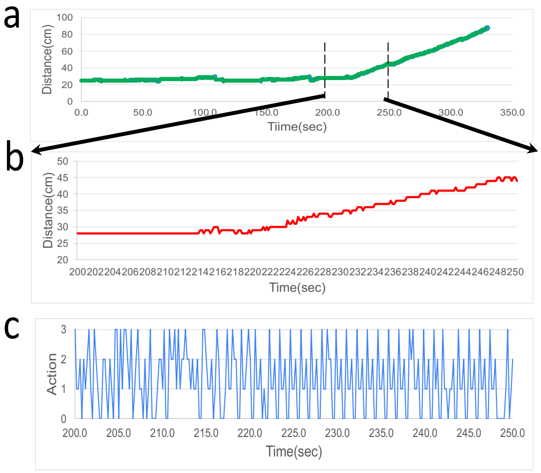

Fig. 3. The learning curve. (a) distance versus time (b) distance versus time during a transition from "learning in progress" to the "learned" optimal solution (c) Action versus time.

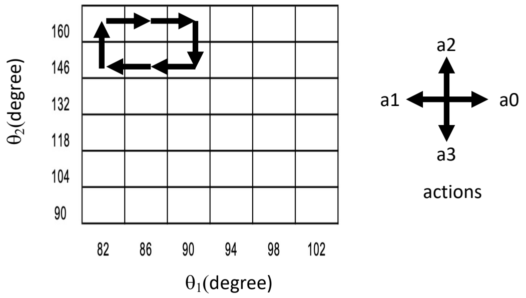

Fig. 4. The trajectory in the state space corresponding to the optimal solution

ACKNOWLEDGEMENT

The authors would like to acknowledge the microcontroller code used in this work is modified from Dr. Andrew August's code in his website. The authors would like to acknowledge funding support from National Science and Technology Council under the grant number 113-2218-E-007-019, Ministry of Education under the grant number PEE1120593, National Tsing Hua University under grant numbers 113Q2703E1.

REFERENCES

[1] R. E. Bellman, *Dynamic Programming*. Princeton University Press, Princeton, 1957.

[2] D. P. Bertsekas, and J. N. Tsitsiklis *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA , 1996.

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

[4] G. Reddy, A. Celani, T. Sejnowski, T. and M. Vergassola, "Learning to soar in turbulent environments." *Proc. Natl Acad. Sci. USA* Vol. 113, pp. E4877–E4884, 2016.

[5] G. Reddy, J. Wong-Ng, A. Celani, A. *et al.* "Glider soaring via reinforcement learning in the field." *Nature* **562**, 236–239 2018.

[6] H. Deng, D. Li, C. Nitroy, A. Wertz, S. Priya, B. Cheng, " Robot motor learning shows emergence of frequency-modulated, robust swimming with an invariant Strouhal number." *J R Soc Interface*. Vol. 21 No. 212:20240036, 2024.

[7] R. Siegwart, I. Reza Nourbakhsh and D. Scaramuzza, Introduction to autonomous mobile robot, MIT Press, Boston, 2011.

[8] R. Boney, J. Sainio, M. Kaivola, A. Solin, J. Kannala, "RealAnt: An Open-Source Low-Cost Quadruped for Education and Research in Real-World Reinforcement Learning," arXiv:2011.03085, 2020.

[9] C. J. C. H. Watkins and P. Dayan. "Q-learning." *Machine Learning* Vol. 8, pp.279–292, 1992.

[10] H. Kimura and S. Kobayashi. Reinforcement learning for locomotion of a two-linked robot arm. In *Proceedings of the 6th European Workshop on Learning Robots*, pages 144–153, 1997.

[11] Y. Kassahun and G. Sommer, "Evolutionary Reinforcement Learning for Simulated Locomotion of a Robot with a Two-link Arm." Intelligent Autonomous Systems 9 - IAS-9, Proceedings of the 9th International Conference on Intelligent Autonomous Systems, University of Tokyo, 2006.

[12] Z. Liu, W. Zhan, X. Liu, X. *et al.* A wireless controlled robotic insect with ultrafast untethered running speeds. *Nat Commun.* Vol. 15, 3815, 2024.

[13]R. Brühwiler, R. et al. "Feedback control of a legged microrobot with on-board sensing." in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 5727–5733, 2015

[14]B. Goldberg, N. Doshi, and R. J. Wood, "High speed trajectory control using an experimental maneuverability model for an insect-scale legged robot." in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 3538–3545, 2017.

[15] J.-S. Koh and K.-J. Cho, ''Omega-shaped inchworm-inspired crawl- ing robot with Large-Index-and-Pitch (LIP) SMA spring actuators,'' *IEEE/ASME Trans. Mechatronics*, Vol. 18, No. 2, pp. 419–429, 013.

[16] W. Wang, K. Wang, and H. Zhang, ''Crawling gait realization of the mini-modular climbing caterpillar robot,'' *Prog. Natural Sci.*, Vol. 19, No. 12, pp. 1821–1829, 2009.

[17] K. Kotay and D. Rus, ''The Inchworm Robot: A Multi-Functional System,'' *Auto. Robots*, Vol. 8, No. 1, pp.53–69, 2000.