Adaptive Navigation of a High Speed Autonomous Underwater Vehicle using Low Cost Sensors for Low-Altitude Survey

Yukiyasu Noguchi, Yoshinori Kuranaga, Toshihiro Maki
Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba, Megro-ku, 153-8505, Japan
(Tel : +81-3-5452-6489; E-mail: yukiyasu@iis.u-tokyo.ac.jp)

Abstract - In order to get highly detailed information of seafloor with low-cost, the authors have been developing high speed autonomous underwater vehicle (AUV) using low cost scanning sonar. It is difficult to design the AUV navigation system functioning properly in various seafloor environment because the sonar echo levels from the seafloor are relatively weak and differ with the survey area. In this study, the authors propose the AUV navigation method adaptive to the various seafloor environment using reinforcement learning in which AUV learns how to move by itself online and test the algorism in a simulation

Keywords – Reinforcement Learning, Autonomous Underwater Vehicles, Low-Altitude Survey, High Speed Survey, Scanning Sonar.

1. Introduction

Low-altitude survey for gathering highly detailed and reliable information of seafloor is important in various fields, such as resource survey and biological research.

These days, Autonomous Underwater Vehicles (AUVs) hold a very important role for this survey. T. Maki et al [1] measured the three-dimensional distribution of tubeworm colonies using an AUV through low-altitude (about 2m) surveys. S. McPhail et al [2] developed an algorism for flight-class AUV to conduct low-altitude terrain following and collision avoidance in order to widen the survey range.

The authors have been developing a low cost and high speed AUV for optical survey of seafloor. "Hattori" is the prototype AUV the authors made for the purpose. Hattori's surge speed is max. ~ 2.0 m/s. Hattori has only two sensors to observe surrounding environment. One is a scanning sonar (tritech Micron). The scanning sonar can emit only one direction sonic beam and it cannot emit multiple beams at the same time. The max. range is 100 m. The other is a camera (with a laser) looking downward.

It is a challenging mission for Hattori to take seafloor photos at low altitude because of its high speed and limited sensors. If Hattori fails to detect the obstacles, Hattori can crash the obstacles. Hattori can often cause this problem because of the uncertainty of the scanning sonar response. The echo level from the obstacles depends on the survey seafloor environment and the noise level from the AUV's thruster are high. When the AUV is at low-altitude, the incidence angle becomes large, making it more difficult to obtain strong echo from terrain. It is difficult to design a control program which can detect any obstacles in any seafloor environment.

Y. Kuranaga [3] developed an algorism using potential method for the AUV navigation. The algorism uses all scanning sonar data and allows the AUV to take seafloor photos in low-altitude. However, the algorism needs setting some parameters which depend on the survey seafloor echo levels. If you fail to preset appropriate parameters, the AUV travels at too high or too low altitude.

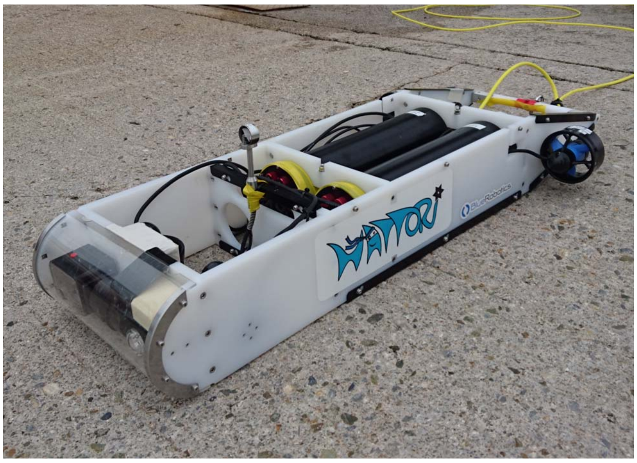


Fig. 1. AUV Hattori

Table 1 Specifications of Hattori

Size	1.02 m (L) x 0.2 m (H) x 0.48 m (W)
Mass	16 kg
Speed	~ 2.0 m/s

In this study, the authors aim to develop a navigation algorism of an high speed AUV using low-cost sensors for low-altitude survey, which is adaptive to the various seafloor environment. The algorism uses reinforcement learning and the AUV learns how to move online considering the real sensors response. The authors expect that the algorism can result in high robustness to the uncertainty of the seafloor environment.

In this paper, the authors made Hattori have the adaptive navigation algorism and tested the control system in the simulator.

2. Methods

2.1 AUV Navigation by reinforcement learning

Reinforcement Learning is a kind of machine learning. Reinforcement learning is a learning algorism in which an agent (in our study case, an AUV) tries to maximize the reward in the interaction with the environment. One of the noteworthy features of reinforcement learning is that the major tasks humans have to do is the definition of the reward. Humans do not have to know the environment in detail. There are many reinforcement learning methods, such as Q-learning [4] or Deep Q-Network (DQN) [5] and many studies to apply these methods to real world robots.

In this study, the authors used sarsa(λ) [6] agent for AUV Navigation. The definitions of the variables and functions located in the Sarsa(λ) in Fig. 2 are as follows:

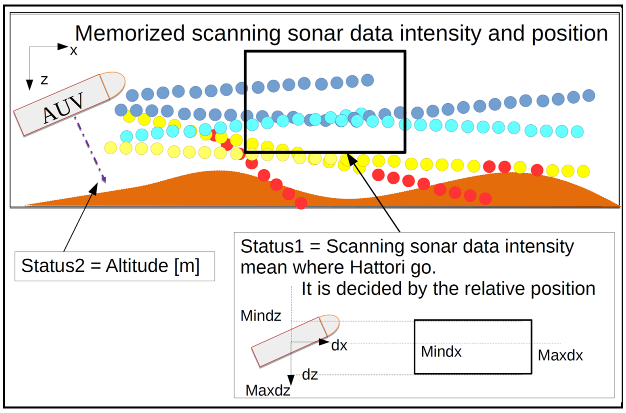
- s, state
- a, action
- δ,TD error
- r, reward
- γ, discount rate
- α, step-size parameter
- Q(s,a), action-value function
- e(s,a), eligibility trace
- λ, trace-decay parameter

```
Initialize Q(s', a')
For each episode:
    Initialize e(s, a), s, a
    For cach timc stcp:
        Perform a, and observe r, s'
        Choice a'
        $\delta \leftarrow r + \gamma Q(s', a') - Q(s, a)$
        $e(s, a) \leftarrow e(s, a) + 1$
        For all s, a:
            $Q(s, a) \leftarrow Q(s, a) + \alpha\delta e(s, a)$
            $e(s, a) \leftarrow \gamma\lambda e(s, a)$
        $s \leftarrow s'$ ; $a \leftarrow a'$
```

Fig. 2. Sarsa(λ) algorism

A. Status

Sarsa(λ) agent in the AUV navigation uses two status value. One is the mean intensity of the echo of the scanning sonar in a certain area in front of the AUV and the other is the altitude. Figure 3. shows the definition of the status.



Memorized scanning sonar data intensity and position
x
z
AUV
Status2 = Altitude [m]
Status1 = Scanning sonar data intensity mean where Hattori go.
It is decided by the relative position
Mindz
dx
dz
Maxdz
Mindx
Maxdx

Fig. 3. Definition of the status

B. Action

In Sarsa(λ) algorism, action choices have to be discrete. In this study, the action choices are defined as

a = {a_0 : Pitch reference = UP_pitch [deg], a_1 : Pitch reference = 0 [deg], a_2 : Pitch reference = Down_pitch [deg] }. The agent choices one action from the 3 actions.

C. Reward

Reward is designed in order to maintain the constant altitude from seafloor without collisions with any obstacle. If the AUV crashed or the altitude becomes higher than the limit altitude, the agent receives minus reward. If the AUV kept the reference altitude to take seafloor photos, the agent receives plus reward.

```
If Hattori crashes obstacles:
    r ← (Crash penalty)
If altitude > (Max limit altitude to take photo):
    r ← (No photo penalty)
Else:
    r ← (Photo value)
```

Fig. 4. Definition of the reward.

2.2 Simulation

The authors tested the algorithm in computer simulation with the model of Hattori. Gazebo simulator [7] was used for compatibility with the existing software of Hattori, those are written using Robot Operation System (ROS). Figure 5 shows the simulated seafloor and the measurement of the scanning sonar at a certain moment.

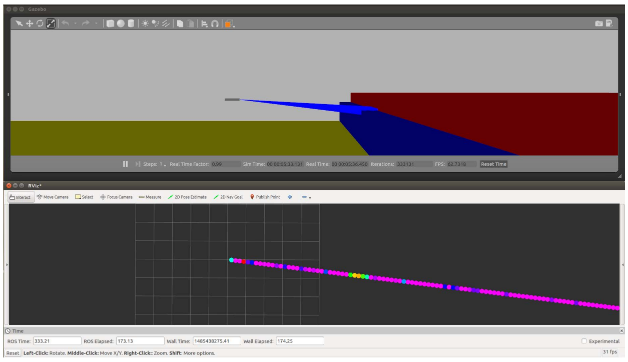


Fig. 5. Simulated seafloor and measurements of the scanning sonar

In this simulation, Hattori moved with constant yaw, roll and surge speed reference. Yaw and roll reference was 0 deg. Surge reference was 2.0 m/s. Some rectangular objects were placed randomly on the seafloor. The objects' height was 1 m or 2 m. The total height of the piled objects were often 5 m. The course length is about 1 km. When Hattori finished the course (the authors call it "Finishing the episode"), Hattori was instantly moved to the initial position and all the rectangular objects are rearranged (the authors call it "Starting new episode").

Pitch reference was decided by the agent all time except the following two cases. If the altitude was over 8 m, pitch reference was -5 deg. If Hattori clashed obstacle, Hattori went back and pitch reference was set to -70 deg for 15 seconds and then went up with the 70 degree pitch reference for 10 seconds.

In the simulation, sarsa(λ) agent parameters were as follows:

- $\alpha = 0.4$
- $\gamma = 0.8$
- $\lambda = 0.8$

The agent chose the action which action-value was the highest in that time. However, the agent chose the action randomly with a 0.1% possibility.

Parameters for status and action were as follows:

- Mindx = 1.1 [m]
- Maxdx = 4.1 [m]
- Mindz = -3.0 [m]
- Maxdz = 2.0 [m]
- Up_pitch = 30 [deg]
- Down_pitch = -10 [deg]

Parameters for the reward were as follows:

- Max altitude to take a seafloor photo = 4.0 [m]
- Crash penalty = -100
- No photo penalty = - 0.02

Photo value was decided by Eq. (1)

$$(\text{Photo value}) = 0.1 * (2.0 - \text{absolute}\ (2.0 - \text{altitude})) \quad (1)$$

When Hattori's altitude was 2.0 m, the agent got highest reward 0.2.

The authors conducted the simulation in two conditions. Case 1: the measurements of the scanning sonar is used. Case 2: the measurements of the scanning sonar is not used. The agent in Case 2 learned how to move only from the altitude.

3. Result

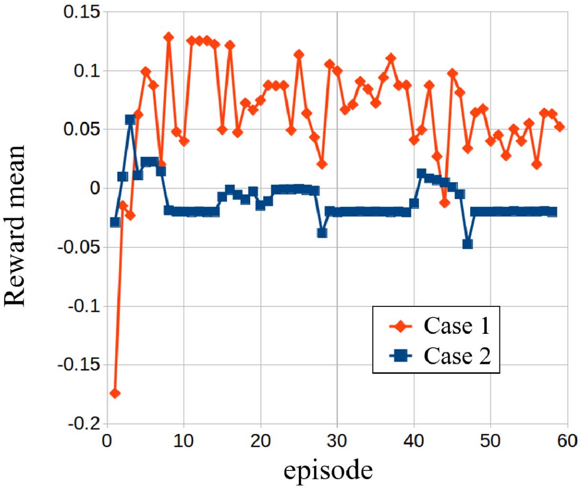


Fig. 6. Mean value of the reward

Figure 6 shows the mean value of the reward in each episode. In case 1, the reward was low at the initial episodes, but it quickly increased and was kept plus during almost all the episodes. In case 2, the reward was relatively high at the initial episodes, but after that it became minus in average.

4. Discussion

The agent in Case 1 got minus reward at the initial episodes. However, the agent got plus reward in average after the first 3 episodes. On the other hand, the agent in Case 2 got higher reward at first than in Case 1, however, the mean reward is minus. This shows the agent in Case 1 learned how to move in the unknown environment by using the scanning sonar data. As the agent in Case 2 cannot detect forward obstacles, it made its altitude high in order to reduce the collision risk.

Hattori in Case 1 would get lower reward at initial episodes than that in Case 2 simply because the dimension size of the action value space. It usually takes longer time to get 2-dimension action value space. It would be possible to say that the agent can get higher reward by increasing the status dimension, however it takes longer time to learn.

The agent in Case 1 succeeded in getting plus reward in average. However, some problems might happen in the real world. The first problem is that the AUV inevitably crashes to the seafloor, especially during the initial stage. The AUV should avoid all obstacles not to damage itself. Second, it takes long time to learn. In Case 1, Hattori had to travel about 3 km before the mean of the reward becomes plus.

One of the approaches for the problems are trainings in the simulator before AUV dives in the real world. Other approach would be to increase the status dimension and the number of the action choice. More information and the outputs may lead the increase of the performance. It is important to make the learning time shorter in the future study. One option to reduce the learning time is linear approximation of the action value function.

5. Conclusion

In order to realize low-altitude high-speed survey using an AUV, the navigation method using a scanning sonar and reinforcement learning was proposed. In the simulation, the Sarsa(λ) agent in AUV Hattori learned how to move by using the scanning sonar in unknown environment.

References

[1] T. Maki, A. Kume and T. Ura, "Volumetoric mapping of tubeworm colonies in Kagoshima Bay through autonomous robotics surveys," Deep-Sea Research I, 58, pp. 757-767, 2011.

[2] S. McPhail, M. Furlong and M. Pebody, "Low-altitude terrain following and collision avoidance in a flight-class autonomous underwater vehicle," Proceedings of the Institution of Mechanical Engineers, Vol. 224, Part M: Journal of Engineering for the Maritime Environment, pp. 279-292, 2010.

[3] Y. Kuranaga, Scanning sonar based algorithm for high speed and low altitude terrain following with cruising type AUV, Master Thesis, The University of Tokyo, 2017. (in Japanese)

[4] S. Mikami, M. Minagawa, "Kyouka Gakushu," Morikawa, 2000.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J, Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg and D. Hassabis, "Human-level control through deep reinforcement learning," Nature 518 (7549), pp. 529-533, 2015.

[6] Rummery, G. A., and Niranjan, M. "On-line Q-learning using connectionist systems," Technical Report CUED/F-INFENG/TR 166, 1994.

[7] Gazebo simulator, http://gazebosim.org/