ICROS-SICE International Joint Conference 2009
August 18-21, 2009, Fukuoka International Congress Center, Japan

Intelligent Navigation and Control of an Autonomous Underwater Vehicle based on Q-Learning and Self-organizing Control

Namhoon Kim[1], Gyeong-Hwan Yoon[2], and Doheon Lee[3]
[1]Robotics Program, KAIST
(Tel: +82-42-350-4353; nhkim@biosoft.kaist.ac.kr)
[2]Robotics Program, KAIST, DAEYANG Electric CO.,LTD
(Tel: +82-42-350-4356; smartAUV@gmail.com)
[3]Robotics Program, KAIST, Dept. of Bio and Brain Engineering, KAIST
(Tel: +82-42-350-4316; dhleebit@gmail.com)

Abstract : An autonomous underwater vehicle(AUV) is developed to explore and patrol in underwater environments. To accomplish these objectives, an autonomous navigation and control system is essential to an AUV. An intelligent navigation system produces safe paths from the start point to the target point by itself, and the control system makes the vehicle follow the planned path. In this paper, we propose an autonomous navigation and control system for AUVs based on reinforcement learning scheme.

Keyword: reinforcement learning, autonomous underwater vehicle, intelligent system

1. Introduction

An intelligent system is defined as a system that perceives its environment and take actions which maximize it chance of success.[1] For an autonomous system, an intelligence is an essential element. Perception stands for the capability of acquiring and using knowledge about the environment and itself. And an action should be taken without involving human beings. Intelligence means that the robot is able to make decisions with learning and inference capabilities. The objective of the intelligent system for an autonomous underwater vehicle(AUV) is minimizing human interventions while executing its desired missions. Especially, in underwater system, it is required that self-organizing and self-learning scheme. Underwater environment is one of the most hostile environments in the earth. In controlling of AUV, there are many difficulties such as time-delay effects, external disturbances, drag forces and poorly modeled plant. And in reinforcement learning scheme, the perception can be accomplished by means of observing state and getting rewards. Also, the robot can take actions by its own policy which is generated from repetitive tasks. In section II, we will show the overview of the autonomous underwater vehicle system. In section III, we present the navigation and control system based on Q-Learning algorithm and self-organizing control scheme. In section IV, we simulate our proposed system, and in section V, we will conclude this paper.

2. The intelligent system overwiew

In our design, AUV system consists of two systems: an autonomous navigation system and a vehicle control system. To simplify our problem, we define the objective of the system is travel from the start point to the target point without hitting any obstacles. The start point should be selected arbitrary and the target point is decided before beginning the mission. In autonomous navigation system, we used Q-learning scheme to generate a global path to the target point. Figure 1 shows the block diagram of our designed system. The system has two phases. In first phase, an autonomous navigation system starts learning the optimal policy to find the path to the target point. In this phase, the system uses Q-Learning algorithm to make the policy. After the first phase, the fuzzy controller starts control the vehicle. The vehicle gives its location to the autonomous navigation system, and then the navigation system provides the references to the fuzzy controller. This procedure is repeated until the vehicle arrives the target point.

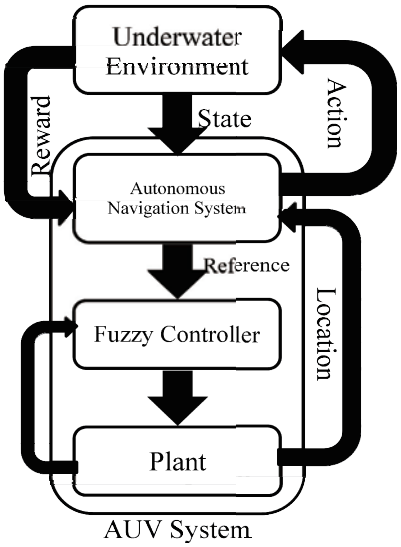


Figure 1. Block diagram

3. The navigation and control of the vehicle

3.1. Autonomous Navigation system

Reinforcement learning is learning what to do so as to maximize a numerical reward signal. The learning agent is not told which actions to take, but discover which actions yield the most reward by trying them. In the scheme, the agent interacts with environment. The agent observes its state from environment, and takes an action derived from its own policy. After that, the agent's state is changed and the agent gets the reward. The agent always takes actions to maximize its reward at the terminal state. Beyond the agent and the environment, there are four 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 behavior. A reward function defines the goal in a reinforcement learning problem. The reward function tells what the good and bad events are for the agent. A value function specifies what is good in the long run. Whereas rewards are immediate, values indicate the long-term desirability. A model of the environment mimics the behavior of the environment. [2]

First, we need to define the elements of the reinforcement learning system. Before we define the elements, we make an assumption that the environment is stationary. We define the environment is three dimensional Cartesian coordinate system. The learning agent is our autonomous underwater vehicle(AUV). The actions are defined as six motions in three dimensional spaces. Figure 2 shows the defined actions. States are the *(x, y, z)* locations in environment.

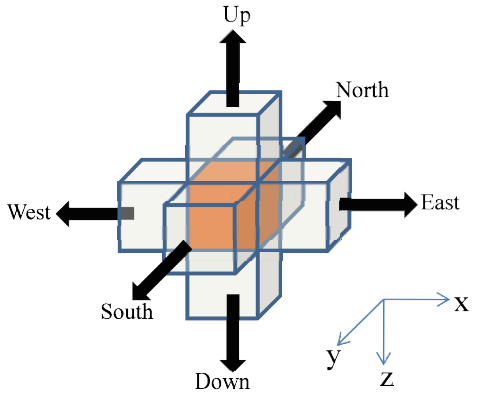


Figure 2. Defined actions

Our tasks are defined as episodic tasks, and the episode is traveling from the start point to the target point without hitting any obstacles. The reward function is following.

$$r = \begin{cases} 10 & \text{if the robot arrives the target point} \\ -3 & \text{if the robot hits obstacles} \\ -1 & \text{otherwise} \end{cases}$$

where r represents the reward in the system.

For the elementary solution method, we use temporal difference learning. Because our objective is improving the value function, we do not need to define a model of the environment, which is called direct reinforcement learning.

Q-Learning algorithm that we used to training our vehicle is following.

1. Initialize all *Q(s,a)*
2. Initialize *s* with arbitrary position
3. Choose *a* using policy* derived from *Q*
4. Take action *a*, observe *r* and *s'*
5. Update *Q(s,a)*
6. $s \leftarrow s'$
7. repeat 3-6 until *s* is terminal state

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

In the above algorithm, *Q(s,a)* represents the learned action-value function, *s* represents state observed from the environment, *a* represents an action which is defined in our design procedure.

The policy derived from *Q* uses exploration and exploitation. The agent prefers actions that has tried in the past and found to be effective in producing reward. But to discover such actions the agent needs to try actions which has not selected before. This procedure is exploration to find better actions for the future. And the exploitation is taking actions which are already known to maximize the reward.

3.2. Self-organizing control

In our system, the most important objective is minimizing human interventions. The conventional controller which is based on the mathematical model of the plant requires the dynamics of the model and the fine tunings of the control parameters. In the underwater environment, the vehicle is highly nonlinear system and the disturbances such as ocean currents and the convection exist. In addition, due to the absence of the tether cable the autonomous underwater vehicle(AUV) should be adaptive for the environment during the mission. To satisfy these requirements, the ability of modifying controller according to changing circumstances.

The self-organizing controller is a table-based controller which has the performance measure unit and the modifying unit. In our design, the controller has two loops: the inner loop and the outer loop. The inner loop is a fuzzy incremental controller and the outer loop is the modifying mechanism. Figure 3 shows the block diagram of the designed control system. The performance measure unit observes the current performance of the controller. If the performance is undesirable, the control table *F* should be modified. The performance is calculated from the error *e* and the change of the error *ce*.

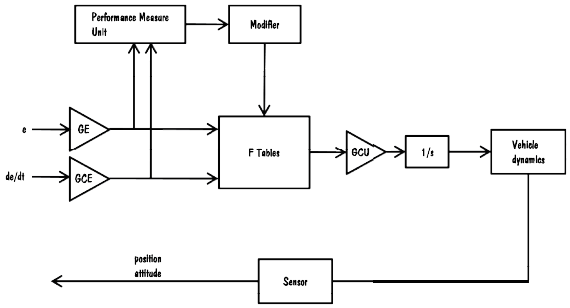


Figure 3. The self-organizing controller

A zero performance specification implies that the state is satisfactory and the non-zero performance indicates that the state is unsatisfactory. Because we used the fuzzy incremental controller, the table *F* contains the change of the control input *cu*. The error *e* and the change of the error *ce* is multiplied by the gain *GE* and *GCE* respectively before entering the rule base block *F*. From the table *F* the table lookup value *cu* is multiplied by the output gain *GCU* and integrated to become the control signal *U*. The outer loop monitors the states, *e* and *ce*, and it modifies table *F* through a modifier *M*. This procedure is repeated until the vehicle arrives the target point.

4. Simulations

For the simulation, we assumed that the environment is 15x15x5 three dimensional Cartesian coordinate space. A cell dimension is 2m*2m*2m cubic cell. We initialized ten start points arbitrary. And a million iterations for each start point. The target point which is (15, 15, 5) in the environment is remained same for all simulations. The path in Figure 4 shows our simulation result with start point (1, 1, 1) with 50 random obstacles. The vehicle used in the simulation is ODIN which is developed by University of Hawaii. It has six degrees of freedom and the weight is 125kg and the radius of the vehicle is 0.3m. We set current effect of the ocean as 0.3m/s along the y axis.

After the learning phase, the vehicle generated its policy to find the safe path to the target point for the given map. The policy maps during the learning phase are shown Figure 5.

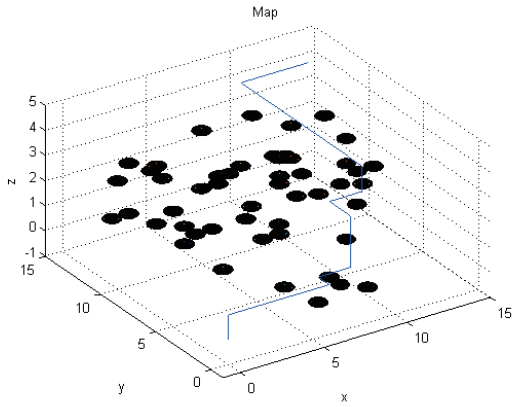
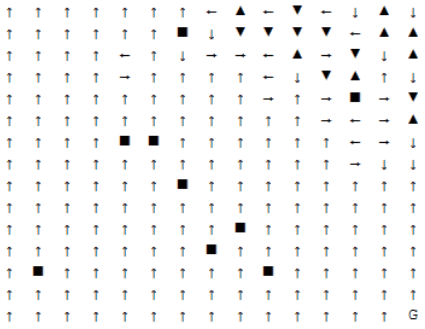
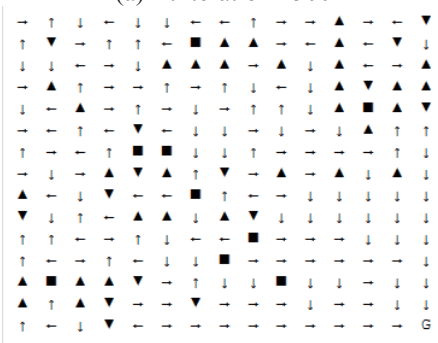
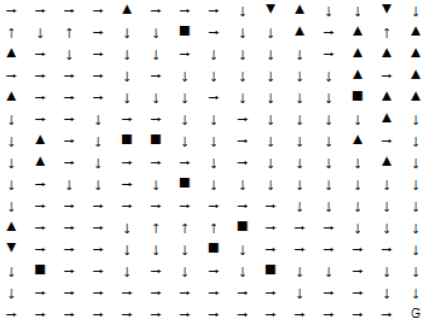


Figure 4. Path generated by the autonomous navigation system from (1, 1, 1) to (15, 15, 5) with 50 random obstacles.


(a) At iteration #500


(b) At iteration #200000


(c) At iteration # 500000

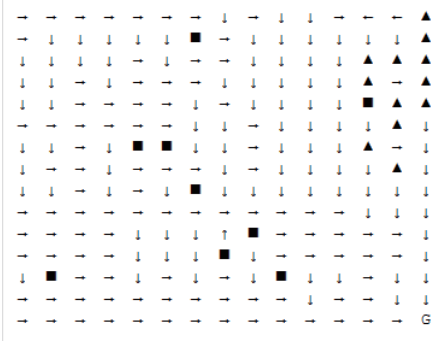

(d) At iteration #1000000

Figure 5. The policy generated by the autonomous navigation system during a million iterations of learning.

After the learning phase, the self-organizing controller starts to drive the vehicle to the target point. Figure 6 shows the actual trajectory and the

desired trajectory of the vehicle.

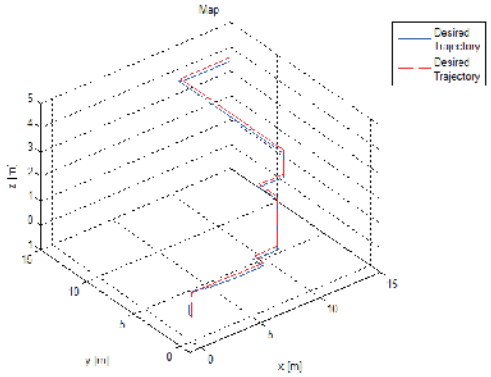
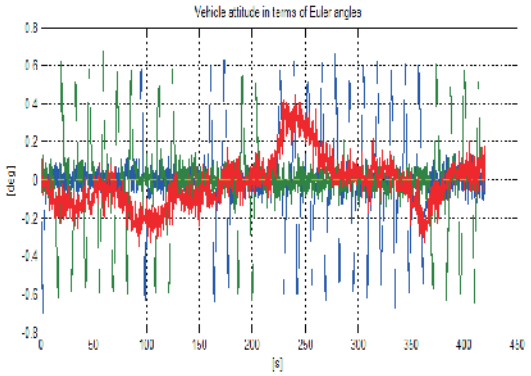


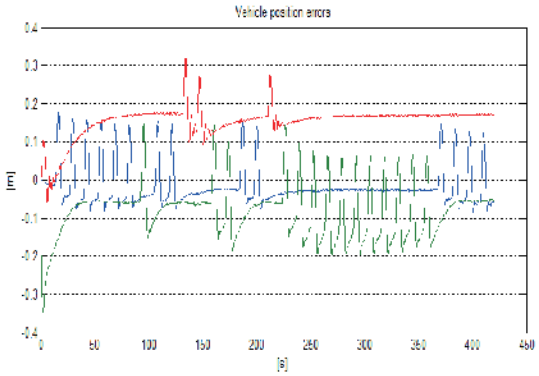
Figure 6. The desired trajectory and the actual trajectory.

The vehicle position and attitudes are shown in Figure 7.

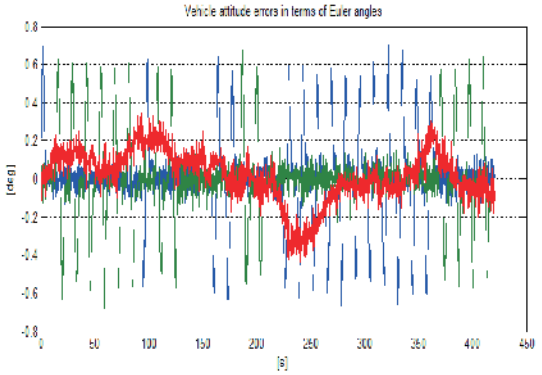
(a) The positions of the vehicle



(b) The attitudes of the vehicle.



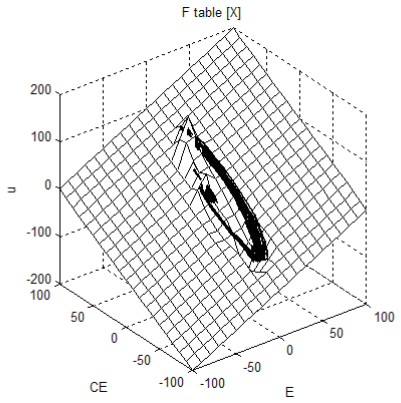
(c) The position errors of the vehicle



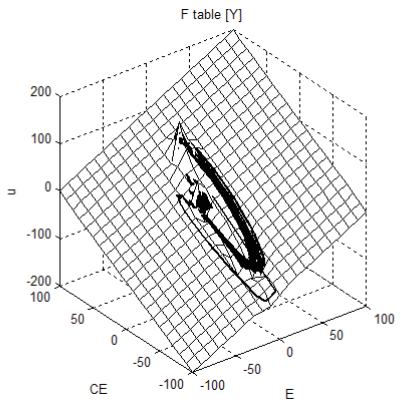
(d) the attitude errors of the vehicle

Figure 7. The positions and the attitudes of the vehicle.

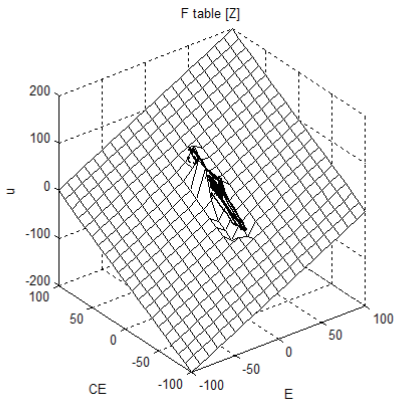
The self-organizing controller modifies its lookup table F to adapt for a given environment. We can see the convergence of the table F in Figure 8.



(a)



(b)



(c)

Figure 8. The convergence of (a) *x* axis (b) *y* axis (c) *z* axis table *F*.

5. Conclusion

The designed system for the AUV produces the safe path through the Q-learning scheme with iterations. After the learning phase the fuzzy controller sends current location to the navigation system, and then the navigation system gives reference signals to the controller. The planned path leads the vehicle to the target points without hitting obstacles. The self-organizing controller adjusts its own table without human intervention to adapt to a given environment. Finally we successfully designed two sub systems for intelligent autonomous underwater vehicle and verified its performance by simulations. In our result the vehicle could find the safe path and arrive the target point with its own intelligence. Future study can be applying this system to the dynamic environments.

ACKNOWLEDGEMENT

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute for Information Technology Advancement) (IITA-2009-C1090-0902-0001)

REFERENCES

[1] Russell, Stuart J.; Norvig, Peter, “Artificial Intelligence: A Modern Approach (2nd ed.)”, Upper Saddle River, NJ: Prentice Hall, 2003

[2] Richard S. Sutton; Andrew G. Barto, “Reinforcement Learning: An Introduction”, Cambridge, MA: The MIT Press, 1998

[3] Ethem Alpaydin, “Introduction to Machine Learning”, Cambridge, MA: The MIT Press, 2004

[4] Jan Jantzen, “Foundations of fuzzy control”, Wiley, 2007

[5] Gianluca Antonelli, “Underwater robots”, Springer, 2003

[6] Geoff Roberts and Robert Sutton, “Advances in Unmanned Marine Vehicles”, The Institution of Electrical Engineers, 2006

[7] Smart. W. D., Pack Kaelbling L., “Effective Reinforcement Learning for Mobile Robots”, Proceedings of the IEEE International Conference on Robotics & Automation, 2002

[8] Chun-Fei Hsu, “Self-Organizing Adaptive Fuzzy Neural Control for a Class of Nonlinear Systems”, IEEE Transactions on Neural Networks, July 2007