Acquisition of Dynamics Control Algorithm of Autonomous Underwater Vehicle by Reinforcement Learning and Teaching Method Considering Thruster Failure

Hiroshi KAWANO
Department of Environmental and Ocean Engineering
Graduate School of Engineering, University of Tokyo
kawanoh@iis.u-tokyo.ac.jp

Tamaki URA
Underwater Technology Reasearch Center
Institute of Industrial Science, University of Tokyo
ura@iis.u-tokyo.ac.jp

Abstract:

A training algorithm for dynamics control of nonholonomic AUV (Autonomous Underwater Vehicle) is proposed in this paper, which can recover from thruster failure during cruising mission. It is based on Q-learning and teaching method. The back up data that represents dynamics model expressed in the form of Baysian Net can be used effectively in this case. In order to overcome difficulties due to making discrete expression of continuous state space of AUV, the algorithm uses multi resolution Q-value tables which is combined in the form of subsumption architechture. Simulation results show high performances of the proposed algorithm for a vertical ascent mission in a severe current condition. It is shown that AUV users can conveniently and quickly train the control algorithm of the AUV by using simulation of dynamics of the vehicle.

Introduction

Nowadays the AUV (Autonomous Underwater Vehicle) is anticipated as a promising new platform for underwater inspection. There is not any constraint on vehicle's motion because it has no tether cable, which is fitted on the ROV (Remotely Operated Vehicle). But development of AUV's control algorithm which should manage various cases of emergency situation, is not easy to develop because of the following reasons:

(1) AUV dynamics is usually a non-holonomic behavior, its lateral movement is very slow compared to that of longitudinal motion, so that it is difficult to control sway motion.
(2) There is severe unknown turbulence in the underwater environment. The water current may run in more than 4kt, while the max speed of AUV is about 3kt (about 1.5 [m/s]).
(3) It is not easy to estimate hydrodynamics coefficients for dynamics description, which require a number of experiments in towing tanks be carried out.
(4) Reliability of actuators fitted on AUV is not perfect, because electronical actuators are used in water of high pressure.

This paper proposes a new training system for the nonholonomic AUV considering a strong current and thruster failures. This new training system uses reinforcement learning algorithm equipped with teaching method to overcome the first, second, and third problems. To solve the fourth problem we introduce Baysian Net to express the dynamics model of the AUV. The proposed system can realize the ability to recover when some of the thrusters fail. Performances of the proposed algorithm are demonstrated through some simulation examples.

History and Related Work for AUV Control

Ishii et al [Ishii,Fujii,Ura] proposed control algorithms for AUVs using artificial neural

networks. These methods have the ability of on-line adaptation during underwater mission. They partially overcame the 3rd problem. But when the AUV encounters large disturbances which the AUV has never known by experience, it may takes a long time to come up with a conclusion by this algorithm. Offline learning before operation is also a time consuming process, and still remains the problem of unstability. The solution is only available for the motion which can be controled by simple feedback algorithm, which is called holonomic motion.

Takai et al [Takai,Fujii,Ura] proposed a self diagnosis system for AUV using artificial neural networks , which can mainly deal with sensor failure problems. When a thruster failure occurred during an underwater mission, the system simply stops the mission, and no additional learning process for motion control can be carried out.

Contribution

In this paper, it is assumed that there is a strong water current while an AUV is under operation. It is considered that the control method can keep on cruising even though some thrusters are failed by using the other active thrusters. It is also considered that the learning algorithm can modify control algorithm rapidly in a limited short time without violating AUV's mission.

To do these, we introduce Q-Learning method for AUV control, which has the ability to learn action policy by interacting with an environment that follows MDP(Markov Decision Process). It is said that Q-Learning is suitable to solve a problem like a path planning in a discrete maze[Sutton,Barto]. In case of the AUV of non-holonomic system in severe water current, velocity control algorithm should contain some procedures like path planning algorithm. For example consider the R-One Robot, which is a torpedo shaped AUV for long range underwater inspection developed by Underwater Technology Research Center at the Institute of Industrial Science of the University of Tokyo [Ura,Obara]. It has one main thruster as shown in Fig.1 equipped with a heading mechanism for surge and yaw motion control, two vertical thrusters for heave motion, and a pair of elevators for pitching and rolling motion control. Its maximum speed is 3.6 kt. Since there is no actuator for sway motion and it cannot go astern, the R-One Robot is a non-holonomic system that has limited heading radius that is not equal to zero.

If there is a water current from lateral direction, zero velocity motion in horizontal plane can not be achieved by a simple feedback control algorithm. To do this, the following procedure is needed.

i) By using the main thruster and its heading rudder, the vehicle's heading direction should be driven against to the current direction.
ii) By using main thruster power control, zero ground velocity should be achieved.

Considering this procedure, Q-Learning could be suitable for non-holonomic AUV control in a severe water current.

Applying Q-Learning algorithm to a practical AUV system has several problems to be solved. First, Q-Learning should be used in discrete state spaces. Since AUV's dynamics state space is continuous, a suitable state partitioning method is needed, which is proposed in the followings. Second, if there are many partitions in every state which should be searched during learning process, state space explosion will occur. Because of high degree of freedom, AUV's state space is too huge to be completely searched. Appropriate heuristic technique for state space searching, therefore, should be introduced. For this purpose, we adopt a teaching method which gives the AUV advices for action selection. Third, the learning response of Q-Learning is not so high that the teaching advice previously mentioned doesn't reflect immediately on modification of the control algorithm. Here, we propose a suitable learning procedure for quick learning based on advices. Fourth, if an action set that can be selected by the AUV is changed during underwater mission due to some thruster failures,

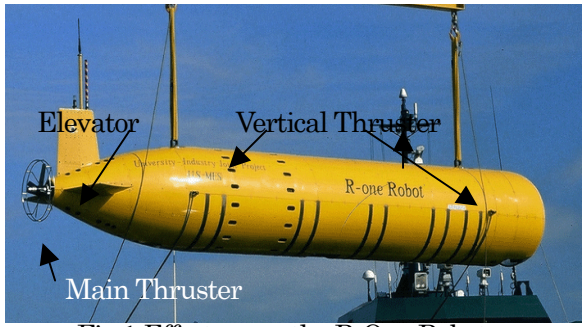


Fig.1 Effecters on the R-One Robot.

Direction of the main thruster can be

the stored Q-Values are no longer applicable. In order to overcome this problem, we propose a method using back up diagram data in the form of Baysian Net, which desribes the dynamics model[Mitchel].

Q-Learning

Reinforcement learning is a method that solves the Markovian sequential decision tasks where a learning agent operates in a stochastic dynamical environment. Dynamics model of the environment is not needed and the performance of the agent is judged by the scalar signal known as the *reward* which is received from the environment as the result of the agent's action. The objective is to determine the best policy which specifies the agent's action in each state in order to maximize a specified cumulative measure of the reward received from the environment.

Q-Learning is one of the most useful reinforcement learning algorithms. In Q-Learning, the state-action value *Q(s,a)* is estimated which is the return in state s when action a is performed. The optimal policy is followed thereafter. The state space is discrete, and a separate *Q(s,a)* exists for each action a. When the agent takes an action a from a state s, the *Q(s,a)* is updated as follows:

$$Q(s,a) \leftarrow Q(s,a)+\alpha(r+\gamma \max_{a' \in A} Q(s',a')-Q(s,a)) \quad (1)$$

where s' is the actual next state, γ is the discount rate, α is the step-size parameter, A is a set of possible actions, and r is the expected value of the reward that the agent receives from the environment by taking action a in state s. When *Q(s,a)* has converged, the greedy policy that selects action value a which maximize *Q(s,a)* is optimal. During learning process, the agent should explore by performing different actions. One way is to select actions according to the Sofmax Action Selection method[Sutton, Barto].

Method for Recovering from Thruster Failure

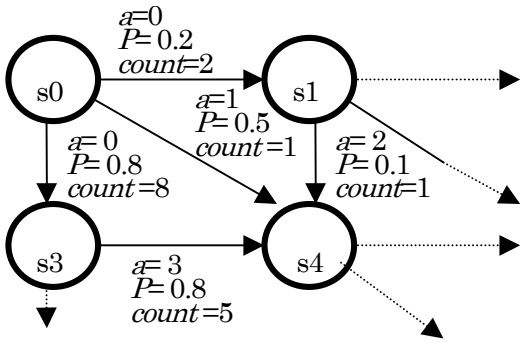
If the action set that the learning agent can select is changed, stored Q-values are no longer applicable. Here, we propose a method that stores dynamics model of learning environment during learning process. This dynamics model is expressed in the form of Baysian Net including state transition and high reward count data as shown in Fig.2 and Fig.3[Mitchel]. In this case, **s** stands for the seabed referenced velocity of AUV and water current in body cordinates. And **a** is actuator commands.

Thruster failure can be recovered as follows:

i) When the available action set is changed due to thruster failure, eliminate the transaction of dynamics model which contains forbidden actions of the thruster failed.

ii) Q-values are recalculated untill they converge using the dynamics model data which was made in i) procedure.

The recalculated Q-Value is the *state-action value* considering the action set which is available after the thruster failure occurrence.



P denotes probability of state transition, and *count* denotes the frequency of transition observation.

Fig 2. Dynamics Model stored during learning process

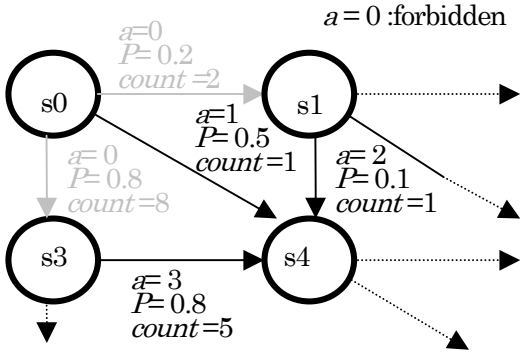


Fig 3. Dynamics Model after elimination of transitions which contain forbidden actions

Multi Resolution Module Learning

Suitable resolution for AUV's state space is generally unknown because the number of parameter of dynamics is indefinite. In the case of velocity control for AUV, some parts of the state space should have high resolution; for example, velocities near to the control destination. On the contrary, some other remaining parts may not need to have such high resolution, as velocities far from the control destination. Considering this situation, we introduce multi Q-Learning modules. Resolutions in state space of each module are different from the others. Each module is placed in a layer as the subsumption architecture controller as illustrated in Fig.4[Brooks]. The cells in each modules stand for the descrete state **s**. The upper layer module has the finest resolution. The output of bottom layer which covers whole state space, can be a random value. Resolution of each layer module becomes week as the rank of subsumption module comes down. If the sensor input is identified as a state which has not been experienced, each layer module outputs no action value. By the proposed method, the AUV can select an action by the analogy of experienced data even if the sensor input is not in experienced state. This mechanism is effective especially in the case that teaching data includes a suitable path to the destination.

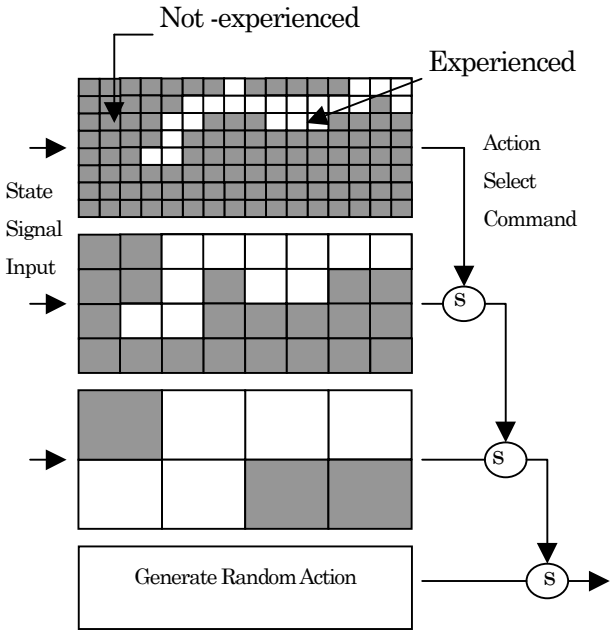


Fig 4. Structure of Multi Resolution Controller

Teaching Method

Although the state space of AUV consists of many discrete states, there are many states that do not have significant meanings among them to guide the AUV to the destination. In order to make the AUV be capable of searching in state space, we introduce the following teaching procedure. An operator gives some advices about action selection to AUV if necessary. When the AUV receives teaching input, it stops selecting action for several times and follows teaching input[Clouse,Utgoff]. If there is no advice, the AUV selects actions at which $Q(s,a)$ is maximum so as to show the best behavior to the operator.

Learning speed of Q-Learning is low in case that it involves many state transition to reach the high reward point. Efficiency of the path to the high reward point is not evaluated till the AUV traces the same path again and again. So that the teaching advices don't immediately change the AUV's action selection policy. The operator may feel impatient because the AUV requires same advices so many times. To avoid this situation, we adopt the model-learning method using dynamics model data. When the AUV reaches to the destination state, Q-Values are recalculated using dynamics model as in the same way as the case of thruster failure. It has the same effect as tracing the successful path for many times.

Simulation

We consider the R-One Robot as a prototype of the AUV system, and show simulation results of the proposed method. Assume a simple mission that the AUV ascends vertical upwards in a strong current. This mission is important because the non-holonomic AUV must have the ability to safely escape upwards even if there are some thrusetr failures in strong water current. Speed of water current is assumed to be 1.5kt(about 0.85[m/s]). The mission starts from the configuration in which the AUV receives the current from backward direction. Structure of the controller is illustrated in Fig.5. Multi Resolution Controller for horizontal motion consists of 4 layers as shown in Fig. 4. Table 1 shows the maximum and minimum values of the state space variables, and target velocity of AUV. State space axes are divided into 20,10,5 discrete

values in terms of

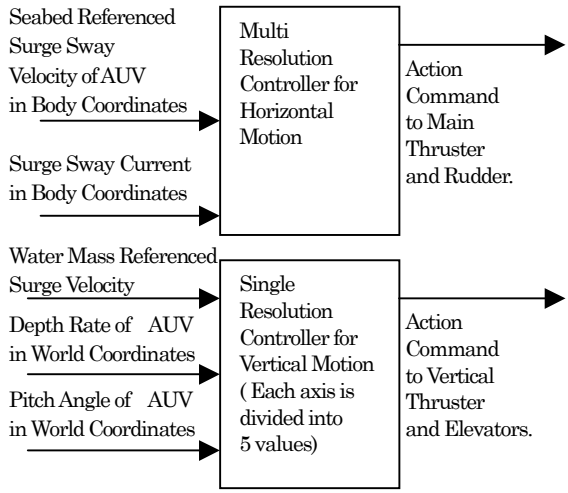


Fig. 5 Structure of Demonstration System

Table 1 Maximum, Minimum values of discrete state space and Target Velocity of AUV

state space axis	Max	Min	Target	Tolerant Error
Surge velocity [m/s]	1.5	-1.5	0.0	0.2
Sway velocity [m/s]	1.0	-1.0	0.0	0.2
Heave velocity [m/s]	0.3	-0.3	-0.2	0.1
Pitch Angle [deg]	15.0	15.0	-	-
Surge current [m/s]	0.9	-0.9	-	-
Sway current [m/s]	0.7	-0.7	-	-

resolution. Teaching simulation is carried out under the assumption that speed of current is changing. Here γ and α are 0.9 and 0.1, respectively. High reward $r = 1$ is given when AUV reaches the target velocity. Table 2 shows the teaching interruption rate which is defined as the proportion of action selection counts by teaching to the all action selection counts in training process of horizontal motion. The proposed multi layer controller is used in case 1 and normal single resolution controller in case 2. In case 2, state space axes are divided into 20 values. Teaching number in the table shows the execution number of Q-Value calculation using dynamics model.

It shows that the multi resolution controller requires less number of repetition of teaching input than the single resolution controller. Figs.6 and 8 show motion of the AUV and action without thruster failure after teaching process shown in Table 2. It demonstrates that the AUV acquired the proper motion control ability by the proposed algorithm.

The training is continued after a thruster failure occurs. Not only the optimal actions the operator knows ,but also the actions in the case of rudder failure in horizontal motion, are taught explicitly. But minimal teaching advices are given to the vertical motion controller, and learning is carried out mainly by self Q-Learning. Figs. 7 and 9 show the motion and actions in case that the vertical

Table 2 Result of Teaching Interruption Rate for Horizontal Motion

teaching No.	current speed [m/s]	teaching interruption rate	
		Case 1 Multi layers	Case 2 Single layer
1	0.7	0.80	0.87
2	0.85	0.18	0.70
3	0.55	0.23	0.64
4	0.60	0.13	0.75
5	0.77	0.00	0.20

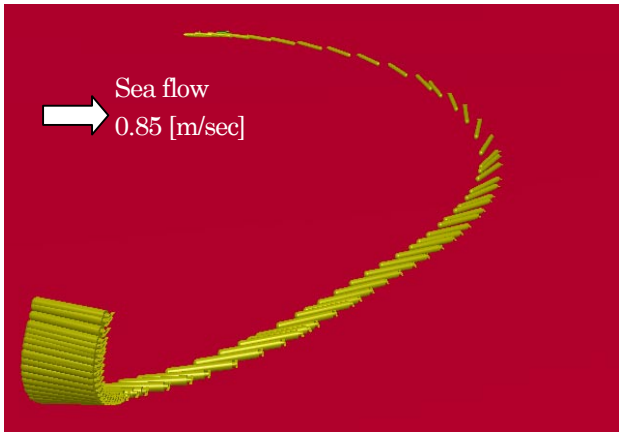


Fig. 6 3D-View of Vertical Ascent Motion in the case of No Thruster Failure

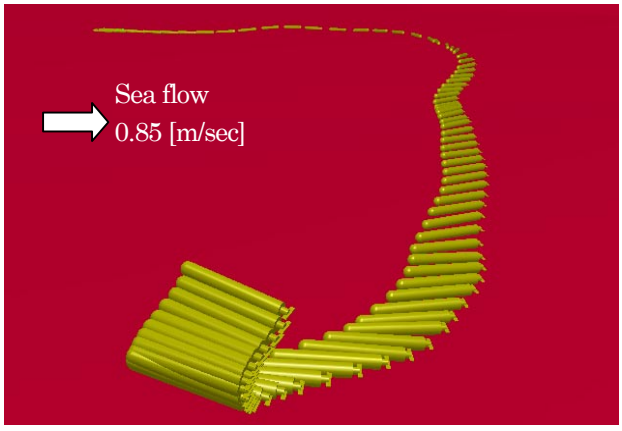


Fig. 7 3D-View of Vertical Ascent Motion in the case of Thruster Failed

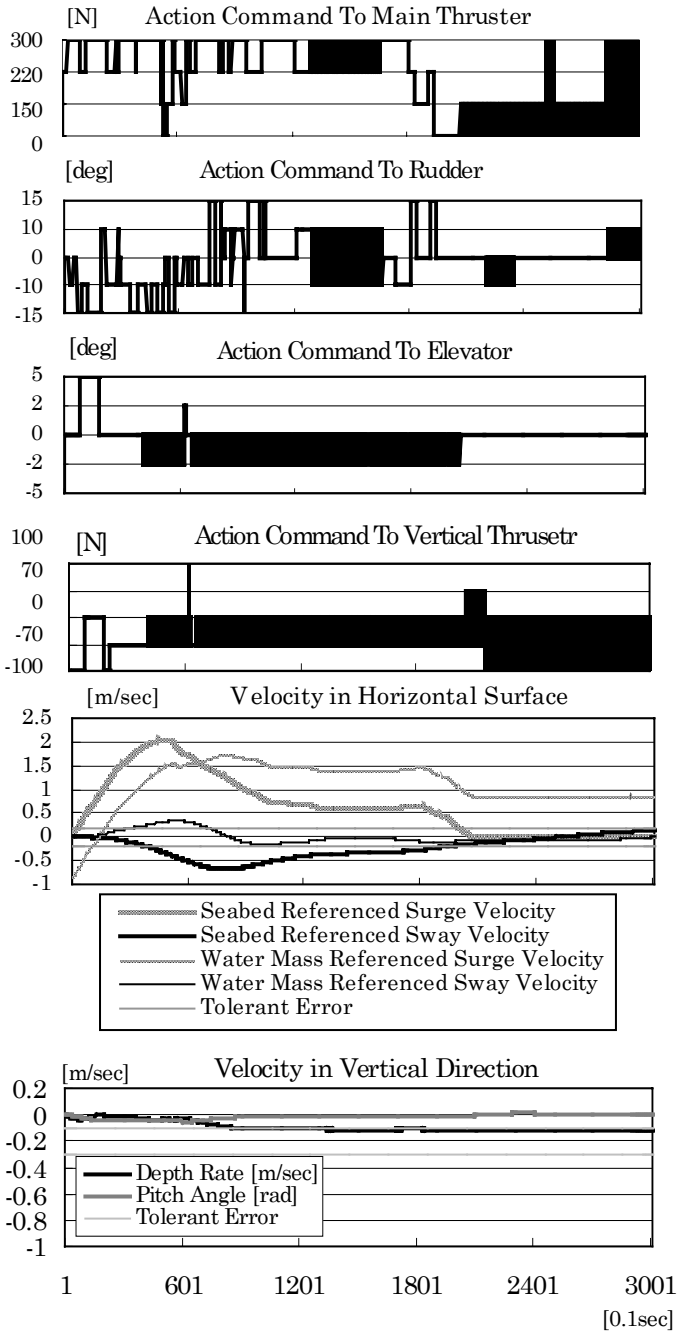


Fig. 8 Action Commands and Velocity in the case of No Thruster Failure

thrusters and the rudder are defeated;i.e. the vertical thrusters stop, and the maximum angle of rudder is limited up to 10[deg]. The AUV uses a pair of elevators for control of its vertical motion instead of the vertical thrusters. As a result the pitch angle of the AUV rises to 0.2[rad]. Caused by rudder angle limitation, radius of the trajectory curve is larger than that of the case of no rudder failure.

Since the vertical motion controller outputs do not include action command to the main thruster, it can not control surge motion. It was not certain before new learning for the vertical motion controller that the AUV can reach the target velocity in vertical motion only by the pair of elevators. But the vertical motion controller learns that the AUV can achieve vertical ascent motion by setting elevator angle in 5 or 2[deg].

The AUV succeeds in position keeping against the current in a horizontal plane while ascending vertically. It takes several hours to train the AUV model in the simulation. These results show that the operator can easily train the alternative motion by teaching, in case that thruster failure

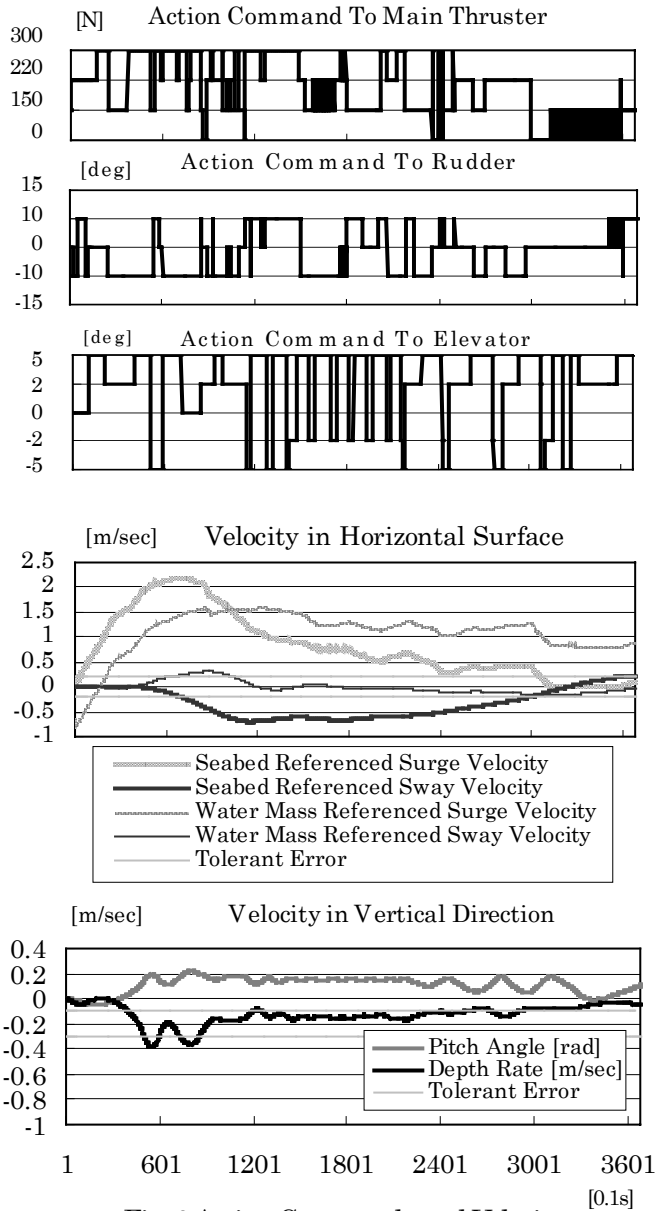


Fig. 9 Action Commands and Velocity in the case of Thruster Failure

occurs explicitly. The proposed algorithm can also provide the ability to recover from thruster failure by self learning implicitly.

Conclusion

A training algorithm for dynamics controller of AUV considering thruster failure and severe turbulence is proposed. The proposed system is based on Q-Learning and teaching method. By teaching method, learning process of Q-Learning can be accelerate and avoid state space explosion. The use of multi resolution state space for Q-Learning is efficient in the case that some teaching advices are given to the AUV. Back up data of dynamics model based on Baysian Net can realize the ability to recover from thruster failure. By the proposed algorithm the AUV's user can easily train dynamics controller and acquired control algorithm can be applied in the case of thruster failure.

REFERENCE

[Ishiii,Fujii,Ura] Ishii, K.; Fujii, T.; Ura, T.: "An on-line adaptation method in a neural network based control system for AUVs " IEEE Journal of Oceanic Engineering, Volume: 20 Issue: 3 , pp221 -228 July 1995

[Takai,Fujii,Ura] Takai,M.; Fujii,T., Ura,T.: "Development of a System to Diagnose the Autonomous Underwater Vehicle", International Journal of System Science on Unmanned Underwater Vehicle Control, Vol. 30 No.9 pp981-988, September 1999

[Sutton,Barto] Sutton, S.R.; and Barto, G.A.: "Reinforcement Learning" , The MIT Press

[Clouse,Utgoff] Clouse, A.J.; Utgoff, P.E.: "A Teaching Method for Reinforcement Learning", Proc of the Ninth International Conference on Machine Learning pp92-101, 1992

[Brooks] Brooks, R. A.: "A robust layered control system for a mobile robot.", IEEE Journal of Robotics and Automation., RA-2, p14-23. April,1986

[Mitchell] Tom M. Mitchell "Machine Learning", WCB/McGraw-Hill pp154-199.

[Ura,Obara] Ura, T; Obara, T;: "Sea Trials of AUV "R-One Robot" Equipped with a Closed Cycle Diesel Engine System", Proc of IEEE OCEANS'97, Oct. 1997