Engng Applic. Artif. Intell. Vol. 4, No. 4, pp. 309–318, 1991
Printed in Great Britain. 

0952-1976/91 $3.00+0.00


*Contributed Paper*

# Neural-Network-Based Adaptive Control Systems for AUVs

TERUO FUJII
University of Tokyo

TAMAKI URA
University of Tokyo

*In this paper, two types of learning systems, the supervised learning system and the unsupervised learning system, are introduced to construct neural-network-based control systems. Both approaches are applied to longitudinal motion control of the free-swimming vehicle "PTEROA".*

*The supervised learning system is based on the simple concept of learning the behavior of the supervisor controller. It is implemented along with a fuzzy controller as the supervisor, and evaluated through numerical simulations and experiments. It is shown that the characteristics of the neural networks, such as flexibility of the I/O selection and saturation of the outputs, provide favorable performance to the control system for AUVs.*

*The unsupervised learning system, which is called "SONCS", is introduced as an adaptive control system. The subsystems and the organizing process of the controller are described in detail. The SONCS is applied to the control problem of the untethered test-bed vehicle PW45, and its performance is evaluated through free-swimming tank tests. It is shown that after several times of adaptation, the SONCS succeeds in organizing an appropriate controller for horizontal swimming at a desired depth.*

Keywords: Adaptive control, self organization, learning control, intelligent motion control, error back-propagation, neural networks, underwater vehicles, multilayered neural networks, supervised and unsupervised learning, robustness, fuzzy control.

## INTRODUCTION

The basic concept of artificial neural networks originated from the efforts to model brain behaviors in the 1940's.[1] In the past decade their fascinating characteristics as an information processing tool have been demonstrated, especially in the field of pattern recognition and optimization problems.[2–4] Their distinctive features depend on massive parallelism, nonlinear operation and learning ability.

As artificial neural networks execute through parallel distributed processes, they can deal with multi-input multi-output (MIMO) systems like Autonomous Underwater Vehicles (AUVs). It is expected that they can follow the nonlinearity of the hydrodynamic forces acting on the vehicle body and the complex interactions between the thrusters, the control surfaces, etc. Moreover, even if the system dynamics are unknown or difficult to represent mathematically, they may be able to handle these kind of complex dynamics adaptively using their learning ability. Since AUVs are operated in an unstructured environment, these features are attractive for an intelligent control system.

Several attempts have been carried out to make a nonlinear adaptive control system using neural networks.[5–8] A framework of representation of sequential adaptive behaviors using neural networks was first suggested by Jordan.[5] A mathematical aproach which is similar to MRAS (Model Referenced Adaptive System) was discussed in detail from the system theoretic point of view by Narendra.[8] On control problems of the real system, however, there are very few examples of implementation of artificial neural networks.

This paper introduces two types of learning systems which utilize neural networks as controllers. The first one is a so-called "supervised" learning system which is based on a simple concept of learning I/O (input–output) relations provided by the supervisor controller which has been designed with conventional theories.[9] This system was investigated as the first step to get to the second one, that is, an unsupervised learning

Correspondence should be sent to: T. Fujii, Institute of Industrial Science, University of Tokyo, 7-22-1 Roppongi Minato-ku Tokyo, 106, Japan

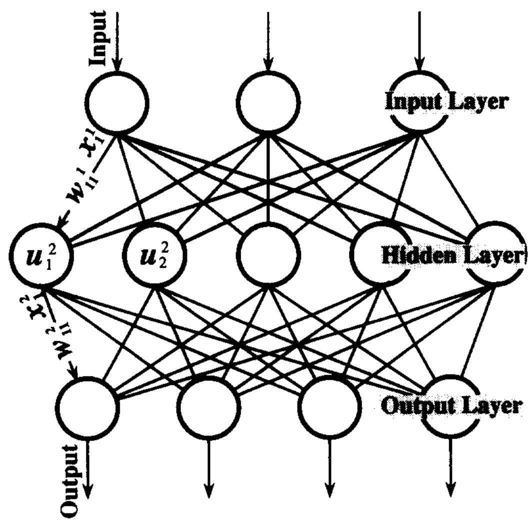


Fig. 1. The connectionist model.

system which can automatically learn the control actions which should be taken without a supervisor. A general architecture and a procedure of unsupervised learning were introduced by the authors[10, 11] and the system was named "Self-Organizing Neural-Net-Controller System (SONCS)". This system includes a neural network called a "Forward Model Network" which represents the dynamics of the controlled object. The controller network is adaptively adjusted according to error information, which is given by a specified evaluation function of motion, calculated from the outputs of the forward model.

Generally speaking, performance of the robotic system should be investigated in the real world, not just via simulations. The proposed neural-network-based control systems are, therefore, applied to longitudinal motion control of an actual free-swimming vehicle, the "PTEROA"[12] and evaluated through tank tests.

## MULTILAYERED NEURAL NETWORKS

A multilayered neural network, which is called "the Connectionist Model",[13] is used as a basic structure in this paper. The network consists of some layers, namely, an input layer, hidden layers and an output layer. Each layer includes several numbers of neurons (cf. Fig. 1). When there are no recurrent connections, the processing of the $i$-th neuron of the $n$-th layer is given by:

$$u_i^n = \sum_j w_{ij}^{n-1} x_j^{n-1} + h_i^n, \qquad x_i^n = f(u_i^n), \tag{1}$$

where $u_i^n$ is the membrane potential, $x_i^n$ is the output, $w_{ij}^n$ is the synaptic weight of the connection from the $j$-th neuron of the $n$-th layer, and $h_i^n$ is the threshold value. $f(u)$ is the output function of the neuron, which is sigmoidal except in the input layer, and is defined by:

$$f(u) = \begin{cases} u & : \text{input layer}, \\ 1/[1+\exp(-u)] & : \text{others}. \end{cases} \tag{2}$$

The behavior of the network is determined by the set of synaptic weights $w_{ij}^n$ and thresholds values $h_i^n$. These values are changed in order to implement a specific function. This process is called learning, which can be executed with the following Error Back-Propagation method.[13] Let the potential function $E$ represent the summation of the output errors as:

$$E = \tfrac{1}{2} \sum_i (t_i - o_i)^2. \tag{3}$$

Here, $o_i$ is the output of the network corresponding to a certain set of inputs and $t_i$ is the desired output. The objective of learning is to minimize the potential function $E$. According to the maximum gradient scheme, each synaptic weight $w_{ij}^n$ should be changed by $\Delta w_{ij}^n$.

$$\Delta w_{ij}^n = -\eta \frac{\partial E}{\partial w_{ij}^n} = \eta \delta_i^n x_i^{n-1}, \tag{4}$$

where $\eta$ is the parameter which determines the speed of learning. $\delta_i^n$ is the error signal based on the Generalized Delta Rule[13] and is obtained by substituting equations (1), (2) and (3) into eqn (4) as:

$$\delta_i^n = \begin{cases} (t_i - o_i) f'(u_i^n) & : \text{output layer}, \\ f'(u_i^n) \sum_k \delta_k^{n+1} w_{ki}^n & : \text{others}. \end{cases} \tag{5}$$

To stabilize the iterative calculation of $\Delta w_{ij}^n$ in eqn (4), the technique proposed by Rumelhart *et al.*[13] is used and is given by:

$$\Delta w_{ij}^{n-1}(p+1) = \eta \delta_i^n x_i^{n-1} + \alpha \Delta w_{ij}^{n-1}(p), \tag{6}$$

where $\Delta w_{ij}^n(p)$ is the changing rate of the synaptic weight at the $p$-th step of the iterative calculation, $\alpha$ is the parameter which determines momentum of learning. $h_i^n$ should be changed in the same manner as eqn (6) by substituting 1 for $x_i^{n-1}$.

Thus, multilayered neural networks, after training, come to represent a nonlinear mapping between input and output patterns. When the learning procedure is implemented as a built-in mechanism, neural networks can automatically acquire new information-processing abilities as specified by the given desired outputs.

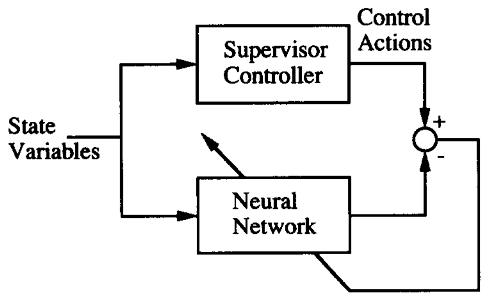


Fig. 2. Supervised learning system.

## SUPERVISED LEARNING SYSTEM

A neural network can be used as a controller of the MIMO system when it involves a mapping between state vectors of the system and corresponding control actions. In the simplest case, a neural network can acquire the ability to represent such a mapping by learning the I/O relationship from an appropriate controller (cf. Fig. 2). Since this controller performs in the role of a supervisor, this type of learning system is called a "Supervised Learning System".

The procedures for typical supervised learning are:

(1) Taking data of motion of the controlled object and on control actions of the supervisor,
(2) Assembling teaching samples consisting of a certain number of sets of inputs and desired outputs which have been derived from the data of motion and control action, and
(3) Updating the synaptic weights by the Error Back-Propagation Method.

## SUPERVISED LEARNING FOR PTEROA60

A neural network for controlling longitudinal motion of the PTEROA60[10] is constructed as an example of such an implementation and its performance is examined experimentally and numerically. PTEROA60 is a tethered test-bed (60 cm in length) which was used for constructing the PTEROA150[12] (cf. Fig. 3). The PTEROA150 is a streamlined cruising-type AUV which has been developed at the Institute of Industrial Science, the University of Tokyo. The shape of the longitudinal cross-section at the center line is the standard wing section NACA0030 and the transverse cross section is approximately oval. Longitudinal motion of the vehicle, such as pitching and heaving, are controlled with a pair of elevators which are fitted aft of the body. The equations of motion of the vehicle were investigated by towing tank tests and numerical analysis.[14] In the following, the pitching angle, the depth and the trimming angle of elevators are represented by $\theta$, $d$ and $\delta_e$, respectively, as illustrated in Fig. 4.

A simple "fuzzy" controller is selected as a supervisor. Data of the PTEROA60's dynamic behavior are determined through numerical simulations. The fuzzy controller is constructed so as to maneuver the vehicle at the constant depth while minimizing the rate of change in depth and pitching. In the simulation, its membership functions are tuned to attain the ability to at least let the vehicle swim horizontally. The inputs to the supervisor controller are the changing rates of depth $\Delta d$ and pitching angle $\Delta\theta$, and the output is the incremental trimming angle of the elevators $\Delta\delta_e$.

The structure and I/Os of the neural network, which includes one hidden layer with 5 neurons, are shown in Fig. 5. It should be noted that the I/Os are not the same as those of the supervisor. It was decided to have the output of the neural network be the trimming angle of the elevators $\delta_e$. This corresponds to the time integral of the output of the supervisor controller. As this integral is determined by the control calculation of the supervisor and the resulting vehicle's motion, the neural network will learn the maneuvering actions which implictly include information on the vehicle's dynamics (cf. Fig. 6).

Figure 7 shows the results of the numerical simulation with the supervisor fuzzy controller, with initial conditions $\theta = 0.2$ rad, $d = 1$ m and $\delta_e = 0$ rad. Two hundred sets of teaching samples are prepared from motion data. To get a controller with satisfactory performance, synaptic weights are updated 70,000 times by the Error Back-Propagation calculation, i.e. eqn (6).

## RESULTS OF SUPERVISED LEARNING

### Numerical simulation

Figure 8 shows results of the simulation of the vehicle motion as controlled by the learned neural network, given the same initial conditions as used in the teaching samples of Fig. 7. Note that the network reduces the

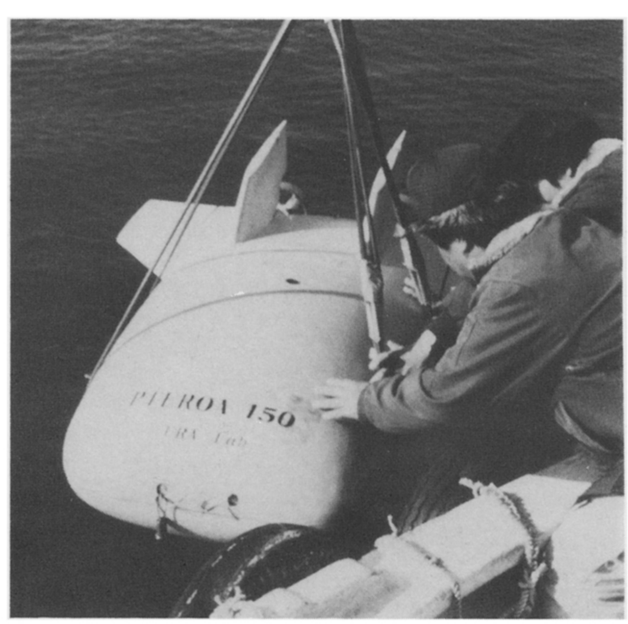

Fig. 3. PTEROA150 vehicle being launched for the sea trial.

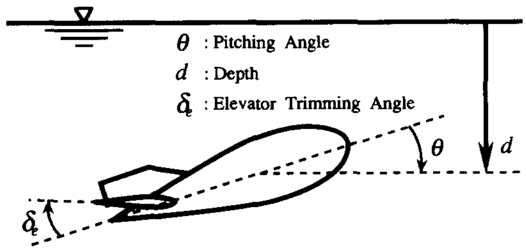


Fig. 4. Longitudinal motion of PTEROA vehicle.

oscillation of the pitching motion, which is induced when controlled by the supervisor. This difference is mainly caused by two reasons: the neural network involves information on the vehicle's dynamics implicitly as mentioned above, and saturation of outputs of neurons due to a sigmoidal feature of the output function [eqn (2)] causes moderation of control actions.

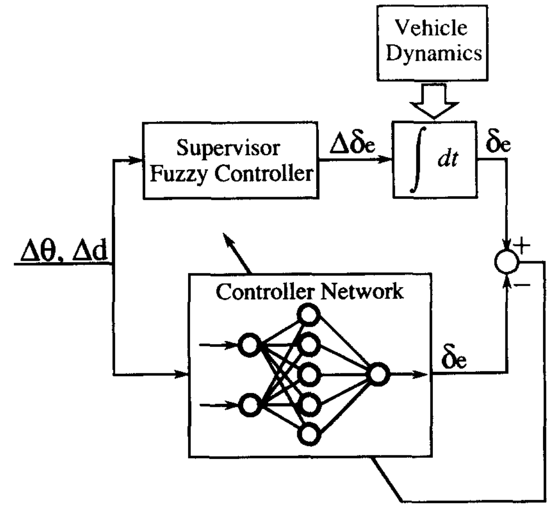


Fig. 6. Supervised learning system for PTEROA60.

**Free-swimming tank test**

In the practical environment, many distrubances exist which reduce the stability of motion of the vehicle. They consist of mechanical and electrical actuator noise, sensor noise, unsteady water flow, etc. In order to investigate the effectiveness of the neural network in the real world, free-swimming tests are carried out with the "PTEROA60" vehicle in a circulating water tank. Figures 9 and 10 show results of the tests controlled by the supervisor fuzzy controller and the learned neural network, respectively. It is interesting that the learned neural network succeeds in controlling the vehicle in the environment where the supervisor fails.

**Robustness against disturbances**

The supervisor fuzzy controller has been mainly tuned based on the designer's instinct and it does not involve the information on the vehicle's dynamics. It is, therefore, predictable that the controller does not succeed in controlling the vehicle in the tank. In contrast, the learned neural network has acquired more robustness against unknown disturbances than the supervisor.

The performance of the learned neural network and the supervisor controller, in particular the behavior against disturbance, are compared in numerical simulations, in which the vehicle is subjected to sinusoidal pitching moments of various amplitudes and frequencies.

Figure 11 shows the level of the response of the vehicle, i.e. the amplitudes of pitching motion divided by the applied moment. The supervisor controller has a clear peak at about 0.2 Hz. The response level diverges to infinity at that frequency when the amplitude of the moment is more than 1.0 kgfm. On the other hand, the learned neural network does not have such a peak. It can be said that the supervisor controller can make the pitching motion unstable under the existence of disturbances with certain frequencies and amplitudes. The learned neural network does not inherit this defect, mainly because of the two reasons described above.

It can be concluded that the remarkable difference between experimental results by the supervisor fuzzy controller and those by the learned neural network is due to the fact that an improved controller was generated which is more robust to the low-frequency disturbances found in a circulating water tank.

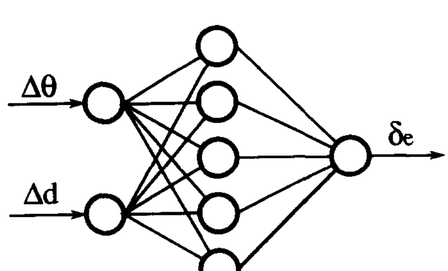


Fig. 5. The controller network for PTEROA60.

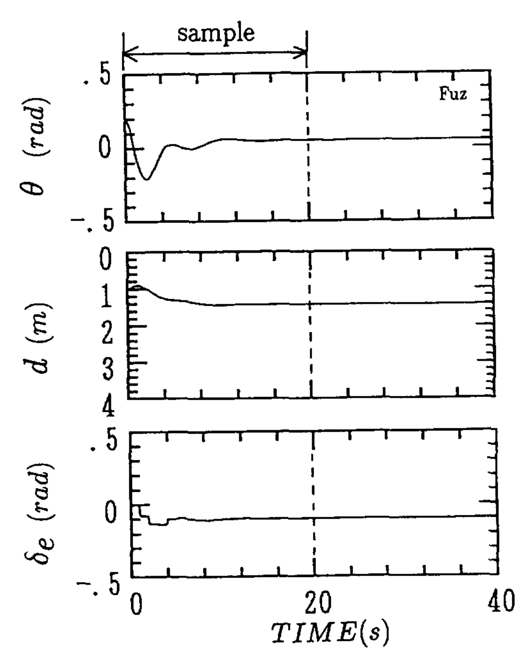


Fig. 7. Numerical simulation of motion of PTEROA60 controlled by the supervisor controller.

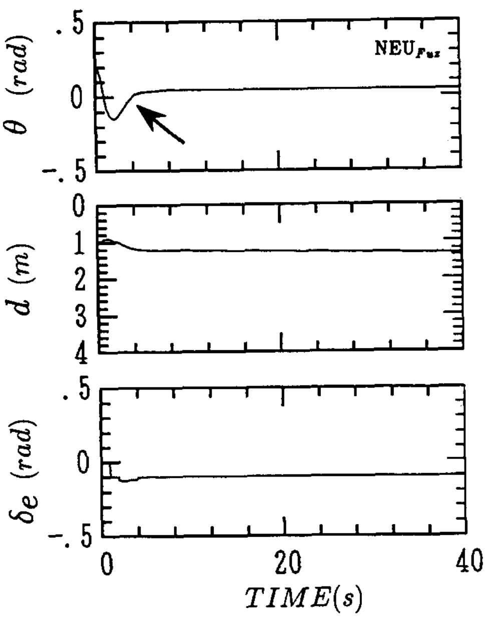


Fig. 8. Numerical simulation of motion of PTEROA60 controlled by the learned neural network.

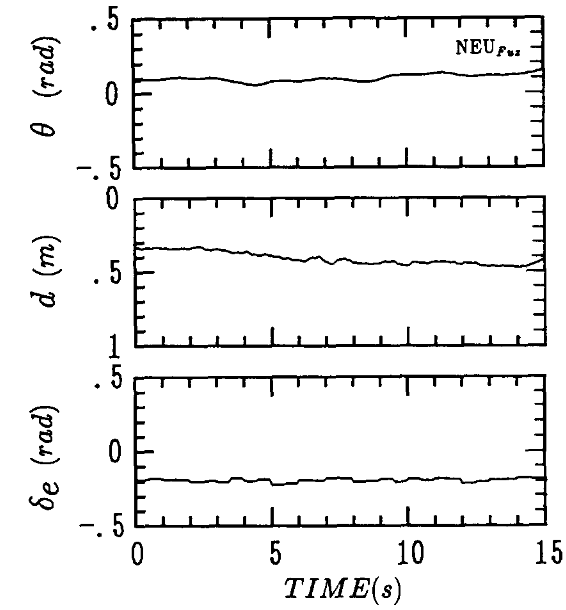


Fig. 10. Experimental results of motion of PTEROA60 controlled by the learned neural network (cruising speed is 0.7 m/s).

## UNSUPERVISED LEARNING SYSTEM

A neural network controller which is only used with a supervised learning system is not a very attractive approach because of the following limitations.

(1) The supervised learning system always needs a supervisor controller.
(2) The neural network cannot adapt to the variation of the controlled object's dynamics caused by changes of environment and a state of motion.
(3) When teaching samples are generated only through computer simulations, the equation of motion of the controlled object must be known accurately.

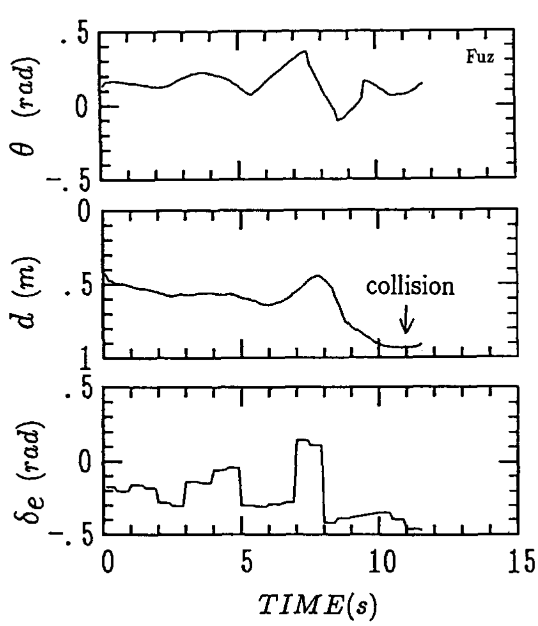


Fig. 9. Experimental results of motion of PTEROA60 controlled by the supervisor controller (cruising speed is 0.7 m/s)—the vehicle collides against the bottom.

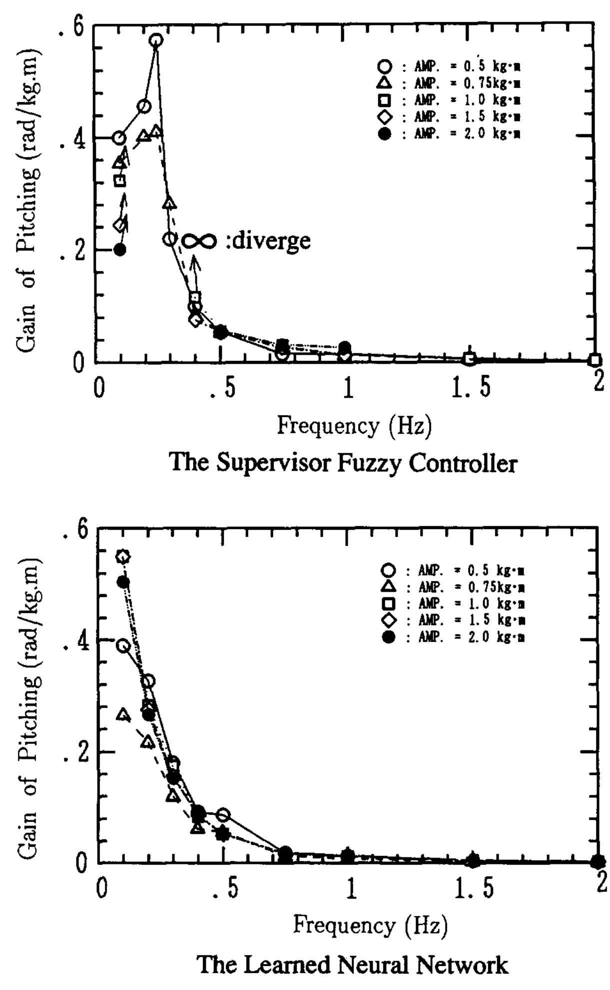


Fig. 11. Response of the control system against disturbance.

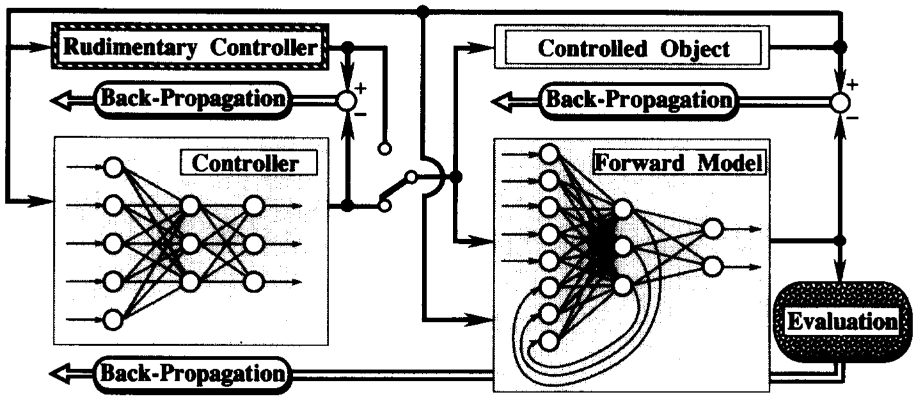


Fig. 12. General architecture of self-organizing neural-net-controller system.

When the dynamics of the controlled object cannot be obtained in advance, or the controlled object is operated in such an environment that the object's dynamics may change, some kind of adaptiveness is required for the control system. This type of control system should be able to know what control actions should be taken even as the dynamics of the controlled object change. The system should have an appropriate adjustment mechanism to keep the controlled object stable for various mission objectives.

With the neural networks, an adaptive control system can be constructed by taking advantage of its inherent learning ability. The SONCS has been proposed as a neural-network-based adaptive control system, and is considered to be an unsupervised learning system because it does not need any supervisor, teaching samples, or reference models once it has been initially trained.

The general architecture of the SONCS is illustrated in Fig. 12. It consists of four subsystems:

(1) Controller Network: controls the controlled object,
(2) Forward Model Network: represents the dynamics of the controlled object,
(3) Evaluation and Adaptation Mechanism: adjusts the controller network and the forward model network, and
(4) Rudimentary Controller: initiates the system.

The basic concept of this system is to adapt the controller network according to backward-propagated signals. These backward-propagated signals are in turn derived by the evaluation of the resultant motion estimated by the forward model network. Functions of each subsystem and the operation of the whole system are described in the following sections.

**Introduction of forward model**

In order to evaluate the results of control actions, it is necessary to establish a subsystem to estimate the controlled object's behavior. A neural network called a "Forward Model Network" (cf. Fig. 13) is introduced to estimate the forward dynamics of the controlled object. The inputs of the network are the state variables of the controlled object and the corresponding control inputs. The outputs are state variables at the next time step, which incude the resulting motions to be evaluated.

When the Connectionist Model has no recurrent connections, it can represent a static mapping from the input patterns to the output. When the input variables completely represent the dynamics of the controlled object, this model can be used as a forward model network. However, it is usually impossible to get all of the input variables. In such a case, the time historical order of the signal patterns should be taken into account. This can be realized by a structure with recurrent connections. There are various types of such connections to which the Error Back-Propagation Method can be applied. Two types of structure were proposed by the authors.[10,11,15]

The first type of the structure includes recurrent connections in the input layer (cf. Fig. 14). Since the output function of the neurons in the input layer is the identity function [cf. eqn (2)], these connections explicitly produce outputs which contain information from the preceding states. For example, the output of the $i$-th neuron in the input layer is calculated as follows:

$$\begin{aligned} O_i(t) &= \mu_i O_i(t-\Delta t) + I_i(t) \\ &= \mu_i^t O_i(0) + \sum_{\tau=0}^{t-1} \mu_i^\tau I_i(t-\tau\Delta t). \end{aligned} \tag{7}$$

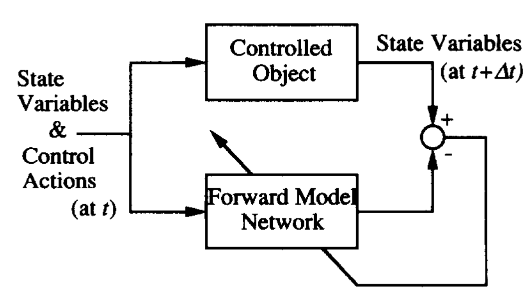


Fig. 13. Learning system of the forward model network.

Here $I_i(t)$, $O_i(t)$ and $\mu_i$ are the input, the output at the time $t$ and the synaptic weight of the recurrent connection, respectively.

The second type includes recurrent connections from the hidden layer to the input layer (cf. Fig. 15). The neurons, which represent the output of the hidden neurons at the previous time step, are added to the input layer. The membrane potential of the $i$-th neuron in the hidden layer is calculated as follows:

$$u_i^2(t) = \sum_j w_{ij}^1 x_j^1(t) + \sum_j w_{ij}^R x_j^2(t - \Delta t), \quad (8)$$

where $w_{ij}^R$s are the synaptic weight of the connection from the added neurons to the neurons in the hidden layer and these from the hidden layer to the input layer are identical. Since $w_{ij}^R$ can be updated in a similar way to the adjustment of the synaptic weights of the normal connection, while $\mu_j$ cannot be, the amount of the effects of the preceding state can be adaptively adjusted according to the properties of each sequence of input variables. Because of this flexibility which the first one does not have, the second type can estimate the output of the modeled object over a wide range of frequency.[15] It should be noted that the first type of structure can be organized as a particular case of the second one.

**Evaluation and adaptation**

Let $E^*$ denote the evaluation potential which is dependent on the desired behavior of the controlled object. The synaptic weights of the controller network are updated so as to reduce $E^*$, i.e. an adaptation of the system. The updating value of the synaptic weights $\Delta w_{ij}^n$ can be calculated as:

$$\Delta w_{ij}^n = -\varepsilon \frac{\partial E^*}{\partial w_{ij}^n}, \quad (9)$$

where $\varepsilon$ is a constant which determines the updating rate.

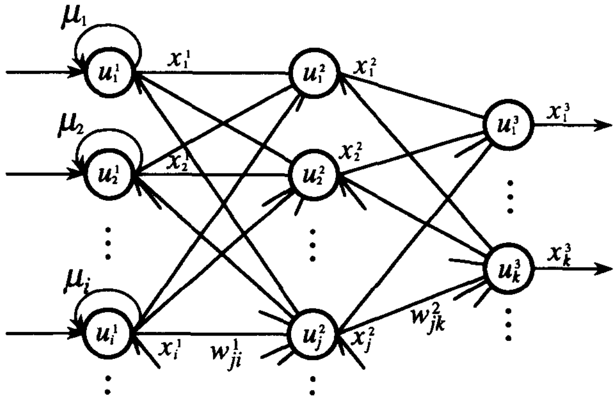

Fig. 14. Forward model network which has recurrent connections in the input layer.

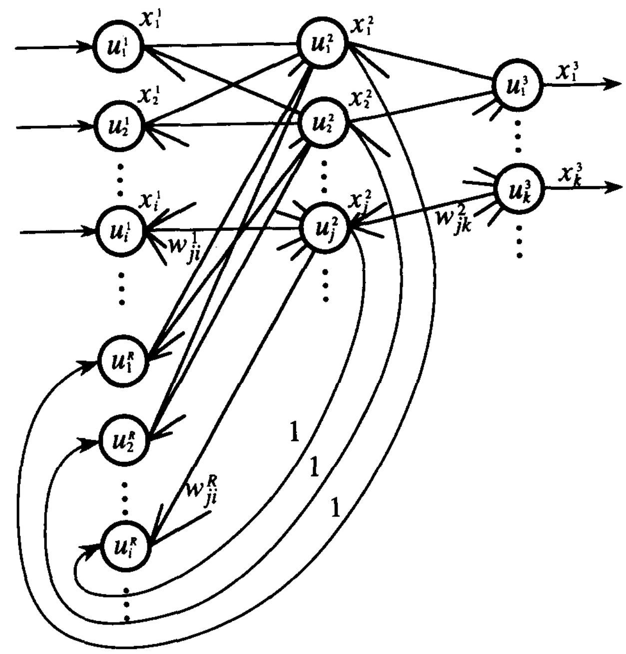

Fig. 15. Forward model network which has recurrent connections from the hidden layer to the input layer.

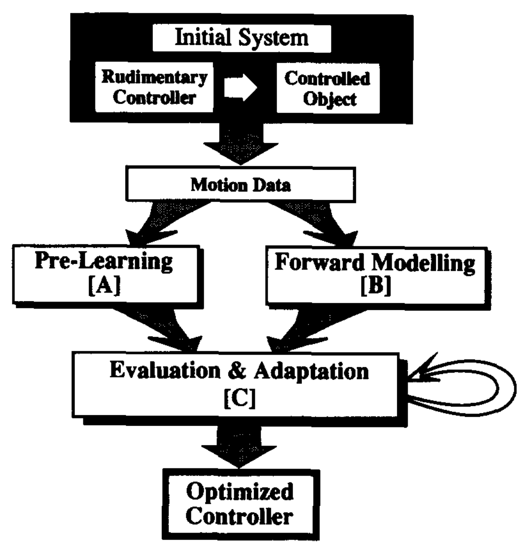


Fig. 16. Organizing process of the controller.

To construct a controller which brings the state vector $\mathbf{x}$ close to the desired state $\mathbf{x}_t$, $E^*$ is given by:

$$E^* = \frac{1}{2} \int^t (\mathbf{x} - \mathbf{x}_t)^T \mathbf{A} (\mathbf{x} - \mathbf{x}_t) \, dt, \quad (10)$$

where $\mathbf{A}$ is a positive definite weighting matrix.

When a controller network and a forward model network are connected with each other, they can be treated as a large single network as shown in Fig. 12. On the assumption that the forward model is sufficiently accurate, their outputs will be equal to the state variables of the controlled object.

By substituting the outputs of the forward model for

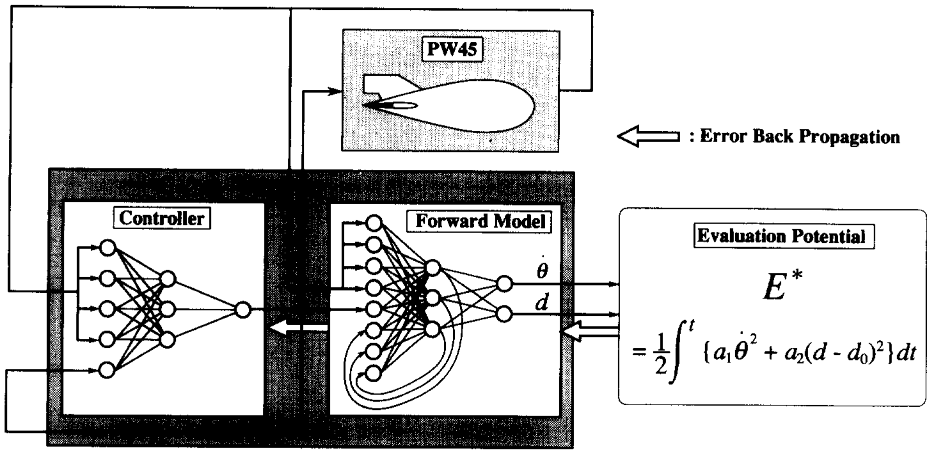


Fig. 17. Controller, forward model and evlauation potential for PW45.

the state vector **x** in eqn (10) and by regarding the desired state vector $\mathbf{x}_t$ as the teaching signal, the evaluation potential is equivalent to the time integral of $E$ in eqn (3) of the network treated as a whole. Therefore, updating of all the synaptic weights in the controller network can be carried out simultaneously, as before,

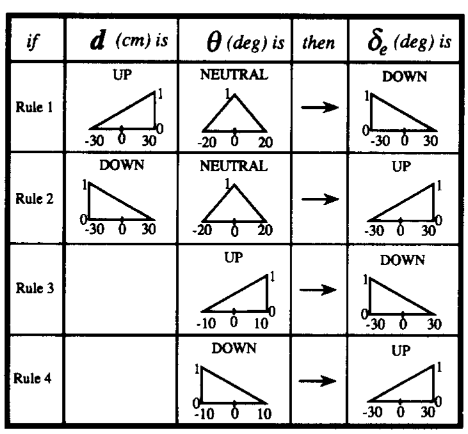


Fig. 18. Fuzzy algorithm for rudimentary control.

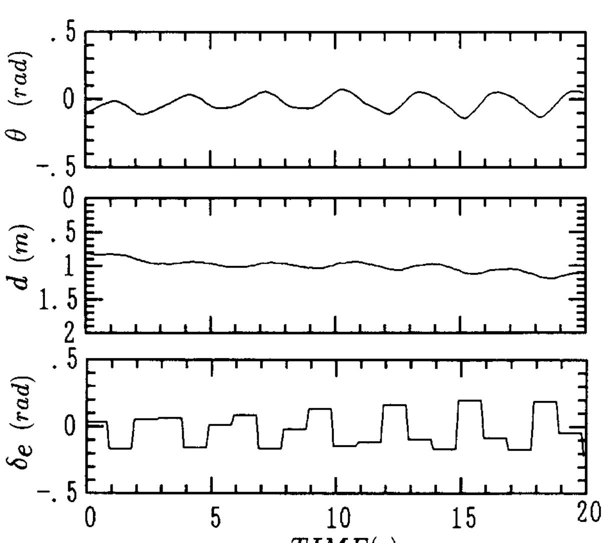


Fig. 19. Experimental results of motion of PW45 controlled by rudimentary fuzzy controller (cruising speed is about 1 m/s).

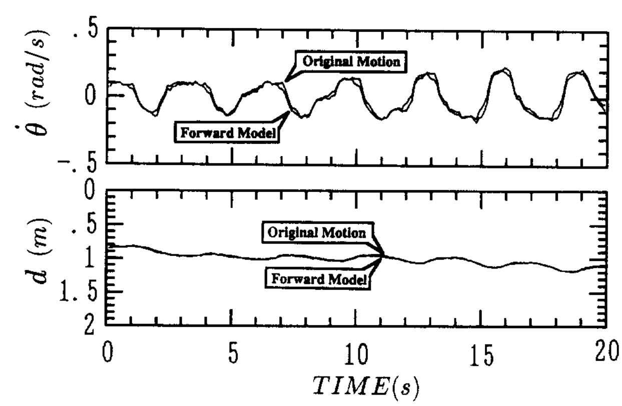


Fig. 20. Estimation of the experimental results by forward model network.

using the Error Back-Propagation Method. The propagated error signals based on the evaluation of motion are calculated through the whole network; however only the synaptic weights of the connections in the controller network should be updated.

When the dynamics of the controlled object are changed, improvement of the forward model network is required. This can be done, independently of the controller's adaptation, by observing the inputs and outputs of the controlled object.

### Organizing process of controller

In the practical environment, the following two schemes for initiation should be included in the system.

(1) Setting the initial values of the synaptic weights of the controller.
(2) Taking the motion data to make the forward model.

For this purpose, an initiation controller is introduced in the SONCS.

During initiation of the controller network and generation of motion data, the state vector of the controlled object should be kept within a reasonably safe range. For this purpose, an initiation controller is used. It is not necessary for this controller to be tuned precisely. This controller is called a "Rudimentary Controller" (cf. Fig. 12).

The organization process of the controller which is initiated by a rudimentary controller is introduced as illustrated in Fig. 16. The followng operations need to occur:

*A. Pre-Learning*: initializing the controller network by learning the control actions from a rudimentary controller,

*B. Forward Modeling*: making the forward model network by learning the motion of the controlled object as controlled by a rudimentary controller, and

*C. Evaluation and Adaptation*: simultaneously adapting the controller network according to the evaluation of motion and improving the forward model network.

## SONCS FOR THE SMALL TEST-BED PW45

The SONCS is applied to the longitudinal motion control of PW45[15] which is a small untethered test-bed (7 kgf in dry weight and 45 cm in length) for investigation of the control system for the PTEROA150 Vehicle. The structure of the SONCS for PW45 is illustrated in Fig. 17. I/O systems of the controller network and the forward model network are shown in Table 1. The objective of the control is to keep the vehicle at the desired depth $d_0$, while keeping the pitching rate $\dot{\theta} = 0$ rad/s (cf. Fig. 4). The following procedures are carried out on the PW45 while it is actually swimming in a tank.

Table 1. I/O system of the neural networks in the SONCS for PW45

| Modules | Controller | Forward model |
|---|---|---|
| Input | $\theta(t)$ $\dot{\theta}(t)$ $d(t)$-$d_0$ $\dot{d}(t)$ $\delta_e(t-1)$ | $\theta(t)$ $\dot{\theta}(t)$ $d(t)$-$d_0$ $\dot{d}(t)$ $\delta_e(t)$ |
| Output | $\delta_e(t)$ | $\dot{\theta}(t+1)$ $d(t+1)$-$d_0$ |
| Number of layers | 3 | 3 |
| Number of neurons | | |
| Input layer | 5 | 8* |
| Hidden layer | 3 | 3 |
| Output layer | 1 | 2 |

* Includes recurrent neurons.

### Rudimentary fuzzy control

A simple fuzzy controller, which is defined by the algorithm shown in Fig. 18, is adopted as a rudimentary controller. Since the function of a fuzzy controller is more understandable than neural networks, it is easier to make the rules for control according to only qualitative knowledge of the vehicle's control actions and motion. The rules are almost the same as those of the supervisor in the previous section, but they have been tuned a little after considering the miserable experimental results (cf. Fig. 9). Figure 19 shows the experimental results of motion of the vehicle controlled by this rudimentary fuzzy controller.

### Pre-learning of controller

The learning steps are executed 500 times on the data set derived from measurements taken while the vehicle is controlled by the rudimentary controller. The data set consists of 20 sets of measurements and their numerical complements in random order. It was found that 500 iterations were required for the neural network controller and that without the use of the numerical complements, substantially more iterations are required. The top of Fig. 21 shows the experimental results of motion of the vehicle controlled by this initial controller network. It should be noted that the controller network has no specific objective at this moment and only tries to imitate the control actions of the rudimentary controller.

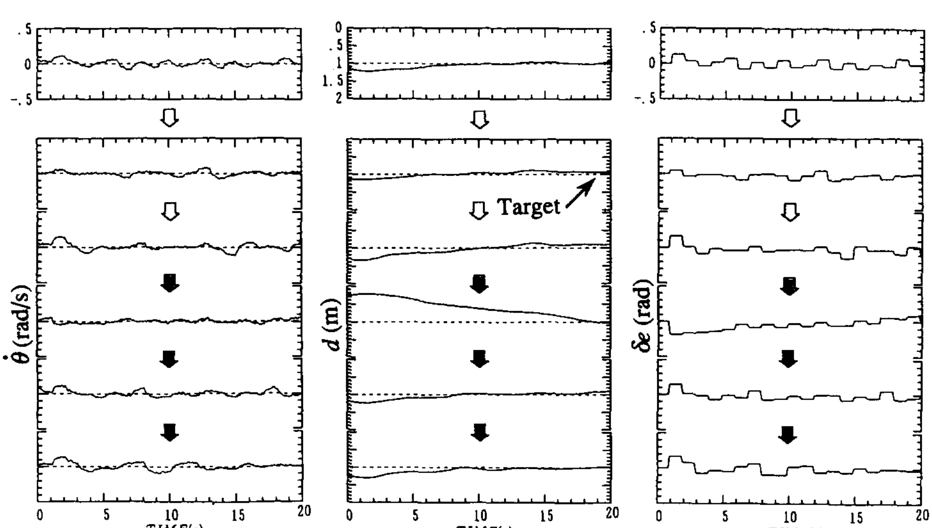


Fig. 21. Adaptation process of the controller network (cruising speed is about 1 m/s).

**Forward modeling**

The forward model network which has recurrent connections from the hidden layer to the input layer is selected for the estimation of the motion of PW45 as illustrated in Fig. 17. The inputs of the forward model are $\theta$, $\dot{\theta}$, $d$, $\dot{d}$ and $\delta_e$ and the outputs are $\dot{\theta}$ and $d$ which are to be evaluated according to the objective of the system. Three thousand times of learning are carried out with the motion data for 20 s (cf. Fig. 19). From Fig. 20, it can be said that the learned forward model network can proficiently estimate the PW45's motion.

**Evaluation and adaptation**

The following evaluation potential is introduced for the adaptation of the controller of PW45's horizontal swimming at a desired depth $d_0$.

$$E^* = \frac{1}{2}\int^{t} \{a_1\dot{\theta}^2 + a_2(d-d_0)^2\}\,dt, \qquad (11)$$

where $a_1$ and $a_2$ are weighting parameters on $\dot{\theta}$ and $d$, respectively.

Adaptation of the controller network is carried out 5 times with $d_0 = 1.0$ m. Figure 21 shows the experimental results of motion of PW45 controlled by the updated controller network after each adaptation. It is shown that the controller network is gradually gaining the ability to keep the depth of PW45 at the desired value $d_0$. After 5 times of adaptation, the controller network acquires satisfactory ability to control PW45 horizontally at the depth $d_0 = 1.0$ m.

## CONCLUDING REMARKS

In this paper two types of neural-network-based control systems are introduced and applied to the longitudinal motion control of the test-bed vehicles of PTEROA.

The supervised learning approach is implemented along with a fuzzy controller as a supervisor. It is shown that the advantages of the neural networks, such as flexibility of the I/O selection and saturating characteristics, yield moderation of the control actions and an improvement of the robustness against disturbances.

In order to deal with unknown dynamics of the controlled object and generate an appropriate controller, the unsupervised learning system, which is called "SONCS", is introduced. The SONCS is applied to the control problem of the untethered test-bed vehicle PW45 and has succeeded in generating an appropriate controller for horizontal swimming at a desired depth. Though a large number of iterations are necessary for the pre-learning and the forward modelling, it is shown that the adaptation proceeds quickly to the goal.

A neural network is an attractive tool for realization of a control system which is required to be both autonomous and able to deal with uncertainty of the real world. The SONCS is one realization of the neural-network-based adaptive control system. It is hoped that this is the first step to the establishment of an ideal control system for underwater vehicles, which includes skilled intelligence.

**Acknowledgements**—The free-swimming tests were carried out in the test basin at the Ship Research Institute. The authors would like to express great thanks to Dr Iwao Watanabe for his kind arrangement. We also would like to thank the Central Workshop of the Institute of Industrial Science, the University of Tokyo for their assistance during the construction of the test-bed vehicles.

## REFERENCES

1. McCulloch W. S. and Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**, 115–133 (1943).
2. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. natn. Acad. Sci.* **79**, 2554–2558 (1982).
3. Fukushima K., Miyake S. and Ito T. Neocognitron: a neural network model for a mechanism of visual pattern recognition. *IEEE Trans. Sys. Man Cybern.* **SMC-13**, 826–834 (1983).
4. White H. Economic prediction using neural networks: the case of IBM daily stock returns. *Proc. IEEE Int. Conf. on Neural Networks*, Vol. II, San Diego, pp. 451–459 (1988).
5. Jordan M. I. Serial order: a parallel Distributed approach, ICS report 8604, University of California, San Diego (1986).
6. Kawato M., Uno, Y., Isobe M. and Suzuki R. Hierarchical neural network for voluntary movement with application to robotics. *IEEE Cont. Sys. Mag.* **8**, (3), 8–16 (1988).
7. Nguyen D. H. and Widrow B. Neural networks for self-learning control systems. *IEEE Cont. Sys. Mag.* **10**, (3), 18–23 (1990).
8. Narendra K. S. and Parthasarathy K. Identification and control of dynamical systems using neural networks. *IEEE Trans. N.N.* **1**, (1), 4–27 (1990).
9. Fujii T. and Ura T. Control with neural network for autonomous underwater vehicle. *J. Soc. Naval Arch. Jap.* **166**, 503–511 (1989). (In Japanese.)
10. Fujii T. and Ura T. Development of motion control system for AUV using neural nets. *Proc. IEEE Symp. on AUV Tech.*, Washington DC, pp. 81–86 (1990).
11. Fujii T., Ura T. and Kuroda Y. Development of self-organizing neural-net-controller system and its application to underwater vehicles. *J. Soc. Naval Arch. Jap.* **168**, 275–281 (1990). (In Japanese.)
12. Ura T. Free swimming vehicle PTEROA for deep sea survey. *Proc. ROV'89*, San Diego, pp. 263–268 (1989).
13. Rumelhart D. E., McClelland J. L. and The PDP Research Group. *Parallel Distributed Processing Vol. 1: Foundations*. The MIT Press, Cambridge, MA (1986).
14. Ura T. and Otubo S. Design of unmanned untethered submersibles for quick swimming. *J. Soc. Naval Arch. Jap.* **162**, 117–124 (1987). (In Japanese.)
15. Fujii T., Ura T., Kuroda Y. and Nose Y. Development of self-organizing neural-net-controller system and its application to underwater vehicles: 2nd report. *J. Soc. Naval Arch. Jap.* **169**, 477–486 (1991). (In Japanese.)