A Sensory Network for Fault Tolerance of An Intelligent Robot

Setsuo HASHIMOTO*, Fumio KOJIMA**, and Naoyuki KUBOTA***

*Faculty of Economics, Kyoto Gakuen University
1-1 Ootani, Nanjyo, Sogabe-cho, Kameoka, Kyoto, 621-8551 Japan
email: setsuo-h@kyotogakuen.ac.jp

** Dept. of System Function Science, Graduate School of Science and Technology, Kobe University
1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, JAPAN
email: kojima@cs.kobe-u.ac.jp

***Dept. of Mechanical Engineering, Tokyo Metropolitan University
1-1 Minami-Ohsawa, Hachioji, Tokyo 192-0397, JAPAN
***"Interaction and Intelligence," PRESTO, Japan Science and Technology Corporation (JST)
email: kubota@comp.metro-u.ac.jp

***Abstract* - This paper discusses fault tolerance in perception-based robotics from the viewpoint of ecological psychology. A prediction-based sensory network using neural networks is proposed for detecting a fault in sensing systems. A transformation matrix extracts perceptual information from the sensory inputs that might include fault inputs owing to breakdown. Furthermore, evolutionary computation is applied for the learning of sensory network. We apply the proposed method to a mobile robot. Computer simulations show the proposed method can detect the fault of sensors and can extract perceptual information used for decision making.**

I. INTRODUCTION

Robotic intelligence has been discussed with the development of artificial intelligence (AI) and cognitive science from the point of historical view [1-3]. In classical AI, world modeling, problem solving, task planning, and others have been discussed, but representational and inferential frame problems were pointed out [1]. To avoid the frame problems, R.Brooks proposed subsumption architecture directly using couplings of sensory inputs and action outputs without generating a complete world model [4]. Afterward, behavior-based robotics and evolutionary robotics have been discussed by using intelligent techniques such as fuzzy, neural, and evolutionary computing, as well as reinforcement learning [4-6]. The behavior-based robotics realizes a real-time control based on reactive motions, but it is pointed out that a robot cannot perform sequential or complicated tasks. Therefore, hierarchical methodology has been proposed to solve this problem. The behavior coordination explicitly selects or combines some of reactive motions according to the facing situation. Because this kind of methods requires inference about all behaviors, high computational cost is required. However, we don't consider all possible behaviors, but we consider a few of specific behaviors concerning the facing situation. This indicates a decision making system or an action system is restricted by the perceptual system. Especially, this kind of concept is very important to avoid frame problems of the classical AI. Accordingly, intelligent robotics should be discussed from this kind of viewpoint.

It is very difficult to design robotic intelligence beforehand, because an environment of a robot is unknown. Therefore, the adaptation capability is required for a robot. For example, perception can be performed by interpreting sensory inputs, but we assume sensory inputs are correct and complete. A robot would not work well if some sensors break down or if the sensory inputs are incomplete or inaccurate. On the other hand, the human perceptual system extracts information from incomplete sensory inputs. This kind of discussion has been done in the field of ecological psychology. In ecological psychology, the smallest unit of analysis must be *the perceiving-acting cycle situated in an intentional context* [2,3]. This indicates the human extracts information not from the only current sensory inputs, but from the time-series of sensory inputs and action outputs according to spatiotemporal context.

In this paper, we apply the concept of sensory network (SN) for realizing fault tolerance for the sensing systems of a mobile robot. The sensory inputs are changeable, but inputs from normal sensors can have a specific pattern because robotic motion to its environment also has a specific pattern. We assume a sensor is normal if the sensory inputs can have a specific pattern. Otherwise, we assume a sensor is faulty. Fault detection has been studied so far because of the demands on reliability and safety of technical plants [7]. In general, fault detection is done by using fault-sensitive filters which monitor measurable signals of mathematical models of process based on theoretical techniques. Fault detection is done where the sensory inputs for monitoring are not faulty. In this paper, we propose a fault detection method from sensory inputs and decision making method using perceptual information including incorrect sensory inputs. Furthermore, we apply evolutionary computation for the learning of the perceptual system. Section 2 explains the concept of perception-based robotics, and proposes a prediction-based sensory network, an error recovery method, and learning method of perceptual system. Section 3 shows simulation results of a mobile robot and discusses the effectiveness of the proposed method.

II. Perception-based Robotics

A. A Sensory Network for A Mobile Robot

Behavior-based robotics directly uses the couplings of sensory inputs and action outputs. However, a specific perception of an object depends on the situation in its facing environment. The perceptual system doesn't extract all features of the object, but picks up the specific information of the object according to the spatiotemporal context of the situation. Consequently, the perceptual system does not construct a complete world model, but makes ready beforehand for a next specific perception according to the situation. In addition, the outputs of the action system construct the spatiotemporal context for a specific perception with the dynamics of the environment. Consequently, the perceptual system and action system restrict each other through the interaction with the facing environment. In ecological psychology, this is called *perceiving-acting cycle* [2,3]. We have proposed the concept of perception-based robotics [8,9]. The perception-based robotics emphasizes the importance of a perceptual system for the perceiving-acting cycle. The perception-based robotics is discussed from the viewpoint of information flow (Fig.1).

We consider collision avoiding and target tracing behaviors of a mobile robot shown in Fig.2. The robot has eight range sensors to measure the distance to obstacles (x_i).

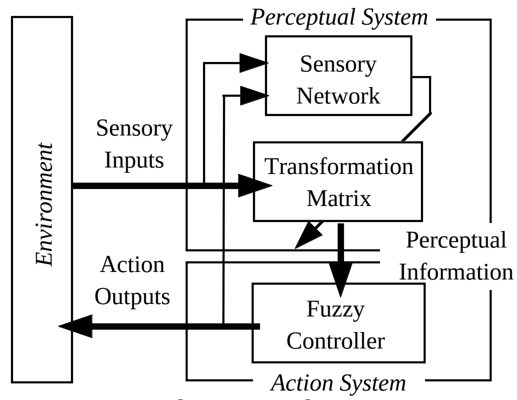


Fig.1 Perceptual system and action system

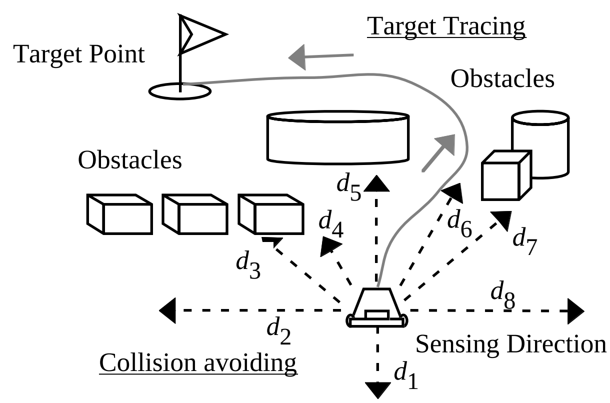


Fig.2 Collision avoiding and target tracing behaviors

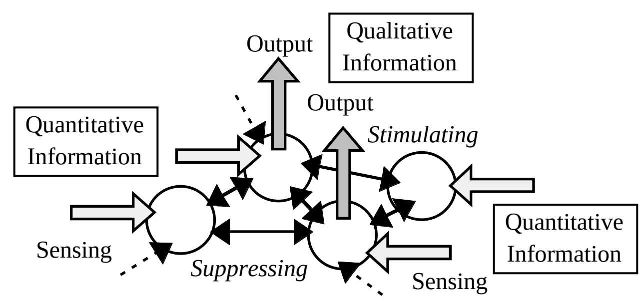


Fig.3 A sensory network (SN) for perceiving environment

Action outputs of the robot are its steering angle (y_1) and its velocity (y_2). A SN is applied to perceive its environment [6]. The robot receives quantitative information of the environment by sensors. Next, the robot extracts qualitative information through interpretation by suppressing or stimulating among sensors (Fig.3). Here the perceptual system reduces distance information into four-dimensional inputs ($\mathbf{p}=(p_1, p_2, \ldots, p_4)^T$) from eight-dimensional inputs ($\mathbf{x}=(x_1, x_2, \ldots, x_8)^T$) by using a following equation;

$$\mathbf{p} = \mathbf{W}\mathbf{x} , \tag{1}$$

$$\mathbf{W} = \begin{pmatrix} 0 & w_{1,2} & w_{1,3} & w_{1,4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_{2,4} & w_{2,5} & w_{2,6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{3,6} & w_{3,7} & w_{3,8} \\ w_{4,1} & w_{4,2} & 0 & 0 & 0 & 0 & 0 & w_{4,8} \end{pmatrix} \quad (2)$$

where $w_{m,n}$ is a weight parameter corresponding to the nth sensor of mth network (Fig.4). Figure 5 shows the structure of a SN based on the relationship between sensory information and perceptual information. The extraction of perceptual information depends on the reliability of sensory inputs. In general, the effective information has a specific pattern of change and persistence, *i.e.*, the change exists in the persistence, while the persistence exists in the change. The sensory information is useful for perceiving its environment if a specific changing pattern exists in the sensory inputs. Therefore, the weight parameter should be large, if the change traces a specific pattern or if the change is large for a specific perception. Accordingly, the flexible perception can be done by updating the above weight parameters according to changing patterns. The extraction of perceptual information depends on the time series of sensory inputs recursively, *i.e.*, the above weight parameters for the perception are self-scaled. In the following, we focus mainly on the reliability of sensory inputs and degree of the change of sensory inputs.

B. Prediction-Based Sensory Network

The predicted values of other sensory inputs can be used for monitoring or detecting the changing patterns of a sensory input, because sensory inputs corresponding to a sensory position can have a specific pattern according to robotic motions and it is difficult to detect the fault of a sensor by itself. We assume a SN composed of three sensory inputs (x_i, x_j, and x_k shown in Fig.6). The predicted distance values from the ith sensory input are defined as $x'_{i,j}$ and $x'_{i,k}$. They are calculated by a following prediction function;

$$f_i(x_i) \rightarrow x'_{i,j},\ x'_{i,k} \,. \quad (3)$$

The error of each prediction is calculated as follows;

$$e_{i,j} = \left| x'_{i,j} - x_j \right| , \quad (4)$$

$$e_{i,k} = \left| x'_{i,k} - x_k \right| . \quad (5)$$

If the ith sensor works well without any problem, the above prediction errors will be small or zero. Otherwise, they will be large. Furthermore, the prediction error of ith sensor from x_j and x_k are also defined as follows;

$$e_{j,i} = \left| x'_{j,i} - x_i \right| , \quad (6)$$

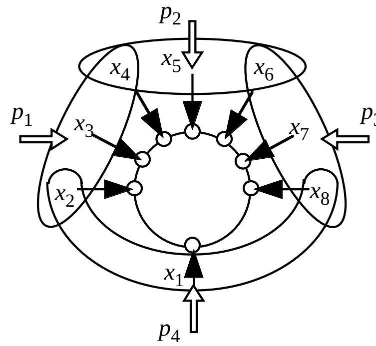


Fig.4 Sensory inputs (x_i) and perceptual inputs (X_j)

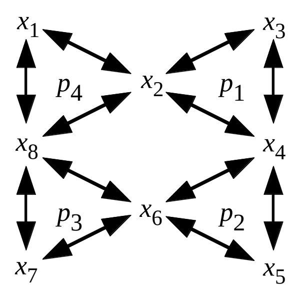


Fig.5 The relationship of perceptual information extracted by a sensory network (SN)

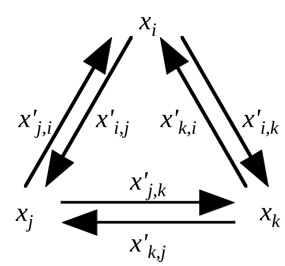


Fig.6 Prediction from each sensory input in a sensory network

$$e_{k,i} = \left| x'_{k,i} - x_i \right| . \quad (7)$$

If the ith sensor breaks down, both of the above prediction errors will be large. This indicates the fault of a sensor can be detected by using the prediction errors. To summarize, if the value of $e_{i,j}$ in eq.(4) is high, we must consider two reasons of the prediction error of $x'_{i,j}$ owing to the fault of the ith sensor and the measurement error of the sensory input of x_j owing to the fault of the jth sensor. Therefore, the weight parameter corresponding to the sensor with relatively high prediction errors should be reduced. If the value of $e_{i,j}$ in eq.(4) is higher than $e_{i,k}$ in eq.(5), the weight parameters are updated as follows;

$$\Delta w_{m,i(i \rightarrow j)} = -\alpha \times e_{i,j} \quad (8)$$

$$\Delta w_{m,j(i)} = -\beta \times e_{i,j} \quad (9)$$

where $\Delta w_{m,i(i\to j)}$ and $\Delta w_{m,j(i)}$ are the updating amounts owing to the prediction error of $x'_{i,j}$ and owing to the measurement error of the jth sensor, respectively. For example, if $e_{i,j} > e_{i,k}$, $e_{j,i} > e_{j,k}$, and $e_{k,i} > e_{k,j}$ when the ith sensor breaks down, then their corresponding updating amounts are $\Delta w_{m,i(i\to j)}$, $\Delta w_{m,j(i)}$, $\Delta w_{m,j(j\to i)}$, $\Delta w_{m,i(j)}$, $\Delta w_{m,k(k\to i)}$, and $\Delta w_{m,i(k)}$, respectively. Therefore, the updating equation with respect to the weight parameter of $w_{m,i}$ is as follows;

$$w'_{m,i} \leftarrow w_{m,i} + \Delta w_{m,i(i\to j)} + \Delta w_{m,i(i\to k)} + \Delta w_{m,i(j)} + \Delta w_{m,i(k)} \quad . \tag{10}$$

Here the number of terms for updating is equal to the number of the arrows with respect to x_i in Fig.6. And then, the weight parameters of the mth SN are normalized as follows;

$$w_{m,i} \leftarrow \frac{w'_{m,i}}{w'_{m,i} + w'_{m,j} + w'_{m,k}} \quad . \tag{11}$$

In this way, if the ith sensor breaks down, the weight parameter is much reduced comparing with other parameters.

Furthermore, the weight parameter is reduced if the change of sensory input is small. The temporally discounted sum of change is calculated as follows;

$$C_i = \sum_{\tau=0}^{\tau_{\max}} \gamma^{\tau} \left| x_i(t-\tau) - x_i(t-\tau-1) \right| \tag{12}$$

where γ is a discount rate; t is a current time step, τ_{max} is the maximal time steps to go back. If this value is smaller than a given threshold, the weight parameter is reduced.

C. Neural Networks for Prediction

Neural networks (NN) have often been used for nonlinear function approximation. NN is applied for predictor in SN, *i.e.*, each NN is used for learning to identify eq.(3) through interaction with the environment. The total number of NN is 12 because each SN includes three predictors (see Fig.5). The output of each neuron is calculated as follows,

$$Y_p^l = S\left(\sum_{q=1}^{N_{l-1}} W_{p,q}^l \cdot Y_q^{l-1} - \theta_p^l \right) \tag{13}$$

where Y_p^l is an output of the pth neuron in the lth layer; S is a Sigmoid function; $W_{p,q}^l$, θ_p^l, and N_l are a weight parameter between the pth neuron of the lth layer and the qth neuron of the (l-1)th layer, threshold of the pth neuron, and the number of neurons of the lth layer, respectively.

NN learns the structural relationship to the sensory inputs of neighboring sensors by using the relationship between sensory inputs and action outputs. The next state of sensory inputs can be predicted by the current sensory inputs and motor outputs like a forward model of a robot. The inputs

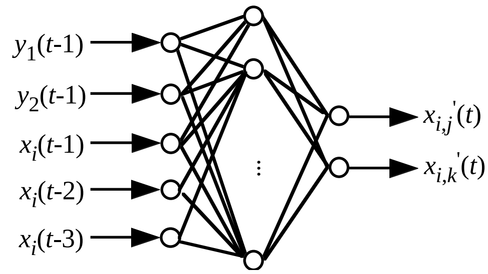


Fig.7 A three-layered neural network for prediction

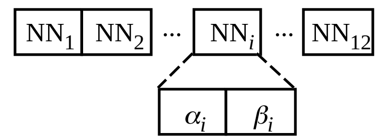


Fig.8 Coding method representing a candidate solution

concerning the ith sensor in the nth SN to a NN are the previous sensory inputs, $x_i(t\text{-}1)$, $x_i(t\text{-}2)$, and $x_i(t\text{-}3)$, action outputs, $y_1(t\text{-}1)$ and $y_2(t\text{-}1)$. The outputs of the NN are the predicted values of other sensory inputs, $x'_{i,j}(t)$ and $x'_{i,k}(t)$ (Fig.7). The learning of each NN is done by the back-propagation learning algorithm [10]. Here, the measured values of the other sensors are used as training data in the learning of each NN.

D. Evolutionary Learning of Perceptual System

Evolutionary computation (EC) is a field of simulating evolution on a computer, and its application is so wide [11, 12]. Especially, the field which uses EC to the adaptation of the robot is called evolutionary robotics [13]. For the learning of the perceptual system, we apply EC.

Candidate solutions are the set of parameters of NNs, α and β, which construct the relationship among sensors in SN (Fig.8). These parameters are the important parameters which determines update amount in eq.(8) and (9). If these parameters are too large, perceptual system will become unstable. On the other hand, if these parameters are too small, adaptation becomes slow. We use the evaluation function to be minimized as follows;

$$E = w_1 \cdot E_{time} + w_2 \cdot E_{distance} \tag{14}$$

where E_{time} and $E_{distance}$ are the time steps and the moving distance required to reach the goal point respectively. w_1 and w_2 are the weight for each evaluation items. Genetic operators are crossover and mutation. The selection and the generation model are based on steady-state genetic algorithm (SSGA). The SSGA simulates the continuous model of the generation, which eliminates and generates a few individuals in a generation

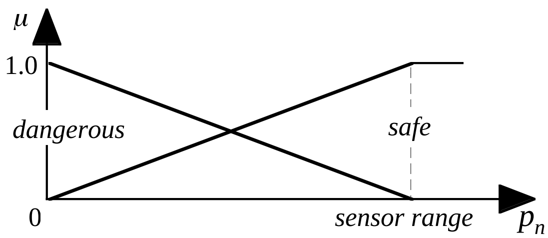


Fig.9 Membership functions

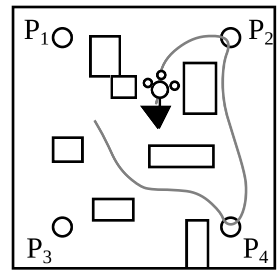


Fig.9 A simulation environment

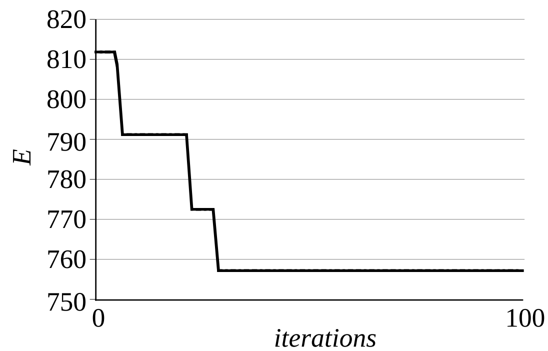


Fig.10 A history of evaluation value (E)

(iteration) [14]. Since the objective of the above evaluation function is minimization, the candidate solution with the maximal value is eliminated in the selection.

E. Action System based on Fuzzy Controller

A behavior of the robot can be represented by using fuzzy rules based on simplified fuzzy inference [10]. The logical structure written by fuzzy rules is easy for humans to understand and to design. In general, a fuzzy if-then rule is described as follows,

If p_1 is $A_{r,1}$ ... and p_M is $A_{r,M}$ **Then** y_1 is $s_{r,1}$... and y_n is $s_{r,O}$

where $A_{r,m}$ and $s_{i,o}$ are is a triangular membership function for the mth input and a singleton for the oth output of the rth rule; M and O are the numbers of inputs and outputs, respectively. Fuzzy inference is generally described by,

$$\mu_{A_{r,m}}(p_m) = \begin{cases} \left(1 - \dfrac{|p_m - a_{r,m}|}{b_{r,m}}\right) & if\,|p_m - a_{r,m}| \le b_{r,m} \\ 0 & otherwise \end{cases} \tag{15}$$

$$\mu_r = \prod_{m=1}^{M} \mu_{A_{r,m}}(p_m) \tag{16}$$

$$y_o = \frac{\sum_{r=1}^{R} \mu_r s_{r,o}}{\sum_{r=1}^{R} \mu_r} \tag{17}$$

where $a_{r,m}$ and $b_{r,m}$ are the central value and the width of the membership function $A_{r,m}$; R is the number of rules.

III. COMPUTER SIMULATIONS

This section shows several computer simulation results of a mobile robot based on perceiving-acting cycles. Here we use two behaviors of target tracing and collision avoiding, and use two linguistic values of "*dangerous*" and "*safe*" for the collision avoiding behavior [6]. The size of the environment is 500*500, when the radius of the robot is depicted as circle is 7. The sensing range is 90. The steering angle is restricted between -30° and 30°. The maximum velocity is 10. Figure 9 shows a simulation environment including seven obstacles and four target points. A robot randomly moves among P_1, P_2, P_3, and P_4. One trial is defined as moves from P_i to P_j. The number of fuzzy rules is 10. The numbers of neurons of input, hidden, and output layers in each NN layers are 5, 10, and 2, respectively. The sensing range is depicted as a broken line. When the mobile robot detects obstacles in the sensing range, it is depicted as a full line.

The breakdown of a sensor occurs at the 11th trial. Until that, every sensor works well. The number of maximal trials is 30. The broken sensor is the 4th one. A random number between 1 and 45 is used as faulty sensory input in sampling.

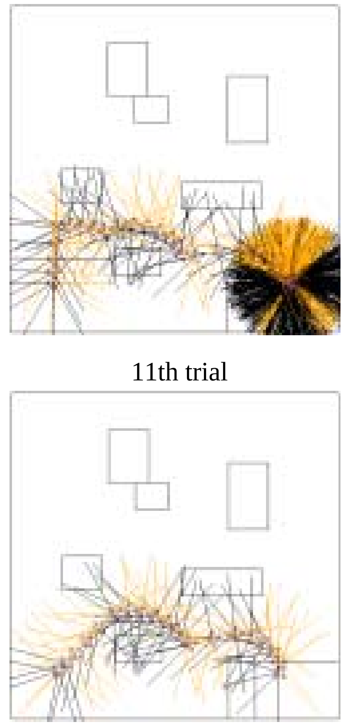
Figure 10 shows a history of evaluation value calculated by eq.(14). Figure 11 shows snapshots of the trajectory of the mobile robot. The mobile robot was able to move from P_4 to P_3 while avoiding collision with obstacles at 11th trial, although the 4th sensor broke down. At the 11th trail, the mobile robot turns around P_4, because the perceptual inputs calculated by the weighted average using the transformation matrix are incorrect owing to the breakdown at fast. Afterward the weight value of the broken sensor fast decreases (Fig.12), because the prediction error concerning the broken sensor is high. In Fig.12, each weight value is plotted every 100 time steps over all trials. Once the weight value of the broken sensor decreases, the prediction of other sensors becomes correct because the mobile robot takes motions before the breakdown.

To summarize, the SN can detect the breakdown according to prediction errors among sensors and the changing history of sensory inputs, and can update weight values for extracting perceptual information.

IV. CONCLUSIONS

This paper proposed a prediction-based sensory network for detecting fault of sensors of a mobile robot. The transformation matrix extracts perceptual information used for decision making. Furthermore, evolutionary computation is applied for the learning of the sensory network. Basically, the degree of correctness in prediction is used for detecting the fault of sensors. Faulty sensory inputs can be detected by using specific patterns of the input-output relationship obtained through interaction with environment. Neural networks are used for learning specific patterns. Computer simulations show the proposed method can detect the fault of sensors and can extract perceptual information used for decision making.

As a future work, we intend to discuss the detail of fault tolerance capability of the proposed method from the mathematical and psychological points of view. In addition, we intend to discuss the relationship between change and persistence of sensory inputs and action outputs in detail. Furthermore, we apply the proposed method to a mobile robot developed by us.



11th trial

18th trial

Fig.11 Trajectories of the mobile robot

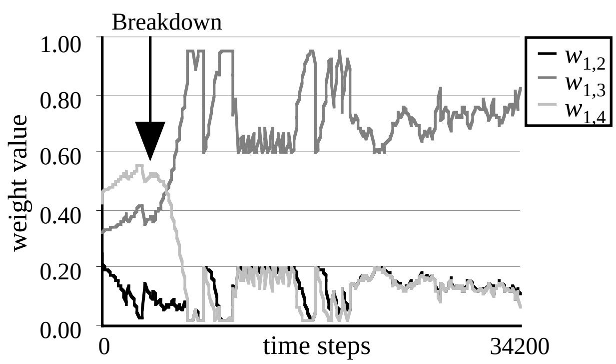


Fig.12 Histories of weight values

REFERENCES

[1] S.J.Russell, P.Norvig. *Artificial Intelligence*, Prentice-Hall, Inc. 1995.

[2] M.T.Turvey and R.E.Shaw. Ecological Foundations of Cognition I. Symmetry and Specificity of Animal-Environment Systems, *Journal of Consciousness Studies*, Vol. 6, No. 11-12, pp. 95-110, 1999.

[3] R.E.Shaw and M.T.Turvey. Ecological Foundations of Cognition II. Degree of Freedom and Conserved Quantities in Animal-Environment Systems, *Journal of Consciousness Studies*, Vol. 6, No. 11-12, pp. 111-123, 1999.

[4] R.Brooks. A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, Vol. 2, No. 1, pp. 14-23, 1986.

[5] J.Tani. Model-Based Learning for Mobile Robot Navigation from the Dynamical Systems Perspective, *IEEE Trans. on Systems, Man, And Cybernetics, Part B: Cybernetics*, Vol. 26, No. 3, pp. 421-436, 1996.

[6] T.Fukuda and N.Kubota. An Intelligent Robotic System Based on A Fuzzy Approach, *Proceedings of The IEEE*, Vol. 87, No. 9, pp. 1448-1470, 1999.

[7] R.Isermann. Process fault detection based on modeling and estimation methods - a survey, *Automatica*, Vol. 20, pp. 387-404, 1984.

[8] S.Hashimoto, N.Kubota, F.Kojima and T.Fukuda. Genetic Programming for Perception-Based Robotics, In *Proc. of The 4th Asian Fuzzy Systems Symposium*, pp. 674-679, 2000.

[9] N.Kubota, S.Hashimoto, and F.Kojima. A Sensory Network for Perception-Based Robotics Using Neural Networks, *Proceedings of the 2003 International Joint Conference on Neural Networks (IJCNN2003)*, pp.3151-3156, 2003.

[10] J.-S.R.Jang, C.-T.Sun, E.Mizutani. *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, Inc., 1997.

[11] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Welsey, 1989.

[12] D.B.Fogel. *Evolutionary Computation*. IEEE Press, 1995.

[13] S. Nolfi, D. Floreano. *Evolutionary Robotics*, The MIT Press, 2000.

[14] G.Syswerda. A Study on Reproduction in Generational and Steady-State Genetic Algorithms, In: *Foundations of Genetic Algorithms*, San Mateo, Morgan Kaufmann Publishers, Inc, 1991.