'88 KACC 1988. 10. 21~22

A METHOD TO CONSTRUCT SELF ORGANIZING SYSTEM IN ROBOTIC APPLICATION

° Hiroshi Noda, Hideki Hashimoto and Fumio Harashima

Institute of Industrial Science
The University of Tokyo

22-1, Roppongi, 7-chome, Minato-ku, Tokyo 106, JAPAN
(TEL) 03-402-6231 Ext. 2360, (Fax) 03-423-1484

Abstract : The author propose a method to realize a self organization in the artificial system. In self organizing system, sub-systems are not constructed as functional parts of the system but cooperate with one another to realize the total system. Each subsystem obtains the local purpose from the global purpose by learning. This function is realized by using a neural network. The validity of this method is confirmed by some simulations.

1. Introduction

Usually a large artificial system is composed of many sub-systems. A sub-system is constructed as a functional part of the artificial system, and those sub-systems are controlled by a master controller. The artificial system has to have more subsystems in proportion to complexities of its function. In that case, it is difficult to control the artificial system, because complexities and uncertainties of dividing it to sub-systems are increasing. On the other hand, from a concept of "Autonomous Distributed System" [1], the total system is achieved as the result of actions of sub-systems which are not functional divisions. If we can implement the ability of cooperation within sub-systems, the total system is organized autonomously. Each sub-system must have some intelligence, observe environments, and communicate with others to cooperate one another. As a result of cooperation of those sub-systems, the (global) purpose can be achieved. Recently, this distributed system can be easily constructed owing to the remarkable progress of micro-electronics.

In this paper, the process that a total system is organized by cooperation of sub-systems is called "Self organization". The example of self organization can be easily found in the nature --- fluid dynamics, lasers (coherent oscillations), macroscopic patterns in chemistry, and so on. However, the self organization in the nature is a little different from that in artificial systems. This will be discussed in a following section. We use a word "An artificial self organizing system" to indicate an artificial system composed of sub-systems with cooperation ability.

The outline of this paper is as follows : First, an artificial self organizing system is described. Secondly, a neural network to construct an inverse system is presented and importance to solve an inverse problem is discussed. This problem can be solved by using learning capability of a neural network. Thirdly, a method to realize artificial self organization is proposed, and then this system is applied to a multi-linked manipulator. Lastly, simulation results are shown.

2. Self organization in control systems

Self organization is a very general concept. The self organization in artificial systems is a little different from that in the nature. The cooperation of sub-systems brings about spatial, temporal or functional structures on macroscopic scales [2][3]. In the nature, there are quite a large number of sub-systems, each of which functions is very primitive. On the other hand, in an artificial system (Fig. 1), the number of sub-systems is not so large but each sub-system must have some intelligence and ability to communicate with each others.

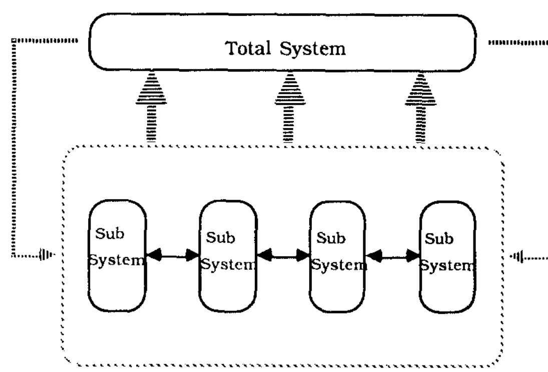


Fig 1. Artificial self organizing system

A fundamental difference between an artificial self organizing system and natural self organizing system is that a purpose is given by human being in an artificial self organizing system. In natural self organization, any intents or purpose of ours are not considered at all, there is only a natural law which we haven't found. In artificial self organization, there is a difficult problem : how we can decide the characteristics of the sub-systems to realize the total system which fits global purpose ? We do not know how an action of a sub-system influences the total system. The action of a sub-system to achieve the global purpose is not stationary, and that, it depends on the action of others. Therefore, it is nearly impossible to implement an algorithm to the sub-systems in advance to satisfy the above requirements. It is necessary that each sub-system gets the cooperation ability by "Learning" to achieve the global purpose . In this paper, a part of this function is realized by using a neural network. (It is discussed in next section.)

For the purpose of having a high-level autonomy, each sub-system has to extract its local purpose from the global purpose given by human being in advance. However, only this ability is not enough to achieve the global purpose, because it is impossible for sub-systems to achieve their local purposes perfectly. Therefore it is necessary for sub-systems to cooperate with others. On the other hand, if the autonomy of each sub-system is primitive, a conflict of communication is occurred because of the frequency of the communication within sub-systems. Hence two important abilities are necessary in artificial self organizing systems, "autonomy" and "cooperation". In this paper, autonomy is defined as the ability to decide the local purpose by observing the purpose of the total system, and cooperation is defined as the ability to cooperate with one another to achieve the global purpose by using the local or global information.

3. Neural Network to Construct an Inverse System

Each sub-system which has an autonomy has to decide its own action (local purpose) to achieve the global purpose. That is "an inverse problem", which means a decomposition of the global purpose into local purposes. A system for solving an inverse problem is called an "inverse system". Usually, an inverse system has the same structure as the original system has. But our neural network system does not have to have the same structure, because it can construct an inverse system by using only the relation between inputs and outputs, i.e.,"Learning". This neural network is independent of the structure of the original system.

Kawato et al. [4] solve the inverse dynamics by using a neural network. However, the inverse system has the same structure as the system has, and the parameters are only calculated by the neural network.

3.1. Configuration of neural network

We use a PDP network [5] , which has a learning capability to map a set of input patterns to a set of output patterns. Pao [6] discuss about the ability of a PDP network to discover the basis for decision and control. The learning algorithm used in this network is called "Back Propagation".

The system is described as a multi-inputs multi-outputs system (MIMO) , but the neural network does not have the same structure as the system has. The neural network is illustrated in Fig. 2.

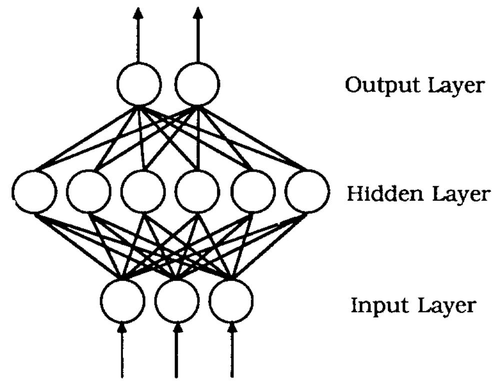


Fig. 2 PDP neural network

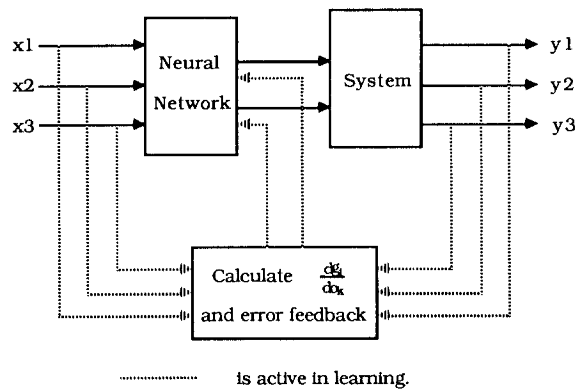


Fig. 3 Total configuration of the inverse system

When it is possible to set an energy function and solve an optimizing problem, a Hopfield network [7][8] is useful. But, in this paper, the solution of an inverse problem must be obtained by "learning". Therefore we use a PDP neural network. This network is independent of every kinds of inverse problems . (A Hopfield network is dependent on problems) .

3.2 Learning --- Convergence ---

A learning rule is as follows:
The system is described as

$$\text{output of the system} := g_i(o_1,o_2,\ldots,o_{k_{max}}) \qquad i = 1,2,\ldots,i_{max} \tag{1}$$

o_k : output of neural network (input to the system)
k_{max} : the number of inputs
i_{max} : the number of outputs .

The number of inputs (outputs) of the neural network is equal to that of outputs (inputs) of the system. Fig. 3 shows the total configuration of the inverse system (This system has two inputs and three outputs). This aim is that the output data y_i coincides with the input data x_i . The input data x_i indicate the desired value and also plays a part of the teaching signal. The error, x_i - y_i , is used for learning .

The neural network is composed of three layers. If o_i are the outputs of neurons in layer i (input layer), the total input to a neuron in layer j (hidden layer) is

$$net_j = \sum_i w_{ij} o_i \tag{2}$$

and output of a neuron in layer j is

$$o_j = f(net_j) \tag{3}$$

where f is a logistic function which is represented by the relation

$$f(x) = \frac{1}{1+\exp(-x)} \tag{4}$$

If o_k are the outputs of neurons in layer k (output layer), the energy function is defined

$$E = \sum_i (d_i - g_i(o_1, o_2, \ldots, o_{k_{max}}))^2 \tag{5}$$

d_i : target input for i-th component of the output pattern

We get the learning rule from the condition : $\dot{E} < 0$

$$\dot{E} = -2\sum_i (d_i - g_i(o_1, o_2, \ldots, o_{k_{max}}))\frac{\partial g_i}{\partial t} \tag{6}$$

$$\begin{aligned}\frac{\partial g_i}{\partial t} &= \sum_k \frac{\partial g_i}{\partial o_k}\frac{\partial o_k}{\partial t} \\ &= \sum_k \frac{\partial g_i}{\partial o_k} f'(net_k)\frac{\partial net_k}{\partial t} \\ &= \sum_k \frac{\partial g_i}{\partial o_k} f'(net_k)\sum_j \left(\frac{\partial w_{jk}}{\partial t} o_j + w_{jk}\frac{\partial o_j}{\partial t}\right) \\ &= \sum_k \frac{\partial g_i}{\partial o_k} f'(net_k)\sum_j \left(\frac{\partial w_{jk}}{\partial t} o_j + w_{jk} f'(net_j)\sum_i \frac{\partial w_{ij}}{\partial t} o_i\right)\end{aligned} \tag{7}$$

Thus the following learning rule is obtained .

$$\frac{\partial w_{jk}}{\partial t} = \eta \delta_k o_j \tag{8}$$

$$\frac{\partial w_{ij}}{\partial t} = \eta\left(\sum_k \delta_k w_{jk}\right) f'(net_j) o_i \tag{9}$$

$$\delta_k = \sum_i (d_i - g_i(o_1, o_2, \ldots, o_{k_{max}}))\, f'(net_k)\frac{\partial g_i}{\partial o_k} \tag{10}$$

η : The learning rate parameter

$\frac{\partial g_i}{\partial o_k}$ is calculated approximately as follows :

$$\frac{\partial g_i}{\partial o_k}[n+1] \cong \frac{g_i(o_1[n+1], \ldots, o_k[n+1], \ldots) - g_i(o_1[n+1], \ldots, o_k[n], \ldots)}{o_k[n+1] - o_k[n]} \tag{11}$$

This neural network can solve some inverse problems, keep their solutions, and interpolate the other points. Thus an inverse system is constructed in the neural network.

3.3 Simulation results

The solutions of the inverse kinematics of a Stanford manipulator are shown here. The hand of this manipulator is illustrated in Fig. 4 . **p** is the position vector of the hand, **n** , **s** , **o** are the unit normal, unit slide, and unit approach vectors of the hand respectively. The outputs of the system is represented by (**p**,**n**,**s**,**o**), and inputs of the system is represented by ($\theta_0, \theta_1, \theta_2, d_3, \theta_4, \theta_5$). ($d_2$ is constant : $d_2 = 0.5$). The result of solution (after 20 presentations of each pattern) is shown in Table 1.

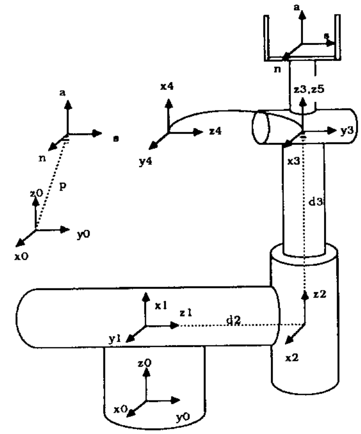


Fig.4 Stanford Manipulator

	Desired	Obtaind
n	(0.5774,0.5774,0.5774)	(0.5628,0.6075,0.5606)
s	(0.40825,- 0.8165,0.4082)	(0.3991,- 0.9097,0.4301)
a	(0.7071, 0 ,- 0.7071)	(0.7005,- 0.0016,-0.7077)
p	(0.7, 0.4 , 0.5)	(0.6267,0.4259,0.5120)

Solution : number of neuron : 45

$\theta_0 = -104.5°$ $\theta_1 = -48.0°$ $\theta_2 = -46.8°$

$\theta_4 = -110.1°$ $\theta_5 = -12.6°$ $d_3 = 0.767$

Table 1 Solution of the Inverse Kinematics

4. Construction of a Self-organizing System

The construction of the sub-systems is shown in Fig. 5 and the total system is shown in Fig. 6 . Each sub-system has a global brain and a local brain. The global brain has a function to extract the local reference (purpose) to achieve a given global purpose. In other words , the global brain solves an inverse problem. In this system, the global brain is constructed by the neural network. The global brain can refer to and analyze the global purpose to extract the local reference. Each sub-system acts on this reference autonomously. If each sub-system has a perfect model of the total system or knows its state perfectly, the global purpose is accomplished only by this mechanism. But, as a matter of fact, it is impossible because there are more uncertainties in proportion to the complexities of the global purpose. Therefore co-operation of the sub-systems is necessary to compensate for such uncertainties. The local brain

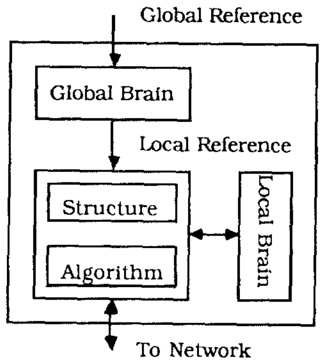


Fig.5 Structure of sub-system

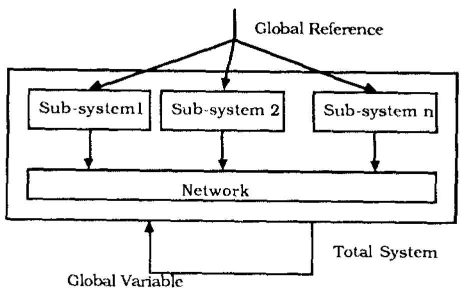


Fig. 6 Realization of total system

changes the structure or the algorithm in the sub-system according to the environment . That is, the global brain has the ability of "autonomy" and the local brain has the ability of "cooperation".

The global purpose in the total system is achieved by both autonomy and cooperation of sub-systems. When the cooperation ability is lost in sub-systems, the influence is dependent on the efficiency of the ability of autonomy.

Sub-systems are connected with each other through the network. In a self organizing system, communication is restricted and each sub-system has some intelligence to make up for the lack of information.

5. An application for a Multi-linked Robotic Manipulator

The geometry of a robotic manipulator with three degrees of freedom is illustrated in Fig. 7 The global purpose is to move the end effector towards a target point. Fig. 8 shows the configuration of the sub-systems applied to the multi-linked robotic arm. The i-th sub-system is composed of a global brain, a local brain, a motor, and a PD controller.

The task of the global brain is to generate the desired angle θ_{di} in each link(sub-system). Each sub-system tries to adjust θ_i to the desired degree θ_{di}, which is the output of the global brain. However some steady errors are left owing to the ambiguity of θ_d and the ignorance of the influence from other links. The local brain obtains the information of other links to compensate for such error. The function of the local brain is constructed automatically through "learning" in a natural self organizing system. In this paper, however, the local brain is implemented in advance in an artificial self organizing system. The local brain for this application is described by the following algorithm.

err = $(x_e - x_d)^2 + (y_e - y_d)^2$;
if (err > ε) {
  $T_i = K_{1i}(\theta_i - \theta_{id}) + K_{2i}\dot{\theta}_i$;
}
else {
  vector_x = $(x_e - x_d,\ y_e - y_d)$;
  vector_y = $(x_{n+1} - x_n,\ y_{n+1} - y_n)$;
  direction = **vector_x** X **vector_y**
  $T_i = K_{1i}(\theta_i - \theta_{id}) + K_{2i}\ \theta_i$
    $+ K_{3i}\ err\ sgn(direction)$;
}

T_i : torque for the *i-th* motor
K_{1i}, K_{2i}, K_{3i} : constant
X : outer product

An algorithm for cooperation

The characteristics of this algorithm are divided into two parts. One part is available in the case that error "err" is larger than ε. Each sub-system can decide its own action by using the information, i.e. an angle of itself θ_i and the desired position θ_{id} , which is output of the global brain. The end effector approaches the neighborhood of the desired point. The other part is available in the case that "err" is smaller than ε. When the position of the end effector reaches within the neighborhood of the desired point, each sub-system acts on information of the position of itself (x_i , y_i) from the external sensor, such as an image sensor. Though each sub-system uses the information about the position of the end effector and desired positions, the other information which the i-th sub-system can obtain from the external sensor is limited to that of only the (i+1) th sub-system. This algorithm has a locality in this point.

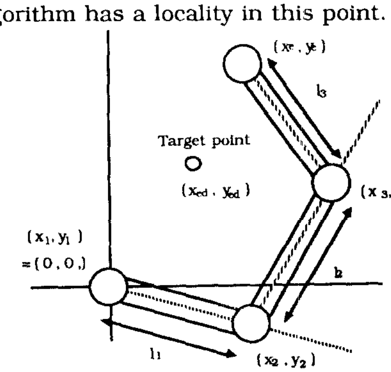


Fig. 7 Robotic manipulator with three degrees of freedom

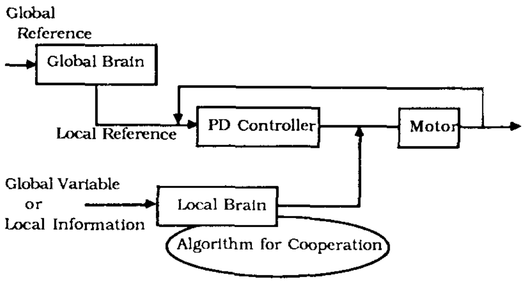


Fig.8 Construction of sub-system

6. Simulation Results

Simulation results for the robotic manipulator with three degrees of freedom are shown in this section. First, the authors show simulation results for learning in global brain. Secondly, simulation results for point to point motion are shown.

6.1 Learning in global brain

A purpose of the global brain is to solve an inverse kinematics. The kinematic equations for the manipulator with three degrees of freedom are

$$x_e = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \tag{12}$$

$$y_e = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \tag{13}$$

x_e , y_e : position of the end effector
x_{ed} , y_{ed} : desired position of the end effector
θ_i : an angle of the i-th link
l_i : a length of the i-th link.

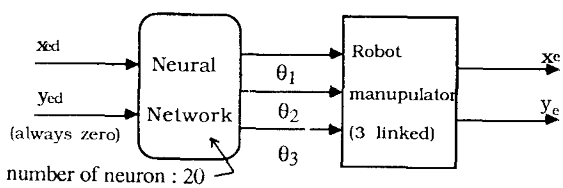


Fig. 9 Configuration of the inverse system for the 3-linked manipulator

x_{ed}	0	0.4	0.8	1.2	1.6	2.0	2.4	2.8
y_{ed}	0	0	0	0	0	0	0	0

Table 2 Training patterns

The configuration of the inverse system is illustrated in Fig.9 . The training patterns are shown in Table 2. The angle of the i-th link ,θ_i, can be obtained as follows :

$$\theta_i = (2o_i - 1)\pi \quad \text{for } i = 1, 2, 3 \tag{14}$$

where o_i (i = 1,2,3) is the i-th output of the neural network ($0 < o_i < 1$). The response of the energy is shown in Fig. 10. The convergence of the learning is quite slow because of the redundancy of the system. To make the convergence fast, we have to give the features of the desired solution to the neural network in advance. There are various patterns to move the end effector to the target point. The typical patterns are shown in Fig. 11. The pattern no.3 is regarded as the desirable pattern in this simulation. To implove the convergence of learning, we let two patterns(shown in Fig.12) learn in advance in neural network. It is no problem how long it takes in the pre-learning because the number of the pre-learning patterns is only two. The response of the energy during learning (not pre-learning) is shown in Fig. 13 . Fig. 14 (a) and (b) show the relation between x_{ed} and x_e,y_e after pre-learning and after learning respectively.

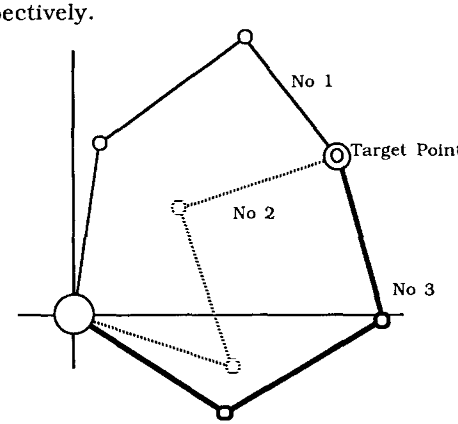


Fig. 11 Typical patterns to move the end effector to the target point

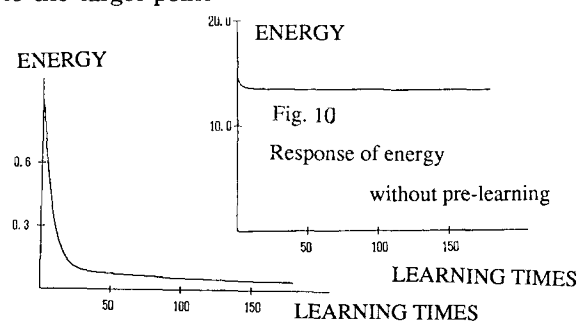


Fig. 10 Response of energy without pre-learning

Fig. 13 Response of energy with pre-learning

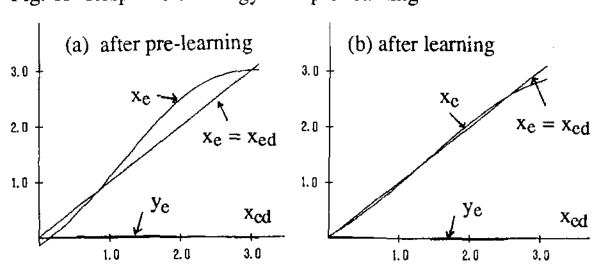


Fig. 14 Relation between x_{ed} and x_e,y_e

Case 1)

Case 2)

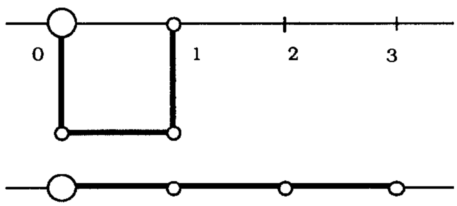


Fig. 12 Training pattern for pre-learning

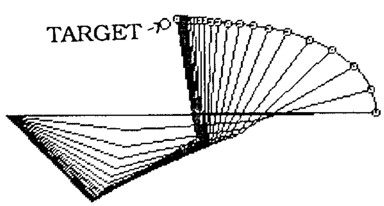
5.2 Point to point (PTP) motion

The purpose of our control system is to move the end effector to the desired point. The desired point is given as (θ_d , r_d). In this simulation, new θ_1 is obtained as follows :

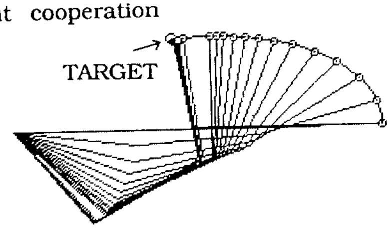
$$\theta_1 := \theta_1 \text{ (output of the global brain in the above simulation)} + \theta_d.$$

The motion of the end effector is shown in Fig. 15 and Fig. 16. In each figures, figure (a) shows the result without using the algorithm for cooperation, and figure (b) shows the result using it. Fig. 15 (a) indicates that the objective cannot be achieved because of the uncertainty of the outputs of the global brain. Fig. 16 (a) also indicates that the speed to achieve the objective is quite slow though the objective is nearly achieved at the end. In both cases, the achievement of the objective is improved by using the algorithm for cooperation. These results are shown in Fig. 15 (b) and Fig. 16 (b).

Fig. 15

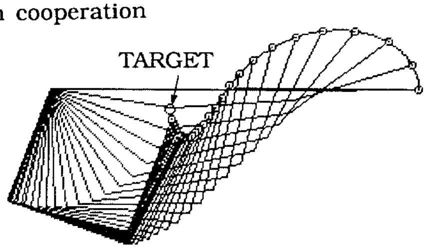


(a) without cooperation

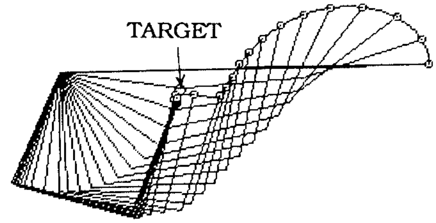


(b) with cooperation

Fig. 16



(a) without cooperation



(b) with cooperation

7. Conclusion

A method to construct an artificial self organizing system is presented. Each sub-system has an autonomy and a cooperation ability. Each sub-system extracts a local purpose from a global purpose to act autonomously. This function is realized partially by using the neural network which the authors presented.

The validity of this approach is confirmed by simulations of an application for robotic manipulator. This approach is effective because this artificial self organizing system decomposes the global purpose into local purposes automatically by learning.

We are planning to apply this approach for a more complex system, i.e. a dynamical system, and to obtain the strategy for cooperation (which function belongs to the local brain) by learning.

8. References

[1] H. Ihara and K. Mori, "Autonomous Decentralized Computer", Control Systems", IEEE Computer, 57/66, Aug., 1984

[2] H. Harken, "Synergetics, An Introduction", Springer-Verlag, 1977

[3] H. Harken, "Advanced Synergetics", Springer-Verlag, 1984

[4] M. Kawato, K. Fukukawa & R. Suzuki, " A Hierarchical Neural-Network Model for Control and Learning of Voluntary Movement", Biol.Cybern., 1987

[5] D. E. Rumelhart et al., "Parallel Distributed Processing", Bradford Books, 1986

[6] Yoh-Han-Pao, "A Connectionist-net Approach to Autonomous Machine Learning of Effective Process Control Strategies", The International Conference on the Manufacturing Science and Technology of the Future, Cambridge,MA,June 3-5,1987.

[7] J. J. Hopfield and D.W.Tank, "Neurons with Graded Response Have Collective Computational Properties like those of Two State Neurons", Proc. Nat. Acad. Aci. USA, May 1984 Biophysics

[8] J. J. Hopfield and D. W. Tank, "Neural Computation of Decisions in Optimization Problems", Biol. Cybern. 52, 141/152 1985