**Proceedings of the 13th
KACC, October 1998**

# Artificial Neural Network and Application in Temperature Control System

**Masanori Sugisaka and Zhijun Liu**
Department of Electrical and Electronic Engineering,
Faculty of Engineering, Oita University.
700 Dannoharu, Oita Shi, 870-1192, JAPAN.
Tel: +81-97-554-7831
Fax: +81-97-554-7841
E-mail: msugi@cc.oita-u.ac.jp, zjliu@cc.oita-u.jp

**Abstract**

In this paper, we implemented the neuro-computer called MY-NEUPOWER in our research to carry out the artificial neural networks (ANN) calculating. An application software was developed based on a neural network using back-propagation (BP) algorithm under the UNIX platform by the specified computer language named MYPARAL. This neural network model was used as an auxiliary controller in the temperature control of sinter cooler system in steel plant which is a nonlinear system. The neural controller was trained off-line using the real input-output data as training pairs. We also made the system description of adaptive neural controller on the same temperature control system. We will carry out the whole system simulation to verify the suitability of neural controller in improving the system features.

## 1. Introduction

### 1.1. Artificial Neural Network

Artificial Neural Network (ANN) is composed of numerous neurons, the outputs of which are the weighted summation of all the inputs by the active function (generally is nonlinear function). The active function for each neuron may be the same one or not depending on the architecture of ANN. Also, the full connection between neurons are not necessary although most of ANNs recently developed are fully connected.

The neural networks in which the network outputs do not feedback to the input or the hidden layer are called feed forward networks. Otherwise, they are called feedback or recurrent networks. We here mainly consider the feedforward neural networks and the training algorithms and their applications in control system.

The most popular training algorithm implemented is the back- propagation (BP) algorithm. It has found that the fixed training rate may lead the convergence to be very slow. Using dynamic rate is a solution [1]. This approach does not increasing the computational burden as largely as the cases applying the second order methods [2], e.g. the Newton method and the Levenberg-Marquardt method.

It has been shown that ANN with partial connection architecture can meet the system requirements. There are numerous approaches concerning the automatically evolving the neural network architectures. For example the evolutionary programming [3], the genetic algorithm [4] [5].

### 1.2. Neural Network Based on Control

Neural network theory has been the object of intense study and application, especially in the last decade. Obviously the introduction of neural network control into areas in which the analysis and design of control systems are traditionally performed using techniques whose effectiveness is well established, is a wise selection.

As regards neural networks, the research activity essentially addressed the study of properties of multilayer feed forward neural network. It has been proven that this kind of network has the ability of mapping non-linear continuous function to arbitrary degree of approximation. The most widely used training algorithm is the backpropagation because of its simplicity.

It can be seen as a natural step to use ANN in control systems to meet new challenges. ANN has found growing success in many kinds of industrial control application. For example, using neural controller in active vision system [6], ANN was used as state feedback controller that learns the time-optimal actions by means of optimization process [7]. Also ANN was used as a nonlinear self-tuning adaptive controller[8], and in system identification [9].

The neurocomputer was applied in our research to carry out the artificial neural network calculating. At first, the neural network as auxiliary controller is constructed. An application software was developed based on back-propagation (BP) algorithm under the UNIX platform using the specified computer language MYPARAL [10] [11] [12]. The application software of BP algorithm using the neurocomputer was used to simulate the cooler system in steel plant [13]. The simulation was carried out and simulation result was analyzed. The standard deviation between the model output and the real output $\sigma$ is equal to 0.045. This neural controller will be used as auxiliary controller in the temperature control of cooler system in sinter plant. The whole system simulation will be carried out to verify the effectiveness of neural controller.

## 2. Control System Description

### 2.1. Neural Network as Auxiliary Controller

Successful industrial applications and favorable comparisons with conventional algorithms have motivated the development of ANN control system. From the every beginning, it has been realized by system theorists that most dynamical systems are nonlinear and time-varying. ANN has been proven be capable of approximating arbitrary nonlinear functions, learning through examples and has the ability of combining large amount of data to form decisions or pattern recognition.

Agarwal M. [14] and Antsaklis P.J. [15] made a comprehensive systematic classification of the neural network based on control schemes proposed in literature. There are mainly two kinds of neural controller, i.e. the auxiliary neural controller and the adaptive neural controller [16]. At first, we give out the system description and neural network training algorithm about the control system based on the auxiliary neural controller. Next, we will consider the topology and training algorithms of adaptive neural controller.

A feed forward neural network was implemented as an off-line controller in the temperature control of cooler system in steel plant. The whole system is illustrated as Fig.1.

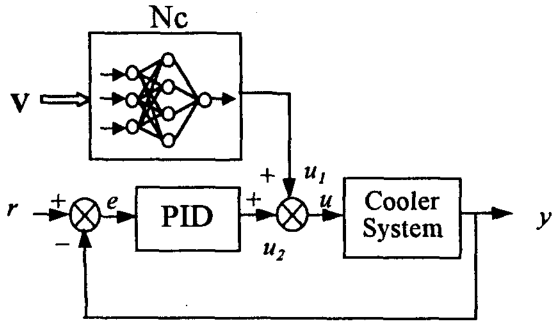


Fig. 1 Auxiliary neural control system

This is a nonlinear control system. The control process is that given the desired output temperature $r$ of cooler system, adjusting the damper slot $u$ to enable the plant output $y$ tracking the desired output $r$. If we do not apply the neural controller Nc in this system, it is simply a single input and single output (SISO) control system using the conventional PID controller.

The fundamental strategy of designing a PID controller for a given plant is to determine the parameters by means of time domain algorithm or frequency domain algorithm. If the plant model is correct i.e. the model architecture is adequate and model parameters are accurate enough, the nonlinear properties of system (the nonlinear inputs and disturbances) are small, the PID controller will perform well. But for cases where the system nonlinear properties can not be ignored, or the system external and internal condition have changed, the conventional PID controller will not guarantee good performance. For this reason, a neural controller Nc is added to the plant as an auxiliary controller to compensate the shortage of PID controller when the system nonlinear properties are apparent.

The neural controller Nc is a feed forward neural network with one hidden layer as shown in Fig.2.

There are 6 inputs in input layer, 10 neurons in hidden layer and one output $u_1$ in output layer. The input vector is V = Col [$v_1$ ... $v_6$ ]. In real cooler temperature control system, the inputs are Mass-input ($v_1$), Cooler-level ($v_2$), Cooler-rotate ($v_3$), Out-temperature ($v_4$), Atmosphere-temperature ($v_5$), WB-

20-temperature ($v_6$), respectively. The neural network

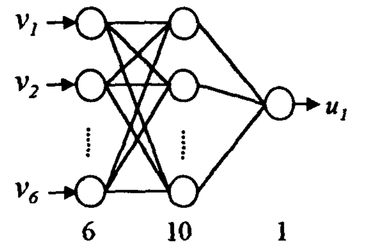


Fig.2 Architecture of neural controller Nc

output is the Damper-slot ($u_1$). The sigmoid function is used as active function in each neuron in hidden layer and output layer.

We use back propagation algorithm (BP) to train the neural network. The training algorithm is based on adjusting the weights of neural network in direction of minimizing the value of objection function. The objection function is defined as the summation of square errors between the desired outputs and the neural network outputs shown as Eq. (1).

$$J= (1/2P) \sum_{p=1}^{P} (u_{1t,p} - u_{1,p})^2 \quad (1)$$

where $J$ is the objective function, $u_{1t,p}$ and $u_{1,p}$ is the desired output and the model output of p-th pattern, respectively It then employs a steepest-descent search algorithm to seek the minimum of objective function defined by Eq.(1). For any weight $w_{ij}$ in neural network, we derive its adjustment in training process, see Eq.(2).

$$w_{ij}(k+1)= w_{ij}(k) - \eta \frac{\partial J}{\partial w_{ij}(k)} + \alpha \Delta w_{ij}(k) \quad (2)$$

where $\alpha$, $\eta$ are training coefficients, $w_{ij}(k+1)$, $w_{ij}(k)$ is ij-th weight at $(k+1)$-th and $k$-th training step, respectively, $\Delta w_{ij}(k)$ is the error of $w_{ij}(k)$ at $k$-th training step.

## 2.2. Neural Network as Adaptive Controller

The auxiliary neural controller described above is trained off-line. It can only perform well in the training state space. If the state of controlled system (in this paper the temperature control system) exceeds the training space, the neural controller will not guarantee the good performance. So it can only act as auxiliary controller to compensate the shortage of conventional controller. A kind of adaptive neural controller is shown as Fig.3.

This is a directive adaptive neural control system. The cooler system is assumed as second order system. The parameters of cooler system are

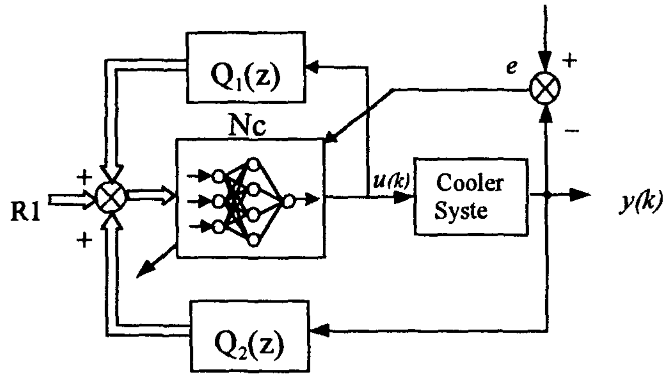


Fig. 3 Adaptive neural control system

unknown. The neural network is trained on-line to decrease the tracking error between the desired output and the system real output. In Fig.3, $r(k)$ is the system reference input, $y(k)$ is the system output, $u(k)$ is the output of neural controller Nc, $e$ is the error between the desired output and the real output of cooler system, R1 = Col [ $v_1$ ... $v_6$, $r(k)$], $Q_1(z)$ and $Q_2(z)$ represent the time delays, $Q_1(z) = z^{-1}$, $Q_2(z)$=Col [1, $z^{-1}$ ]. $v_1$ ... $v_6$ are same as the defined variables in 2.1.

The neural controller Nc in Fig. 3 is also a feed forward neural network with 10 inputs as R = [$v_1$ ... $v_6$ , $r(k)$, $y(k)$, $y(k-1)$, $u(k-1)$], 10 neurons in the hidden layer, one neuron in the output layer. The training of neural network is also in the direction of decreasing the value of objective function $J$ defined by Eq.(3).

$$J = (1/2)\, e^2 = (1/2)\, (r(k) - y(k))^2 \quad (3)$$

Comparing Eq.(1) with Eq.(3), we may find the difference between the two objective functions. In Eq.(1), the objective function is the error between the desired output and the real output of neural network. Whereas in Eq.(3), the objective function is the error between the desired output and the real output of cooler system. The neural network is trained on-line to decrease the system error. For any weight $w_{ij}$ in neural network, the adjustment in training process is defined by Eq.(4).

$$w_{ij}(k+1)= w_{ij}(k) - \eta \frac{\partial J}{\partial w_{ij}(k)} + \alpha \Delta w_{ij}(k) \quad (4)$$

Using the chain rule, the partial derivative of objective function $J$ with respect to the weight $w_{ij}$ is calculated as following.

$$\frac{\partial J}{\partial w_{ij}(k)} = -e\frac{\partial y(k)}{\partial w_{ij}(k)} = -e\frac{\partial y(k)}{\partial u(k)}\frac{\partial u(k)}{\partial w_{ij}(k)} \qquad (5)$$

where $e = r(k) - y(k)$, the partial derivative $\partial u(k)/\partial w_{ij}(k)$ can be derived in the same way described in 2.1. using BP algorithm. Because the cooler system model is unknown, the partial derivative $\partial y(k)/\partial u(k)$ can not derived directly. We use the sign function of $(\partial y(k)/\partial u(k))$ in training of the neural network [17]. So that the Eq.(4) is rewritten as Eq.(6).

$$w_{ij}(k+1) = w_{ij}(k) + e\,\eta\,sign\left(\frac{\partial y(k)}{\partial u(k)}\right)\frac{\partial u(k)}{\partial w_{ij}(k)} + \alpha\,\Delta w_{ij}(k) \qquad (6)$$

## 3. Simulation Result

We carried out the training of neural network as auxiliary controller described in 2.1. The training of adaptive neural controller described in 2.2. will carry out in the near future.

Generally, the data normalization is needed before carrying out the training process. We normalized the input and output data of the real system by dividing the maximum value of the original ones. After normalization, the values of input and output of neural network range between 0 and 1. The 100 sets of real data from the cooler system are selected as the training pairs in training the neural controller Nc. Fig. 4 is the output of neural network $u_l$ compared with the desired output $u_{lt}$. Fig. 5 is the training standard deviation σ.

From the simulation results shown by Fig.4 and Fig.5, we can see that the standard deviation σ is equal to 0.045. The value of standard deviation σ is still little large.

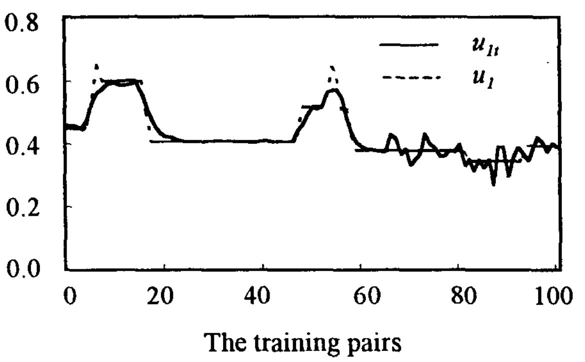


Fig.4 The output of the neural network

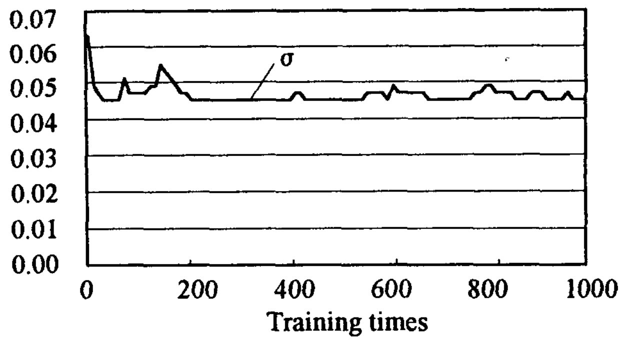


Fig.5 The training standard deviation

The first reason is that the input and output data length is limited to 10 bits in this neurocomputer, less than the length of original data. The second one is that maybe the structure of the neural network is not complex enough.

## 4. Conclution

In this paper, two kinds of neural controllers, auxiliary neural controller and adaptive neural controller, are discussed in the temperature control of cooler system in steel plant. The neural network as auxiliary neural controller was trained of-line using BP algorithm in neurocomputer called MICROCO-MPUTER under the special computer language called MYPARAL. In next step, we will improve the program and the model structure to get more accurate result and to apply this model to the cooler control system. Also, we plan to use evolutionary algorithm to optimizing the neural network architectures.

Although there are numerous successful ANN based control systems, most of the architectures are the kind of full-connected and fixed ones. It means that the structure of a neural network can not be changed if it has been trained for a definite goal. This kind of neural network will perform worse when the plant (controlled system) structure changed. Although a large scale neural network with more hidden neurons than real requirements can be applied to meet the potential demands of systems, the computation intensity may become much more obvious than ever, especially for on-line training algorithm. So that, the optimization of neural networks architectures is necessary for control systems.

Evolutionary system is one of novel schemes for evolving ANN. It combines the architecture evolving (i.e. the number of inputs, hidden neurons and the connection density, etc) with the weights training. Given some performance criteria, e.g., minimum error of training, fastest learning, lowest complexity, it tries to search the optimal architecture of ANN and train the weights of neural network at the same time. We will make efforts to carry out the research on the optimization of neural networks architectures.

## Reference


[1] X. H. Yu, G .A. Chen, S. X. Chen, " Dynamic Learning Rate Optimization of the Back-propagation Algorithm", *IEEE Transaction on Neural Networks,* Vol. 6, No. 3, pp. 669 – 677, 1995.

[2] R. Parisi, E. D. D. Claudio, G. Oriandi and B. D. Rao, "A Generalized Learning Paradigm Exploiting the Structure of Feedforward Neural Networks", *IEEE Transaction on Neural Networks,* Vol. 7, No. 6, pp. 1450–1459, 1996.

[3] X.Yao and Y.Liu, "A new evolutionary system for evolving artificial neural networks", *IEEE Transactions on Neural Networks,* Vol.8, No.3, pp. 694 –713, 1997.

[4] V. Maniezzo, "Genetic Evolution of the Topology and Weight Distribution of Neural Networks", *IEEE Transactions on Neural Networks,* Vol.5, No.1, pp.39–53, 1994.

[5] J. Murata, K. Tanaka, M. Koga and K. Hirasawa, "Neural Network Structure Design Using Genetic Algorithm", *Proceeding of Korea Automatic Control Conference(KACC '95),* Pusan, Korea, pp. 187- 190, 1995.

[6] N.Srinivasa and R.Sharma, "Execution of saccade for active Vision Using a neuro-controller", *IEEE Control Systems,* Vol. 17, No. 2, pp. 18-29, 1997

[7] T.R Niesler. and J.J. Duplessis, "Time-optimal control by means of neural networks", *IEEE Control Systems,* Vol. 15, No. 5, pp.23-33, 1995.

[8] F. C. Chen, "Back-propagation neural networks for nonlinear self-tuning adaptive control", *IEEE Control Systems Magazine,* Vol. 10, No. 3, pp. 44-48, 1990.

[9] S. I. Mistry and S. S. Nair, "Identification and control experiments using neural designs", *IEEE Control Systems,* Vol.14, No.3, pp.48-57, 1994.

[10] M. Sugisaka and Z. J. Liu, "Artificial neural network models using neurocomputer", *Proceeding of the 2nd Int. Workshop on Advanced Mechatronics,* Nagasaki, Japan, pp.80-83, 1997.

[11] M. Sugisaka and Z. J. Liu, "The application of neurocomputer for a control problem", *Proceeding. of the Third Int. Symp. on Artificial Life and Robotics(AROB 3rd '98),* Beppu, Oita, Japan, pp. 444-487, 1998.

[12] M. Sugisaka and Z. J. Liu, "The simulation of artificial neural network methods in cooler control using neurocomputer", *Proceeding of the Third Workshop of Int. Institute for General Systems Studies,* Qinhuangdao, China, pp.140-143, 1998.

[13] Tokuda, "The research on the temperature control of cooler", *Research Report,* Nippon steel plant in Oita, pp. 1-9, 1996.

[14] M. Agarwal, "A systematic classification of neural-network-based control", *IEEE Control Systems,* Vol. 17, No. 2, pp. 75-93, 1997.

[15] P. J. Antsaklis, "Neural networks in control systems", *IEEE Control Systems Magazine,* Vol. 10, No. 3, pp. 3-5, 1990.

[16] Y. Zhang, P. Sen and G.E Hearn., "An on-line trained adaptive neural controller", *IEEE Control Systems,* Vol.15, No.5, pp.67-75, 1995.

[17] Y. Zhang, P. Sen and G. E Hearn., "An on-line trained adaptive neural controller", *IEEE Control Systems,* Vol.15, No.5, pp.67-75, 1995.