

System Identification Using a Neuro – Fuzzy Identifier

Min – ho Lee

Department of Electrical Engineering, Korea Maritime University, Pusan, Korea

Abstract

A neuro – fuzzy identifier for fuzzy modeling of a system is explained. The proposed neuro – fuzzy identifier contains not only an adaptive clustering process for determining center points of the inputs and virtual output membership functions but also an adaptive process for deciding the shapes of the input membership functions. Moreover, linguistic fuzzy rules of a system can be obtained from the proposed neuro – fuzzy identifier. Computer simulation results show that the neuro – fuzzy identification is very effective in modeling fuzzy systems with the fuzzy rules of which cannot be obtained easily.

1. Introduction

Fuzzy logic, based on the possibility theory that allows functions to have real values from zero to one, has been successfully applied to many industrial areas such as process control¹⁾, robot arm control²⁾, servo motor position control³⁾, stabilization of the inverted pendulum system⁴⁾ and complex decision making or diagnosis systems⁵⁾ etc. When fuzzy logic is used, input data in the form of linguistic variables are represented by membership functions, which are used to determine the fuzzy set of a crisp value and degrees of membership in this set. The linguistic fuzzy if – then rules which are determined from a priori knowledge of a system generate the appropriate output value through a suitable reasoning process and a defuzzification process as shown in Fig. 1. In this process, many parameters of fuzzy logic

must be designed by an expert of fuzzy logic but their design processes depend on the trial – and – error methods or some heuristic algorithms⁶⁾. Moreover, someone who knows the characteristics of the system is also needed for setting up the initial rules. Recently, some results show techniques to create fuzzy rules and modify them based on experience^{6,7,8)}. Among these results, a self organizing fuzzy controller can create and modify the rules^{1,6)}, and a clustering algorithm for fuzzy partition of the input data space⁷⁾ and a least mean square algorithm for determination of the consequence parameters^{7,8)}. Although these methods give interesting results, they are still some-

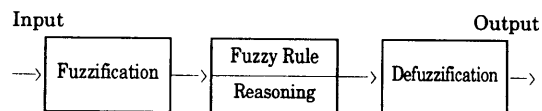


Fig. 1 General structure of fuzzy logic.

what heuristic, and the choice of membership functions depends on the trial – and – error procedure. Hence, lots of methods utilizing learning capability of artificial neural networks have been successfully tried in recent^{9,10,11,12}. Multilayer neural networks with learning algorithms, such as error back propagation learning algorithm, reinforcement learning algorithm, unsupervised learning algorithm, or combined learning algorithm among these algorithms, have been used for extraction fuzzy rules or tuning finely the membership functions and degrees of membership. Most of these researches use a multilayer neural network for resolving the inconvenience of using fuzzy logic like tuning of fuzzy rules, and membership functions with the weight values that represent directly the fuzzy parameters. This approach of using neural networks has the limited number of degrees of freedom because the number of weights depend on a priori knowledge of fuzzy logic such as the number of clusters of input and output data on the universe of discourse and the number of fuzzy rules, and thus well – defined a priori knowledge of the system is required for success in learning process of the neuro – fuzzy network.

In this paper, a new neuro – fuzzy identifier is developed by incorporation the fuzzy logic into the neural network so that the proposed neuro – fuzzy can handle not only unstructured numerical data but also the structured informations like fuzzy logic. The neuro – fuzzy identifier based on the fuzzy relational equation¹³, which will be explained in section 2.1, consists of a fuzzification block, a multilayer neural network that has enough number of degrees of freedom for finding the fuzzy rules of the unknown system, and a defuzzification block^{14,15,16}. The proposed neuro – fuzzy identifier has optimal cluster center values and shapes

of membership functions by training the weight values and neuron characteristics of the neural network for minimizing a performance criterion function. Moreover, it can compensate for the fuzzification – and – defuzzification error^{14,15,16} and extract an implication fuzzy rule table of the system in the form of linguistic fuzzy relationships between input and output data of the system¹⁵. Its effectiveness is shown by computer simulation of some examples in section 2.2.

2. Fuzzy modeling using neuro – fuzzy network

2.1 Neuro – fuzzy identifier

In order to design a system processor for handling knowledge information represented in the linguistic or uncertain numerical form of data, we need a fuzzy model of the system. Fuzzy modeling is the method of describing the characteristics of a system using fuzzy rules⁷, and it can express complex nonlinear dynamic systems by linguistic if – then rules. As mentioned by Takage & Sugeno⁸, various structures of fuzzy models may be used according to their usage. If an operator can not tell linguistically what kind of action he takes in some situation, then it is quite useful to model his control actions using numerical data. On the other hand, if linguistic representation of system characteristics is required, an implication fuzzy rule table of input and output characteristics may be useful as a model. But, there are some difficulties in designing the parameters of fuzzy logic which satisfy a criterion. Therefore, neuro – fuzzy networks are useful for fuzzy modeling in both cases.

In this section, the new neuro – fuzzy identifiers and the learning algorithm are introduced^{14,15,16}. The proposed neuro – fuzzy identifier can express a system characteristic not only

in the form of linguistic fuzzy rule table but also in the form of numerical data. It is shown in Park¹⁷⁾ that a multilayer neural network with error back propagation learning algorithm can approximate an arbitrary nonlinear function in $L^{p(18)}$ and has the generalization characteristics¹⁹⁾. Since a fuzzy model of the system is a function between input and output fuzzy variables, we can use a multilayer neural network for finding a fuzzy relationship of the system¹⁷⁾. Fig. 2 shows a method for obtaining a fuzzy model of an unknown system^{14,15,16)}. A fuzzification process of the input signal with gaussian membership functions is shown in Fig. 3. In other words, the fuzzy sets, \tilde{X} in Eq.(1), are constructed by fuzzy relational Eq.(2) that calculates by max – product operation the possibilities that the input and output data as fuzzy singleton $\delta(x - x_k)$ belong to the predefined reference fuzzy numbers in the uni-

verse of discoures¹⁹⁾.

$$\begin{aligned} \tilde{X}_k &= \text{fuzzification}(x_k) \\ &= \{x_k^1/cx^1, x_k^2/cx^2, \dots, x_k^M/cx^M\} \end{aligned} \quad (1)$$

and

$$\begin{aligned} x_k^i &= \text{possibility}(x_k | \tilde{R}X^i) \\ &= \max_{x \in X} (\tilde{R}X^i \cdot \delta(x - x_k)) \end{aligned} \quad (2)$$

where $\tilde{R}X^i$ represents the i -th reference fuzzy number, x_k^i the membership function at time k in the i -th reference fuzzy number whose center point is cx^i , and $\delta(\cdot)$ a Kronecker delta function. The elements of fuzzy set \tilde{X}_k , that is x_k^1, x_k^2, \dots , and x_k^M , are used for the inputs of the neural network. The output y of the system is also fuzzified according to Eq.(1) and Eq. (2) with $\tilde{R}Y^i$ instead of $\tilde{R}X^i$ and compared with the output \hat{y} of the identification neural network. The neural network is trained with error back propagation learning algorithm²⁰⁾ to minimize the output error that is difference between the output of the neural network and the fuzzified output of the system. The center of gravity method in Eq.(3) is used for the defuzzification process,

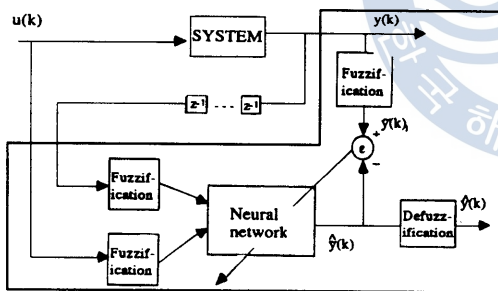


Fig. 2 Neuro – fuzzy identification structure of dynamic systems.

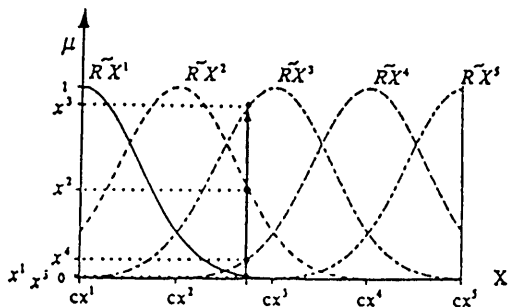


Fig. 3 Fuzzification of crisp data with some reference fuzzy numbers.

$$\hat{y}_k = \text{defuzzy}(\hat{Y}_k) = \frac{\sum c y^i \hat{y}_k^i}{\sum \hat{y}_k^i} \quad (3)$$

In general, the structure as shown in Fig. 4 can have some errors due to the fuzzification and defuzzification processes^{15,16)}. Even though the neural network for finding fuzzy rules in Fig. 2 was trained well, the defuzzified output and the system output may have some discrepancy in their values. This error mainly depends on the fuzzification and defuzzification process and adds to the identification error.

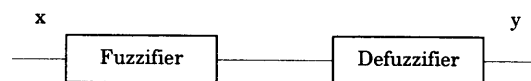


Fig. 4 Fuzzification and defuzzification process.

Fig. 5 shows the difference between fuzzified - and - defuzzified value and original crisp value of data with gaussian membership functions in Fig. 3 and center of gravity method in Eq.(3), where solid lines represent the input x and dashed lines the fuzzified - and - defuzzified output value y . Fuzzification - and - defuzzification error can be compensated with

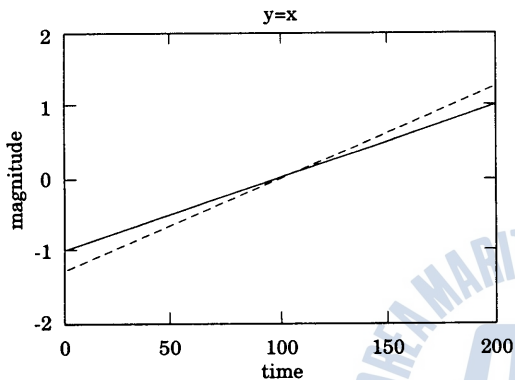


Fig. 5 Simulation results of fuzzification and defuzzification process.

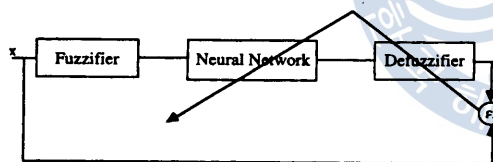


Fig. 6 Compensation of fuzzification and defuzzification process using neural networks.

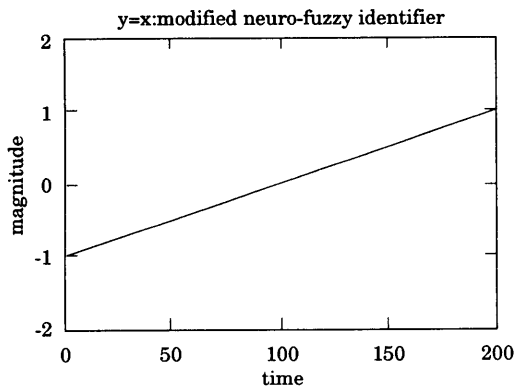


Fig. 7 Simulation results when neural network is used.

neural networks operation in between fuzzification and defuzzification processes. In Fig. 6, neural networks are trained to minimize fuzzification - and - defuzzification error by back - propagation the error through defuzzifier. Fig. 7 shows the result of compensated error in Fig. 5 by neural network. When we use triangular membership functions as shown in Fig. 8, there is no difference between fuzzified - and defuzzified value and original crisp value of data with center of gravity method in Eq. (3) as a defuzzification process. However, when output cluster center values are different from input cluster center values, simple fuzzification and defuzzification processes with triangular membership functions also yield error as shown in Fig. 9. In

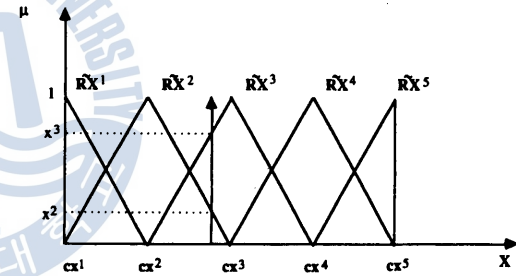


Fig. 8 Fuzzification of crisp data using reference fuzzy numbers with triangular membership functions.

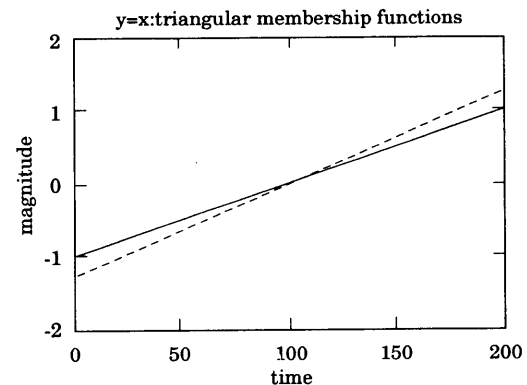


Fig. 9 Comparison of fuzzified - and - defuzzified values and original crisp data for output y equal to x , when triangular membership functions are used.

this case, neural networks can be used for compensation of this fuzzification - and - defuzzification error as in Fig. 6, and the result is shown in Fig. 10.

Fig. 11 shows the modified neuro - fuzzy identifier for compensation of general fuzzification - and - defuzzification error in the output of neuro - fuzzy identifier of Fig. 2. The output of

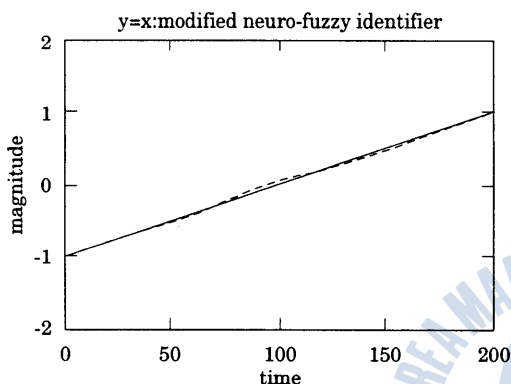


Fig. 10 Compensated result by neural network which is located between the fuzzification block and defuzzification block.

the modified neuro - fuzzy identifier is given as the defuzzification Eq.(3) where cy^p represents the center point of the p - th output fuzzy number in Fig. 3 and behaves as fixed weight of the additional layer in the defuzzification block of Fig. 11 during the learning process of the identifier. The output of the neural network becomes a set of virtual membership values in fuzzy numbers in this case. We can reduce this fuzzification - and - defuzzification error of the neuro - fuzzy identifier by taking the output error after the defuzzification block^{14,15}. Moreover, the center points and shapes of input membership functions are also trained by error back propagation learning algorithm, and this additional learning process not only reduces the learning time and the size of the neural network but also gives optimal center points and shapes of membership functions.

To obtain a fuzzy model represented by lin-

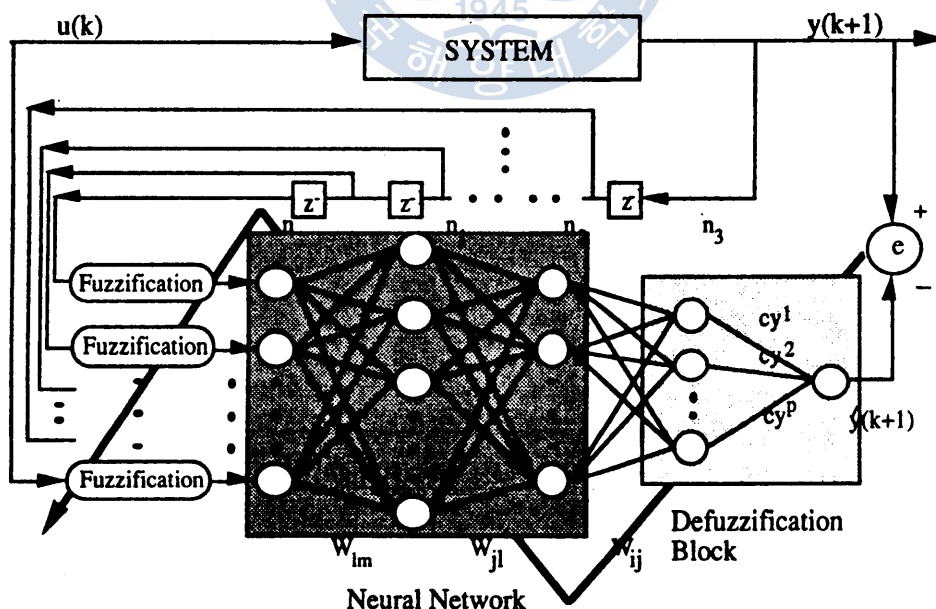


Fig. 11 Modified neuro - fuzzy identification structure.

guistic implication fuzzy rules of input and output data, an additional consideration must be taken. In Fig. 11, since the outputs of the neural network generate the virtual output membership values according to the fuzzy input variables, and these values may violate a convexity property of membership values during the error back propagation learning process, we suggest a learning algorithm to obtain a fuzzy model represented by linguistic fuzzy rules with the error measure for minimization of output errors and maintenance of convexity of output membership values in the following equation,

$$E(k+1) = \frac{1}{2} [\{y(k+1) - \hat{y}(k+1)\}^2 + \sum_p^{n_3} \delta_p^2(k)^2] \quad (4)$$

and

$$\hat{y}(k+1) = \frac{\sum_p^{n_3} CY^p \hat{y}^p(k)}{\sum_p^{n_3} \hat{y}^p(k)} = \frac{\sum_p^{n_3} CY^p f_1(\sum_j^{n_2} W_{pj} h_j^2)}{\sum_p^{n_3} \hat{y}^p(k)} \quad (5)$$

and CY^p are the center values of reference output membership function, n_i shows the number of neurons in the i -th layer, and $\delta_p(k)$ is a measure of violation of convexity of output membership values,

$$\delta_p(k) = \begin{cases} \max\{\hat{y}^p(k) - \hat{y}^{p+1}(k), 0\} & \text{when } p < P_{max} \\ \max\{\hat{y}^p(k) - \hat{y}^{p-1}(k), 0\} & \text{when } p > P_{max} \end{cases}$$

where p_{max} shows the p -the neuron having the maximum value of the output of the neural network. In Eq.(4), the first term is the error for the measure of identification accuracy and the second term is for the measure of convexity of output membership values. The weight values of the neural network are trained by usual error back propagation learning algorithm with the modified error function in Eq.(4), and additional parameters of the neuro-fuzzy network are trained according to Eqs. (6) - (11).

$$CY^p(k+1) = CY^p(k) + \alpha \{y(k+1) - \hat{y}(k+1)\} \frac{\hat{y}^p(k)}{\sum_p^{n_3} \hat{y}^p(k)} \quad (6)$$

and

$$m_i(k+1) = m_i(k) + \alpha_m \varepsilon_m(k+1) x_i \quad (7)$$

where

$$\varepsilon_m(k+1) = \varepsilon(k+1) \sum_j^{n_2} W_{pj} f' \left(\sum_j^{n_1} W_{jl} h_l^1 \right) \sum_l^{n_1} W_{jl} f' \left(\sum_m^n W_{lm} x_m \right) \frac{(x_i - m_i)}{\sigma_i^2(k)} \quad (8)$$

and

$$\sigma_i^2(k+1) = \sigma_i^2(k) + \sigma_\sigma \varepsilon_\sigma(k+1) x_i \quad (9)$$

where

$$\varepsilon_\sigma(k+1) = \varepsilon(k+1) \sum_j^{n_2} W_{pj} f' \left(\sum_j^{n_1} W_{jl} h_l^1 \right) \sum_l^{n_1} W_{jl} f' \left(\sum_m^n W_{lm} x_m \right) \frac{(x_i - m_i)^2}{\sigma_i^3(k)} \quad (10)$$

and where

$$\varepsilon(k+1) = (y(k+1) - \hat{y}(k+1)) \frac{CY^p \sum_p^{n_3} \hat{y}^p(k) - \sum_p^{n_3} CY^p \hat{y}^p(k)}{(\sum_p^{n_3} \hat{y}^p(k))^2} f'_1 \left(\sum_p^{n_2} W_{pj} h_j^2 \right) + \sum_p^{n_2} \delta_p^2(k) f'_1 \left(\sum_p^{n_2} W_{pj} h_j^2 \right) \quad (11)$$

and

$$f_1(r) = \frac{1}{1+e^{-ar}}, \quad f_2(r) = \frac{2}{1+e^{-ar}} - 1 \quad (12)$$

where, because the outputs of the neural network show the virtual membership values which are real values in $[0, 1]$, the function $f_1(r)$ is used as the output sigmoid function of the neural network, $f_2(r)$ as the transfer function of hidden neurons, and $f'_1(\cdot)$ and $f'_2(\cdot)$ show the derivatives of $f_1(\cdot)$ and $f_2(\cdot)$, respectively, The notation of W_{uv} represents the

weight values in the neural network block between the v – th neuron of a layer and the u – th neuron of the next layer as shown in Fig. 11, and $m_i(k)$ and $\sigma_i(k)$ show the i – th center value and variance of input membership functions, respectively. The CY^p is the center value of the p – th virtual output membership function and α shows the learning rate. When the learning process is finished, we use the modified center values of reference input fuzzy numbers as the test signals for finding linguistic fuzzy rules. As shown in Fig. 3, the fuzzy input sets are constructed by fuzzy relational equation using these center values of input membership functions and then these fuzzy values are applied as input values to a trained neural network which represents fuzzy relationship. The virtual output membership function values are obtained from the output of the neural network, and these input and output values of the neural network construct implication fuzzy rules of the system.

When the fuzzy rule of a system is known, it can be used for design of a fuzzy model of the system. That is, the proposed neuro – fuzzy identifier learns the initial implication fuzzy rules by learning the system directly as shown in Fig. 13. So, we can not only reduce the size of the neural network and the number of training data but also speed up the learning process if the initial fuzzy rules of the system are more or less accurate.

The following section presents the computer-simulation results of some example models.

2.2 Simulation results

We consider the following two examples of nonlinear dynamic systems,

Model 1 :

$$y_p(k+1) = \frac{y_p(k)}{1+y_p^2(k)} + u(k) \quad (13)$$

Model 2 :

$$y_p(k+1) = \frac{y_p(k)}{1+y_p^2(k)} + u^3(k) \quad (14)$$

These models have two inputs that are the external input $u(k)$ and the delayed system output $y(k)$. Model 1 has nonlinear characteristics in the delayed output, and model 2 has the nonlinearities both in the system output and in the input but these terms are combined linearly. From generalization capability of the neural network, about 10W learning data were sufficient for identification of the system where W denotes the number of weights of neural network. Learning for identification is performed with 7,000 random data for the neuro – fuzzy identifiers shown in Figs 2, 11, and 12 and with 4,000 random data for fine – tuning process of fuzzy rules in Fig. 13. Learning data are bounded by $[-1.65, 1.65]$, and identifiers are tested with the input signal, $\sin(2\pi k/250)$. Neural network models for finding fuzzy rules have 2 hidden layers. Input and hidden layers have additional neurons for adaptive thresholds.

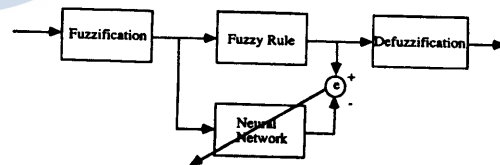


Fig. 12 Neural identification structure of initial fuzzy rules.

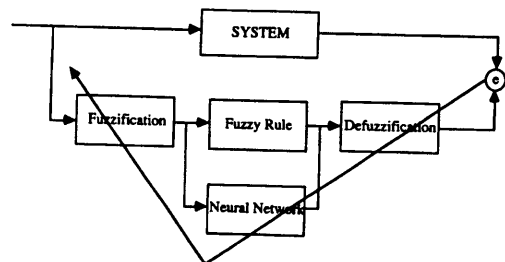


Fig. 13 Compensation for the error of initial fuzzy rules by the modified neurofuzzy identifier.

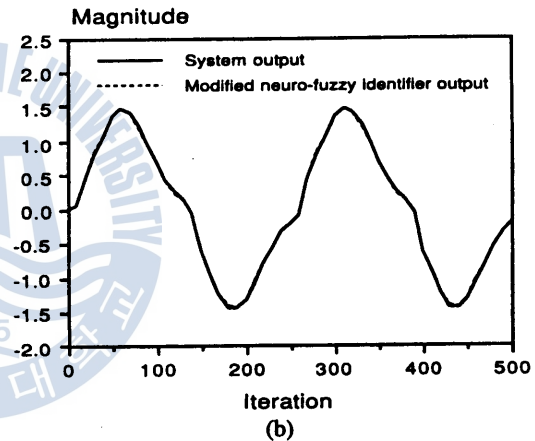
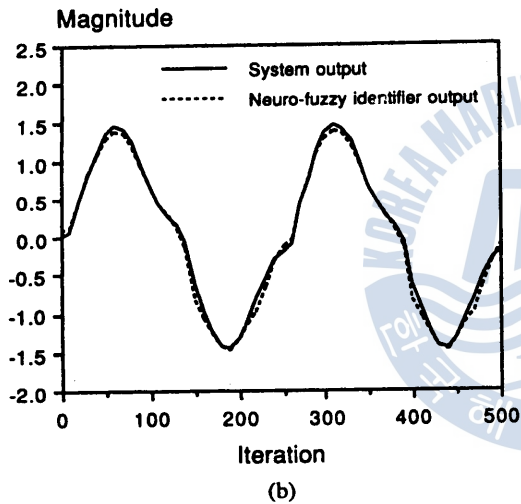
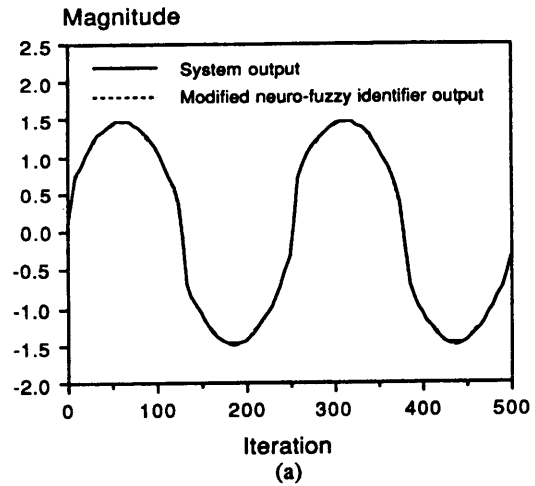
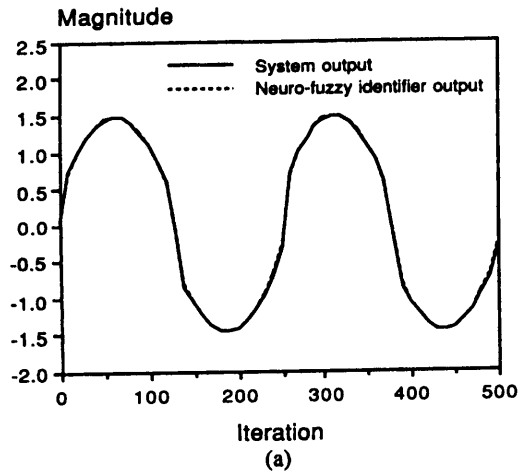


Fig. 14 Simulation results of neuro - fuzzy identifier (a) model 1, (b) model 2.

Fig. 15 Simulation results of the modified neuro - fuzzy identifier that creates the fuzzy rules of the system (a) Model 1, (b) Model 2.

Originally, we use gaussian functions with mean values of -1.5 , -0.75 , 0 , 0.75 , and 1.5 and variance 0.2 as reference input fuzzy numbers and gaussian functions with mean values of -2 , -1 , 0 , 1 , and 2 and variance 0.35 as reference output fuzzy numbers in Fig. 2. In case of the modified neuro - fuzzy identifier shown in Fig. 11, mean values and variance values of input and output fuzzy numbers are used as initial weight values of the additional layer in the fuzzification and defuzzification blocks of

Fig. 11 and output fuzzy membership values are produced by neural network.

We compare the simulation results of the neuro - fuzzy identifier shown in Fig. 2 with those of the modified neuro - fuzzy identifier shown in Fig. 11. Fig. 14 and 15 show the simulation results of the neuro - fuzzy identifier and the simulation results of the modified neuro - fuzzy identifier that creates the fuzzy rules of system from random initial weight values, respectively. Solid and dashed lines represent

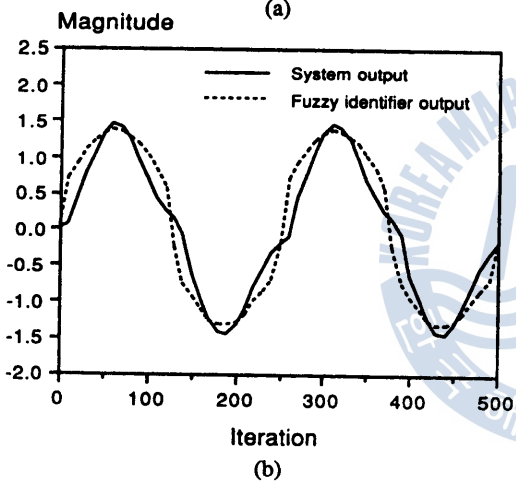
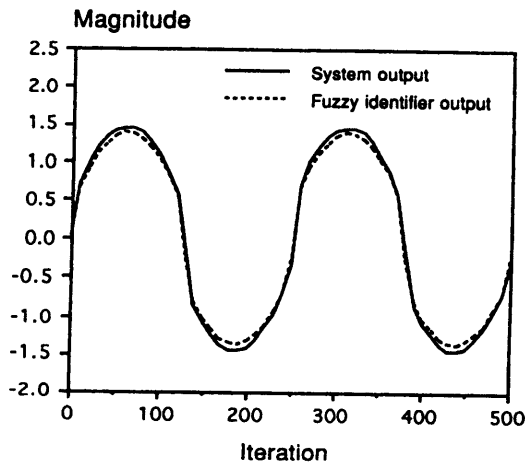


Fig. 16 Simulation results of the fuzzy identifiers.

the real system outputs and the identifier outputs, respectively. The modified neuro – fuzzy identifier is shown to have better performance than the neuro – fuzzy identifier by reduction the fuzzification – and – defuzzification error without much increase of complexity of the network structure and the learning method and also having the additional adaptive process of membership functions.

Fig. 16 shows the simulation results of fuzzy modeling by max – min direct reasoning method and implication fuzzy rules of the system shown in Table 1, which is obtained from

Table 1 Fuzzy rules of models 1 and 2(a) Model 1, (b) Model 2.

U(k) \ Y(k)	NB	NM	Z	PM	PB
NB	NB	NB	NB	NM	NM
NM	NM	NM	NM	Z	Z
Z	NM	NM	Z	PM	PM
PM	Z	Z	PM	PM	PM
PB	PM	PM	PB	PB	PB

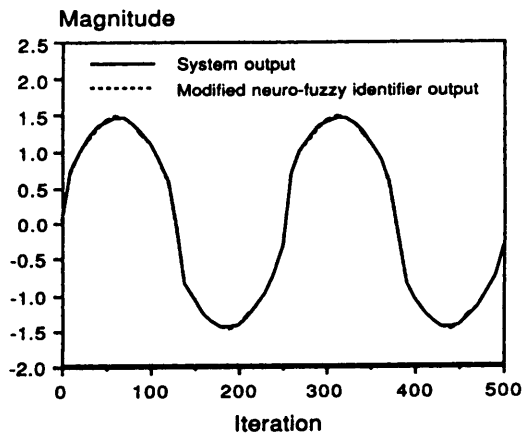
(a)

U(k) \ Y(k)	NB	NM	Z	PM	PB
NB	NB	NB	NB	NB	NB
NM	NM	NM	NM	Z	Z
Z	Z	NM	Z	PM	PM
PM	Z	Z	PM	PM	PM
PB	PB	PB	PB	PB	PB

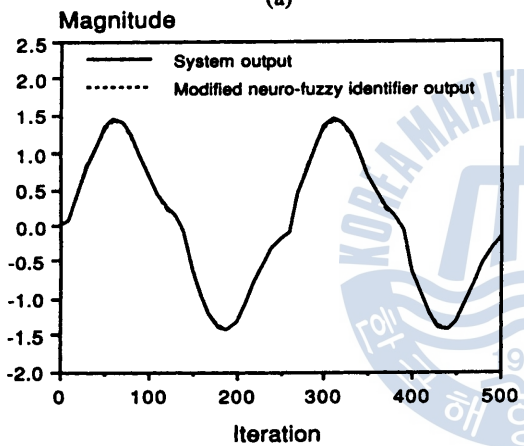
(b)

ad hoc method. Solid and dashed lines represent the output of the system and the output of the fuzzy identifier, respectively. Fig. 17 shows the simulation results in case that the modified neuro – fuzzy identifier learns the initial fuzzy rules as shown in Fig. 12 and then modifies the initial fuzzy rules as shown in Fig. 13. Solid lines and dashed lines represent the output of the system and the output of the modified neuro – fuzzy identifier, respectively.

Tables 2 and 3 show the modified fuzzy rules and the created fuzzy rules, respectively. The modified fuzzy rules are different from the created fuzzy rules in model 2 because the initial weight values have different values and the trained weight values and the optimized parameters of membership functions have some differences in two cases of Table 2 and 3. Simulation results show that the modified neuro – fuzzy identifier not only gives vary accurate modeling results and the optimal center values and shapes of membership functions but also can extract the implication fuzzy rules or correct



(a)



(b)

Fig. 17 Simulation results of the modified neuro - fuzzy identifiers that modifies the initial fuzzy rules of the system (a) Model 1, Model 2.

the fuzzy rules of the system. Fig. 18 compares the learning speed of the case of training center points and variances of input membership functions with that of using the fixed values of them. Solid and dashed lines represent the total errors at each iteration in case of training the input membership functions and of having fixed values of the parameters in input membership functions, respectively. Additional learning of the input membership functions gives a faster learning speed and smaller error values

Table 2 Modified fuzzy rules of models 1 and 2(a) Model 1, (b) Model 2.

Y(k) \ U(k)	NB	NM	Z	PM	PB
NB	NB	NB	NB	NM	NB
NM	NB	NB	Z	Z	Z
Z	Z	Z	Z	PM	PM
PM	Z	Z	PM	PB	PB
PB	PM	PB	PB	PB	PB

(a)

Y(k) \ U(k)	NB	NM	Z	PM	PB
NB	NM	NM	NM	NM	NM
NM	NM	NM	Z	Z	Z
Z	Z	NM	Z	Z	Z
PM	Z	Z	Z	PM	PM
PB	PB	PB	PB	PB	PB

(b)

Table 3 Created fuzzy rules of models 1 and 2 (a) Model 1, (b) Model 2.

Y(k) \ U(k)	NB	NM	Z	PM	PB
NB	NB	NB	NB	NM	NB
NM	NB	NB	Z	Z	Z
Z	Z	Z	Z	PM	PM
PM	Z	Z	PM	PB	PB
PB	PM	PB	PB	PB	PB

(a)

Y(k) \ U(k)	NB	NM	Z	PM	PB
NB	NM	NM	NM	Z	Z
NM	NM	NM	Z	Z	Z
Z	Z	NM	Z	Z	Z
PM	Z	Z	Z	PM	PM
PB	PB	PB	PB	PB	PB

(b)

than the other case.

3. Discussion

the proposed neuro - fuzzy identifier gives a very accurate fuzzy modeling of unknown sys-

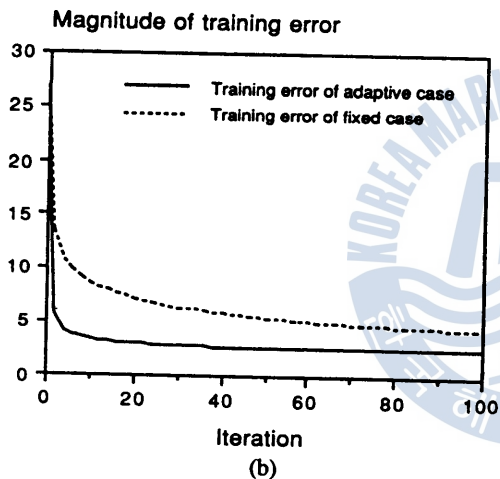
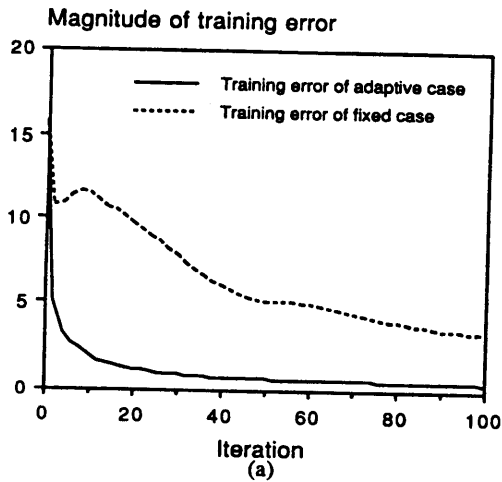


Fig. 18 Comparison of the learning speed of the case of training the center points and variances of input membership functions and that of using fixed values of them (a) Model 1, (b) Model 2.

tems and has optimal fuzzy variables, such as the center points and shapes of input and output fuzzy membership functions. Also, the modified neuro - fuzzy network can learn a priori knowledge of the system. Studies on the design method of intelligent controller using neuro - fuzzy identifier is under investigation.

References

- 1) S. Shao, "Fuzzy self-organizing controller and its application for dynamic processes," *Fuzzy Sets Syst.*, Vol. 26, pp. 151 - 164, 1988.
- 2) E. M. Scharf and N. J. Mandic, "The application of a fuzzy controller to the control of a multi-degree - freedom robot arm," *Industry Application of Fuzzy Control*, pp. 41 - 62, 1989.
- 3) Y. F. Li and C. C. Lan, "Development of fuzzy algorithm for servo systems," *IEEE Contr. Syst. Mag.*, pp. 65 - 72, 1989.
- 4) T. Yamakawa, "Stabilization of an inverted pendulum by a high - speed fuzzy logic controller hardware system," *Fuzzy Sets and Systems*, Vol. 32, pp. 161 - 180, 1989.
- 5) H. Takagi, "Fusion technology of fuzzy theory and neural networks - survey and future directions," *Procs. of the Intr. Conf. on Fuzzy Logic and neural Networks*, pp. 13 - 26, 1990.
- 6) R. Tanscheit and E. M. Scharf, "Experiments with the use of a rule - based selforganizing controller for robotics applications," *Fuzzy Sets Syst.*, Vol. 26, pp. 195 - 214, 1988.
- 7) M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets Syst.*, Vol. 28, pp. 15 - 33, 1988.
- 8) T. Tzkgagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 15, No. 1, pp. 166 - 182, 1985.
- 9) J. S. R. Jang, "Self learning fuzzy controllers based on temporal back propagation," *IEEE Trans. on Neural Networks*, Vol. 3, No. 5, pp. 714 - 723, 1992.
- 10) S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On the fuzzy modeling using fuzzy neural networks with the back - propagation algorithm," *IEEE trans. on Neural Networks*, Vol. 3, No. 5, pp. 801 - 806, 1992.
- 11) M. Mukaidono and M. Yamaoka, "A learning method of fuzzy inference rules with neural networks and its application," *Procs. of the Intr. Conf. on Fuzzy Logic and Neural Networks*, pp.

- 185 - 187, 1992.
- 12) T. Yamakawa, E. Uchino, T. Miki, H. Kusanagi, "A neo fuzzy neuron and its application to system identification and prediction of the system behavior," *Procs. of the 2nd Inter. Conf. on Fuzzy Logic and Neural networks*, Vol. 1, pp. 477 - 484, 1992.
 - 13) W. Pedricz, "Numerical and application aspects of fuzzy relational equations," *Fuzzy Sets and Systems*, Vol. 11, pp. 1 - 18, 1993.
 - 14) M. Lee, S. Y. Lee, and C. H. Park, "A new neuro - fuzzy identification model of nonlinear dynamic systems," *Intr. J. of Approximate Reasoning*, Vol. 10, No. 1, pp. 29 - 44, 1994.
 - 15) M. Lee, S. Y. Lee, and C. H. Park, "Neuro - fuzzy identifiers and controlers," *Journal of Intelligent and Fuzzy systems*, Vol. 2, No. 1, pp. 1 - 14, 1994.
 - 16) M. Lee, S. Y. Lee, and C. H. park, "Neuro - uzzzy identification model of fuzzy system," *Proc. of Korea - Japan joint Conf. on fuzzy Stsytems and Engineering*, pp. 145 - 148, 1992.
 - 17) C. H. Park, "Neuro - fuzzy information processing," *Proc. of 91' Korean Scientist and Engineer Summer Symposium*, pp. 90 - 94, 1991.
 - 18) K. Hornik, "Approximation capabilities of multi-layer feedforward networks," *Neural networks*, Vol. 4, No. 2, pp. 251 - 257, 1991.
 - 19) E. B. Baum and D. Hausser, "What size net gives valid generalization?," *Neural Information Processing Systems*, Vol. 1, pp. 81 - 90, 1989.
 - 20) D. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representation by error propagation," *Parallel Distributed Processing*, D. Rumelhart and J. McClelland(Eds.), Vol. 1, MIT Press, 1986.

