Journal of the Korean Society of Marine Engineers, Vol.28, No.2, pp.228~236, 2004(ISSN 1226-9549)

A Study on the Neuro-Fuzzy Control and Its Application

Myung-Ok So† · Heui-Han Yoo* · Sun-Ho Jin**

(Manuscript : DEC 3, 2003 ; Revised JAN 4, 2004)

Abstract : In this paper, we present a neuro-fuzzy controller which unifies both fuzzy logic and multi-layered feedforward neural networks. Fuzzy logic provides a means for converting linguistic control knowledge into control actions. On the other hand, feedforward neural networks provide salient features, such as learning and parallelism. In the proposed neuro-fuzzy controller, the parameters of membership functions in the antecedent part of fuzzy inference rules are identified by using the error backpropagation algorithm as a learning rule, while the coefficients of the linear combination of input variables in the consequent part are determined by using the least square estimation method. Finally, the effectiveness of the proposed controller is verified through computer simulation for an inverted pole system

Key words : Neuro-fuzzy control, Fuzzy rule, Membership function, Fuzzy inference, Multi-layered feedforward networks, Back propagation rule

1. Introduction

Automatic Control problems have been implemented rapidly for realization in the many kinds of industrial fields with the advanced technology of science. Most of the conventional approaches in the design of an automatic control system often involve the construction of a mathematical model describing the dynamic behavior of the plant to be controlled. Usually, such a mathematical model consists of a set of linear or nonlinear differential equation, most of which are derived using some form of approximation and simplification.

The traditional model based control techniques break down, however, when a representative model is difficult to obtain due to uncertainty or complexity. So, the optimal control and MRAC(Model Reference Adaptive Control) schemes are adopted in order to compensate the traditional method. But, these kinds of control theories have the difficulties in the complicated control theory to the change of parameter depending on the

† Comesponding Author(Department of Mechatronics Engineering, Korea Maritime University), E-mail : smo@mail.hhu.ac.kr, Tel : 051)410-4248
* Department of Mechatronics Engineering, Korea Maritime University
** Department of Marine Engineering, Graduate School, Korea Maritime University

change of control environment, adaptability to the disturbance and system non-linearity[1],[2].

Recently, intelligent controller[3] which have the thinking ability and adaptability human-like have been studied to solve above mentioned difficulties. For instance, there is neural network fuzzy controller by fuzzy logic and neural network method. Fuzzy control system has more degrees of freedom than other kinds of controller to be designed on the basis of proper modeling or PID controller, and has used as the logic controller using linguistic control rule[4] in the many practical fields[5]-[7]. Generally, the fuzzy controller consists of fuzzification interface, knowledge base, fuzzy inference engine and defuzzification interface. Fuzzy control can be considered as a special type of rule based control system. But, the simple fuzzy control algorithm above mentioned is depend on the knowledge or experience of experts with respect to the rule base and data base of fuzzy logic controller. Therefore, the optimal design of fuzzy controller is one of the difficult problems. Because the combination of fuzzy logic with the function of inference and multi-layered feedforward network with the function of learning and adaptability grants flexibility to the control system[8]-[10], recently these control techniques have been successfully applied to control of complex and ill-defined systems.

In this paper, we present a neuro-fuzzy controller which unifies both the advantages of fuzzy logic and multi-layered feedforward neural networks. This neuro-fuzzy controller adapts the learning function of multi-layered feedforward network to the fuzzy controller. Hybrid learning rule which is combination of error back-propagation learning rule and least squares estimation is adopted as a learning method. The parameters of membership functions in the antecedent part of fuzzy rules are identified by using the error back-propagation algorithm as a learning rule, while the coefficients of the linear combination of input variables in the consequent part are determined by using the least squares estimation method. Finally, the effectiveness of the proposed controller is verified through computer simulation for an inverted pole system.

2. Neuro-fuzzy controller

2.1 Design of neuro-fuzzy controller

Fuzzy control is a special control technique on the basis of fuzzy set theory, "IF-THEN" rule and fuzzy inference. Recently, this technique has been successfully applied to control of complex and ill-defined systems in a wide variety of areas, such as robot, water purification, automatic train operation system and automatic container crane operation system, etc. Fig. 1 shows fuzzy controller which consists of four main parts such as fuzzification interface, knowledge base, fuzzy inference engine and defuzzification interface. The function of fuzzification interface is to transform input data to fuzzy sets properly. The knowledge base consists of data bases which have the information on fuzzy sets, the boundary of the fuzzy space and "IF-THEN" rule bases. The function of fuzzy inference engine is to infer the fuzzy output from fuzzy input using fuzzy "IF-THEN" rules. The function of defuzzification interface is to transform fuzzy output value to the crisp value. Among the above mentioned parts, the fuzzy inference engine is a core of fuzzy controller which imitates a human-like inference process.

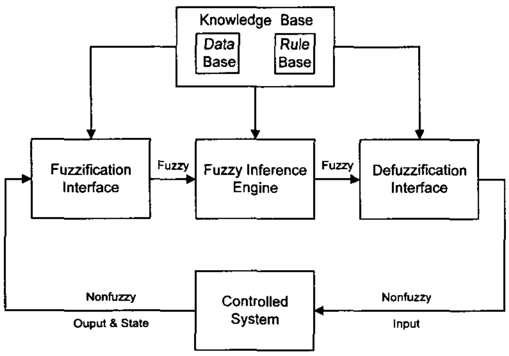


Fig. 1 Basic structure of a fuzzy logic controller

In the case of the fuzzy controller for multi-input single output, fuzzy IF-THEN rule R_i is expressed as following.

$$R_i: \text{IF } x_1 \text{ is } A_{1,i} \text{ and} x_2 \text{ is } A_{2,i} \text{ and} \cdots \text{and} x_n \text{ is } A_{n,j}$$
$$\text{THEN } z \text{ is } C_i \qquad (2.1)$$

Where, $R_i(i=1,2,\cdots,r)$ means fuzzy control rule, and r means the number of fuzzy control rules. $x_j(j=1,2,\cdots,n)$ and z are linguistic variables which take a certain value within $X_j(j=1,2,\cdots,n)$ and Z respectively. $A_{j,i}(j=1,2,\cdots,n,\ ;\ i=1,2,\cdots,r)$ and C_i are fuzzy sets to be characterized by the membership functions such as triangular type, bell type, sigmoidal type and trapezoidal type. In general, the antecedent part of fuzzy rule in fuzzy controller is partitioned, and the consequent part of fuzzy rule has been studied diversely.

In the Mandani-fuzzy model, the consequent part is a membership function. In the Tsukamoto-fuzzy model it has a membership function of a exponential type. Also, in the Sugeno-fuzzy model it is divided into two ways, namely, zero order fuzzy model and first order fuzzy model.

This paper adopted first-order Sugeno model. When considering general multi-input single output(MISO) fuzzy system, fuzzy "IF-THEN" rule R_i can be described as follows.

$$R_i: \text{IF } x_1 \text{ is } A_{1,i} \text{ and } x_2 \text{ is } A_{2,i} \text{ and} \cdots \text{and } x_n \text{ is } A_{n,i}$$
$$\text{THEN } u_i = \sum_{j=1}^{n} (p_{i,j}\ x_j + p_{i,0}) \qquad (2.2)$$

Where, u_i is a ouput of fuzzy rule $R_i(i=1,2,\cdots,r)$, $p_{i,j}$ and $p_{i,0}$ are parameters that exist in the consequent parts. Fig. 2 shows structure of a neuro-fuzzy control system.

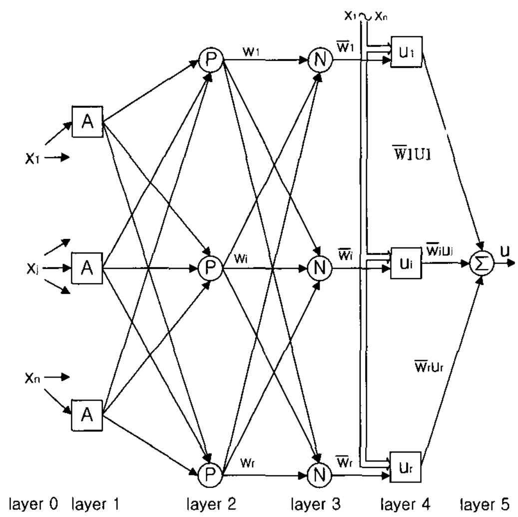


Fig. 2 Structure of a neuro-fuzzy controller

In this paper, fuzzy inference process is followed below.

layer 1 : In this layer, every node i is an adaptive node with a node output defined by equation(2.3). The parameters in the nodes of this layer are referred to as antecedent parameters.

$$O^1_i = A_{j,i}(x_j) \tag{2.3}$$

where x_j is the input to the node , i in this layer, and $A_{j,i}(x_j)$ is a membership function of a linguistic variables related to these node functions such as "positive big", "zero", and "negative big". In other words, outputs of this layer are a certain value of membership function in the antecedent part. The membership functions such as triangular type, bell type and trapezoidal type can be used as the node functions in this layer. For instance, $A_{j,i}(x_j)$ of bell type membership function like equation (2.4) can be used.

$$A_{j,i}(x_j) = \frac{1}{1+\left(\frac{x_j - c_i}{a_i}\right)^{2b_i}} \tag{2.4}$$

where, $\{a_i, b_i, c_i\}$ is the parameter set of membership function in the antecedent part. By adjusting the above parameter, membership function for linguistic label $A_{j,i}$ can be transformed into some kind of bell type membership function.

layer 2 : In this layer, every node is a fixed node labeled Ⓟ, which multiplies the input signals and outputs the product.

$$O^2_i = w_i = A_{1,i}(x_1) \cdot A_{2,i}(x_2) \cdots A_{n,i}(x_n) \tag{2.5}$$

layer 3 : In this layer, every node is a fixed node labeled Ⓝ. Node output in this layer is equation(2.6).

$$O^3_i = \overline{w_i} = \frac{w_i}{\sum_{i=1}^{n} w_i} \tag{2.6}$$

layer 4 : In this layer, every node is an adaptive node labeled ⓤ . Node output in this layer is equation(2.7)

$$O^4_i = \overline{w_i} \cdot u_i = \overline{w_i} \cdot \sum_{j=1}^{n} (p_{i,j} x_j + p_{i,0}) \tag{2.7}$$

where, $\overline{w_i}$ is the output of layer 3, $\{p_{i,j}, p_{i,0}\}$ is the parameter set in the consequent part.

layer 5 : single node in this layer is a fixed node labeled Σ. Overall output is summation of all input signals. Output is expressed by the equation (2.8)

$$O^5_i = \sum_{i=1}^{r} \overline{w_i} \cdot u_i = \frac{\sum_{i=1}^{r} w_i \cdot u_i}{\sum_{i=1}^{r} w_i} \tag{2.8}$$

2.2 Neural network and hybrid learning rule.

If feedforward adaptive network has L layers, and layer $l(l=0,1,2,\cdots L)$ have $N(l)$ nodes, then the output and function of node $i(i=1,2,\cdots,N(l))$ in the layer l can be expressed as O^l_i and g^l_i, respectively, where $l=0$ and $l=L$ mean input and output layer. Since the node output depends on the input signals and the parameter set of the node, the node function g^l_i can be expressed by equation(2.9).

$$O_i^l = g_i^l(O_1^{l-1}, \cdots, O_{N(l-1)}^{l-1}, \theta_1, \theta_2, \cdots) \quad (2.9)$$

Where, θ_1 and θ_2 are the parameters of this node. Suppose that the number of pairs of given training data is P_1, and an error index for the Pth $(1 \le p \le P)$ pair of the training data is defined as equation(2.10). Total error index leads to equation (2.11)

$$E_p = \sum_{k=1}^{N(L)} (d_{k,p}^L - O_{k,p}^L)^2 \quad (2.10)$$

$$E = \sum_{p=1}^{P} E_p = \sum_{p=1}^{P} \sum_{k=1}^{N(L)} (d_{k,p}^L - O_{k,p}^L)^2 \quad (2.11)$$

Where, $d_{k,p}^L$ is k th component of the p th desired output vector and $O_{k,p}^L$ is the k th component of the actual output vector. We define the error signal $e_{i,p}^l$ as the derivative of the error index E_p with respect to the output of node i in the layer l. The error signal for the i th output node at layer l can be calculated directly.

$$e_{i,p}^l = \frac{\partial E_p}{\partial O_{i,p}^l} \quad (2.12)$$

Also, the variation rate $e_{i,p}^L$ of the error signal at output layer L can be calculated by equation (2.13).

$$e_{i,p}^L = \frac{\partial E_p}{\partial O_{i,p}^L} = -2(d_{i,p}^L - O_{i,p}^L) \quad (2.13)$$

For the internal node at the ith position of layer l, the variation rate of error signal can be derived by the chain rule.

$$e_{i,p}^l = \frac{\partial E_p}{\partial O_{i,p}^l} = \sum_{m=1}^{N(l+1)} \frac{\partial E_p}{\partial O_{m,p}^{l+1}} \cdot \frac{\partial O_{m,p}^{l+1}}{\partial O_{i,p}^l}$$

$$= \sum_{m=1}^{N(l+1)} e_{m,p}^{l+1} \frac{\partial O_{m,p}^{l+1}}{\partial O_{i,p}^l} \quad (2.14)$$

Where, $o \le l \le L-1$. Namely, the error signal variation rate of an internal node at layer l can be expressed as a linear combination of the error signal variation rates of the internal nodes at layer $(l+1)$. Therefore, for arbitrary layer l and $i(0 \le l \le L, 1 \le i \le N(l))$, we can find by first $e_{i,p}^l$ applying equation (2.13) to get error signals at the output layer, and then applying equation (2.14) iteratively until we reach the desired layer l. Since the error signals are obtained sequentially from the output layer back to the input layer, this is called the back-propagation learning rule. The gradient vector is defined as the derivative of the error index with respect to each parameter. If θ is a parameter of ith node at layer l, we have equation (2.15).

$$\frac{\partial E_p}{\partial \theta} = \frac{\partial E_p}{\partial O_{m,p}^l} \cdot \frac{\partial O_{m,p}^l}{\partial \theta}$$

$$= e_{m,p}^l \frac{\partial E_p}{\partial O_{i,p}^l} \quad (2.15)$$

Therefore, the derivative of overall error index E with respect to θ is equation (2.16), the update formula for the generic parameter is equation (2.17) with respect to

$$\frac{\partial E}{\partial \theta} = \sum_{p=1}^{P} \frac{\partial E_p}{\partial \theta} \quad (2.16)$$

$$\Delta\theta = -\eta \frac{\partial E}{\partial \theta} \tag{2.17}$$

Where, η is learning rate.

In this paper, hybrid learning rule[11] which combines the gradient method(GM) and the least squares estimation(LSE) [12] is adopted in order to identify parameters. Assuming that multi-layered feedforward network has only one output variable by input variables and parameters, and the values of nonlinear parameters are given, matrix equation like equation(2.18) is obtained by using P pairs of training data.

$$A\theta = B \tag{2.18}$$

Where, A and B are matrices of dimension $P\times M$ and $P\times 1$ respectively. θ is an unknown parameter vector of dimension $M\times 1$ in the set θ of linear parameters, P and M indicate the number of pairs of training data and linear parameters respectively. This P is generally bigger than M. Therefore, there is a certain solution of θ to satisfy equation(2.18). Hence, and instead of θ, θ^* of equation (2.19) which minimizes the squares of error, $\| A\theta - B \|^2$ is used.

$$\theta^* = (A^T A)^{-1} A^T B \tag{2.19}$$

3. Simulation and analysis

3.1 An inverted pole system

An inverted pole system like Fig. 3 can be expressed as nonlinear system equation (2.20).

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{g\sin x_1 - \frac{ml}{M+m} x_2^2 \sin x_1 \cos x_1 - \frac{u\cos x_1}{M+m}}{l\left(\frac{4}{3} - \frac{m\cos^2 x_1}{M+m}\right)} \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{u - ml(\dot{x}_2 \cos x_1 - x_2^2 \sin x_1)}{M+m} \\
y &= x_1
\end{aligned} \tag{2.20}$$

Where, x_1[rad] is angular position of the pole to vertical line, x_2[rad/s] is angular velocity, x_3[m] is cart position, x_4[m/s] is cart velocity, M[kg] is mass of cart, m [kg] is mass of pole, l[m] is half length of pole, g[m/s^2] is gravity acceleration and u[N] is force.

The cart and pole can be moved on the rail tracks to its right or left depending on the force, the pole is hinged to a cart through a free one degree of freedom. The purpose of control for inverted pole system is to move to the desired position maintaining the balance of the pole when the force u is applied to the cart. If membership functions $A_{1,i}$, $A_{2,i}$, $A_{3,i}$ and $A_{4,i}$ corresponding to fuzzy variables x_1, x_2, x_3 and x_4 of the antecedent part are partitioned by the two kinds of N(Negative) and P(positive) bell type, the number of node in each layer l is as follows.

When l is 0 to 5, $N(l)$ are $N(0)=4$, $N(1)=8$, $N(2)=16$, $N(3)=16$, $N(4)=16$ and $N(5)=1$. Therefore, the number of total nodes including input and output layer is 61, and the number of total rules is 16, and the number of parameters of in antecedent part is 24, and the number of parameters in consequent part is 80. So, u_i equals $\sum_{j=1}^{4}(p_{i,j}x_j+p_{i,0})$.

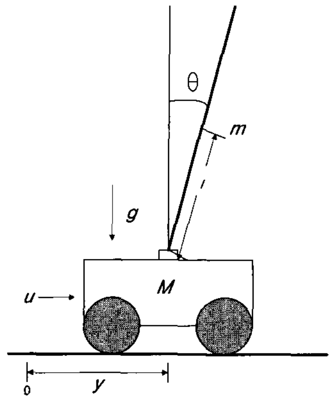


Fig. 3 An inverted pole system

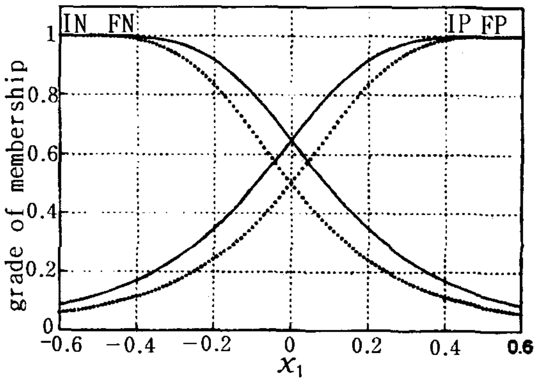


Fig. 4 Membership function of the term sets over pole angle x_1

Fig. 4 indicates initial membership function(IN, IP) before learning and final membership function(FN, FP) after learning.

3.2 Simulation and analysis

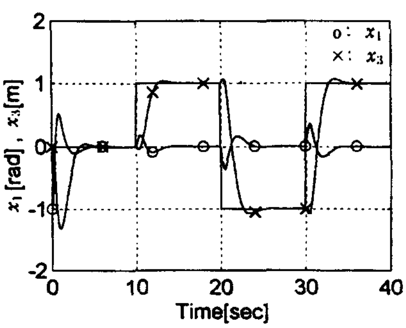
The values to be used in simulation are as follows.

$M=1$[kg], $m=0.1$[kg], $2l=1$[m], $g=9.8$[m/s^2].

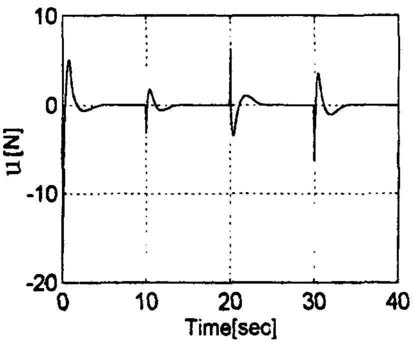
Table 1 The values of linear parameters in the consequent part after learning

Antecedents					Consequents[$p_{i,j}$]				
					j				
i	x_1	x_2	x_3	x_4	0	1	2	3	4
1	N	N	N	N	-14.37	30.13	4.46	11.18	7.04
2	N	N	N	P	0.89	44.08	11.38	2.42	4.74
3	N	N	P	N	-12.09	26.91	12.09	4.85	3.24
4	N	N	P	P	6.68	40.09	7.75	2.01	2.06
5	N	P	N	N	7.81	44.40	15.09	6.64	7.11
6	N	P	N	P	-9.66	11.80	4.13	-1.12	1.49
7	N	P	P	N	-2.23	41.07	8.36	3.90	3.34
8	N	P	P	P	-3.66	10.03	3.72	-5.85	-1.68
9	P	N	N	N	3.66	10.03	3.72	-5.85	-1.68
10	P	N	N	P	2.23	41.07	8.36	3.90	3.34
11	P	N	P	N	9.66	11.80	4.13	-1.12	1.49
12	P	N	P	P	-7.81	44.40	15.09	6.64	7.11
13	P	P	N	N	-6.68	40.09	7.75	2.01	2.06
14	P	P	N	P	12.09	26.91	12.09	4.85	3.24
15	P	P	P	N	-0.89	44.08	11.38	2.42	4.74
16	P	P	P	P	14.37	30.13	4.46	11.18	7.04

Fig. 5(a) shows the angle of pole and the position of cart. Where, x_1 [rad] indicates angle of inverted pole, and x_3[m] indicates the position of cart. Before control of cart with pole, suppose that the position of cart is in the center of the rail, and the inverted pole is inclined left side from the vertical line by 0.5[rad]. If the control starts, the cart moves toward left side, then moves to initial starting point again keeping the vertical state of pole, after 10 seconds, the cart moves to the right side until reaching 1 meter from the starting point, after 20 seconds, the cart moves to left side until reaching 2 meter from previous cart position, after 30 seconds, the cart moves to right side until reaching 2 meter from the previous cart position.



a) responses of pole angle x_1 and cart position x_2



(b) Control input u

Fig. 5 Simulation results

Fig. 5(b) shows the force to be applied to the cart in case of above mentioned control of cart with pole. For the more understanding, x_1 in the Fig. 5(a) express 2 times of actual values.

4. Conclusion

In this paper, Neuro-fuzzy controller (NFC) which combined the strong point of fuzzy logic and multi-layered feedforward network was suggested. We decided the parameters of membership function in antecedent part and linear parameters in consequent part through the hybrid learning rule. Hybrid learning method which combines the error back-propagation learning rule and least squares estimation method was used. Finally, as the results of application of the proposed controller to the control of a inverted pole system, we confirmed that a satisfactory response characteristics were obtained, and the proposed method also had a robustness in case of variation of control environment.

References

[1] M. Drouin, H. H. Kandil and M. Marition, Control of Complex System, Plenum Press, 1991.

[2] R. E. Skelton, Dynamic Systems Control, John Wiley & Sons, 1988.

[3] G. G. Jin, Intelligent Fuzzy Logic Control of Processes with Time Delays, Ph.D. Thesis, University of Wales Cardiff, United Kingdom, 1995.

[4] B. P. Graham and R. B. Newell, Fuzzy Identification and Control of a Liquid Level Rig , Fuzzy Sets and Syst., Vol. 26, pp. 255-273, 1988.

[5] M. Sugeno and M. Nishida, Fuzzy Control of Model Car , Fuzzy Sets and Syst., Vol. 16, pp. 103-113, 1985.

[6] Y. F. Li and C. C. Lau, Developement of Fuzzy Algorithms for Servo Systems, IEEE Control Syst. Mag., pp.65-72, 1989.

[7] J. A. Bernard, Use of a Rule-based System for Process Control, IEEE Control Syst. Mag., pp. 3-13, 1988.

[8] Y. Dote, Fuzzy and Neural Network Controller, Proc. of IECON'90, Vol. 2, pp. 1315-1343, 1990.

[9] S. Horikawa, T. Furuhasi, S. Okuma and Y. Uchikawa, Composition Methods of Fuzzy Neural Networks, Proc. of IECON ′90, Vol. 2, pp. 253-1258, 1990.

[10] T. Takagi and M. Sugeno, Fuzzy Identification of Systems and its Applications to Modeling and Control, IEEE Trans. Systems, Man. and Cybernetics, Vol. 15, pp. 116-132, 1985.

[11] J. S. R. Jang, Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm , Proc.9th Nat. Conf.on Aritf. Intell.(AAAI-91), pp. 762-767, July 1991.

[12] T. C. Hsia, System Identification : Least-Squares Methods , New York : Heath, 1977.