'92 KACC 1992. 10. 19~21

# A Neuro-Fuzzy Adaptive Controller

Hee Tae Chung, Hyun Cheol Lee, and Gi Joon Jeon

Department of Electronics, Kyungpook National University, Taegu, Korea and
Engr. Research Center for Adv. Control and Instr. of SNU by KOSEF

ABSTRACT

This paper proposes a neuro-fuzzy adaptive controller which includes the procedure of initializing the identification neural network(INN) and that of learning the control neural network(CNN). The identification neural network is initialized with the informations of the plant which are obtained by a fuzzy controller and the conrtol neural network is trained by the weight informations of the identification neural network during on-line operation.

## 1. INTRODUCTION

Modern control theory has been successfully applied to a plant which is are well defined by mathematical equations. However, when the structure of the plant is unknown, or the plant includes model uncertainties such as parameter variations or nonlinearities, the effectiveness of modern control theory diminishes. Furthermore, goals and constraints in control problems are easier to be represented by linguistic variables than numerical values. In those situations, fuzzy set theory, introduced by Zadeh in 1965[1], can be effective means of dealing with such linguistically specified objectives. The linguistic terms such as small and large may be defined as fuzzy sets. Therefore, by incoporating linguistic control rules and the fuzzy set theory the control objectives in some cases can be achieved more effectively than by the modern control theory. When fuzzy controllers are designed, the human experience is used and a set of heuristic rules are implemented insted of mathematical models.

In these days, there is a tendency to construct controllers using some kind of intelligent algorithm[2]. When intelligent algorithms are mentioned, fuzzy reasoning and neural networks are involved here and the reason is because they have the capability to learn from experience. Moreover, to avoid the modeling difficulties, a number of multilayered neural networks based controllers have been proposed[3,4]. Anderson et al.[5] assumed the plant dynamics were unknown and used unsupervised learning to stabilize a cart-pole system. Geuz and Selinsky[6] applied error back propagation in the design of a trainable adaptive controller. Ku and Lee[7] proposed diagonal recurrent neural networks for nonlinear system control. However, in these neuro-controller design methods, they can utilize little or no prior knowledge of the plant dynamics. Recently Selinsky et al.[8] employed a priori structural knowledge of robot dynamics in a neurocontroller design.

In this paper it is assumed that making the best use of existing knowledge will improve the control performance and we present an on-line control algorithm for the control neural network. We use a priori knowledge of the plant as the linguistic descriptions, which are implemented by a fuzzy controller. And the identification neural network is trained with the data of the plant which are derived by the fuzzy controller. Then, the control neural network learns control strategy through the weight informations of the identification neural network. The proposed adaptive control system shows a good performance on the control of linear and nonlinear plants.

## 2. FUZZY LOGIC CONTROLLER

Based on expert's experiences and engineering judgements, linguistic rules may be used as a specification of control laws in real control problems. Here, we use general linguistic rules to achieve control objectives of the plant by computer simulations. To estabilish the fuzzy controller it is necessary to interpret rules and quantize the qualitative statement. Following linguistic sets are assigned[9]:

| | |
|---|---|
| POSITIVE LARGE | ( PL ) |
| POSITIVE SMALL | ( PS ) |
| ZERO | ( ZE ) |
| NEGATIVE SMALL | ( NS ) |
| NEGATIVE LARGE | ( NL ) |

The next step is to define the membership function of the linguistic sets. Table 1 is a membership matrix for a membership function. The membership matrix consists of five linguistic sets, including PL, PS, ZE, NS, and NL, and each set consists of nine elements of quantized levels. All errors, error changes, and control input variables are quantized to these nine levels.

Table 1. Member-ship matrix

| LINGUISTIC SETS | QUANTIZED LEVELS | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 |
| PL | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.6 | 1.0 |
| PS | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.6 | 1.0 | 0.6 | 0.0 |
| ZE | 0.0 | 0.0 | 0.0 | 0.6 | 1.0 | 0.6 | 0.0 | 0.0 | 0.0 |
| NS | 0.0 | 0.6 | 1.0 | 0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| NL | 1.0 | 0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

A fuzzy control algorithm consists of situation and action pairs[10]. Conditional rules expressed in IF and THEN statements are generally used. In tracking control problems, the objective is to minimize the tracking error($e_c[k]$). The control input is a function of error. Also the change of tracking error($e_r[k]$) is inportant in determining the control input. Therefore, ER and EC are selected as linguistic variables to be used in the premise of control rules and are defined as

$$e_c[K] = ( SP[K] - Y[K] ) \quad (1)$$

$$e_r[k] = ( e_r[K] - e_r[K-1] ) \quad (2)$$

$$ER = e_c * GR \quad (3)$$

$$EC = e_r * GC \quad (4)$$

where k is time step, Y[k] the plant output, SP[k] a reference signal, and GR and GC constant gain. The consequent of each control rule is defined in terms of the control input U[k]. If we consider the control rules and the fuzzy reasoning as a nonlinear mapping from ER[k] and EC[k] to U[k], U[k] can be represented as

$$U[k] = f(ER, EC) \quad (5)$$

The control laws U[k] are described by linguistic rules which are shown in table 2. Those rules are the general fuzzy control rules which will be used in simulations later.

Table 2. Rule base for the fuzzy controller

| ER EC | NL | NS | ZE | PS | PL |
|---|---|---|---|---|---|
| NL | NL | NL | NS | NS | ZE |
| NS | NL | NS | NS | ZE | PS |
| ZE | NS | NS | ZE | PS | PS |
| PS | NS | ZE | PS | PS | PL |
| PL | ZE | PS | PS | PL | PL |

From the fuzzy control algorithm, the control input U[k] is obtained from the following equation:

$$U[k] = \frac{\sum_{i=1}^{n} w_i * u_i}{\sum_{i=1}^{n} w_i} * GU \quad (6)$$

where $w_i$ and $u_i$ denote the weights of the implication and the inference outcome of each control rule respectively, and GU is the output scaling factor of control action.

3. THE STRUCTURE OF NEURO-FUZZY ADAPTIVE CONTROL

Initially the fuzzy logic controller(FLC) operates and INN is trained with input-output values of the plant. The control and learning structure is shown in Fig. 1.

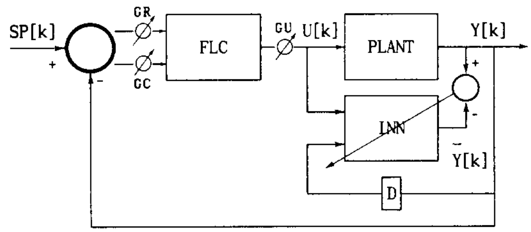


Fig. 1. Fuzzy control and learning structure.

The feedforward network is a static mapping and is not convenient in representing a dynamic response in time domain. Hence, we use a diagonal recurrent neural network which is believed to be more suitable for a dynamical system than the feedforward network[11]. The structure of the dynamical recurrent neural network is shown in Fig. 2.

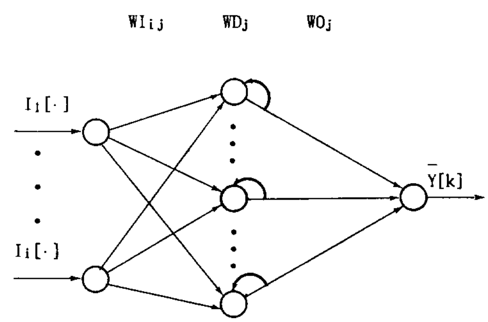


Fig. 2. Diagonal recurrent neural networks

To train the identification neural network, we define the error function and error for a training cycle as follows:

$$E_I = \frac{1}{2} ( Y[k] - \bar{Y}[k] )^2 \quad (7)$$

$$e_m = Y[k] - \bar{Y}[k] \quad (8)$$

where $\bar{Y}[k]$ is the output of the neural network and $e_m$ is the error between the plant and the neural network. The mathematical model for diagonal recurrent neural network is shown below:

$$\bar{Y}[k] = F(W, Y[k-1], \cdots, Y[k-n], U[k-1], \cdots, U[k-m]) \quad (9)$$

$$\bar{Y}[k] = \sum_j WO_j \cdot X_j[k] \quad (10)$$

$$X_j[k] = f(S_j[k]) \quad (11)$$

$$S_j[k] = WD_j \cdot X_j[k-1] + \sum_i WI_{ij} \cdot I_i[k] \quad (12)$$

where $I_i[k]$ is the $i^{th}$ input to the diagonal recurrent neural network, $S_j[k]$ is the sum of inputs to the $j^{th}$ recurrent neuron, and $X_j[k]$ is the output of the $j^{th}$ recurrent neuron. Here $f(\cdot)$ is the sigmoid function, and $WI_{ij}$, $WD_j$, $WO_j$ are input, recurrent, and output weight, respectively. The weights can be adjusted by the following steepest descent method.

$$WO_j(n+1) = WO_j(n) + \eta_I \left( -\frac{\partial E_I}{\partial WO_j} \right) + \alpha \Delta WO_j(n) \quad (13)$$

$$WD_j(n+1) = WD_j(n) + \eta_I \left( -\frac{\partial E_I}{\partial WD_j} \right) + \alpha \Delta WD_j(n) \quad (14)$$

$$WI_{ij}(n+1) = WI_{ij}(n) + \eta_I \left( -\frac{\partial E_I}{\partial WI_{ij}} \right) + \alpha \Delta WI_{ij}(n) \quad (15)$$

where $\eta_I$ is the learning rate, $\alpha$ is the momentum factor, and $\Delta W(n)$ represents the change of weights in the $n^{th}$ iteration. The gradient of error in eqn. (7) with respect to each weight is represented by:

$$-\frac{\partial E_I}{\partial WO_j} = e_m \cdot X_j[k] \quad (16)$$

$$-\frac{\partial E_I}{\partial WD_j} = e_m \cdot WO_j \cdot \frac{\partial X_j[k]}{\partial WD_j} \quad (17)$$

$$-\frac{\partial E_I}{\partial WI_{ij}} = \sum_j e_m \cdot WO_j \cdot \frac{\partial X[k]}{\partial WI_{ij}} \quad (18)$$

$$\frac{\partial \bar{Y}[k]}{\partial WD_j} = WO_j \cdot f'(S_j[k])\left( X_j[k-1] + WD_j \cdot \frac{\partial X_j[k-1]}{\partial WD_j} \right),$$

$$\frac{\partial X_j[0]}{\partial WD_j} = 0.0 \quad (19)$$

$$\frac{\partial \bar{Y}[k]}{\partial WI_{ij}} = \sum_j WO_j \cdot f'(S_j[k])\left( I_i[k] + WD_j \cdot \frac{\partial X_j[k-1]}{\partial WI_{ij}} \right),$$

$$\frac{\partial X_j[0]}{\partial WI_{ij}} = 0.0 \quad (20)$$

After the INN is sufficiently learned with the informations of the plant which are generated by the fuzzy controler, the INN and the CNN are trained concurrently in on-line operation. The final control structure is shown in Fig. 3.

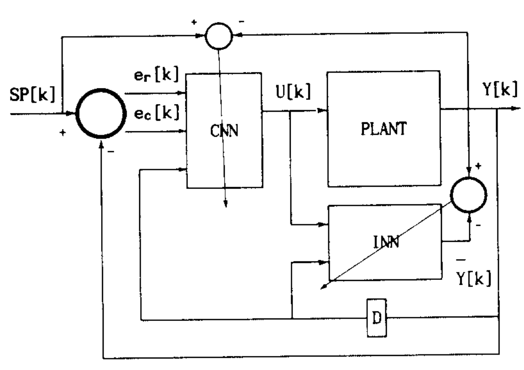


Fig. 3. The Structure of the adaptive control system.

We define the error function and control error of the CNN to train the CNN as follows:

$$E_C = \frac{1}{2} ( SP[k] - Y[k] )^2 \quad (21)$$

$$e_c[k] = SP[k] - Y[k] \quad (22)$$

From pre-learning of the INN, we use the relation of $Y[k] \cong \bar{Y}[k]$. Therefore, the CNN is trained with the weights information of the INN. The structure and learning method of the CNN are same as the INN. Hence, we define the gradient of error in eqn. (21) with respect to the CNN weights as follows:

$$-\frac{\partial E_C}{\partial CWO_j} = e_c[k] \cdot \frac{\partial Y[k]}{\partial U[k]} \cdot \frac{\partial U[k]}{\partial CWO_j} \cong e_c[k] \cdot \frac{\partial \bar{Y}[k]}{\partial U[k]} \cdot CX_j[k] \quad (23)$$

$$-\frac{\partial E_C}{\partial CWD_j} = e_c[k] \cdot \frac{\partial Y[k]}{\partial U[k]} \cdot \frac{\partial U[k]}{\partial CWD_j} \cong e_c[k] \cdot CWO_j \cdot \frac{\partial \bar{Y}[k]}{\partial U[k]} \cdot \frac{\partial CX_j[k]}{\partial CWD_j} \quad (24)$$

$$-\frac{\partial E_C}{\partial CWI_{ij}} = e_c[k] \cdot \frac{\partial Y[k]}{\partial U[k]} \cdot \frac{\partial U[k]}{\partial CWI_{ij}} \cong \sum_j e_c[k] \cdot \frac{\partial \bar{Y}[k]}{\partial U[k]} \cdot CWO_j \cdot \frac{\partial CX_j[k]}{\partial CWI_{ij}} \quad (25)$$

$$\frac{\partial \bar{Y}[k]}{\partial U[k]} = \sum_j WO_j \cdot f'(S_j[k]) \cdot WI_{1j} \quad (26)$$

where $CI_i[k]$ is the $i^{th}$ input to the diagonal recurrent control neural network, $CS_j[k]$ is the sum of inputs to the $j^{th}$ recurrent neuron, and $CX_j[k]$ is the output of the $j^{th}$ recurrent neuron. $CWI_{ij}$, $CWD_j$, $CWO_j$ are input, recurrent, and output weight, respectively.

## 4. SIMULATION RESULTS

The proposed algorithm is tested on a linear and a nonlinear control problem.

Plant 1:

A linear discrete-time plant with nonlinear time invariant component is given by

$$Y[k+1] = 1.4*Y[k] - 0.4*Y[k-1] + 0.165*U[k] + 0.005*(1.0 - \exp(Y[k-1]))$$

The result of fuzzy control only with scaling factor GR=GC=4.0 and GU=1.0 is shown in Fig. 4.

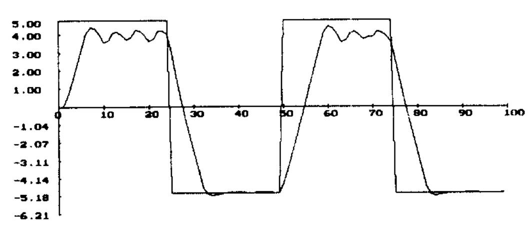


Fig. 4. Time response of the plant 1 with the fuzzy controller.

Initially the INN is trained 10 epochs using the input-output informations which are generated by the fuzzy controlled plant with $\eta_I=0.01, \alpha_I=0.005$ and 7 hidden layer neurons. Then the CNN and the INN are trained 100 epochs with $\eta_I=0.01, \alpha_I=0.005$, 7 hidden layer neurons and $\eta_C=0.02, \alpha_C=0.01$, 9 hidden layer neurons. The simulation result is shown in Fig. 5.

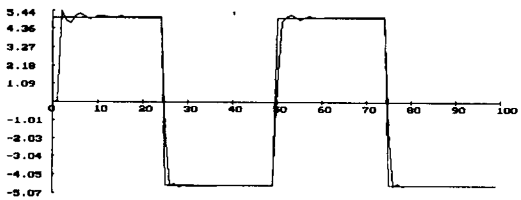


Fig. 5. Time response of the plant 1 with the neural network controller.

Plant 2:

The nonlinear discrete time plant is described by

$$Y[k+1] = Y[k]/(1.0 + Y^2[k]) + U^3[k]$$

The result of fuzzy control only with scaling factor GR=GC=4.0 and GU=2.4 is shown in Fig. 6.

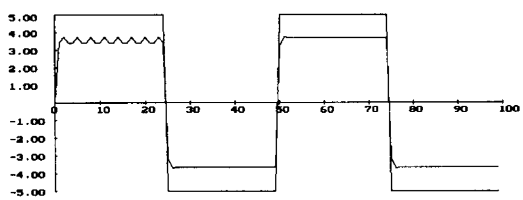


Fig. 6. Time response of the plant 2 with the fuzzy controller.

The same parameters as plant 1 are used except that the INN and the CNN have 5 and 7 hiddern neurons, respectively. The time response of plant 2 is shown in Fig. 7.

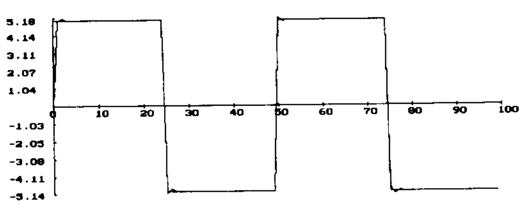


Fig. 7. Time response of the plant 2 with the neural network controller.

To test the adaptability, when the plant 2 is trained 50 epochs, we change the input parameter at time step 35 as follows:

$$Y[k+1] = Y[k]/( 1.0 + Y^2[k]\ b ) + \underline{1.2}*U^3[k]$$

The simulation result given in Fig. 8 shows a good adaptability of the proposed neuro-fuzzy adaptive controller.

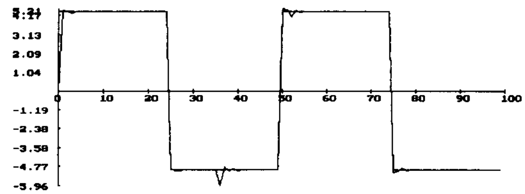


Fig. 8. Time response of the plant 2 with parameter change.

## 5. CONCLUSIONS

We have described a neuro-fuzzy adaptive control system. The proposed control system was verified on unknown linear and nonlinear discrete-time plants which have the priori linguistic control rules. The simulation results show a good performance on the adaptive property to variations of plant parameters. There has also been an effort to develop neural networks with the objective of reproducing the behaviors of a human being, especially his learning capacity.

## 6. REFERENCES

[1] L. A. Zedah, "Fuzzy sets," *Inform. Contr.*, vol. 8, no. 3, pp. 338-353. June 1965.

[2] L.Pavel and M.Chelaru, "Neural fuzzy architecture for adaptive control," *IEEE Int. Conf. on Fuzzy System,* Mar. 1992.

[3] D. Psaltis, A.Sideris, and A. A. Yamamura, "A multilayered neural network controller," *IEEE Contr. Syst. Mag.*, vol. 8, pp. 17-21, Apr. 1988.

[4] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4-27, Mar. 1990.

[5] C. W. Anderson, "Learning to control an inverted pendulum using neural networks," *IEEE Contr. Syst. Mag.*, vol. 8, pp. 17-21, 1989.

[6] A. Guez and J. W. Selinsky, "A trainable neuromorphic controller," *J. Robotic Syst.*, vol. 5, no. 4, pp. 363-388, Apr. 1988.

[7] Chao-Chee Ku and Kwang Y. Lee, "Diagonal recurrent neural networks for nonlinear system control," *IEEE Int. Joint Conf. on Neural Networks*, vol. 1, pp. 315-320, 1992.

[8] J. W. Selinsky and A. Guez, "The role of a priori knowledge of plant dynamics in neurocontroller design," in *Proc. 28th Decision and Control*, pp. 1754-1758, 1989.

[9] Y.F.Li and C.C.Lau, "Development of fuzzy algorithms for servo systems," *IEEE Contr. Syst. Mag.*, Apr. 1989.

[10] Liang-Jong Huang and Masayoshi Tomizuda, "A self-paced fuzzy tracking controller for two dimensional motion control," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 20, no. 5, Sept./Oct. 1990.

[11] Chao-Chee Ku and Kwang Y. Lee, "Nonlinear system identification using diagonal recurrent neural networks," *IEEE Int. Joint Conf. on Neural Networks*, vol. 3, pp. 839-844, 1992.