

# A Study on an Adaptive Membership Function for Fuzzy Inference System

\*Eun-Oh Bang and \*Sang-Bae Lee

\* Dept. of Electronic & Commucatuion Eng., Korea Maritime University  
1, Dongsam-dong, Youngdo-gu, Pusan, 606-791, KOREA

## Abstract

In this paper, a new adaptive fuzzy inference method using neural network based fuzzy reasoning is proposed to make a fuzzy logic control system more adaptive and more effective. In most cases, the design of a fuzzy inference system rely on the method in which an expert or a skilled human operator would operate in that special domain. However, if he has not expert knowledge for any nonlinear environment, it is difficult to control in order to optimize. Thus, using the proposed adaptive structure for the fuzzy reasoning system can controled more adaptive and more effective in nonlinear environment for changing input membership functions and output membership functions.

The proposed fuzzy inference algorithm is called adaptive neuro-fuzzy control(ANFC).

ANFC can adapt a proper membership function for nonlinear plant, based upon a minimum number of rules and an initial approximate membership function. Nonlinear function approximation and rotary inverted pendulum control system are employed to demonstrate the viability of the proposed ANFC.

## I . INTRODUCTION

Fuzzy theory was initiated by Lotfi A. Zadeh in 1965 with his seminal paper "Fuzzy Sets" (Zadeh [1965]). Fuzzy inference systems have been successfully applied in various areas for intelligent control to expert system [2][11][12][18].

In most cases, the design of a fuzzy inference system is rely on the method in which an expert or a skilled human operator would operate in that special domain [11][12][18]. However, if he has not expert knowledge for any nonlinear environment, it is difficult to control in order to optimize. Thus, using the proposed adaptive structure for the fuzzy reasoning system can controled more adaptive and more effective in nonlinear environment for changing input membership functions and output membership functions.

Several algorithm of an adaptive the fuzzy membership functions have been proposed

in [2]-[8].

With rapid development of techniques for neural networks and fuzzy logic systems, the neuro-fuzzy systems are attracting more and more interest since they are more efficient and more powerful than either neural networks or fuzzy logic system. Many effective learning algorithm of neuro-fuzzy systems were developed and many structure of neuro-fuzzy systems were proposed. For example, Wang's several adaptive fuzzy systems [2], Jang's adaptive network based fuzzy inference systems(ANFIS) [3], Lin's neural networked based fuzzy logic control and decision system [6], etc.

In the application aspect, neuro-fuzzy systems have been widely used in control system, Intelligent robot, pattern recognition, consumer products, medicine, expert systems, fuzzy mathematics, Information retrieval, etc.

This paper is organized as follows.

In section II, we propose a new adaptive neuro-fuzzy control(ANFC) for the structure, operations, and learning algorithm.

In section III, we compares the proposed ANFC with the conventional control systems for simulation result. Nonlinear function approximation and rotary inverted pendulum control system are employed to demonstrate the viability of the proposed ANFC.

The conclusions are given in section IV.

## II. LEARNING ALGORITHM OF AN ADAPTIVE FUZZY INFERENCE SYSTEM

This section represents the structure and learning algorithm of the adaptive network which is fact a superset of all kinds of feedforward neural networks with supervised learning capability. An adaptive network is a network structure consisting of nodes and directional links through which the nodes are connected. Moreover, part or all of the nodes are adaptive, which means each output of these nodes depends on the parameters pertaining to this node, and the learning rule specifies how these parameters should be changed to minimize a prescribed error measure.

Neural networks are essentially low-level computational structures and algorithm that offer good performance in dealing with sensory data, while fuzzy logic techniques often deal with issues such as reasoning on a higher level than neural networks. However, since fuzzy systems do not have much learning capability it is difficult for a human operator to tune the fuzzy rule and membership functions from the training data set. Thus, a promising approach for reaping the benefits of both fuzzy systems and neural networks is to merge or fuse them into an integrated system.

The proposed ANFC aim at a proper membership function for nonlinear plant, based upon a minimum number of rules and an initial approximate membership function.

Now, ANFC algorithm represents as below.

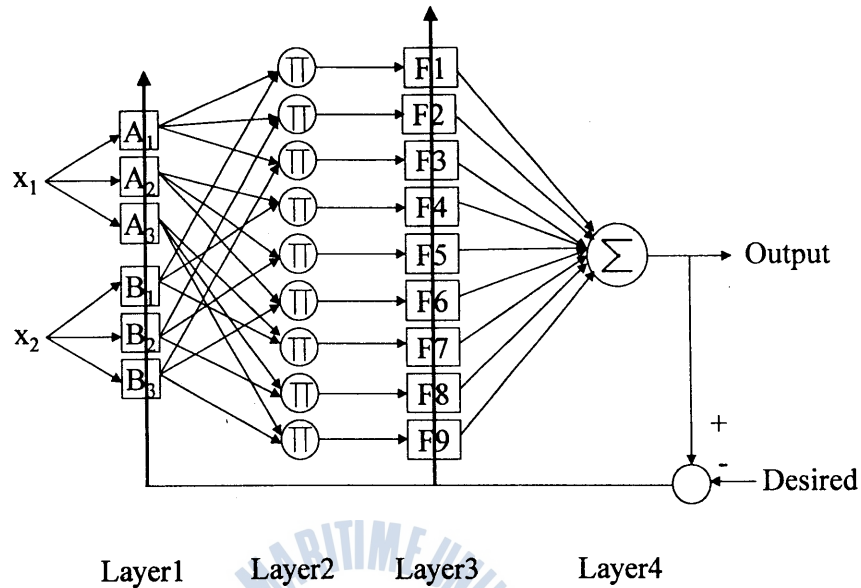


Fig. 1. Adaptive Neuro-Fuzzy Control(ANFC)

The function of each layer describe the following algorithm for two input  $x_1, x_2$ , respectively.

**Layer 1** This layer is a square node  $i$  with a node function for input membership function depends on incoming input signals.

$$O_{1,i} = \mu_{A_i}(x_1) \quad (1)$$

$$O_{1,i} = \mu_{B_i}(x_2) \quad (2)$$

Where  $x_1, x_2$  are the input to node  $i$ , and  $A_i, B_i$  are the linguistic label for node function. Namely,  $O_{1,i}$  is the membership function of  $A_i, B_i$  and it specify the degree to which the given  $x_1, x_2$  satisfy the quantifier  $A_i, B_i$ , respectively.

$\mu_{A_i}(x_1), \mu_{B_i}(x_2)$  choose to be gaussian membership function for input  $x_1, x_2$ . Each node represents one membership function for one linguistic term.

Input membership functions in this layer are an defined by

$$O_{1,i} = \mu_{A_i}(x_1) = \exp \left[ -\left( \frac{x_1 - c_i}{\alpha_i} \right)^2 \right] \quad \text{for } i = 1, 2, \dots, k \quad (3)$$

$$O_{1,i} = \mu_{B_i}(x_2) = \exp \left[ -\left( \frac{x_2 - m_i}{\beta_i} \right)^2 \right] \quad \text{for } i = 1, 2, \dots, l \quad (4)$$

Where  $c_i$ ,  $a_i$ ,  $m_i$ ,  $\beta_i$  are centers and widths of  $\mu_{A_i}(x_1)$ ,  $\mu_{B_i}(x_2)$ , respectively.

**Layer 2** This layer labeled  $\Pi$  performs fuzzy AND operation.

Every node in this layer is a fixed node, which operates the incoming signal from every set of the membership function nodes for their corresponding input. Each node output indicates the firing strength of a rule. For instance,

$$O_{2,i} = \min(\mu_{A_i}(x_1), \mu_{B_i}(x_2)) \quad (5)$$

**Layer 3** Every node in this layer performs fuzzy OR operation. for a output membership function.

An adaptive node function is a gaussian membership function for the output layer. Each node represents a simple output membership function. The link weight of this layer is equal to 1.

$$O_{3,i} = \max(O_{2,i}) \quad \text{for } i = 1, 2, \dots, k \times l \quad (6)$$

**Layer 4** The single node in this layer is a fixed node labeled  $\Sigma$ , which performs the overall output as the summation of all incoming signals. It transits the fuzzy output to crisp signal used for control. This function can be used to computes the center of area defuzzification method. The function of the node can be describe by

$$O_{4,i} = \frac{\sum_{i=1}^{k \times l} s_i \sigma_i O_{3,i}}{\sum_{i=1}^{k \times l} \sigma_i O_{3,i}} \quad \text{for } i = 1, 2, \dots, k \times l \quad (7)$$

### Tuning Parameter

The network structure are adjust the parameter of input and output membership functions optimally. The back-propagation learning algorithm is employed to fine-tune the membership functions for desired output. Considering a single output, the aim is to minimize the error function by

$$E = \sum_{i=1}^n \frac{1}{2} (T_i - O_i)^2 \quad (8)$$

Where  $T_i$ ,  $O_i$  are the desired output and the measured output, respectively.

To represent the learning rule, we shall show the computation of  $\frac{\partial E}{\partial O}$ , layer by layer, starting at the output nodes, and we will use gaussian membership function with center  $s_i$  and width  $\sigma_i$  as the adjustable parameters for these computations.

**Layer 4** Every node in this layer needs to be tuned the center and width of the output membership function. The adaptive rule of the center  $s_i$  is derived as

$$\frac{\partial E}{\partial s_i} = \frac{\partial E}{\partial O_{4,i}} \frac{\partial O_{4,i}}{\partial s_i} = - [T_i(t) - O_i(t)] \frac{\sigma_i O_{3,i}}{\sum \sigma_i O_{3,i}} \quad (9)$$

Hence, the center parameter is updated by

$$s_i(t+1) = s_i(t) + \eta [T_i(t) - O_i(t)] \frac{\sigma_i O_{3,i}}{\sum \sigma_i O_{3,i}} \quad (10)$$

Where  $\eta$  is the learning parameter.

Similarly, the adaptive rule of the width  $\sigma_i$  is derived as

$$\begin{aligned} \frac{\partial E}{\partial \sigma_i} &= \frac{\partial E}{\partial O_{4,i}} \frac{\partial O_{4,i}}{\partial \sigma_i} \\ &= - [T_i(t) - O_i(t)] \frac{s_i O_{3,i} (\sum \sigma_i O_{3,i}) - (\sum s_i \sigma_i O_{3,i}) O_{3,i}}{(\sum \sigma_i O_{3,i})^2} \end{aligned} \quad (11)$$

Hence, the center parameter is updated by

$$\begin{aligned} \sigma_i(t+1) &= \sigma_i(t) + \eta [T_i(t) - O_i(t)] \\ &\quad \frac{s_i O_{3,i} (\sum \sigma_i O_{3,i}) - (\sum s_i \sigma_i O_{3,i}) O_{3,i}}{(\sum \sigma_i O_{3,i})^2} \end{aligned} \quad (12)$$

The error to be propagated to preceding layer is derived as

$$\delta_4 = - \frac{\partial E}{\partial O_{4,i}} = T_i(t) - O_i(t) \quad (13)$$

**Layer 3** In this layer, no parameter needs to be regulated. Only the error need to be computed.

$$\delta_{3,i} = - \frac{\partial E}{\partial O_{3,i}} = - \frac{\partial E}{\partial O_{4,i}} \frac{\partial O_{4,i}}{\partial O_{3,i}} \quad (14)$$

$$\frac{\partial O_{4,i}}{\partial O_{3,i}} = \frac{s_i \sigma_i (\sum \sigma_i O_{3,i}) - (\sum s_i \sigma_i O_{3,i}) \sigma_i}{(\sum \sigma_i O_{3,i})^2} \quad (15)$$

$$- \frac{\partial E}{\partial O_{4,i}} = T_i(t) - O_i(t) \quad (16)$$

Hence, the error signal is

$$\delta_{3,i}(t) = [T_i(t) - O_i(t)] \frac{s_i \sigma_i (\sum \sigma_i O_{3,i}) - (\sum s_i \sigma_i O_{3,i}) \sigma_i}{(\sum \sigma_i O_{3,i})^2} \quad (17)$$

**Layer 2** In this layer, only the error need to be computed.

$$\delta_{2,i} = -\frac{\partial E}{\partial O_{2,i}} = -\frac{\partial E}{\partial O_{3,i}} \frac{\partial O_{3,i}}{\partial O_{2,i}} = -\frac{\partial E}{\partial O_{3,i}} = \delta_{3,i} \quad (18)$$

**Layer 1** Every node in this layer needs to be tuned the center and width of the input membership function. The adaptive rule of the center  $c_i$  is derived as

$$\frac{\partial E}{\partial c_i} = \frac{\partial E}{\partial O_{1,i}} \frac{\partial O_{1,i}}{\partial c_i} = \frac{\partial E}{\partial O_{1,i}} \frac{2(x_1 - c_i)}{\alpha_i^2} \quad (19)$$

Where

$$\frac{\partial E}{\partial O_{1,i}} = \frac{\partial E}{\partial O_{2,i}} \frac{\partial O_{2,i}}{\partial O_{1,i}} \quad (20)$$

Where from equation (16),

$$\frac{\partial E}{\partial O_{2,i}} = -\delta_{3,i} \quad (21)$$

$$\frac{\partial O_{2,i}}{\partial O_{1,i}} = \delta_{1,i} \quad (22)$$

$$= \begin{cases} 1 & \text{if } O_{1,i} = \min(\text{input node } i) \\ 0 & \text{otherwise} \end{cases}$$

$$c_i(t+1) = c_i(t) + \eta \delta_{1,i} \frac{2(x_1 - c_i)}{\alpha_i^2} \quad (23)$$

So the update rule of  $c_i$  is defined by

$$c_i(t+1) = c_i(t) - \eta [T_i(t) - O_i(t)] \frac{2(x_1 - c_i)}{\alpha_i^2} \quad (24)$$

Similarly, the update rule of  $\alpha_i$  is derived as

$$\frac{\partial E}{\partial \alpha_i} = \frac{\partial E}{\partial O_{1,i}} \frac{\partial O_{1,i}}{\partial \alpha_i} = \frac{\partial E}{\partial O_{1,i}} \frac{2(x_1 - c_i)}{\alpha_i^3} \quad (25)$$

$$\alpha_i(t+1) = \alpha_i(t) + \eta \delta_{1,i} \frac{2(x_1 - c_i)}{\alpha_i^3} \quad (26)$$

Hence, the update rule of  $\alpha_i$  becomes

$$\alpha_i(t+1) = \alpha_i(t) - \eta [T_i(t) - O_i(t)] \frac{2(x_1 - c_i)^2}{\alpha_i^3} \quad (27)$$

Similarly, the update rule of  $m_i, \beta_i$  are derived as

$$m_i(t+1) = m_i(t) - \eta [T_i(t) - O_i(t)] \frac{2(x_2 - m_i)}{\beta_i^2} \quad (28)$$

$$\beta_i(t+1) = \beta_i(t) - \eta [T_i(t) - O_i(t)] \frac{2(x_2 - m_i)}{\beta_i^3} \quad (29)$$

### III. SIMULATION RESULTS

In this section, we apply the adaptive neuro-fuzzy control (ANFC) proposed in section II to the nonlinear function approximation and the rotary inverted pendulum stabilizing problem. First, a nonlinear function approximation result is depicted as below.

**Example 1 :** For a first Example, Nonlinear function defined by

$$\begin{aligned} u(k)_1 &= 2 \sin(k/40) \\ y(k) &= 3 \sin(k/40) + 2 \cos(k/20) \end{aligned} \quad (30)$$

Where  $u(k)_2$  is delayed input  $u(k)_1$ . Learning rate,  $\eta$  is 0.005.

Fig. 2 shows the proposed ANFC result (bold line) satisfies the approximation for nonlinear function at 62 iterations.

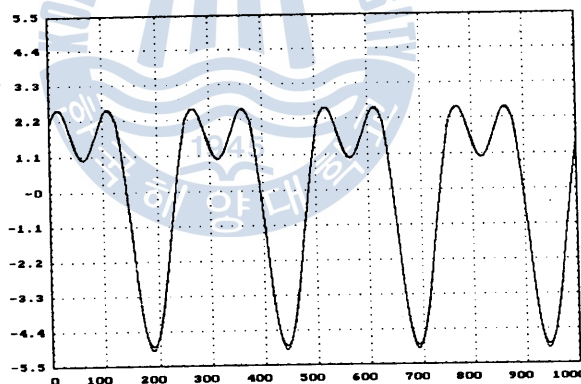


Fig. 2. Proposed ANFC result for the nonlinear function approximation

Fig. 3 shows the five input membership functions for input  $x_1$ ,  $x_2$ , respectively.

Fig. 4 illustrates the altering the five input membership functions after training by proposed algorithm. In five input membership functions, it must be eliminate membership function which is big similarity since it does not affect the output. Thus, proposed ANFC have a priori performance than the expert knowledge.

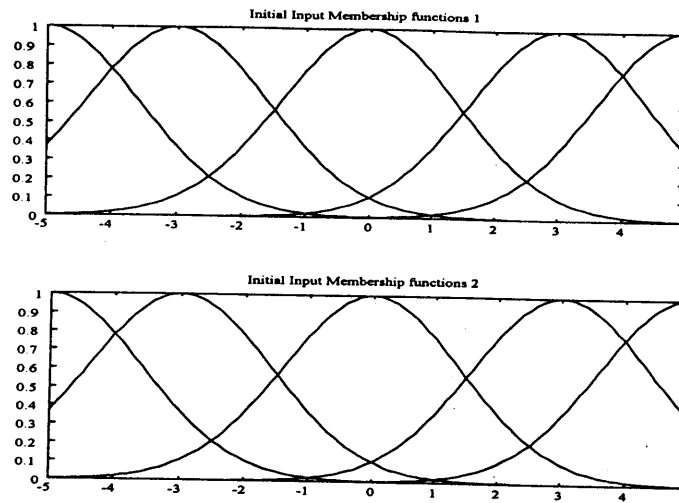


Fig. 3. Five Input membership functions for input  $x_1$ ,  $x_2$ , respectively.

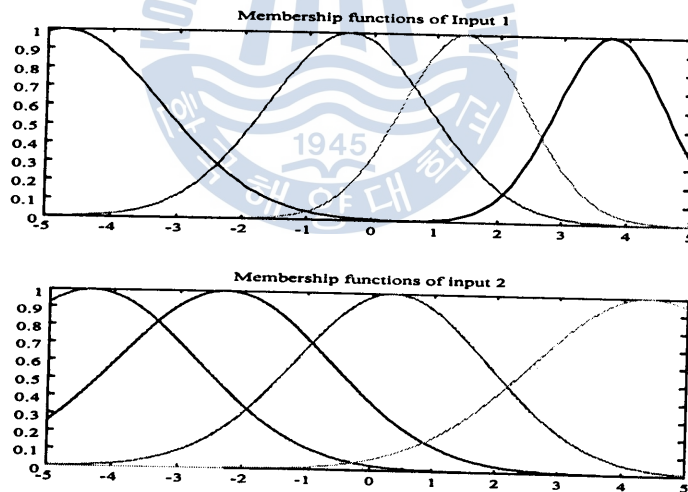


Fig. 4. Regulated five Input membership functions for input  $x_1$ ,  $x_2$ , respectively after training.

**Example 2 :** For a second example, the plant is assumed to be defined by

$$y_p(k+1) = f[y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)] \quad (32)$$



The input to the plant and the identified model is given by

$$\begin{aligned} u(k) &= \sin(2\pi k/250) && \text{for } k \leq 500 \\ u(k) &= 0.8\sin(2\pi k/250) && \text{for } k > 500 \end{aligned} \tag{33}$$

Learning rate,  $\eta$  is 0.005.

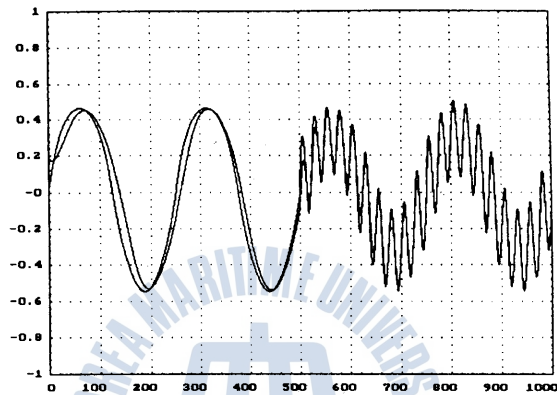


Fig. 5. Result of the Neural Network

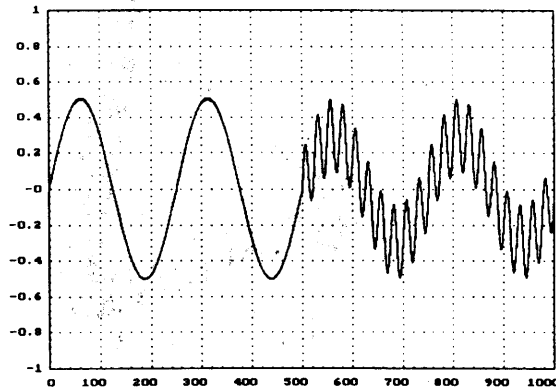


Fig. 6. Result of the proposed ANFC

Fig. 6 shows the result of the three input membership functions and nine output membership functions. The proposed algorithm result(bold line in Fig. 6) affects than the conventional neural network result(tended line in Fig. 5) for the approximation and the convergence time. In Fig. 6, five input membership functions have more fast convergence time than three input membership functions as a same learning rate.

**Example 3 :** We apply the adaptive neuro-fuzzy control(ANFC) proposed in section II to the rotary inverted pendulum stabilizing problem. The rotary inverted pendulum includes nonlinear dynamics which is difficult to control.

We must be satisfied two conditions for control of the rotary inverted pendulum.

First, It is a position control that any position moves to the objective position.

Second, It is a balance control that stabilizing the pole to inverted forward.

The proposed ANFC(Fig. 8) performs the minimization for the conventional controller (LQR) output.

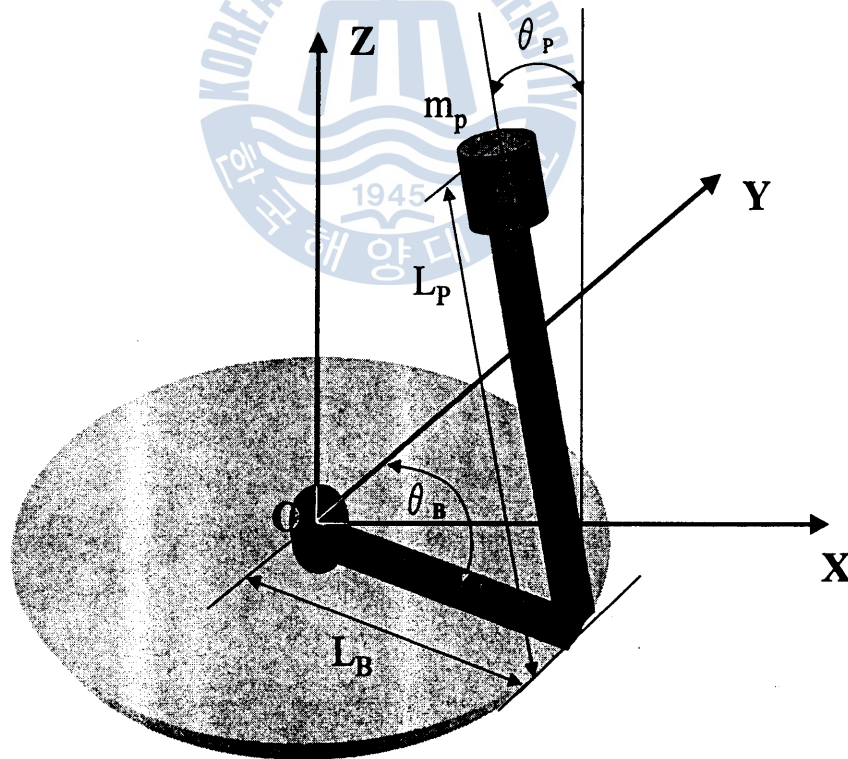


Fig. 7. Rotary Inverted Pendulum System

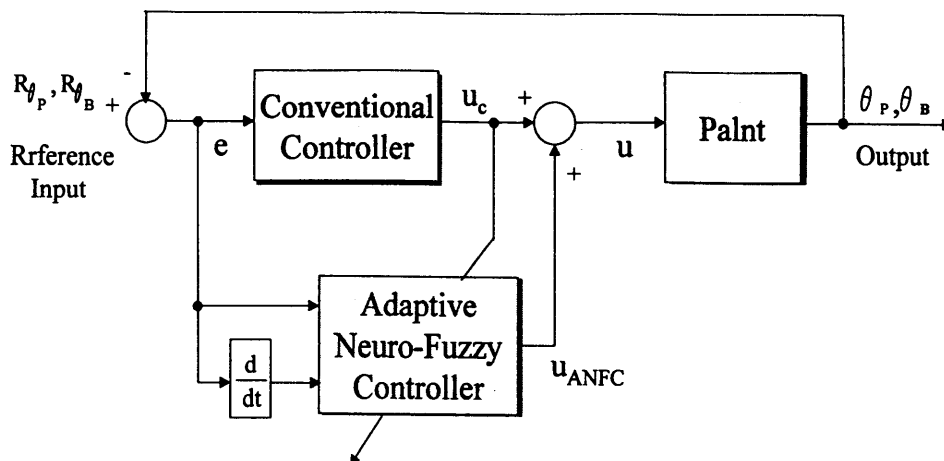


Fig. 8. Proposed Adaptive Neuro-Fuzzy Controller

The dynamic equation of the rotary inverted pendulum system are represented by

$$K_P = \frac{1}{2} m_p [L_m \dot{\theta}_m + L_P \dot{\theta}_P \cos(\theta_P)^2 + (L_P \dot{\theta}_P \sin(\theta_P))^2] \quad (34)$$

$$K_B = \frac{1}{2} J_B \dot{\theta}_m^2 \quad (35)$$

$$P_P = m_p g L_P \cos(\theta_P) \quad (36)$$

Where  $K_P$ ,  $K_B$ ,  $P_P$  are the kinetic energy for the pole of the rotary inverted pendulum, the kinetic energy for the base link of the rotary inverted pendulum, and the potential energy for the pole of the rotary inverted pendulum, respectively.  $m_p$  is mass of pole.

By using a above equation, we obtained Lagrangian equation by

$$(m_p L_B^2 + J_B) \ddot{\theta}_B + m_p L_B \ddot{\theta}_P L_P \cos(\theta_P) - m_p L_B \dot{\theta}_P L_P \sin(\theta_P) = T \quad (37)$$

$$m_p L_B L_P \ddot{\theta}_P \cos(\theta_P) - m_p L_P L_B \dot{\theta}_P \dot{\theta}_m \sin(\theta_P) + m_p L_P^2 \ddot{\theta}_P - m_p g L_P \sin(\theta_P) = 0 \quad (38)$$

A torque  $T$  by DC motor is given the following form. That is

$$u = V = I R_m + K_m \omega_m \quad (39)$$

$$T = I K_m = \frac{K_m}{R_m} V - \frac{K_m^2}{R_m} \omega_m \quad (40)$$

Where  $u$ ,  $\omega_m$  are the plant input and rotate speed[rad/s] of the motor. Thus, we can find the state equation as a following system parameter(Table 1). That is

$$\begin{bmatrix} \dot{\theta}_B \\ \theta_P \\ \dot{\theta}_B \\ \dot{\theta}_P \end{bmatrix} = \begin{bmatrix} 1 & -0.0031 & 0.0088 & 0 \\ 0 & -1 & 0.0006 & 0.01 \\ -0.5935 & 0.7787 & 0 & -0.0031 \\ 0.6036 & 1 & 0.1063 & 1.031 \end{bmatrix} \begin{bmatrix} \theta_B \\ \theta_P \\ \dot{\theta}_B \\ \dot{\theta}_P \end{bmatrix} + \begin{bmatrix} 0.0022 \\ 1 \\ 0.4161 \\ -0.1942 \end{bmatrix} u \quad (41)$$

where sampling time is 0.01sec.

Parameter	Symbol	Value	Unit
Length of Pole	$L_P$	0.305	m
Mass of Pole	$m_P$	0.210	Kg
Length of Base Link	$L_B$	0.145	m
Inertia of Base Link	$J_B$	0.0044	$\text{Kgm}^2$
Motor Armature Resistance	$R_m$	2.6	$\Omega$
Motor Voltage Constant	V	$\pm 5.0$	V
Motor Torque Constant	$K_m$	0.00767	V/rads

Table 1. System parameter of the rotary inverted pendulum

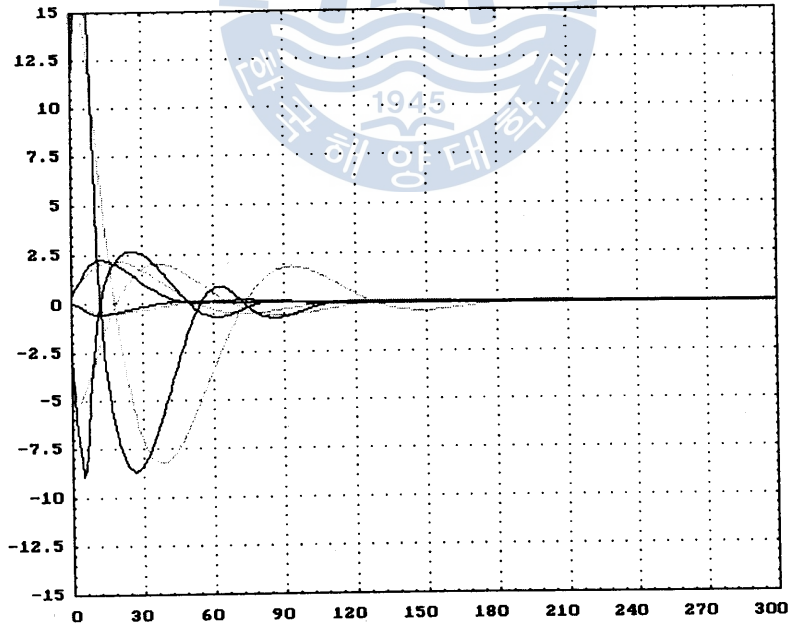


Fig. 9. Result of proposed ANFC for the rotary inverted pendulum

Fig. 9 shows the result of the rotary inverted pendulum by control to the proposed ANFC. For the proposed ANFC, parameter used 7 input membership functions and 49 output membership functions. In the initial parameter,  $\theta_B$ ,  $\theta_P$ ,  $\dot{\theta}_B$ ,  $\dot{\theta}_P$  is 0.5, 0.1, 5.0, 0, respectively. Learning rate,  $\eta$  is 0.03.

The proposed ANFC result (strong line) affects than the conventional controller (LQR) result (tender line) for the convergence time. As compared to conventional controller, it provides improved performance. Thus, proposed method can be controlled more adaptive and more effective in rotary inverted pendulum for changing an input membership functions and output membership functions.

#### IV. Conclusion

In this paper, we proposed adaptive neuro-fuzzy control (ANFC) to nonlinear function approximation and rotary inverted pendulum. With a simulation result, proposed method can be controlled more adaptive and more effective in nonlinear environment for changing an input membership functions and output membership functions. As it were, the proposed ANFC has learning ability which can be used in the fine tuning of membership functions to minimize the output error of the control. Moreover, the rule base can be generated automatically by the proposed ANFC. Thus, this proposed algorithm is very useful for fuzzy logic control because the needs for the expert knowledge is relative much lower compare to conventional fuzzy logic control. However, this algorithm must be attend learning rate,  $\eta$  which effect for learning speed and the convergence time. Further work, we will establish more effect and the more reinforcement algorithm.

#### Reference

- [1] K. S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks", IEEE, Trans. on Neural Networks. Vol. 1, No. 1, Pp 4-27, 1990
- [2] L.-X. Wang, "Adaptive Fuzzy Systems and Control", Englewood Cliffs, NJ, Prentice Hall, 1994
- [3] Jyh-Shing Roger Jang, ANFIS: "Adaptive Network Based Fuzzy Inference system", IEEE Trans. on system, Man, And Cyb. Vol. 23, No. 3, Pp. 665-685, 1993
- [4] J.-S.R. Jang and C.-T. Sun, "Neuro-Fuzzy Modeling and Control", Proc. IEEE, March 1995
- [5] Chin-Teng Lin and C.S. George Lee, "Neural fuzzy systems", Prentice Hall, 1996

- [6] C.-T. Lin and C.S. George. Lee, "Neural networked based fuzzy logic control and decision system", IEEE, Trans. Comput., Vol. 40, Pp 1320-1336, Dec. 1991
- [7] D. Nauck and R. Kruse, "A fuzzy neural networks learning fuzzy control rules and membership functions by fuzzy error backpropagation", in proc. IEEE Int. Conf. Neural Networks, Pp. 1022-1027, 1993
- [8] Yan-qing ZhYang and Abraham Kandel, "Compensatory Neurofuzzy Systems with Fast Learning Algorithm", IEEE, Trans. on Neural Networks, Vol. 9, No. 1, Pp. 83-105, 1998
- [9] K. S. Narendra and A. M. Annaswamy, "Stable Adaptive Systems." Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [10] P. G. Gallman and K. S. Narendra, "Identification of nonlinear systems using a Uryson model", Becton Center, Yale University, New Haven, CT, tech. rep. CT-38, Apr. 1971; also Automatica, Nov. 1976.
- [11] Li-Xin Wang, "A Course in Fuzzy Systems and Control", Prentice Hall, 1997
- [12] Bart Kosko, " Fuzzy Engineering", Prentice Hall, 1997
- [13] Jacek M. Zurada, "Artificial Neural Systems", West Info Access, 1992
- [14] Simon Haykin, "Neural Networks", Macmillan, 1994
- [15] Sigeru Omatu, Marzuki Khalid and Rubiyah Yusof, "Neuro-control and its Application", Springer, 1996
- [16] K. S. Narendra and K. Parthasarathy, neural networks and dynamical systems. Part I: A gradient approach to Hopfield networks", Center Syst. Sci., Dept. Electrical Eng., Yale University, New Haver, CT, tech. rep. 8820, Oct. 1988.
- [17] Richard C. Port, Robert H. Bishop, "Modern Control Systems 7th edition", Addition-Wesley Publishing Company. Inc, pp. 30-169, 1995.
- [18] Chen Wei Ji, Fang Lei and Lei Kam Kin, "Fuzzy Logic Controller for An Inverted Pendulum Systems", IEEE, Int. Conf. on intelligent Processing systems, Pp. 185-189, 1997