The Design of Fuzzy Controller by Means of Genetic Optimization and Estimation Algorithms

Sung-Kwun Oh and Seok-Beom Rho

Abstract - In this paper, a new design methodology of the fuzzy controller is presented. The performance of the fuzzy controller is sensitive to the variety of scaling factors. The design procedure is based on evolutionary computing (more specifically, a genetic algorithm) and estimation algorithm to adjust and estimate scaling factors respectively. The tuning of the scaling factors of the fuzzy controller is essential to the entire optimization process. And then we estimate scaling factors of the fuzzy controller by means of two types of estimation algorithms such as HCM (Hard C-Means) and Neuro-Fuzzy model[7]. The validity and effectiveness of the proposed estimation algorithm for the fuzzy controller are demonstrated by the inverted pendulum system.

Keywords - fuzzy controller, evolutionary computing, estimation algorithm, Hard C-Means (HCM) Clustering, neuro-fuzzy model

1. Introduction

The ongoing challenge for advanced system control has resulted in a diversity of design methodologies and detailed algorithms. Fuzzy controllers have positioned themselves in the dominant role at the knowledge-rich spectrum of control algorithms. The advantages of the fuzzy controllers manifest by their suitability for nonlinear systems (as they are nonlinear mappings in the first place) and for high deviations from the set point, and by easy capturing the approximate and qualitative aspects of human knowledge and reasoning [3][4]. The intent of this study is to develop, optimize and experiment with the fuzzy controller (the fuzzy PD controller or the fuzzy PID controller). One of the difficulties in controlling complex systems is to derive the optimal control parameters such as linguistic control rules, scaling factors, and membership functions of the fuzzy controller. With this regard, genetic algorithms (GAs) have already started playing an important role as a mechanism of global search of the optimal parameters of such controllers. However, in controlling a nonlinear plant such as the inverted pendulum of which initial states change in each case, the performance of controllers may become poor, since the control parameters of the fuzzy controller cannot be easily adapted to the changing initial states such as angular position and angular velocity. To alleviate the above shortcoming, we use two types of estimation algorithms such as HCM (Hard C-Means) clustering method and Neuro-fuzzy model and then estimate the parameters of the controller in each case. The paper includes the experimental study dealing the inverted pendulum. The performance of systems under control is evaluated from the viewpoint of ITAE (Integral of the Time multiplied by the Absolute value of Error) and overshoot [1].

This work was supported by a grant No. R02-2000-00284 from Korea Science & Engineering Foundation.

Manuscript received; Aug. 22, 2001 accepted: Nov. 18, 2001.

Sung-Kwun Oh and Seok-Beom Rho are with School of Electrical, Electronic and Information Engineering, Wonkwang University, 344-2, Shinyong-Dong, Iksan, Chon-Buk, 570-749, Korea.

2. The Fuzzy PID Controller

The block diagram of fuzzy PID controller is shown in Fig. 1. Referring to Fig. 1, we confine to the following notation. e denotes the error between reference and response (output of the system under control), Δe is the first-order difference of error signal while $\Delta^2 e$ is the second-order difference of the error. Note that the input variables to the fuzzy controller are transformed by the scaling factors (GE, GD, GH, and GC) whose role is to allow the fuzzy controller to "see" the external world to be controlled.

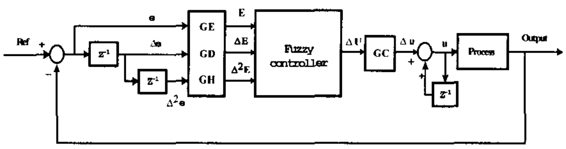


Fig. 1 An overall architecture of the fuzzy PID controller

The above fuzzy PID controller consists of rules of the form [9][10]

R_j : if E is A_{1j} and ΔE is A_{2j} and $\Delta^2 E$ is A_{3j} then ΔU_j is D_j

The capital letters standing in the rule (R_j) denote fuzzy variables (linguistic terms) whereas D is a numeric value (singleton) of the control action. In each control rule, a level of its activation is computed in a standard fashion (1). The inferred value of consequence part is converted into numeric values with the aid of (2-1)[9].

$$\omega_i = \min\{\mu_{A_i}(E), \mu_{B_i}(\Delta E), \mu_{C_i}(\Delta^2 E)\} \quad (1)$$

$$\Delta U^* = \frac{\sum_{i=1}^{n} \omega_i D_i}{\sum_{i=1}^{n} \omega_i} \quad (2\text{-}1)$$

$$\Delta u(k) = \Delta U^*(k) \cdot GC \quad (2\text{-}2)$$

An overall operation of a fuzzy PID controller can be described by

$$u(k) = u(k-1) + \Delta u(k) \quad (3)$$

In the case the input variables are E and ΔE, their membership functions are as follows

NB : Negative Big, NM : Negative Medium, NS : Negative Small,

ZO : Zero, PS : Positive Small, PM : Positive Medium, and

PB : Positive Big.

When dealing with the three input variables of the fuzzy controller, namely E, ΔE, and $\Delta^2 E$, the membership functions are denoted as follows

N : Negative, Z : Zero, and P : Positive.

The membership functions of the output variable of the controller, that is the changes of control are NB(-m3), NM(-m2), NS(-m1), ZO(0), PS(m1), PM(m2) and PB(m3). The initial parameters of these membership functions are equal to m1, m2, and m3, respectively. The collection of the rules is shown in Table 1.

Table 1 Fuzzy control rules

(a) 2 input variables

		ΔE						
		NB	NM	NS	ZO	PS	PM	PB
E	NB	-m3	-m3	-m3	-m3	-m2	-m1	0
	NM	-m3	-m3	-m3	-m2	-m1	0	m1
	NS	-m3	-m3	-m2	-m1	0	m1	m2
	ZO	-m3	-m2	-m1	0	m1	m2	m3
	PS	-m2	-m1	0	m1	m2	m3	m3
	PM	-m1	0	m1	m2	m3	m3	m3
	PB	0	m1	m2	m3	m3	m3	m3

(b) 3input variables

$\Delta^2 E$=N

		ΔE		
		N	Z	P
E	N	-m3	-m3	-m2
	Z	-m2	-m1	0
	P	0	m1	m3

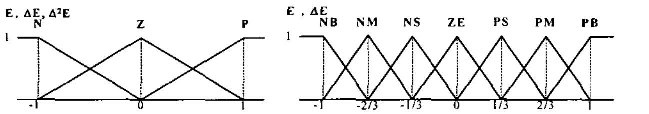
$\Delta^2 E$ = Z

		ΔE		
		N	Z	P
E	N	-m3	-m2	-m2
	Z	-m2	0	m1
	P	m2	m2	m3

$\Delta^2 E$ =P

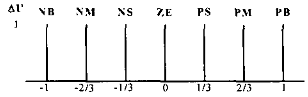
		ΔE		
		N	Z	P
E	N	-m2	-m1	0
	Z	0	m1	m2
	P	m2	m3	m3

We use triangular membership functions defined in the input and output spaces; see Fig. 2 and 3. Here these spaces are normalized to the [-1, 1] interval.



(a) In case of E,ΔE and Δ^2E (b) In case of E and ΔE

Fig. 2 Membership functions of the premise input variables



(a) In case of training (b) In case of evaluation

Fig. 3 Membership functions (singletons) defined in the consequence variable, ΔU

3. Auto-tuning of the fuzzy controller by GAs

Genetic algorithms (GAs) are the search algorithms inspired by Nature in the sense that we exploit a fundamental concept of a survival of the fittest as being encountered in selection mechanisms among species. In GAs, the search variables are encoded in bit strings called chromosomes. They deal with a population of chromosomes with each representing a possible solution for a given problem. A chromosome has a fitness value that indicates how good a solution represented by it is. In control applications, the chromosome represents the controller's adjustable parameters and fitness value is a quantitative measure of the performance of the controller.

In general, the population size, a number of bits used for binary coding, crossover rate, and mutation rate are specified in advance. The genetic search is guided by a reproduction, mutation, and crossover. Each of these phases comes with a set of specific numeric parameters characterizing the phase. In this study, the number of generations is set to 100, crossover rate is equal to 0.6, while the mutation rate is taken as 0.35. The number of bits used in the coding is equal to 10.

Fig. 4 portrays an overall auto-tuning scheme. Let us recall that this involves tuning of the scaling factors and a construction of the control rules. These are genetically optimized. We set the initial individuals of GAs using three types of parameter estimation modes such as a basic mode, contraction mode and expansion mode. In the case of a basic mode (BM), we use scaling parameters that normalize error between reference and output, one level error difference and two level error difference by [-1, 1] for the initial individuals in the GA. In a contraction mode (CM), we use scaling parameters reduced by 25% in relation to the basic mode. While in the expansion mode (EM), we use scaling parameters enlarged by 25% from a basic mode. The standard ITAE expressed for the reference and the output of the system under control is treated as a fitness function [2].

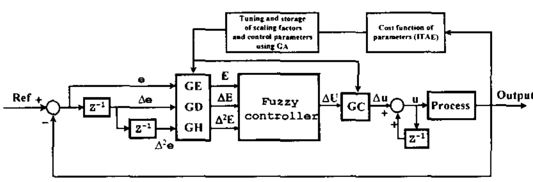


Fig. 4 The scheme of auto-tuning of the fuzzy PID controller involving estimation of the scaling factors

The overall design procedure of the fuzzy PID controller realized by means of GAs is illustrated in Fig. 4. It consists of the following steps

[step 1] Select the general structure of the fuzzy controller according to the purpose of control and dynamics of the process. In particular, we consider architectural options. (PID, FPD(Fuzzy PD), and FPID(Fuzzy PID) controller)

[step 2] Define the number of fuzzy sets for each variable and set up initial control rules, refer to Fig. 2 and 3.

[step 3] Form a collection of initial individuals of GAs. This involves the following

1. set the initial individuals of GAs for the scaling factor of fuzzy controller. The scaling factors can be described as normalized coefficients. Each scaling factor is expressed by (4).

Fig. 5 illustrates three types of estimation modes of the scaling factor being used in setting the initial individuals of GAs describing the fuzzy controller.

$$\begin{aligned} &E(kT) = error \times GE \\ &\Delta E(kT) = [error(kT) - error((k-1)T)] \times GD \\ &\Delta^2 E(kT) = [error(kT) - 2error((k-1)T) + error((k-2)T)] \times GH \\ &U(kT) = U((k-1)T) + \Delta U(kT) \times GC \end{aligned} \tag{4}$$

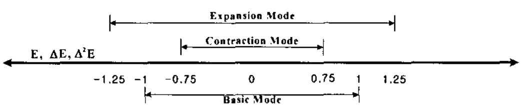


Fig. 5 Three types of estimation modes for the scaling factors: basic, expansion, and contraction

[step 4] Here, all the control parameters such as the scaling factors GE, GD, GH and GC are tuned at the same time.

4. The Estimation Algorithms

4.1 Algorithm 1 : HCM clustering algorithm and Polynomial

In this algorithm, we use HCM clustering algorithm to classify the data and identify the divided data on each cluster by means of LMS method. We use a type of such polynomial as (5), and estimate coefficients of the polynomial.

$$\hat{y}(i) = C_0 + C_1\theta(i) + C_2\theta(i)^2 + \cdots + C_n\theta(i)^n \tag{5}$$

Given a set of data $X=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$, where $\mathbf{x}_k = [x_{k1}, \ldots, x_{km}]$, n is the number of data and m is the number of variables[3]. Let P(X) be the power set of X, that is, the set of all the subsets of X. A hard c-partition of X is the family $\{A_i \in P(X) : 1 \le i \le c\}$ such that $\bigcup_{i=1}^{c} A_i = X$ and $A_i \cap A_j = \phi$ for $1 \le i \ne j \le c$. Each A_i is viewed as a cluster, so $\{A_1, \ldots, A_c\}$ partitions X into c clusters. The hard c-partition can be reformulated through the characteristic (membership) function of the element $\mathbf{x}_k$ in A_i. Specifically, define

$$u_{ik} = \begin{cases} 1, & \mathbf{x}_k \in A_i \\ 0, & \mathbf{x}_k \notin A_i \end{cases} \tag{6}$$

where $\mathbf{x}_k \in X$, $A_i \in P(X)$, i=1,2,…,n. Clearly, u_{ik}=1 means that $\mathbf{x}_k$ belongs to cluster A_i. Given the value of u_{ik}, we can uniquely determine a hard c-partition of X, and vice versa. The u_{ik}'s should satisfy the following three conditions:

$$u_{ik} \in \{0,1\},\ 1 \le i \le c,\ 1 \le k \le n \tag{7}$$

$$\sum_{i=1}^{c} u_{ik} = 1,\ \forall k \in \{1,2,\ldots,n\} \tag{8}$$

$$0 < \sum_{k=1}^{n} u_{ik} < n,\ \forall i \in \{1,2,\ldots,c\} \tag{9}$$

(7) and (8) together mean that each $\mathbf{x}_k \in X$ should belong to one and only one cluster. (9) requires that each cluster A_i must contain at least one and at most n-1 data point. Collecting u_{ik} with $1 \le i \le c$ and $1 \le k \le n$ into a c×n matrix $\mathbf{U}=[u_{ik}]$. We obtain the matrix representation for hard c-partition, defined as follows.

$$M_C = \left\{ \mathbf{U} \mid u_{ik} \in \{0,1\},\ \sum_{i=1}^{c} u_{ik} = 1,\ 0 < \sum_{k=1}^{n} u_{ik} < n \right\} \tag{10}$$

Step 1 : Fix the number of clusters $c(2 \le c < n)$ and initialize the partition matrix $\mathbf{U}^{(0)} \in M_C$

Step 2 : Calculate the center vectors $\mathbf{v}_i$ of each cluster :

$$\mathbf{v}_i^{(r)} = \{v_{i1}, v_{i2}, \cdots, v_{ij}, \cdots, v_{im}\} \tag{11}$$

$$v_{ij}^{(r)} = \frac{\sum_{k=1}^{n} u_{ik}^{(r)} \cdot x_{kj}}{\sum_{k=1}^{n} u_{ik}^{(r)}} \tag{12}$$

where, $[u_{ik}]= \mathbf{U}^{(r)}$, i = 1, 2, …,c, j=1, 2, …,m.

Step 3 : Update the partition matrix $\mathbf{U}^{(r)}$; these modifications are based on the standard Euclidean distance function between the data points and the prototypes

$$d_{ik} = d(\mathbf{x}_k - \mathbf{v}_i) = \|\mathbf{x}_k - \mathbf{v}_i\| = \left[\sum_{j=1}^{m} (x_{kj} - v_{ij})^2 \right]^{1/2} \tag{13}$$

$$u_{ik}^{(r+1)} = \begin{cases} 1 & d_{ik}^{(r)} = \min\{d_{jk}^{(r)}\} \text{ for all } j \in c \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

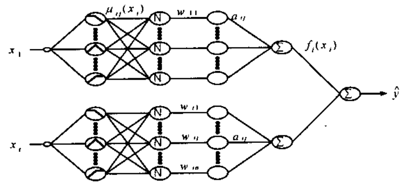
Step 4 : Check a termination criterion.

$$\text{If } \| \mathbf{U}^{(r+1)} - \mathbf{U}^{(r)} \| \le \varepsilon \text{ (tolerance level)} \tag{15}$$

stop ; otherwise set r = r + 1 and return to step 2.

4.2 Algorithm 2: Neuro-fuzzy model

As visualized in Fig. 6, NFN can be designed by using space partitioning in terms of individual input variables. Its topology is concerned with a granulation carried out in terms of fuzzy sets being defined for each input variable.



(23)

Fig. 6 NFN architecture

In Fig. 6, the "circles" denotes units of the NFN, "N" describes a normalization procedure applied to the membership grades of the input variable x_i. The output $f_i(x_i)$ of the "Σ" neuron is governed by some non-linear function f_i. Finally, the output of the FNN $\hat{y}$ reads as

$$\hat{y} = f_1(x_1) + f_2(x_2) + \cdots + f_m(x_m) = \sum_{i=1}^{m} f_i(x_i) \qquad (16)$$

with m being the number of the input variables(viz. the number of the outputs f_i's of the "Σ" neurons in the network). Here we can regard each f_i given by (16) as the following mappings (rules).

$$R^1 : \text{If } x_i \text{ is } A_{i1} \text{ then } y_{i1} = w_{i1}$$
$$\vdots \qquad \vdots$$
$$R^n : \text{If } x_i \text{ is } A_{in} \text{ then } y_{in} = w_{in} \qquad (17)$$

The learning of the NFN is realized by adjusting connections of the neurons and as such it follows a standard Back-Propagation (BP) algorithm. In this study, we use the Euclidean error distances

$$E_p = (y_p - \hat{y}_p)^2 \qquad (18)$$

$$E = \sum_{p=1}^{N} (y_p - \hat{y}_p)^2 \qquad (19)$$

where E_p is an error measure for the *p-th* data, y_p is the *p-th* target output data, $\hat{y}$ stands for the *p-th* actual output of the model for this specific data point, N is total input-output data pairs, and E is a sum of the errors.

As far as learning is concerned, the connections change as follows

$$w(\text{new}) = w(\text{old}) + \Delta w \qquad (20)$$

where the update formula follows the gradient descent method

$$\Delta w_{ij} = \eta \cdot \left(-\frac{\partial E_p}{\partial w_{ij}} \right) = -\eta \frac{\partial E_p}{\partial \hat{y}_p} \cdot \frac{\partial \hat{y}_p}{\partial f_i(x_i)} \cdot \frac{\partial f_i(x_i)}{\partial w_{ij}}$$
$$= 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \mu_{ij}(x_i) \qquad (21)$$

with η being a positive learning rate.

Quite commonly to accelerate convergence, a momentum term is being added to the learning expression. Combining (21) and a momentum term, the complete update formula combining the already discussed components is

$$\Delta w_{ij} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \mu_{ij}(x_i) + \alpha(w_{ij}(t) - w_{ij}(t-1)) \qquad (22)$$

(Here the momentum coefficient, α, is constrained to the unit interval).

Fig. 7 depicts the detailed flowchart of the complete tuning and estimating process.

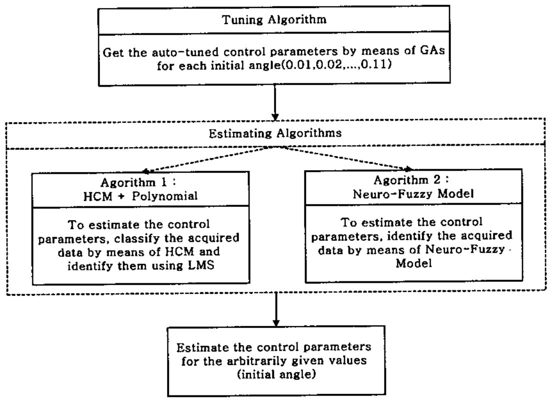


Fig. 7 Overall tuning and estimating process

5. Simulation Study

The proposed control scheme can be applied to a variety of control problems. In this section, we demonstrate the effectiveness of the fuzzy controller by applying it to the inverted pendulum system (Fig. 8).

The inverted pendulum system is composed of a rigid pole and a cart on which the pole is hinged [4][5]. The cart moves on the rail tracks to its right or left, depending on the force exerted on the cart. The pole is hinged to the car through a frictionless free joint such that it has only one degree of freedom. The control goal is to balance the pole starting from nonzero conditions by supplying appropriate force to the cart. In this study, the dynamics of the inverted pendulum system are characterized by two state variables: θ (angle of the pole with respect to the vertical axis), $\dot{\theta}$ (angular velocity of the pole). The behavior of these two state variables is governed by the following second-order equation.

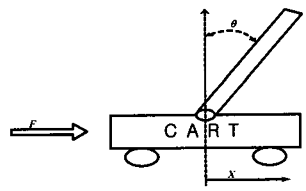


Fig. 8 The inverted pendulum system

The dynamic equation of the inverted pendulum is shown as the following.

$$\ddot{\theta} = \frac{g\sin\theta + \cos\theta\left(\dfrac{-F - ml\theta^2 \sin\theta}{m_c + m}\right)}{l\left(\dfrac{4}{3} - \dfrac{m\cos^2\theta}{m_c + m}\right)} \qquad (23)$$

Where g (acceleration due to gravity) is $9.8m/s^2$, m_c (mass of cart) is $1.0kg$, m (mass of pole) is $0.1kg$, and F is the applied force in newtons.

(24) is the linearized form from (23).

$$\ddot{\theta} = \frac{g\theta - \left(\frac{F}{m_c + m}\right)}{l\left(\frac{4}{3} - \frac{m}{m_c + m}\right)} \tag{24}$$

From the above (24), we can find the state equation.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 15.776 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1.463 \end{bmatrix} U \tag{25}$$

Where, x_1 is θ, x_2 is $\dot{\theta}$ and U is F.

Our control goal here is to balance the pole without regard to the cart's position and velocity, and we compare the fuzzy PID controller and the fuzzy PD controller with the conventional PID controller under same conditions to validate the fuzzy PID controller and the fuzzy PD controller.

- Tuning of control parameters and estimation

We get control parameters such as GE, GD, GH, and GC tuned by GAs because the control parameters have an effect on the performance of controller. GAs are powerful nonlinear optimization techniques. However, the powerful performance is obtained at the cost of expensive computational requirements and much time. To overcome this weak point, first, we select several initial angular positions and obtain the auto-tuned control parameters by means of GAs according to the change of each selected initial angular positions, then build a table. Secondly, we use three algorithms to estimate the control parameters, which are explained next section. We consider the ITAE (Integral of the Time multiplied by the Absolute value of Error) and Overshoot as the PI (Performance Index) of the controller. The initial angular positions are selected that are 0.01rad., 0.02rad. ,..., 0.11rad., and 0.115rad. We tune(adjust) the control parameters of each controller (fuzzy PID controller, fuzzy PD controller and PID controller)

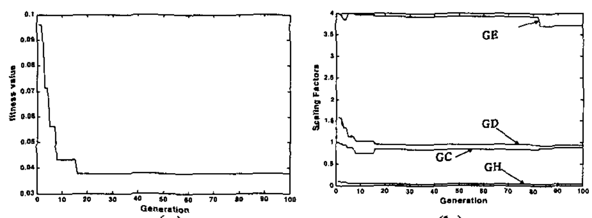
Table 2. shows the scaling factors of the fuzzy controller tuned by using GAs, ITAE and OverShoot(%) in case that the initial angular position of the inverted pendulum is 0.01rad., 0.02rad. ,..., 0.11rad., and 0.115rad. respectively. Using these 21 data, the auto-tuned values of scaling factors are obtained by using GAs for estimating control parameters.

The control parameter and performance index (ITAE, Overshoot(%)) of the fuzzy PID controller after genetic optimization in case of $\theta = 0.01,\ldots,0.115$(rad)

Fig. 9 shows (a) the performance of a fitness function in case of $\theta = 0.06$(rad) and (b) the tuning procedure of scaling factors such as GE, GD, GC and GH according to successive generation with the aid of GAs. Refer to Table 2.

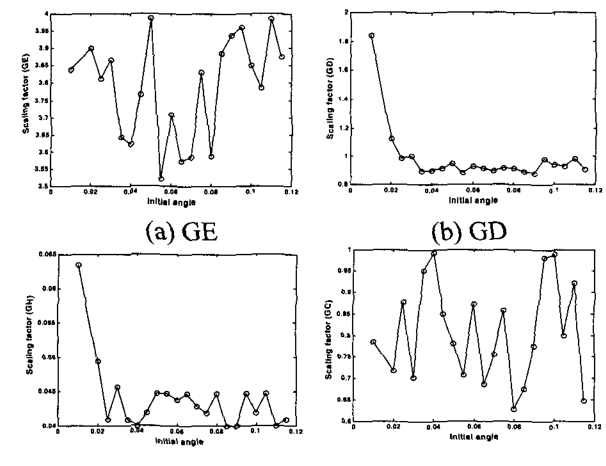
Table 2 The control parameter and performance index (ITAE, Overshoot(%)) of the fuzzy PID controller after genetic optimization in case of $\theta = 0.01,\ldots,0.115$(rad)

Initial Angle (rad)	Scaling Factors				ITAE	Over Shoot(%)
	GE	GD	GH	GC		
0.01	3.838221	1.845552	0.063460	0.784946	0.000773	0.069411
0.02	3.900293	1.124145	0.049384	0.718475	0.004156	0.0
0.025	3.812805	0.987292	0.040938	0.878299	0.005924	0.0
0.03	3.866080	1.004888	0.045630	0.702346	0.011615	0.0
0.035	3.644184	0.901271	0.040938	0.951630	0.011605	0.172315
0.04	3.623656	0.903226	0.04	0.992180	0.016847	0.007811
0.045	3.767351	0.916911	0.041877	0.850440	0.020891	0.006007
0.05	3.985826	0.948192	0.044692	0.780547	0.024612	0.147663
0.055	3.5210170	0.887586	0.04692	0.708211	0.031033	0.131664
0.06	3.707722	0.930596	0.043754	0.873412	0.037682	0.068068
0.065	3.573314	0.981866	4.46920	0.686217	0.045051	0.0
0.07	3.585044	0.899316	0.042815	0.754643	0.047059	0.0
0.075	3.830889	0.916911	0.041877	0.857771	0.047319	0.017514
0.08	3.588465	0.918866	0.044692	0.629032	0.060604	0.0
0.085	3.884653	0.891496	0.04	0.675953	0.051839	0.0
0.09	3.937928	0.883657	0.04	0.776149	0.051433	0.0
0.095	3.960411	0.983382	0.044692	0.981916	0.071910	0.082461
0.1	3.849642	0.942326	0.041877	0.989247	0.074951	0.023834
0.105	3.785435	0.932551	0.046920	0.799120	0.079358	0.103983
0.11	3.985337	0.983382	0.04	0.921799	0.094146	0.0
0.115	3.875367	0.909091	0.040938	0.647116	0.081462	0.0



(a) (b)

Fig. 9 (a) A fitness function (b) tuning procedure of scaling factors in successive generations($\theta = 0.06$rad)



(a) GE (b) GD

(c)GH (d) GC

Fig. 10 Auto-tuned scaling factors according to the change of initial angles (a)GE, (b) GD, (c) GH and (d) GC

Fig. 10 shows auto-tuned scaling factors according to the change of initial angles of the inverted pendulum.

Table 3 summarizes the scaling factors of the fuzzy PD controller that are tuned by using GAs under the same initial condition as those of the fuzzy PID controller, ITAE and OverShoot (%).

Table 3 The control parameter and performance index (ITAE, Overshoot(%)) of the fuzzy PD controller after genetic optimization in case of θ = 0.01,...,0.115(rad)

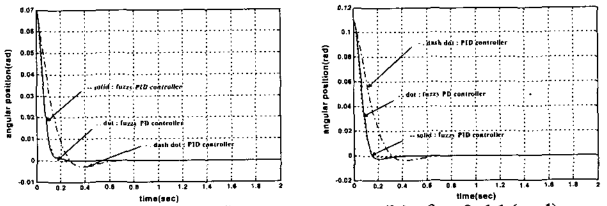
Initial Angle (radian)	Scaling Factors			ITAE	Over Shoot(%)
	GE	GD	GC		
0.01	8.712610	1.137439	1.004888	0.000399	0.058256
0.02	8.250244	0.698925	1.414956	0.001367	0.047433
0.025	8.328445	0.643206	1.497067	0.002415	0.017458
0.03	8.361681	0.621701	1.494135	0.003908	0.088196
0.035	8.308895	0.610948	1.497067	0.005679	0.086351
0.04	9.061584	0.630499	1.488270	0.007141	0.082944
0.045	9.687194	0.674487	1.392962	0.009562	0.051975
0.05	8.494624	0.630499	1.456012	0.012641	0.020423
0.055	9.198436	0.654936	1.394428	0.013748	0.082748
0.06	9.755621	0.659824	1.431085	0.014174	0.090579
0.065	9.120234	0.659824	1.391496	0.018474	0.032252
0.07	8.993157	0.630499	1.472141	0.018803	0.059627
0.075	9.462365	0.669599	1.441349	0.022051	0.000610
0.08	9.384164	0.640274	1.489736	0.021564	0.036157
0.085	9.863148	0.684262	1.379765	0.024754	0.032981
0.09	9.931574	0.679374	1.395894	0.025784	0.044760
0.095	9.921799	0.659824	1.450147	0.025784	0.076515
0.1	9.442815	0.659824	1.482405	0.031177	0.000149
0.105	9.892473	0.6940.97	1.457478	0.034580	0.0
0.11	9.892473	0.708700	1.309384	0.037549	0.22831
0.115	9.892473	0.645161	1.486804	0.032466	0.118137

Table 4 shows the control parameters of the PID controller that are tuned by using GAs under the same initial condition as those of the fuzzy PID controller, ITAE and Overshoot (%).

Table 4 The control parameter and performance index ITAE, Overshoot(%)) of the PID controller after genetic optimization in case of θ = 0.01,..., 0.115(rad)

Initial Angle(rad)	Control Parameters			ITAE	Over Shoot(%)
	K	Ti	Td		
0.01	64.692085	62.883675	0.400616	0.001093	0.048586
0.02	59.670185	62.563278	0.244526	0.008048	0.103347
0.025	78.739006	75.542526	0.174428	0.012917	0.132199
0.03	69.522324	59.279228	0.156579	0.028170	5.087282
0.035	75.934036	56.542522	0.156579	0.031596	0.602903
0.04	68.740410	65.847321	0.156579	0.041281	1.242897
0.045	68.896797	77.888832	0.127263	0.060246	8.373180
0.05	72.884567	57.168053	0.127263	0.065878	5.888550
0.055	70.773392	50.912727	0.127263	0.080656	5.893166
0.06	74.682976	75.777657	0.127263	0.079899	4.355784
0.065	66.238281	70.773392	0.127263	0.105081	6.278503
0.07	73.901054	77.732445	0.127263	0.101790	4.075428
0.075	75.934036	67.880302	0.127263	0.113617	3.552615
0.08	68.896797	57.715397	0.127263	0.141023	4.962779
0.085	75.543076	79.843620	0.127263	0.132412	3.431820
0.09	79.687233	75.777657	0.127263	0.139150	2.691088
0.095	68.114883	39.262180	0.127263	0.201815	4.986214
0.1	75.152122	71.477112	0.127263	0.170988	3.382015
0.105	70.304245	41.920692	0.127263	0.221192	4.401833
0.11	74.682976	66.551048	0.127263	0.198607	3.425961
0.115	79.374466	60.530293	0.127263	0.205189	2.672773

Fig. 11(a) and (b) show the dynamics of output of the system controlled by each controller after genetic optimization in case that the initial angle is 0.07(rad) and 0.11(rad) respectively.



a) θ=0.07(rad) (b) θ= 0.11(rad)

Fig. 11 The dynamics of output of the system controlled by each optimized controller in case of(a) θ = 0.07(rad) and *(b)* θ= 0.11(rad) respectively.

In this Fig., we know that the fuzzy PID controller and fuzzy PD controller are superior to the conventional PID controller from the viewpoint of ITAE and Overshoot percentage.

Now, we consider the case in which the initial angular positions of the inverted pendulum are not included in Table 2,3 and 4 (in other words, selected arbitrarily within the given range). Here we show that the control parameters under the arbitrarily selected initial angular position are not tuned by the GAs and the control parameters of each controller are estimated by using the following two types of estimation algorithms.

Algorithms 1 – HCM and Polynomial (Least Means Square)

We cluster the acquired control parameters by using HCM method. Each cluster is identified by LMS (Least Mean Square) method. The type of polynomial used in LMS method is as the following.

$$P_{estimate} = C_{0j} + C_{1j}\theta_{initial} + C_{2j}\theta^2_{initial} + C_{3j}\theta^3_{initial} \quad (26)$$

where, $P_{estimate}$ is the estimated controller parameter, C_{0j} , C_{1j} , C_{2j} , and C_{3j} are the coefficients of polynomial, which belong to j_{th} cluster, and $\theta_{initial}$ is the angular position variable of the inverted pendulum. Whatever initial angular position is given, we are able to estimate the control parameters by using HCM and the above polynomial.

To estimate the control parameters, the control parameters obtained from Table 2,3 and 4 are classified by HCM method and identified by LMS method. The structure of center vector is shown as the following.

$$V = (\theta, \text{Control Parameter}) \quad (27)$$

The number of the scaling factors for the fuzzy PID controller, which belong to each cluster, is shown Table 5.

After classified by HCM method, the scaling factors of the fuzzy PD controller are divided into three clusters. The number of the scaling factors, which belong to each cluster, is shown Table 6.

Table 5 The number of parameters of each cluster for the fuzzy PID controller

	GE	GD	GH	GC
Cluster 1	7	1	9	7
Cluster 2	10	15	5	7
Cluster 3	4	5	7	7

Table 6 The number of parameters of each cluster for the fuzzy PD controller

	GE	GD	GC
Cluster 1	6	11	8
Cluster 2	8	9	12
Cluster 3	7	1	1

After classified, the control parameters of PID controller are divided into two clusters. The number of the scaling factors, which belong to each cluster, is shown Table 7.

Table 7 The number of parameters of each cluster for the PID controller

	K	Ti	Td
Cluster 1	11	15	13
Cluster 2	6	3	6

In case that the initial angular position is 0.034(rad) or 0.102(rad) not shown in Table 2, we should estimate the scaling factor for we can not acquire the tuned scaling factors from Table 2,3 and 4. We estimate the control parameters of each controller by means of HCM method and LMS method.

Table 8 shows the estimated scaling factors of the fuzzy PID controller and describes performance index(ITAE, Overshoot(%)) of the fuzzy PID controller with the estimated scaling factors in case that the initial angle of inverted pendulum is 0.034(rad) and 0.102(rad) respectively.

Table 8 The estimated parameters and performance index(ITAE, Overshoot(%)) of the fuzzy PID controller in the case of θ=0.034(rad) and 0.102(rad) respectively

Case	Initial Angle (rad)	GE	GD	GH	GC	ITAE	Over Shoot(%)
1	0.034	3.841056	0.913392	0.040402	0.941209	0.010015	0.143057
2	0.102	3.841056	0.913392	0.042440	0.800517	0.069981	0.011672

In case of the fuzzy PD controller, the estimated scaling factors and performance index are shown in Table 9 when the initial angular position is 0.034(rad) and 0.102(rad) respectively.

Table 9 The estimated parameters, ITAE and Overshoot (%) of the fuzzy PD controller in the case of θ=0.034(rad) and 0.102(rad) respectively

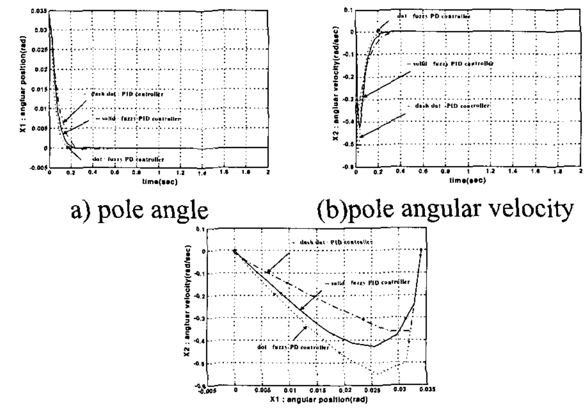
Case	Initial Angle (rad)	GE	GD	GC	ITAE	Over Shoot(%)
1	0.034	8.409417	0.646850	1.388746	0.027212	0.027212
2	0.102	9.854593	0.671228	1.476051	0.030337	0.000357

In case of the PID controller[8], the estimated scaling factors are shown in Table 10 when the initial angular position is 0.034(rad) and 0.102(rad) respectively.

Table 10 The estimated parameters, ITAE and overshoot (%) of the PID controller in the case of θ=0.034(rad) and 0.102(rad) respectively

Case	Initial Angle(rad)	K	Ti	Td	ITAE	Over Shoot(%)
1	0.034	75.990829	72.399765	0.153718	0.027240	1.224719
2	0.102	77.300262	71.467155	0.128666	0.172103	2.716259

Fig. 12 demonstrates (a)pole angle (b)pole angular velocity (c)state space for initial angle θ = 0.034(rad) for each controller (Case 1).

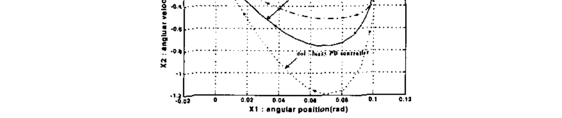


a) pole angle (b)pole angular velocity

(c)state space

Fig. 12 (a)pole angle (b)pole angular velocity (c)state space for initial angle θ= 0.034(rad) (Case 1)

Fig. 13 demonstrates (a)pole angle (b)pole angular velocity (c)state space for initial angle θ= 0.102(rad) (Case 2).



(a)pole angle (b)pole angular velocity

(c)state space

Fig. 13 (a) pole angle (b)pole angular velocity (c)state space for initial angle θ= 0.102(rad) (Case 3)

Algorithms 2 : Neuro-fuzzy model

We implement Neuro-Fuzzy model for parameter estimation. Table 11 shows the estimated scaling factors of fuzzy PID controller and describes performance index (ITAE, Overshoot(%)) of the fuzzy PID controller with the estimated scaling factors in case that the initial angle of inverted pendulum is 0.034(rad) and 0.102(rad) respectively.

Table 11 The estimated parameters by means of neuro-fuzzy model and performance index(ITAE, Overshoot(%)) of the fuzzy PID controller in the case of θ=0.034(rad) and 0.102(rad) respectively

Case	Initial Angle(rad)	GE	GD	GH	GC	ITAE	Over Shoot(%)
1	0.034	3.738344	0.943769	0.043576	0.822975	0.012271	0.15698
2	0.102	3.843736	0.952163	0.043012	0.944191	0.079213	0.027158

In case of the fuzzy PD controller, the estimated scaling factors and performance index are shown in Table 12 when the initial angular position is 0.034(rad), 0.067(rad), and 0.102(rad) respectively.

Table 12 The estimated parameters, ITAE and Overshoot (%) of the fuzzy PD controller in the case of θ=0.034(rad) and 0.102(rad) respectively

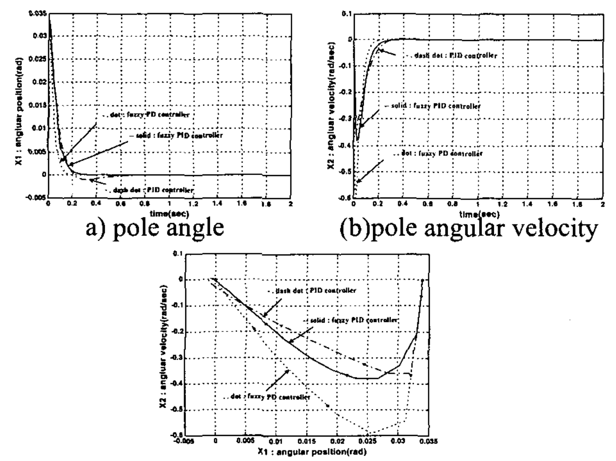
Case	Initial Angle(rad)	GE	GD	GC	ITAE	Over Shoot(%)
1	0.034	8.504756	0.617285	1.498769	0.005214	0.087426
2	0.102	9.751844	0.688485	1.431968	0.033265	0.0

In case of the PID controller, the estimated scaling factors by means of neuro-fuzzy model are shown in Table 13 when the initial angular position is 0.034(rad) and 0.102(rad) respectively.

Table 13 The estimated parameters, ITAE and overshoot (%) of the PID controller in the case of θ=0.034(rad) and 0.102(rad) respectively

Case	Initial Angle(rad)	K	Ti	Td	ITAE	Over Shoot(%)
1	0.034	74.179962	62.227455	0.148222	0.030588	3.174667
2	0.102	71.612167	54.232288	0.127264	0.195189	4.086237

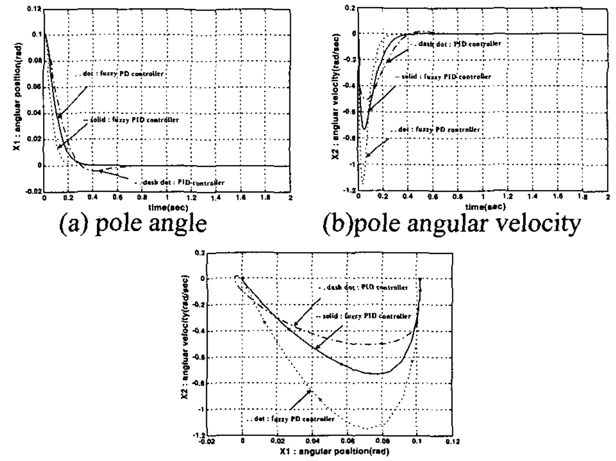
Fig. 14 demonstrates (a)pole angle (b)pole angular velocity (c)state space for initial angle $\theta = 0.034$(rad) (Case 1).


a) pole angle (b)pole angular velocity

(c)state space

Fig. 14 (a)pole angle (b)pole angular velocity (c)state space for initial angle $\theta = 0.034$(rad) (Case 1)

Fig. 15 demonstrates (a)pole angle (b)pole angular velocity (c)state space for initial angle $\theta = 0.102$(rad) (Case 2).


(a) pole angle (b)pole angular velocity

(c)state space

Fig. 15 (a)pole angle (b)pole angular velocity (c)state space for initial angle $\theta = 0.102$(rad) (Case 2).

From the above Fig.s (Fig.s 12~15), we know that the fuzzy PD and the fuzzy PID controller is superior to the PID controller from the viewpoint of performance index.

The PID controller is suitable to linear plant and the fuzzy PD and the fuzzy PID controller have the nonlinear characteristic. So the fuzzy controllers are superior. Fig. 16 depicts the nonlinear characteristic of the fuzzy PD controller in case that GE, GD, and GC is equal to 1.

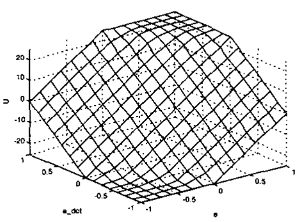


Fig. 16 The input-output relation of the fuzzy PD controller (GE, GD, GC = 1)

Fig. 17 visualizes the input-output relation of the fuzzy PD controller in case of using case 2 of algorithm 1. Note that the fuzzy PD comes with a significant nonlinear mapping between the inputs and output.

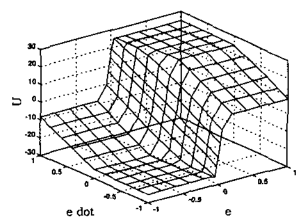


Fig. 17 The input-output relation of fuzzy PD controller (Case 2 of Algorithm 1)

Fig. 18 shows that the input-output relation of the fuzzy PID controller when GE, GD, GH, and GC is equal to 1.

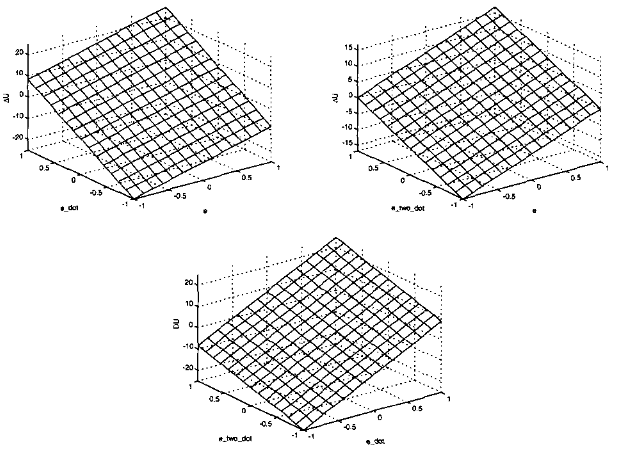


Fig. 18 The surface of the fuzzy PID controller (GE, GD, GH, GC =1)

Fig. 19 visualizes the input-output relation of each of the fuzzy PID controller in case of using case 2 of algorithm 1. Note that the fuzzy PID comes with a significant nonlinear mapping between the inputs and output.

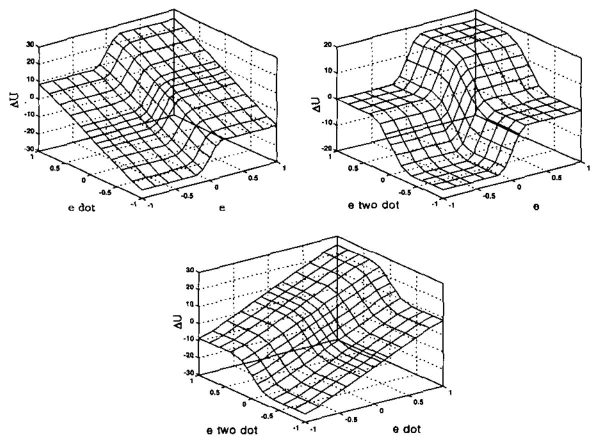


Fig. 19 The surface of the fuzzy PID controller (Case 2 of Algorithm 1)

6. Conclusions

In this paper, we propose the Fuzzy controller design based on the methodology of tuning of control parameters using GAs and estimating of control parameters using two types of estimation algorithms. First, to set the initial individual of GAs applied to controllers, we utilize the scaling factor estimation modes such as BM, CM and EM. Scaling factor estimation modes such as BM, CM and EM which are determined by means of relation between reference, process error and gain respectively is used to set the initial individual of GAs for fuzzy controller. Second, we estimate the control parameters such as GE, GD, GH, and GC by using two types of estimation algorithms so that we may improve the control performance of the fuzzy controller in case that the initial states of the inverted pendulum change.

From the simulation studies, using genetic optimization by scaling factor estimation modes and two types of estimation algorithms, we show that the fuzzy controllers (fuzzy PID controller and fuzzy PD controller) control effectively the inverted pendulum system.

Based on this study, for the performance improvement of output of the inverted pendulum we can consider the advanced estimation algorithms mentioned in the following.

Adopt FCM method to estimate the control parameters.

Consider angular velocity as well as angular position as initial values and adopt three algorithms described in this study to estimate the control parameters.

Use MIMO (Multi Input Multi Output) Neuro-Fuzzy Model to estimate the control parameters

Acknowledgment

This work was supported by a grant No. R02-2000-00284 from Korea Science & Engineering Foundation.

References

[1] S.-K. Oh, *"Fuzzy model and control system by C programming"*, NAEHA Press, Korea, 2002.

[2] D.E. Goldberg, "*Genetic algorithms in Search, Optimization, and Machine Learning*" Addison-Weatley, 1989.

[3] B.-J. Park, Witold Pedrycz and S.-K. Oh, "Identification of Fuzzy Models with the Aid of Evolutionary Data Granulation", *IEE Proceedings-CTA*, Vol. 148, Issue 05, pp. 406-418, 2001

[4] J.R. Jang, "Self-Learning Fuzzy Controllers Based on Temporal Back Propagation", *IEEE Trans. on Neural Networks*, Vol. 3, NO. 5, September, 1992

[5] L. Wang, "Stable Adaptive Fuzzy Controllers with Application to Inverted Pendulum Tracking", *IEEE Trans. On Systems, Man and Cybernetics-Part B: Cybernetics*, Vol. 26, NO. 5, October, 1996

[6] T. Yamakawa, "A New Effective Learning Algorithm for a Neo Fuzzy Neuron Model," *5th IFSA World Conference*, pp. 1017-1020, 1993.

[7] K.J. Astrom and B. Wittenmark, *"Adaptive Control"*, Addison-Wealtley, 1995.

[8] J.G. Ziegler and N.B. Nichols, "Optimum settings for automatic controllers" *Trans. ASME*, 65, pp.433-444, 1942.

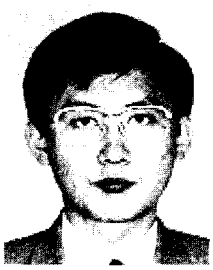
[9] J. Malers and Y.S. Sherif, "Application of fuzzy set theory" *IEEE Trans. on System, Man and Cybernetics.*, vol. SMC-15, No.1, 1985.

[10] T.j. Procyk and E.H. Mamdani, "A linguistic synthe-

sis of fuzzy controller" *Automatica*, vol.15, pp.15-30, 1979.

[11] S.-K. Oh and W. Pedrycz, "The Design of Hybrid Fuzzy Controllers Based on Genetic Algorithms and Estimation Techniques", *Kybernetes,*2002 (to appear)

[12] S.-K. Oh, W. Pedrycz and D.-K. Lee, "The Genetic Design of Hybrid Fuzzy Controllers", *Cybernetics & Systems,* 2001 (submitted)



Sung-Kwun Oh received the B.S., M.S., and Ph. D. degrees in electrical engineering from Yonsei University, Korea, in 1980, 1983, and 1993, respectively. He had worked as a postdoctoral fellow in the Department of Electrical and Computer Eng. Univ. of Manitoba, Canada, from 1996 to 1997. He is currently an Associate Professor in the School of Electrical, Electronic and Information Engineering, Wonkwang University. His research interests include fuzzy and neural systems, intelligent control, and computational intelligence. He currently serves as an Associate Editor of the journal of KIEE and ICASE.
E-mail: ohsk@wonkwang.ac.kr

Seok-Beom Rho received the M.S. degree in control and instrumentation engineering from Wonkwang University, in 1995. He is currently studying for Ph. D. degree at the same institute. His research interests are mainly in the theory and technology of intelligent systems, control, and soft computing to learning and machine intelligence.