DOI: 10.5370/JEET.2010.5.4.653

The Design of Fuzzy Controller Based on Genetic Optimization and Neurofuzzy Networks

Sung-Kwun Oh† and Seok-Beom Roh*

Abstract – In this study, we introduce a neurofuzzy approach to the design of fuzzy controllers. The development process exploits key technologies of Computational Intelligence (CI), namely, genetic algorithms (GA) and neurofuzzy networks. The crux of the design methodology deals with the selection and determination of optimal values of the scaling factors of fuzzy controllers, which are essential to the entire optimization process. First, the tuning of the scaling factors of the fuzzy controller is carried out. Next, we form a nonlinear mapping for the scaling factors, which are realized by GA-based neurofuzzy networks by using a fuzzy set or fuzzy relation. The proposed approach is applied to control nonlinear systems like the inverted pendulum. Results of comprehensive numerical studies are presented through a detailed comparative analysis.

Keywords: Fuzzy controller, Estimation algorithm, Scaling factors, Genetic optimization, FS-based neurofuzzy networks (NFN), FR-based NFN

1. Introduction

The ongoing challenge for designing advanced system control has resulted in a diversity of underlying methodologies, development platforms, and detailed algorithms. Similar to PID controllers that are regarded nowadays as standard control constructs of numeric control [1]-[4], fuzzy controllers have positioned themselves in a similar dominant role at the knowledge-rich end of the entire spectrum of control algorithms. The design goals of PID control and fuzzy control are similar. However, the same problem is approached from two different angles. Clearly, at the final phase of the design phase, two different design facets are being served. PID controllers can excellent deal with linear systems or nonlinear systems with an operation mode confined to a small neighborhood around a given setpoint. For its advantages, fuzzy controllers can be positioned at the opposite end of the scale in order to envision their full strength in nonlinear system settings (i.e., controllers as nonlinear mappings), as well as when dealing with high deviations from the given setpoint. Such advantages of fuzzy controllers stem directly from the nonlinear-type characteristics of linguistic rules and the associated membership functions used in describing corresponding linguistic terms. The objective of this study is to develop, optimize, and experiment with fuzzy controllers (e.g., fuzzy PD controller or fuzzy PID controller) when developing their general design methodology cast in accordance with computational intelligence (CI) settings. However, one of the difficulties in the construction of a fuzzy controller is the derivation of a set of optimal control parameters, such as linguistic control rules, scaling factors, and membership functions of the fuzzy sets, requiring individual sets of rules. In the conventional design method being applied, a control expert introduces some linguistic rules and decides upon the type and parameters of the associated membership functions. With an attempt to enhance the quality of the control knowledge conveyed by the expert, as well as to provides for a more coherent (i.e., this usually relates to the process of calibration of such initial domain knowledge), genetic algorithms (GAs) have since then been applied. Considering a vast number of parameters of the fuzzy controller, GAs are instrumental in carrying out a global search in the overall parameter space.

Many studies have reported the optimal design of control parameters using various EAs, such as GA [18], MPSO [19]-[22], and DE [23]. EAs employ a robust search and optimization methodology, and are able to cope with ill-defined problem domains, such as multimodality, discontinuity, time-variance, randomness, and noise [24].

Evolutionary computing, such as GAs, is computationally intensive and may be a point of concern when dealing with the prescribed time allowed in completing such search. For instance, when controlling a nonlinear plant, such as an inverted pendulum where initial states vary for each controlled case, the search time required by GAs could be prohibitively high, especially when dealing with dynamic systems. Consequently, the parameters of the fuzzy controller cannot be easily adapted to the changing initial states of this system, such as for an angular position and an angular velocity of the pendulum. To overcome this shortcoming, we introduce nonlinear mapping between the initial states of the system and the corresponding optimal values of the

† Corresponding Author: Dept. of Electrical Engineering, The University of Suwon (ohsk@suwon.ac.kr)

* Department of Electrical Engineering, National Institute of Technology, Warangal – 506 004 Andhra Pradesh, India. (vinodkumar.dm@gmail.com)

Received: February 5, 2010; Accepted: April 5, 2010

parameters of the controller. By anticipating possible nonlinearities residing within such transformation, we first consider GA-based neurofuzzy networks (NFN), namely, fuzzy set-based and fuzzy relation-based NFNs. The development process consists of two main phases. First, using genetic optimization, we determine optimal parameters of the fuzzy controller for various initial states (conditions) of the dynamic system. Second, we build on a nonlinear model to capture the relationship between the initial states of the system and the corresponding genetically optimized control parameters. The paper includes the experimental study on inverted pendulum, but with the initial states changed.

The essential core of this study is to reduce computational complexity in order to find the optimal control parameters without reducing control performance.

The characteristic whereby optimal control parameters can be estimated by nonlinear mapping between the initial condition and the control parameters with low computation load assures us that we can obtain the adaptive control parameters for various initial conditions of a given plant online.

We carry out experimentation with several categories of controllers, such as PID, fuzzy PD, and fuzzy PID. The controlled systems performances are evaluated and compared from the viewpoint of several typical criteria, such as ITAE (i.e., integral of the time multiplied by the absolute value of error), overshoot, and rising time [1].

2. The Fuzzy Controller

The block diagram of the fuzzy PID controller is presented in Fig. 1. This paper confines itself to the following standard notation: e denotes the error between reference and response (i.e., output of the controlled system), Δe is the first-order difference of error signal, and Δ^2e is the second-order difference of the error. The input variables to the fuzzy controller are transformed by some scaling factors (GE, GD, GH, and GC), whose roles are to allow the fuzzy controller to "perceive" properly the external world that needs to be controlled.

The above fuzzy PID controller consists of rules in the following form (cf. [9], [10]):

R_j : If E is A_{1j}, ΔE is A_{2j}, and Δ^2E 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, the level of activation is computed in a standard fashion given by (1). Subsequently, the inferred value of the consequence part is converted into numeric values using (2-1) [11].

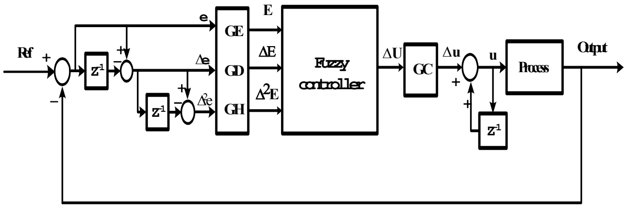


Fig. 1. An overall architecture of the fuzzy PID controller.

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

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

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

An overall operation of a fuzzy PID controller should be described, such that the resulting control is incrementally formed based on the previous value of control:

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

In this paper, the input variables are denoted by *E* and *ΔE,* whereas 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 Δ^2E, the membership functions are denoted as follows:

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

The membership functions of the output variable of the controller corresponding to 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. The set of rules is shown in Table 1.

We employ triangular membership functions defined in the input and output spaces (Figs. 2 and 3). The choice of

Table 1. Fuzzy control rules

(a) The two 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) The three input variables

Δ^2E=N

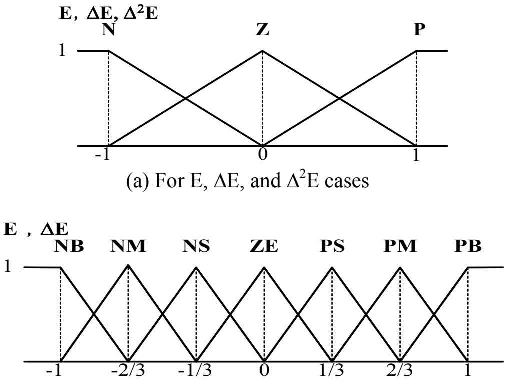
		ΔE		
		N	Z	P
E	N	-m3	-m2	-m1
	Z	-m1	0	m1
	P	m1	m2	m3

Δ^2E=Z

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

Δ^2E=P

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



(a) For E, ΔE, and Δ^2E cases

(b) For E and ΔE cases

Fig. 2. Membership functions of the premise input variables.

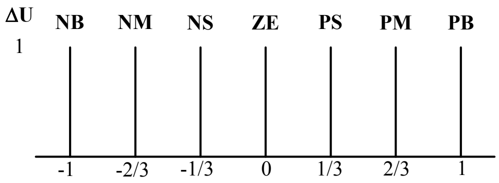


Fig. 3. Membership functions (singletons) defined in the consequence variable, ΔU.

these membership functions has been dictated by their simplicity and obvious interpretation. These spaces are normalized to the [-1, 1] interval.

3. Auto-Tuning of the Fuzzy Controller by GA

Genetic algorithms are search algorithms inspired by nature; that is, it can exploit the fundamental concept of "survival of the fittest" encountered by the selection mechanisms. In GAs, search variables are encoded in bit strings format called chromosomes, whereby each represents a possible solution for a given problem. A chromosome has a fitness value corresponding to how good a solution is. In control applications, the chromosome represents the adjustable parameters of the controller. Fitness value is a quantitative measure of the performance of the controller.

Population size, number of bits used for binary coding, crossover rate, and mutation rate are essential parameters. Their values are specified in advance. The genetic search is guided by reproduction, mutation, and crossover. Each of these phases comes with a set of specific numeric parameters. In this study, the number of generations is 100, crossover rate is 0.6, and mutation rate is 0.1. The number of bits used in the coding is 10. This selection is a result of intensive experiments. These values are similar to those found in literatures [3, 17].

Fig. 4 shows an overall auto-tuning scheme involving the tuned scaling factors and constructed control rules, all of which are genetically optimized. We set the initial individuals of GAs using three types of parameter estimation modes: basic, contraction, and expansion. 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 the contraction mode (CM), we use scaling parameters reduced by 25% in relation to the BM. In the expansion mode (EM), we use scaling parameters enlarged by 25% from the BM. The standard ITAE expressed for the reference and the output of the controlled system are treated as a fitness function [2].

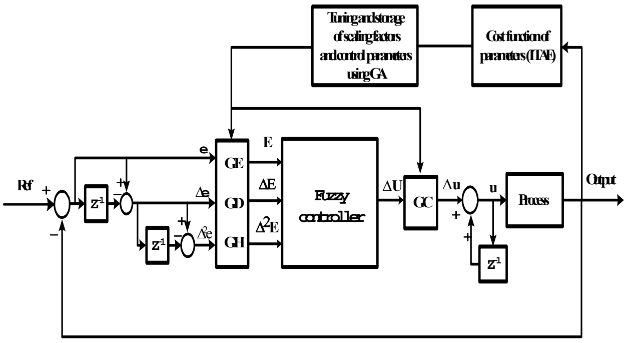

Tuning and storage of scaling factors and control parameters using GA
Cost function of parameters (ITAE)
Rf
+
-
z^{-1}
+
-
e
Δe
Δ^2e
GE
GD
GH
E
ΔE
Δ^2E
Fuzzy controller
ΔU
GC
Δu
u
+
+
z^{-1}
Process
Output

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

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

[Step 1] We select the general structure of the fuzzy controller based on the purpose of the control and dynamics of the process. In particular, we consider architectural options (PID, FPD/fuzzy PD, and FPID/fuzzy PID) controller).

[Step 2] We define the number of fuzzy sets for each variable and set up initial control rules (Figs. 2 and 3).

[Step 3] We form a collection of initial individuals of GAs. Then, we 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.a), (4.b), (4.c), and (4.d).

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

$$\mathrm{E(kT)} = \mathrm{error} \times \mathrm{GE} \tag{4.a}$$

$$\Delta\mathrm{E(kT)} = [\mathrm{error(kT)} - \mathrm{error((k-1)T)}] \times \mathrm{GD} \tag{4.b}$$

$$\Delta^2 E(kT) = [error(kT) - 2error((k-1)T) + error((k-2)T)] \times GH \tag{4.c}$$

$$\mathrm{U(kT) = U((k-1)T) + \Delta U(kT) \times GC} \tag{4.d}$$

[Step 4] All of the control parameters, such as the scaling factors GE, GD, GH, and GC are, tuned simultaneously.

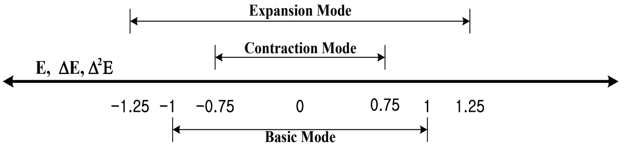


Fig. 5. Three types of estimation modes for the scaling factors: basic, expansion, and contraction.

4. Estimation Algorithm by using GA-based Neurofuzzy Networks

To estimate the control parameters, we use two types of NFNs, such as fuzzy set-based NFNs and fuzzy relation-based NFNs. Both are tuned by GAs.

4.1 Fuzzy Set-based Neurofuzzy Networks (FS-NFN)

As shown in Fig. 6, FS-NFN can be designed using space partitioning realized in terms of individual input variables. Its topology deals with granulation carried out with the aid of fuzzy sets that are defined for each input variable.

In Fig. 6, the circles denote the generic computing units of the FS-NFN, where "N" describes the normalization procedure applied to the membership grades of the input variable x_i. The output $f_i(x_i)$ of the "∑" neuron is governed by nonlinear function f_i. Finally, the output of the FS-NFN $\hat{y}$ can be read as

$$\hat{y} = f_1(x_1) + f_2(x_2) + \cdots + f_m(x_m) = \sum_{i=1}^{m} f_i(x_i) \tag{5}$$

where m is the number of input variables (viz. the number of outputs f_i's of "∑" neurons in the network). We can regard each f_i, given by (5), into the following mappings (rules):

$$\begin{aligned} &R^1 : If \;\; x_i \;\; is \;\; A_{i1} \;\; then \;\; y_{i1} = w_{i1} \\ &\quad\vdots \qquad\qquad\qquad\qquad \vdots \\ &R^j : If \;\; x_i \;\; is \;\; A_{ij} \;\; then \;\; y_{ij} = w_{ij} \\ &\quad\vdots \qquad\qquad\qquad\qquad \vdots \\ &R^n : If \;\; x_i \;\; is \;\; A_{in} \;\; then \;\; y_{in} = w_{in} \end{aligned} \tag{6}$$

The learning of FS-NFN is realized by adjusting the connections of the neurons such that they follow a standard back-propagation (BP) algorithm. In this study, we use the Euclidean error, where the distance between the target and the model is expressed by

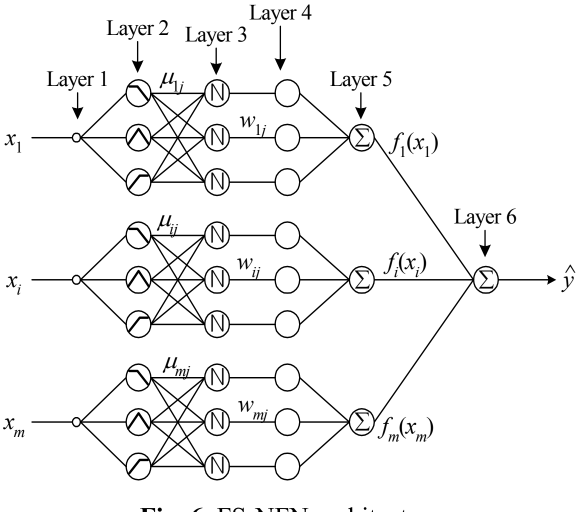


Fig. 6. FS-NFN architecture.

$$E_p = (y_p - \hat{y}_p)^2 \tag{7}$$

$$E = \sum_{p=1}^{N} (y_p - \hat{y}_p)^2 \tag{8}$$

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

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

$$w(new) = w(old) + \Delta w \tag{9}$$

where the update formula follows the gradient descent method,

$$\begin{aligned} \Delta w_{ij} = \eta \cdot \left(-\frac{\partial E_p}{\partial w_{ij}} \right) &= -\eta \frac{\partial E_p}{\partial \hat{y}_p} \cdot \frac{\partial \hat{y}_p}{\partial f_i(x_i)} \cdot \frac{\partial f_i(x_i)}{\partial w_{ij}} \\ &= 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \mu_{ij}(x_i) \end{aligned} \tag{10}$$

where η denotes positive learning rate.

To accelerate convergence, a momentum term is added to the learning expression. By combining (10) with a momentum term, the complete update formula incorporating the discussed components becomes

$$\Delta w_{ij} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \mu_{ij}(x_i) + \alpha(w_{ij}(t) - w_{ij}(t-1)) \tag{11}$$

where the momentum coefficient, α, is constrained to the unit interval.

4.2 Fuzzy Relation-based Neurofuzzy Networks (FR-NFN)

Let us consider an extension of the network with the fuzzy partition realized by fuzzy relations. Fig. 7 demonstrates an architecture of such FR-NFN (i.e., two inputs and one output), where each input assumes three membership functions. The circles denote the processing units of FR-NFN. The node notated by $\prod$ denotes a Cartesian product whose output is the product of all the incoming signals. N denotes the normalization of the membership grades.

In the language of rule-based systems, a structure corresponds to the following collection of rules:

$$\begin{aligned} &R^1: \textit{If } x_1 \textit{ is } A_{11} \textit{ and} \cdots x_k \textit{ is } A_{1k} \textit{ then } y_1 = w_1 \\ &\quad\vdots \qquad\qquad\qquad\qquad \vdots \\ &R^j: \textit{If } x_1 \textit{ is } A_{j1} \textit{ and} \cdots x_k \textit{ is } A_{jk} \textit{ then } y_j = w_j \\ &\quad\vdots \qquad\qquad\qquad\qquad \vdots \\ &R^n: \textit{If } x_1 \textit{ is } A_{n1} \textit{ and} \cdots x_k \textit{ is } A_{nk} \textit{ then } y_n = w_n \end{aligned} \tag{12}$$

The fuzzy rules in (12) constitute overall networks of modified FR-NFN, such as the one included in Fig. 7. The output f_i of each node generates a final output, $\hat{y}$, in the form,

$$\hat{y} = \sum_{i=1}^{n} f_i = \sum_{i=1}^{n} \overline{\mu}_i \cdot w_i = \sum_{i=1}^{n} \frac{\mu_i \cdot w_i}{\sum_{i=1}^{n} \mu_i} \tag{13}$$

The learning of the FR-NFN is realized by adjusting the connections of the neurons. Hence, it follows a standard BP algorithm. As far as learning is concerned, the connections change based on the formula,

$$w(new) = w(old) + \Delta w \tag{14}$$

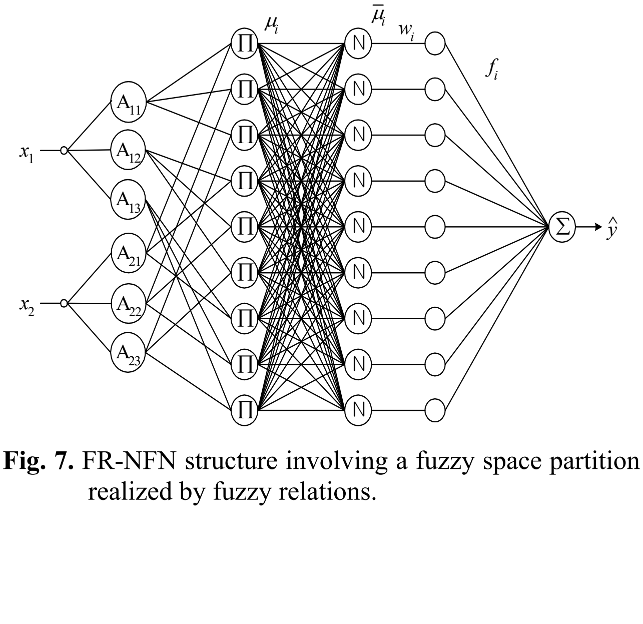


Fig. 7. FR-NFN structure involving a fuzzy space partition realized by fuzzy relations.

where the update formula follows the gradient descent method,

$$\begin{aligned} \Delta w_i = \eta \cdot \left(-\frac{\partial E_p}{\partial w_i} \right) &= -\eta \cdot \frac{\partial E_p}{\partial \hat{y}_p} \cdot \frac{\partial \hat{y}_p}{\partial f_i} \cdot \frac{\partial f_i}{\partial w_i} \\ &= 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \overline{\mu}_i \end{aligned} \tag{15}$$

where η denotes a positive learning rate.

To accelerate convergence, a momentum term is added to the learning expression. By combining (15) and the momentum term, the complete update formula incorporating the discussed components becomes

$$\Delta w_i = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \mu_i + \alpha (w_i(t) - w_i(t-1)) \tag{16}$$

where the momentum coefficient, α, is constrained to the unit interval.

In this algorithm, the momentum term and the fuzzy membership function of the abovementioned FS-NFN and FR-NFN use GA to optimize the learning rate. We use 1000 generations, a population of 60 individuals, 10 bits per string, crossover rate equal to 0.6, and mutation probability equal to 0.1.

Fig. 8 depicts the detailed flowchart of the complete tuning and estimating process.

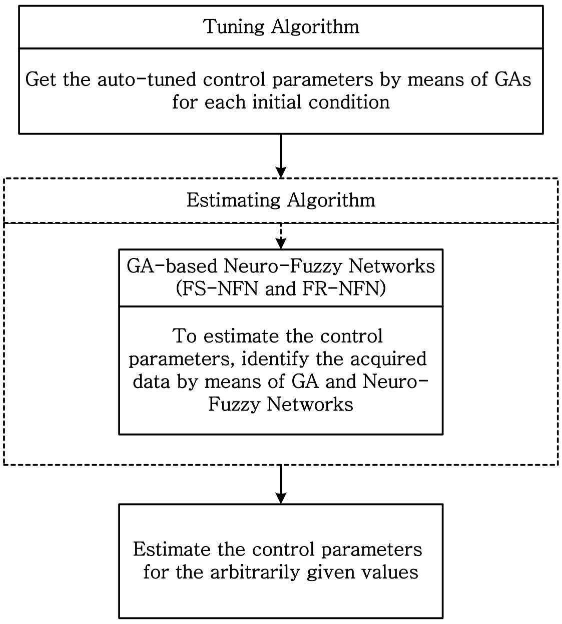

Tuning Algorithm
Get the auto-tuned control parameters by means of GAs for each initial condition
Estimating Algorithm
GA-based Neuro-Fuzzy Networks (FS-NFN and FR-NFN)
To estimate the control parameters, identify the acquired data by means of GA and Neuro-Fuzzy Networks
Estimate the control parameters for the arbitrarily given values

Fig. 8. Overall tuning and estimating process.

5. Experimental Studies

The proposed control scheme is useful for a variety of control problems. In this section, we demonstrate the effectiveness of the fuzzy PID controller by applying it to the inverted pendulum (Fig. 9).

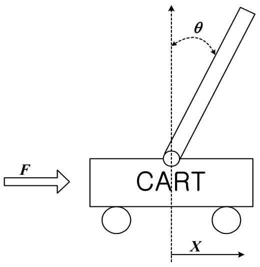


Fig. 9. The inverted pendulum.

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 an 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) and $\dot{\theta}$ (angular velocity of the pole). The behaviors of these two state variables are governed by a second-order equation [i. e., (17)]. The dynamic equation of the inverted pendulum comes in the form

$$\ddot{\theta}=\frac{g\sin\theta+\cos\theta\left(\dfrac{-F-ml\dot{\theta}^2\sin\theta}{m_c+m}\right)}{l\left(\dfrac{4}{3}-\dfrac{m\cos^2\theta}{m_c+m}\right)} \tag{17}$$

where *g* (acceleration due to gravity) is 9.8 m/s^2, m_c (mass of cart) is 1.0 *kg*, *m* (mass of pole) is 0.5 *kg*, and F is the applied force expressed in Newtons.

Our control goal is to balance the pole without considering the position and velocity of the cart. We then compare the fuzzy PID controller and the fuzzy PD controller with the conventional PID controller under similar conditions in order to validate both fuzzy PID and fuzzy PD controllers. The experiment is completed in discrete time. The sampling period is 10 milliseconds.

- Tuning and estimation of control parameters

Control parameters, such as GE, GD, GH, and GC tuned by GAs, are obtained because they affect the performance of the controller.

GAs are powerful nonlinear optimization techniques. However, the performance of genetic optimization is realized at the expense of significant computing overhead. To overcome this deficiency, we first select several initial conditions and determine the auto-tuned control parameters by means of GAs based on changes in each selected initial conditions. Second, we use the algorithms to estimate the control parameters. We consider ITAE and overshoot as the performance index (PI) of the controller. We have completed an experiment for two cases. In the first case, initial angular positions are selected as 0.1 (rad)., 0.12 (rad).,…, 0.78 (rad)., and 0.8 (rad). The second experiment covers the values 0.1 (rad)., 0.2 (rad).,…, 0.7 (rad)., and 0.8 (rad), regarded as a collection of initial angular positions, as well as 0.1 (rad/sec), 0.2 (rad/sec),…, 0.7 (rad/sec), and 0.8 (rad/sec) forming the corresponding family of the values of the initial angular velocity.

In each case, we tune the control parameters of each controller (fuzzy PID, fuzzy PD, and PID). In the two cases, we use two kinds of GA-optimized NFN (FS-NFN and FR-NFN) to estimate the control parameters.

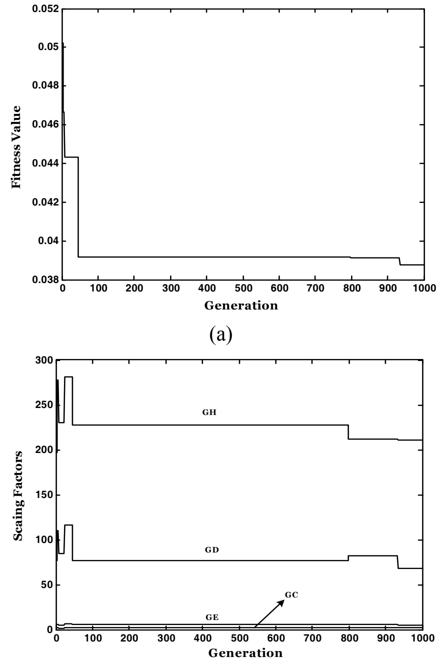
The networks, FS-NFN and FR-NFN, are applied to Cases 1 and 2, respectively. The former is the topology concerned with the granulation carried out in terms of the fuzzy set defined in each input variable, whereas the latter can be used to cover all the variables simultaneously.

In this simulation study, the control parameters of the PID controller are optimized and estimated in the same way. They are used for optimizing and estimating the control parameters of the fuzzy PID controller.

Case 1) Only the angular position is considered for the initial condition

In Case 1, we consider the initial angular position, but not the initial angular velocity. The initial angular velocity is fixed at zero. We change the initial angular position from 0.12 to 0.8 (rad) at an interval of 0.01 (rad).

Fig. 10 shows (a) the performance of a fitness function, where θ=0.14 (rad), and (b) the tuning procedure of scaling factors, such as GE, GD, GH, and GC, in successive generation with the aid of GAs (see also Table 2).



(a)

(b)

Fig. 10. (a) A fitness function and the (b) tuning procedure of scaling factors in successive generations (θ=0.14 rad).

Fig. 11 shows the auto-tuned scaling factors that are completed based on changes in initial angles of the inverted pendulum.

Figs. 12(a) and (b) show the dynamics of the output of the system controlled by each controller after genetic optimization, where the initial angles are 0.3 and 0.7 (rad), respectively.

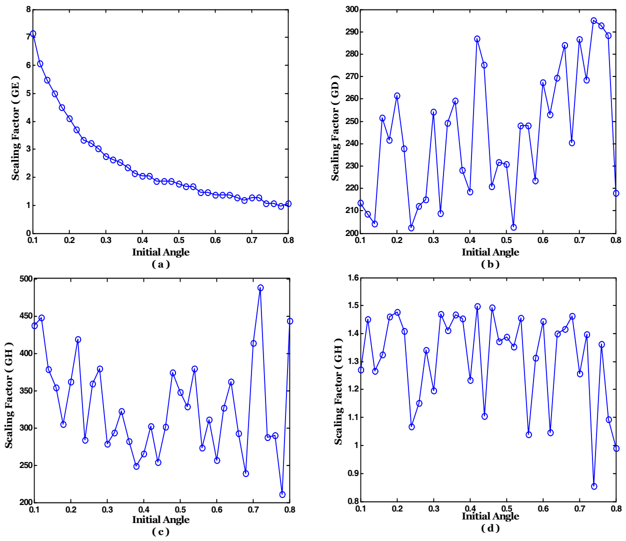


Fig. 11. Auto-tuned scaling factors based on changes in initial angles: (a) GE, (b) GD, (c) GH, and (d) GC.

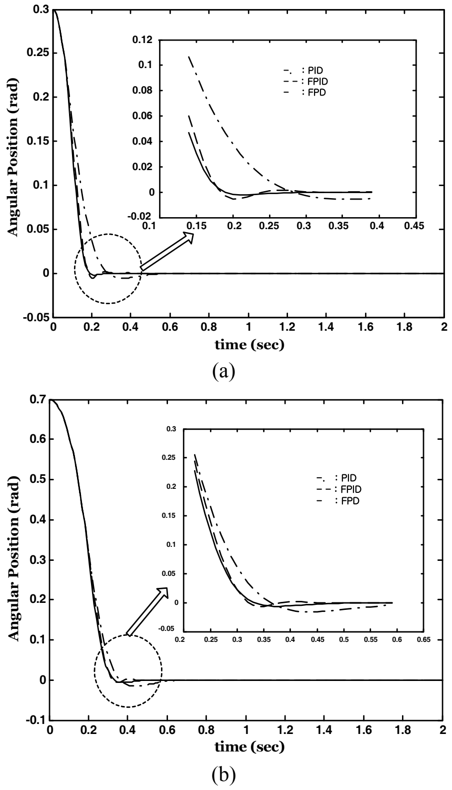


Fig. 12. The dynamics of the output of the system controlled by each optimized controller, where (a) θ=0.3 (rad) and *(b)* θ=0.7 (rad).

As shown in Fig. 12, both fuzzy PID and fuzzy PD controllers are superior to the conventional PID controller. This performance has been quantified from the viewpoint of ITAE, overshoot percentage, and rising time.

Thereafter, we consider a case where the initial angular positions of the inverted pendulum are not obtained through GAs, or those arbitrarily selected within some given range. The control parameters under the arbitrarily selected initial angular position are not tuned by the GAs. The control parameters of each controller are estimated using the estimation algorithm of GA-based FS-NFN.

We implement optimal neurofuzzy networks (i.e., FS-FNN) for parameter estimation using GAs. In this algorithm, we adjust the learning rates, momentum coefficient, and apexes of membership function of NFNs by using GAs.

Table 2 shows the estimated scaling factors of the fuzzy PID controller and describes PI [ITAE, overshoot (%), rising time (sec)] of the fuzzy PID controller with the estimated scaling factors, where the initial angles of inverted pendulum are 0.1849, 0.4854, 0.38878, and 0.7133 (rad).

As for the fuzzy PD controller, the estimated scaling factors and PI are obtained, where the initial angular positions are 0.1849, 0.4854, 0.38878, and 0.7133 (rad) (Table 3).

In the PID controller, the scaling factors estimated by the neurofuzzy model, where the initial angular positions are set as 0.1849, 0.4854, 0.38878, and 0.7133 (rad), as shown in Table 4.

In LQG controller based on Matlab toolbox, the control performances of LQG controller when the initial angular positions are set as 0.1849, 0.4854, 0.38878, and 0.7133 (rad), as shown in Table 5.

Table 2. Estimated parameters based on GA-based NFN and PI [ITAE, overshoot (%), rising time (sec)] of the fuzzy PID controller, where θ=0.1849, 0.4854, 0.38878, and 0.7133 (rad)

Case	Initial Angle (rad)	GE	GD	GH	GC	ITAE	Over Shoot (%)	Rising Time (sec)
1-1	0.185	4.11	65.88	203.59	1.94	0.07	0.0	0.092
1-2	0.485	1.84	60.66	181.51	1.89	0.55	2.29	0.144
1-3	0.389	2.23	62.80	183.07	1.87	0.33	2.61	0.127
1-4	0.713	1.30	49.17	159.89	1.74	1.50	2.29	0.186

Table 3. Estimated parameters based on GA-based NFN and PI [(ITAE, overshoot (%), rising time (sec)] of the fuzzy PD controller, where θ=0.1849, 0.4854, 0.38878, and 0.7133 (rad)

Case	Initial Angle (rad)	GE	GD	GC	ITAE	Over Shoot (%)	Rising Time (sec)
1-1	0.185	8.84	0.40	3.91	0.06	0.0	0.082
1-2	0.485	4.57	0.28	3.87	0.52	0.04	0.143
1-3	0.389	5.32	0.31	3.78	0.31	0.0	0.126
1-4	0.713	3.06	0.21	3.84	1.47	2.03	0.181

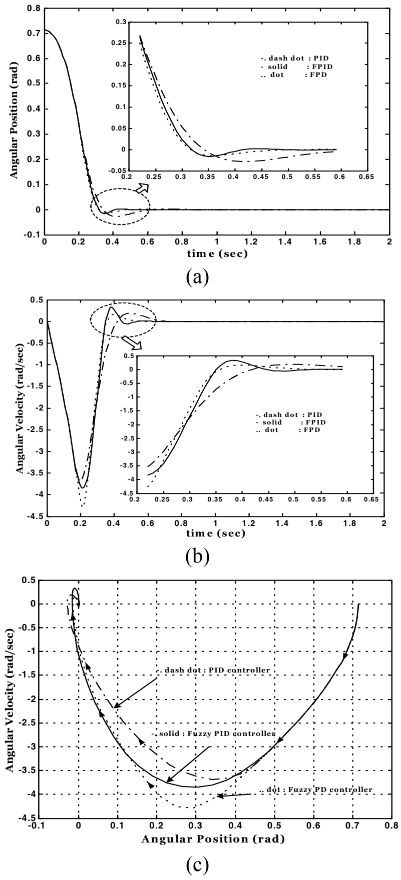
Table 4. Estimated parameters, ITAE, overshoot (%), and rising time of the PID controller, where θ=0.1849, 0.4854, 0.38878, and 0.7133 (rad)

Case	Initial Angle (rad)	K	Ti	Td	ITAE	Over Shoot (%)	Rising Time (sec)
1-1	0.185	129.30	253.00	0.098	0.18	3.13	0.15
1-2	0.485	129.29	248.10	0.101	0.76	2.09	0.18
1-3	0.389	129.52	251.78	0.096	0.51	3.62	0.16
1-4	0.713	128.21	249.20	0.093	1.82	3.97	0.20

Table 5. The estimated parameters, ITAE, overshoot (%), and rising time of the LQG controller, where θ=0.1849, 0.4854, 0.38878, and 0.7133 (rad)

Case	Initial Angle (rad)	ITAE	Over Shoot (%)	Rising Time (sec)
1-1	0.185	3.64	0.42	0.87
1-2	0.485	10.41	0.48	0.89
1-3	0.389	8.03	0.45	0.88
1-4	0.713	17.63	0.60	0.937500

Fig. 13 demonstrates (a) pole angle (b) pole angular velocity, and (c) state space for initial angle θ=0.7133 (rad) (Cases 1-4).



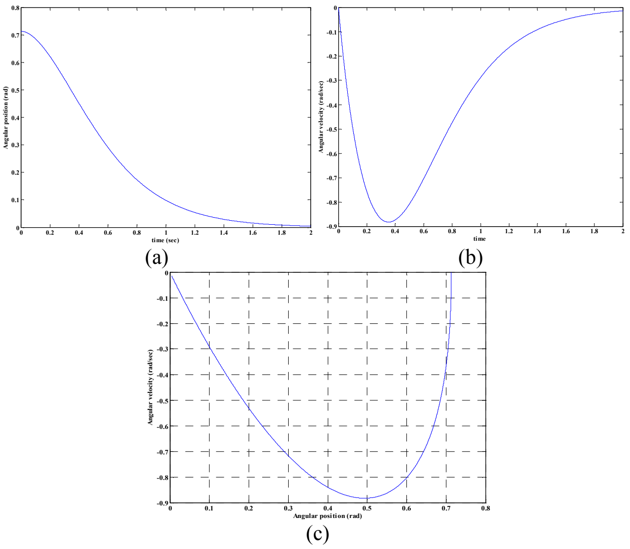
(a)

(b)

(c)

Fig. 13. (a) Pole angle; (b) Pole angular velocity; (c) State space for initial angle θ=0.7133 (rad) (Cases 1-4).

Fig. 14 demonstrates (a) pole angle, (b) pole angular velocity, and (c) state space for initial angle θ=0.7133 (rad) (Cases 1-4) for the LQG controller.



(a)

(b)

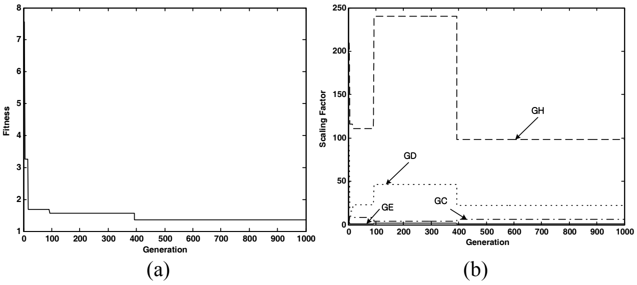
(c)

Fig. 14. (a) Pole angle; (b) Pole angular velocity; (c) State space for initial angle θ=0.7133 (rad) (Cases 1-4 and LQG).

Case 2) The initial angular position and angular velocity simultaneously vary

We consider a more complicated situation under which both the initial angular position and velocity of the inverted pendulum vary within some range. To deal with this variability, more data is needed to estimate the required control parameters. In this study, we vary the initial conditions of the inverted pendulum, such as the initial angular position from 0.1 to 0.8 (rad) at the interval of 0.1 (rad), and the initial angular velocity from 0.1 to 0.8 (rad/sec) at the interval of 0.1 (rad/sec). Then, we tune the control parameters of each controller (fuzzy PID, fuzzy PD, and PID). This optimization completes each initial condition.

Fig. 15 shows the performance represented in terms of fitness function, where θ=0.6 (rad) and $\dot{\theta}$=0.4 (rad/sec). It also presents the tuning procedure of scaling factors, such as GE, GD, GH, and GC, based on successive generation using GAs.



(a)

(b)

Fig. 15. (a) Fitness function and (b) tuning procedure of scaling factors in successive generations (θ=0.6 rad and $\dot{\theta}$=0.4 rad/sec).

Fig. 16 shows the value of the scaling factors treated as functions of initial angular position, as well as the angular velocity of the inverted pendulum in the fuzzy PID controller. Evidently, nonlinear characteristics exist.

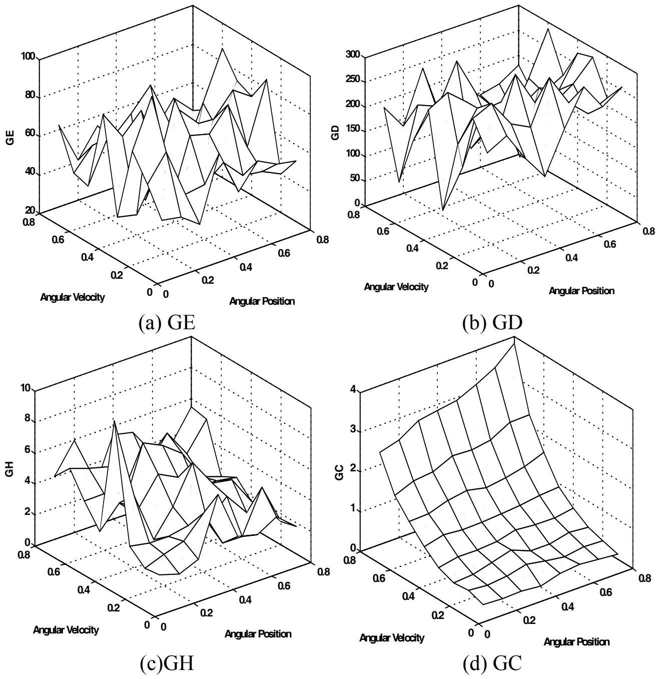
Figs. 17(a) and (b) illustrate the dynamics of the output of the system controlled by each controller after genetic optimization, where (a) θ=0.3 (rad), $\dot{\theta}$=0.7 (rad/sec) and (b) θ=0.7 (rad), $\dot{\theta}$=0.5 (rad/sec), respectively.

In Fig. 17, from the viewpoint of ITAE, overshoot, and rising time, both fuzzy PID and fuzzy PD controllers are evidently superior than the conventional PID controller.

Then, we consider the case in which the initial angular positions and angular velocities of the inverted pendulum are not genetically tuned. Herein, we show that the control parameters under the arbitrarily selected initial condition are not tuned by GAs. The control parameters of each controller are estimated using the estimation algorithm of the GA-based FR-NFN rather than by the GA-based FS-NFN. We implement the optimal NFNs for parameter estimation using GAs, as we have done in the Case 1.

Table 6 shows the estimated scaling factors of the fuzzy PID controller and the values of PI (ITAE, overshoot, and rising time) of the fuzzy PID controller in accordance with the estimated scaling factors, where θ=0.22, 0.45, 0.78 (rad) and $\dot{\theta}$=0.22, 0.45, 0.78 (rad/sec), respectively.

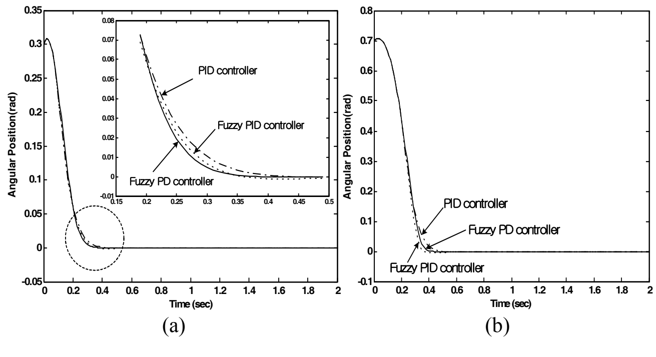
As for the fuzzy PD controller, the estimated scaling factors and PI can be observed when the initial angular positions are 0.22, 0.45, or 0.78 (rad), and the initial angular velocities are 0.22, 0.45, or 0.78 (rad/sec), respectively (Table 7).



(a) GE (b) GD

(c)GH (d) GC

Fig. 16. Auto-tuned scaling factors based on changes in initial angles and angular velocity in the fuzzy PID controller: (a) GE, (b) GD, (c) GH, and (d) GC.



(a) (b)

Fig. 17. The dynamics of output of the system controlled by each optimized controller, where (a) θ=0.3 (rad), $\dot{\theta}$=0.7 (rad/sec) and (b) θ=0.7 (rad), $\dot{\theta}$=0.5 (rad/sec).

Table 6. Estimated parameters completed by the GA-based NFN and the values of the PI (ITAE, overshoot, and rising time) of the fuzzy PID controller for selected combinations of the initial values of angle and their derivatives, θ=0.22, 0.45, 0.78 (rad) and $\dot{\theta}$=0.22, 0.45, 0.78 (rad/sec)

Case	θ (rad)	$\dot{\theta}$	GE	GD	GH	GC	ITAE	Over Shoot (%)	Rising Time (sec)
2-1	0.22	0.22	2.03	61.55	237.39	3.71	0.42	0.0	0.26
2-2	0.22	0.45	2.12	61.29	242.74	3.05	0.40	0.0	0.24
2-3	0.22	0.78	2.19	60.58	250.43	1.44	0.36	0.0	0.21
2-4	0.45	0.22	1.82	57.95	222.71	4.02	0.92	0.0	0.24
2-5	0.45	0.45	1.87	58.78	228.36	3.55	0.90	0.0	0.21
2-6	0.45	0.78	1.91	61.08	236.45	2.41	0.86	0.0	0.17
2-7	0.78	0.22	1.37	47.82	174.74	4.47	2.43	0.0	0.22
2-8	0.78	0.45	1.35	51.71	181.34	4.28	2.91	0.0	0.22
2-9	0.78	0.78	1.33	62.49	190.80	3.80	3.73	0.0	0.21

Table 7. Estimated parameters for GA-based NFN and PI (ITAE, overshoot, and rising time) of the fuzzy PD controller. The values of θ are set as 0.22, 0.45, and 0.78 (rad) and $\dot{\theta}$=0.22, 0.45, and 0.78 (rad/sec)

Case	θ (rad)	$\dot{\theta}$	GE	GD	GC	ITAE	Over Shoot (%)	Rising Time (sec)
2-1	0.22	0.22	7.44	0.53	1.85	0.15	0.0	0.13
2-2	0.22	0.45	7.00	0.51	2.07	0.17	0.0	0.13
2-3	0.22	0.78	6.69	0.50	2.27	0.22	0.0	0.13
2-4	0.45	0.22	4.91	0.33	2.72	0.55	0.16	0.15
2-5	0.45	0.45	4.47	0.32	2.79	0.61	0.13	0.15
2-6	0.45	0.78	4.15	0.31	2.87	0.73	0.10	0.15
2-7	0.78	0.22	3.60	0.25	3.91	2.27	0.03	0.21
2-8	0.78	0.45	3.15	0.23	3.80	2.62	0.08	0.20
2-9	0.78	0.78	2.83	0.22	3.69	3.30	0.14	0.20

As for the PID controller, the estimated scaling factors using GA-based NFN can be observed when initial angular positions are 0.22, 0.45, or 0.78 (rad), and initial angular velocities are 0.22, 0.45, or 0.78 (rad/sec), respectively (Table 8).

As for the LQG controller, based on Matlab toolbox and as shown in Table 9, control performances can be obtained when the initial angular position is 0.22, 0.45, or 0.78 (rad) and the initial angular velocity is 0.22, 0.45, or 0.78 (rad/sec), respectively.

Fig. 18 shows the (a) pole angle, (b) pole angular velocity, and (c) state space for initial angle θ=0.78 (rad) and initial angular velocity $\dot{\theta}$=0.45 (rad/sec) (Cases 2-8).

Fig. 19 shows the (a) pole angle, (b) pole angular velocity, and (c) state space for initial angle θ=0.78 (rad) and initial angular velocity $\dot{\theta}$=0.45 (rad/sec) for the LQG controller (Cases 2-8).

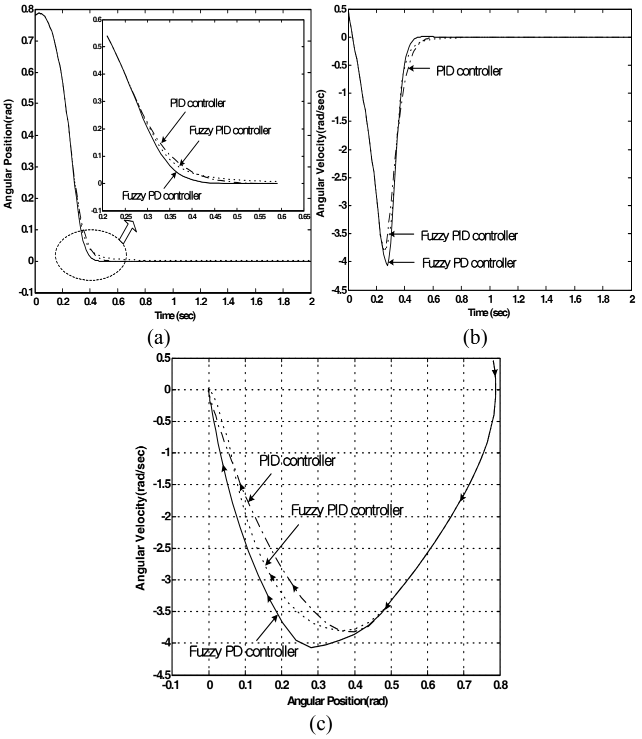
Figs. 14-18 clearly demonstrate that both fuzzy PD and fuzzy PID controllers are superior to the PID controller when performance is assessed from the PI viewpoint. The

Table 8. Estimated parameters, ITAE, overshoot, and rising time of the PID controller, where θ=0.22, 0.45, and 0.78 (rad) and $\dot{\theta}$=0.22, 0.45, and 0.78 (rad/sec)

Case	θ (rad)	$\dot{\theta}$	K	Ti	Td	ITAE	Over Shoot (%)	Rising Time (sec)
2-1	0.22	0.22	168.69	164.72	0.1045	0.25	0.088	0.172
2-2	0.22	0.45	168.68	164.88	0.1047	0.27	0.092	0.172
2-3	0.22	0.78	168.66	164.99	0.1050	0.32	0.100	0.700
2-4	0.45	0.22	168.39	166.31	0.1034	0.74	0.123	0.186
2-5	0.45	0.45	168.42	166.04	0.1033	0.82	0.132	0.185
2-6	0.45	0.78	168.51	165.86	0.1031	0.95	0.147	0.182
2-7	0.78	0.22	167.59	167.40	0.1028	2.67	0.188	0.230
2-8	0.78	0.45	167.71	166.84	0.1025	3.052	0.208	0.228
2-9	0.78	0.78	168.09	166.46	0.1020	3.76	0.242	0.223

Table 9. Estimated parameters, ITAE, overshoot, and rising time of the LQG controller, where θ=0.22, 0.45, 0.78 (rad) and $\dot{\theta}$=0.22, 0.45, 0.78 (rad/sec)

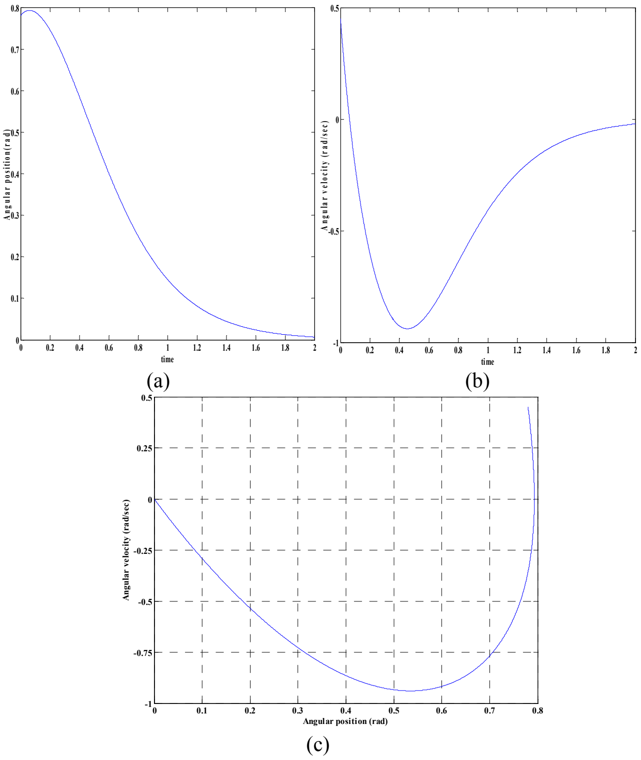
Case	θ (rad)	$\dot{\theta}$	ITAE	Over Shoot (%)	Rising Time (sec)
2-1	0.22	0.22	5.100	0.512	0.860
2-2	0.22	0.45	5.887	0.610	0.844
2-3	0.22	0.78	7.040	0.756	0.822
2-4	0.45	0.22	10.412	0.525	0.886
2-5	0.45	0.45	11.388	0.589	0.879
2-6	0.45	0.78	12.857	0.688	0.866
2-7	0.78	0.22	22.259	0.754	0.956
2-8	0.78	0.45	24.249	0.853	0.951
2-9	0.78	0.78	27.516	1.027	0.943



(a) (b)

(c)

Fig. 18. (a) Pole angle, (b) Pole angular velocity, and (c) State space for initial angle θ=0.78 (rad) and initial angular velocity $\dot{\theta}$=0.45 (rad/sec) (Cases 2-8).



(a) (b)

(c)

Fig. 19. (a) Pole angle, (b) Pole angular velocity, and (c) State space for initial angle, where θ=0.78 (rad) and initial angular velocity $\dot{\theta}$ =0.45 (rad/sec) (Cases 2-8 and LQG).

output performances of the fuzzy controllers (fuzzy PD and fuzzy PID) are also superior to that of the PID controller when the nonlinear dynamic equation of the inverted pendulum is used.

Moreover, the proposed estimation algorithms (GA-based NFNs, such as FS-NFN and FR-NFN) generate the preferred architectures. Fig. 20 shows the nonlinear characteristic of the fuzzy PD controller for cases wherein the values of GE, GD, and GC are all equal to 1.

Fig. 21 shows the input-output relation of the optimized fuzzy PD controller corresponding to the optimization scenarios described earlier (Cases 1-2). The fuzzy PD realizes a highly nonlinear mapping between its inputs and the control output.

Fig. 22 shows the input-output dependency of the fuzzy PID controller for GE, GD, GH, and GC, whose values are equal to 1.

The genetic optimization yields nonlinear characteristics for the fuzzy PID controller, as shown in Fig. 23.

Fig. 24 shows the comparative results of the proposed controllers (fuzzy PID, fuzzy PD, and PID) and LQG in terms of control performances, such as ITAE, overshoot, and rising time.

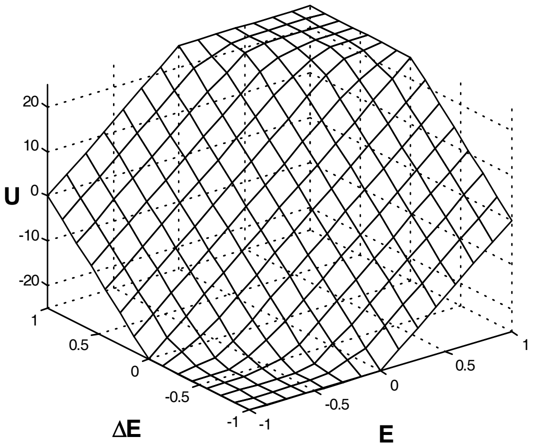


Fig. 20. Input-output relation of the fuzzy PD controller (GE, GD, GC=1).

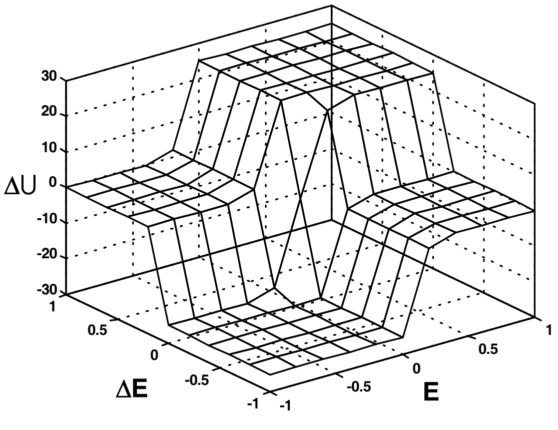
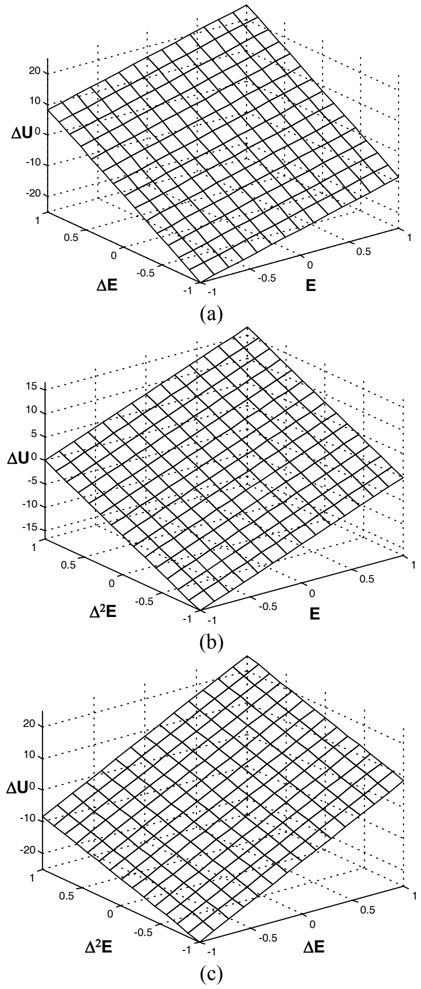


Fig. 21. Input-output relation of the fuzzy PD controller (after genetic optimization).

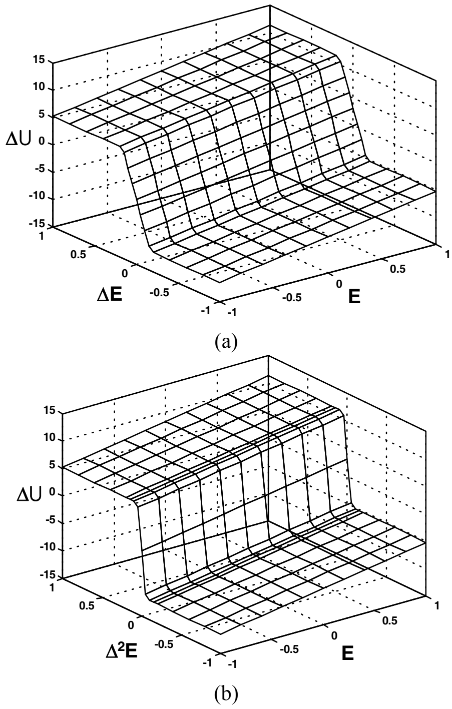


(a)

(b)

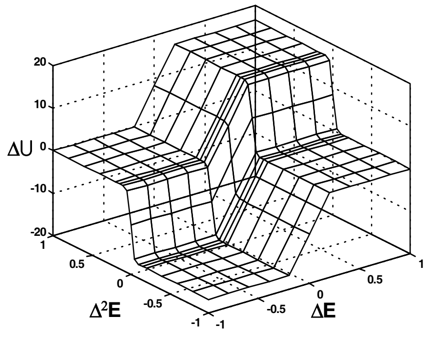
(c)

Fig. 22. The surface of the fuzzy PID controller (GE, GD, GH, GC=1).



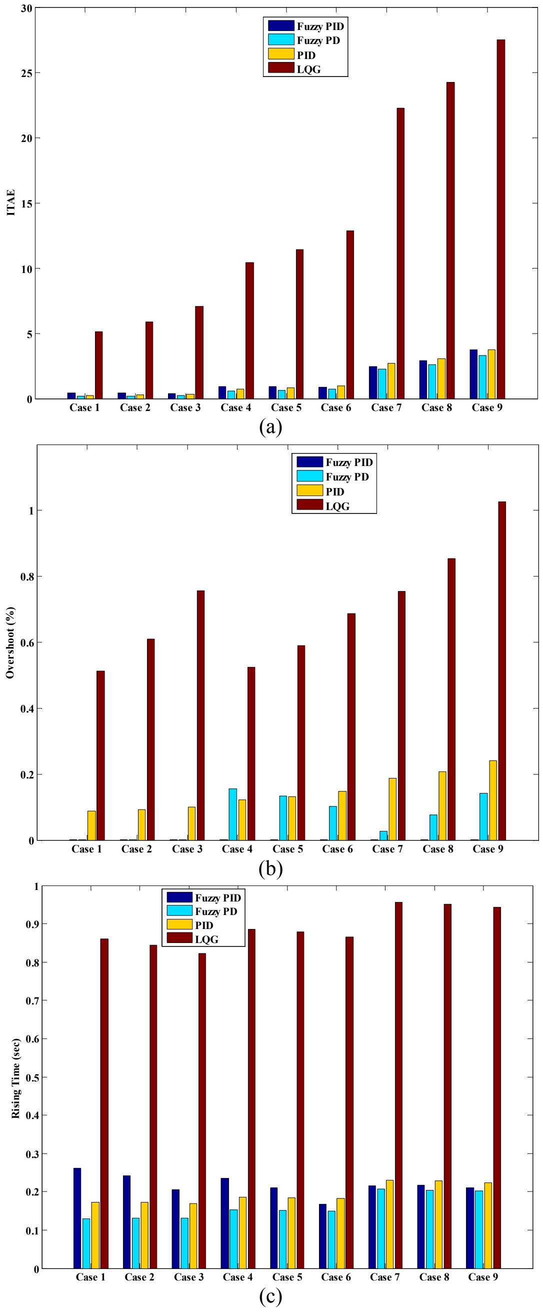
(a)

(b)



(c)

Fig. 23. The surface of the fuzzy PID controller (Cases 1-2).



(a)

(b)

(c)

Fig. 24. Comparative results.

6. Conclusion

In this paper, we propose a two-phase optimization scheme for the fuzzy PID and PD controllers. The parameters under optimization deal with scaling factors of the input and output variables of the controller, which are known to exhibit substantial effect on quality. The first phase of the design of the controller uses genetic computing that aims for global optimization of its scaling factors. These factors are optimized based on the finite collection of initial conditions (i.e., the system under control). In the second phase, we construct nonlinear mapping between the initial conditions of the system and the corresponding values of the scaling factors. We then investigate a number of alternatives by looking at various GA-based NFNs.

Although the study portrays the development of a controller in the experimental control framework for a specific dynamic system (i.e., inverted pendulum), this methodology can still be classified as a general approach and hence can be utilized directly in other control systems. There are a number of modifications worth investigating. For instance, a design of time-varying systems can also benefit from the approach pursued in this study.

Acknowledgements

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2010-D00065) and by the GRRC program of Gyeonggi Province [GRRC SUWON2010-B2, Center for U-city Security & Surveillance Technology].

References

[1] S.K. Oh, "*Fuzzy Model & Control System by C-Programming*," Naeha Press, 2002.

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

[3] B.J. Park, W. 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, pp. 714-723, 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, pp. 677–691 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] T.J. Procyk and E.H.Mamdani, "A linguistic synthesis of fuzzy controller," *Automatica*, Vol. 15, pp.15-30, 1979.

[10] H.X. Li, "A comparative design and tuning for conventional fuzzy control," *IEEE Trans. Syst., Man, Cybern. B*, Vol. 27, pp. 884-889, Oct. 1997.

[11] Bao-Gang Hu, G.K.I. Mann , and R.G. Gosine, "A systematic study of fuzzy PID controllers-function-based evaluation approach," *IEEE Trans., Fuzzy Systems*, Vol. 9, pp. 699-712, Oct. 2001.

[12] Peng Wang and D.P. Kwok , "Analysis and synthesis of an intelligent control system based on fuzzy logic and the PID principle," *Intelligent Systems Engineering*, Vol. 1, pp. 157-171, 1992.

[13] Baogang Hu, G.K.I. Mann, and R.G. Gosine , "New methodology for analytical and optimal design of fuzzy PID controllers," *IEEE Trans. Fuzzy Systems*, Vol. 7, pp. 521-539, Oct. 1999.

[14] S.K. Oh, T. Ahn, H. Hwang, J. Park, and K. Woo, "Design of a Hybrid Fuzzy Controller with the Optimal Auto-tuning Method," *Journal of Control, Automation and Systems Engineering*, Vol. 1, No. 1, pp. 63-70, September, 1995.

[15] B.J. Park, S.K. Oh, and W. Pedrycz, "The Hybrid Multi-layer Inference Architecture and Algorithm of FPNN Based on FNN and PNN," *9th IFSA World Congress*, pp. 1361-1366, 2001.

[16] B.J. Park, W. Pedrycz, and S.K. Oh, "Fuzzy Polynomial Neural Networks: Hybrid Architectures of Fuzzy Modeling," *IEEE Trans. Fuzzy Systems*, Vol. 10, No. 10, pp. 607-621, Oct. 2002.

[17] S.K. Oh and W. Pedrycz, "The Design of Hybrid Fuzzy Controllers Based on Genetic Algorithms and Estimation Techniques," *Kybernetes,* Vol. 31, No. 6, pp. 909-917, 2002.

[18] B.S. Chen, Y.M. Cheng, and C.H. Lee, "A genetic approach to mixed H2/H1 Optimal PID control," *IEEE Control Systems*, Vol. 15, No. 5, pp. 51-60, 1995.

[19] Z.L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Transactions on Energy Conversion*, Vol. 19, No. 2, pp. 384-391, 2004.

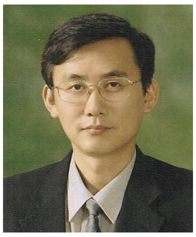
[20] S.P. Ghoshal, "Optimizations of PID gains by particle swarm optimizations in fuzzy based automatic generation control," *Electric Power Systems Research*, Vol. 72, pp. 203-212, 2004.

[21] V. Mukherjee and S.P. Ghoshal, "Intelligent particle swarm optimized fuzzy PID controller for AVR system," *Electric Power Systems Research*, Vol. 77, No. 12, pp. 1689-1698, 2007.

[22] J.S. Wang, Y. Zhang, and W. Wang, "Optimal design of PI/PD controller for non-minimum phase system.," *Transactions of the Institute of Measurement and Control*, Vol. 28, No. 1, pp. 27-35, 2006.

[23] Z. Bingul, "A new PID tuning technique using differential evolution for unstable and integrating processes with time delay," ICONIP, Proceedings Lecture Notes in Computer Science, Vol. 3316, pp. 254-260, 2004.

[24] M. Willjuice Iruthayarajan and S. Baskar, "Evolutionary algorithms based design of multivariable PID controller," Expert Systems with Applications, Vol. 36, pp. 9159-9167, 2009.



Sung-Kwun Oh received his B.Sc., M.Sc., and Ph.D. degrees in Electrical Engineering from Yonsei University, Seoul, Korea, in 1981, 1983, and 1993, respectively. During 1983–1989, he was a Senior Researcher of R&D Lab of Lucky-Goldstar Industrial Systems Co., Ltd. From 1996 to 1997, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada. He is currently a Professor with the Department of Electrical Engineering, University of Suwon, Suwon, South Korea. His research interests include fuzzy system, fuzzy-neural networks, automation systems, advanced computational intelligence, and intelligent control. He currently serves as an Associate Editor of the *KIEE Transactions on Systems and Control*, *International Journal of Fuzzy Logic and Intelligent Systems* of the KFIS, and *International Journal of Control, Automation, and Systems* of the ICASE, South Korea.

Seok-Beom Roh received his B.S., M.S., and PhD. degrees in Control and Instrumentation Engineering from Wonkwang University, Korea in 1994, 1996, and 2006, respectively. He is currently a MOEHRD-KRF research professor at Wonkwang University. His research interests include fuzzy set, neural networks, genetic algorithms, computational intelligence, and statistical learning.