Design of Fuzzy PID Controller Using GAs and Estimation Algorithm

Seok-Beom Roh and Sung-Kwun Oh
School of Electrical and Electronic Engineering, Wonkwang University

Abstract - In this paper, a new approach to estimate scaling factors of fuzzy controllers such as the fuzzy PID controller and the fuzzy PD controller is presented. The performance of the fuzzy controller is sensitive to the variety of scaling factors[1]. The design procedure dwells on the use of evolutionary computing(a genetic algorithm) and estimation algorithm for dynamic systems (the inverted pendulum). 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 Neuro-Fuzzy model, and regression polynomial [7]. This method can be applied to the nonlinear system as the inverted pendulum. Numerical studies are presented and a detailed comparative analysis is also included.

1. 서 론

The ongoing challenge for advanced system control ha s 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. 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 vary 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 three types of estimation algorithms such as HCM (Hard C-Means) clustering method, Neuro-fuzzy model, and regression polynomial, 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].

2.1 Fuzzy PID Controller

The block diagram of fuzzy PID controller is shown in Figure 1. Referring to Figure 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 Δ^2e 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.

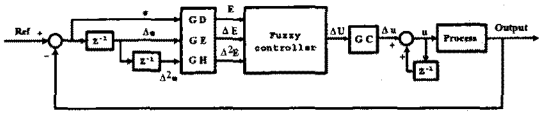

Figure 1. An overall architecture of the fuzzy PID controller

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 Δ^2E is A_{3j} then U_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)[11].

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

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

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

The collection of the rules is shown in Table 1.

Table 1 Fuzzy control rules
(a) In case of 2-fuzzy variables

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

(b) In case of 3-fuzzy variables

Δ^2E = N

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

Δ^2E = Z

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

Δ^2E = P

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

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

2.2 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 controllers 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. 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].

The design procedure 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 Figure 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 (3).

Figure 2 illustrates three types of estimation modes of the scaling factor being used in setting the initial individuals of GAs describing the fuzzy controller.

$$E(kT) = error(kT) \times GE \qquad (3.a)$$

$$\Delta E(kT) = [error(kT) - error(k-1)T] \times GD \qquad (3.b)$$

$$\Delta^2 E(kT) = [error(kT) - 2error(k-1)T + error(k-2)T] \times GH \qquad (3.c)$$

$$U(kT) = U(k-1)T + \Delta U(kT) \times GC \qquad (3.d)$$

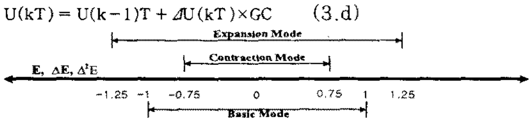

Figure 2. Three types of estimation modes for the scaling factors: basic, expansion, and contraction

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

2.3 The Estimation Algorithm

Algorithm 1-1: Neuro-fuzzy model

Let us consider an extension of the network with the fuzzy partition realized by fuzzy relations. Figure 3 illustrates an architecture of such FNN for two-input and one-output, where each input assumes three membership functions. The node indicated Π denotes a Cartesian product, whose output is the product of all the incoming signals. As before, N denotes the normalization of the membership grades.

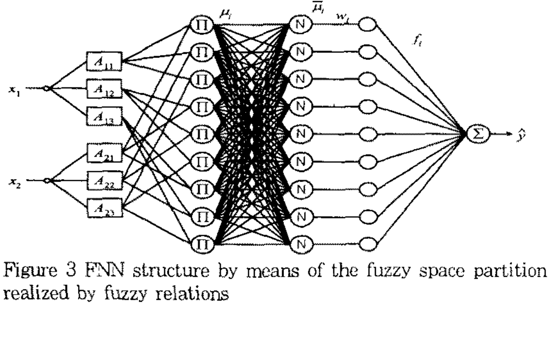

Figure 3 FNN structure by means of the fuzzy space partition realized by fuzzy relations

In the language of the rule-based systems, the structure is equivalent to the following collection of rules.

$$R^1: \text{If } x_1\ A_{11} \text{ and } \cdots x_k \text{ is } A_{1k} \text{ then } y_1 = w_1$$
$$\vdots$$
$$R^j: \text{If } x_j\ A_{j1} \text{ and } \cdots x_k \text{ is } A_{jk} \text{ then } y_j = w_j \qquad (4)$$
$$\vdots$$
$$R^n: \text{If } x_n\ A_{n1} \text{ and } \cdots x_k \text{ is } A_{nk} \text{ then } y_n = w_n$$

The fuzzy rules in equation (4) constitute overall networks of modified FNN such as shown in Figure 3. The output f_i of each node generates a final output $\hat{y}$ fo the form

$$\hat{y} = \sum_{i=1}^{n} f_i = \sum_{i=1}^{n} \bar{\mu}_i \cdot w_i = \sum_{i=1}^{n} \bar{\mu}_i \cdot w_i / \sum_{i=1}^{n} \mu_i \qquad (5)$$

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

$$E = \sum_{p=1}^{N} (y_p - \hat{y}_p)^2 \qquad (6)$$

where $\hat{y}_p$ is the *p-th* target output data, y_p stands for the *p-th* actual output of the model for this specific data points .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.

$$W(new) = w(old) + \Delta w \qquad (7)$$

where the update formula follows the gradient descent method

$$\Delta w_i = \eta \cdot \left(-\frac{\partial E_p}{\partial w_i}\right) = -\eta \cdot \frac{\partial E_p}{\partial \hat{y}_p} \cdot \frac{\partial \hat{y}_p}{\partial f_i} \cdot \frac{\partial f_i}{\partial w_i} = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \bar{\mu}_i \qquad (8)$$

with η being a positive learning rate. Quite commonly to accelerate convergence, a momentum term is being added to the learning expression. Combining (8) and a momentum term, the complete update formula combining the already discussed components is

$$\Delta w_i = 2 \cdot \eta \cdot (y_p - \hat{y}_p) \cdot \bar{\mu}_i + \alpha(w_i(t) - w_i(t-1)) \qquad (9)$$

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

Algorithm 1-2: Genetic Algorithms + Neuro-fuzzy model

In this algorithm, to optimize the learning rate, momentum term and fuzzy membership function of the above NFN we use the genetic algorithm. We use 100 generations, 60 populations, 10 bits per string, crossover rate equal to 0.6, and mutation probability equal to 0.35.

Algorithm 2: Polynomial model

To build a mathematical model we use n-order polynomial and LMS (Least Means Square) method. For this algorithm, we use such type of polynomial as (10), and estimate coefficients of the polynomial.

$$\hat{y}(i) = C_0 + C_1\theta(i) + C_2\theta(i)^2 + \cdots + C_n\theta(i)^n \qquad (10)$$

where $\hat{y}(i)$ is model output, $\theta(i)$ is input variable and C_0 , C_1 , ⋯, C_n are coefficients. The problem is to determine the coefficients in a such a way that the outputs computed from the model in (10) agree as closely as possible with the measured variables *y(i)* in the sense of least squares. That is, the coefficient C should be chosen to minimize the least-squares error function

$$V(\theta, t) = \frac{1}{2}\sum_{i=1}^{t}(y(i) - \hat{y}(i))^2 = \frac{1}{2}\sum_{i=1}^{t}(y(i) - \Theta(i)^T C)^2 \qquad (11)$$

where $\Theta(i)^T = [1\ \theta(i)\ \theta(i)^2 \cdots \theta(i)^n]$, $C = [C_0\ C_1 \cdots C_n]^T$

The f unction of (11 is minimal for coefficients $\hat{C}$ such that

$$\Theta^T \Theta \hat{C} = \Theta^T Y \qquad (12)$$

If the matrix $\Theta^T\Theta$ is nonsingular, the minimum is unique and given as the following

$$\hat{C} = (\Theta^T\Theta)^{-1}\Theta^T Y \qquad (13)$$

Figure 4 depicts the detailed flowchart of the complete tuning and estimating process.

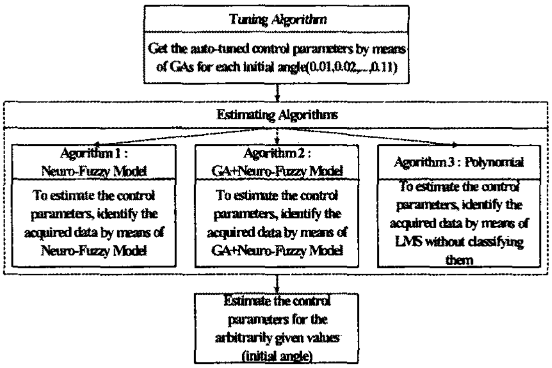

Figure 4. Overall tuning and estimating process

2.3 Simulation Study

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

$$\ddot{\theta}=\frac{g\sin\theta+\cos\theta\left(\frac{-F-ml\dot{\theta}^{2}\sin\theta}{m_c+m}\right)}{l\left(\frac{4}{3}-\frac{m\cos^{2}\theta}{m_c+m}\right)} \qquad (14)$$

Where g (acceleration due to gravity) is $9.8m/s2$, mc (mass of cart) is $1.0kg$, m (mass of pole) is $0.5m$, and F is the applied force in newtons. Figure 5 shows auto-tuned scaling factors according to the change of initial angle and angular velocity of the inverted pendulum.

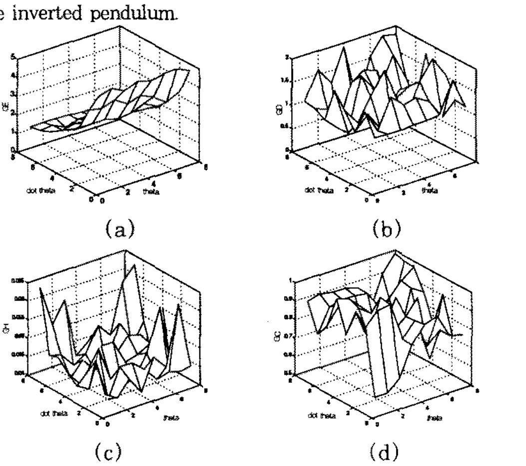

Figure 5. Auto-tuned scaling factors according to the change of initial angles (a) GE, (b) GD, (c) GH and (d) GC

Table 2 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 0.78(rad) and the initial angular velocity is 0.78(rad/sec) respectively.

Table 2 the estimated scaling factors of fuzzy PID controller and describes performance index (ITAE, Overshoot(%)) of the fuzzy PID controller

Estimation algorithm	ITAE	Overshoot
1. Neuro Fuzzy Model	5.908313	4.154205
2. GA+Neuro-Fuzzy Model	5.842924	3.722670
3. Polynomial	6.811258	7.621215

Figure 6 demonstrates (a)pole angle (b)pole angular velocity (c)state space of fuzzy PID controller for initial angle=0.78(rad) and initial angular velocity=0.78(rad/sec) for each estimation algorithm respectively.

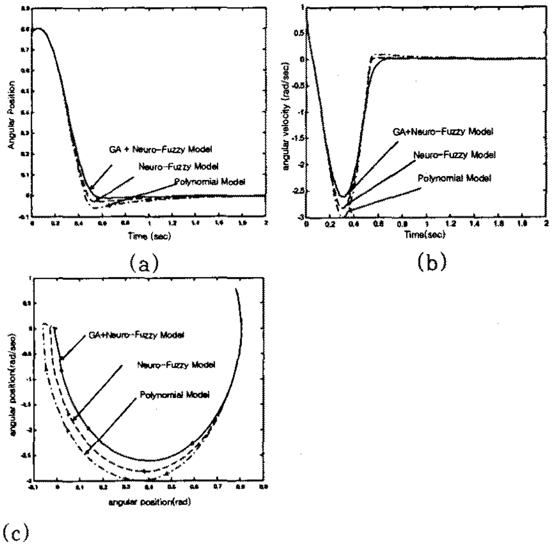

Figure 6 (a)pole angle (b)pole angular velocity (c)state space of fuzzy PID controller for initial angle=0.78(rad) and initial angular velocity=0.78(rad/sec) for each estimation algorithm respectively.

3. 결 론

In this paper, we propose the Fuzzy PID 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 GD by using two types of estimation algorithms so that we may improve the control performance of the fuzzy PID 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 three types of estimation algorithms, we show that whenever the initial values of the inverted pendulum system are changed, the fuzzy PID controller with control parameters such as GE, GD, GH and GC estimated by estimation algorithm controls 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.

1. Adopt FCM method to estimate the control parameters.
2. Use MIMO (Multi Input Multi Output) Neuro-Fuzzy Model to estimate the control parameters.

감사의 글

본 연구는 한국과학재단 목적기초연구 (과제번호 R02-2000-00284) 지원으로 수행되었음.

[참 고 문 헌]

[1] S.K. Oh, *"Fuzzy model, control theory and programming"* KIDARI press, Korea, 1999.
[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] S.-K. Oh, W. Pedrycz and D.-K. Lee, "The Genetic Design of Hybrid Fuzzy Controllers", *Fuzzy Sets and Syst.* (Accepted)
[5] J.R. Jang, Self-Learning Fuzzy Controllers Based on Temporal Back Propagation , *IEEE Trans. On Neural Networks*, Vol. 3, NO. 5, September, 1992
[6] 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
[7] T. Yamakawa, A New Effective Learning Algorithm for a Neo Fuzzy Neuron Model, *5th IFSA World Conference*, pp. 1017-1020, 1993.
[8] K.J. Astrom and B. Wittenmark, *Adaptive Control* , Addison-Wealtley, 1995.
[9] J.G. Ziegler and N.B. Nichols, "Optimum settings for automatic controllers" *Trans. ASME*, 65, pp.433-444, 1942.
[10] J.Malers and Y.S.Sherif, "Application of fuzzy set theory" *IEEE Trans. on System, Man and Cybernetics.*, Vol. Smc-15, No.1, 1985.
[11] T.j.Procyk and E.H.Mamdani, "A linguistic synthesis of fuzzy controller" *Automatica*, Vol.15, pp.15-30, 1979.