A Genetic-Based Design of Auto-Tuning Fuzzy PID Controllers

Chia-Ju Wu, Chia-Nan Ko, Yu-Yi Fu, and Chao-Hsien Tseng

Abstract

This paper presents genetic algorithms (GAs) to perform the optimal design of an auto-tuning fuzzy proportional-integral-derivative (PID) controller and to determine the minimal number of fuzzy rules simultaneously. Different from PID controllers with fixed gains, the fuzzy PID controller is expressed in terms of fuzzy rules, in which the input variables are the error signals and their derivatives, while the output variables are the PID gains. Based on the proposed GAs, the centers and the widths of the Gaussian membership functions, the fuzzy control rules corresponding to every possible combination of input linguistic variables, and the PID gains are chosen as parameters to be determined. When defining the fitness function of the GA, the concept of multi-objective optimization is used such that the fitness function can be defined in a systematic way. To show the effectiveness and validity of the designed fuzzy PID controller, a typical benchmark, a multivariable seesaw system, is used for illustration.

Keywords: Genetic algorithms, fuzzy PID controllers, multi-objective optimization, multivariable systems.

1. Introduction

Conventional PID controllers are widely applied in industry process control for about half a century due to their simplicity in structure and convenience of implementation [1, 2]. To complete such a controller, the proportional gains, the integral gains, and the derivative gains must be determined. Among the existing gain tuning techniques, the method in [3] is probably the most well-known and popular one. However, a conventional PID controller may have poor control performance for nonlinear and/or complex systems that have no precise mathematical models. To overcome these difficulties, various types of modified traditional PID controllers such as auto-tuning and adaptive PID controllers were developed. How to tune the PID controller based on mathematical models is proposed, but complex mathematical computation is generally required in tuning procedures [4, 5]. For the researches [6-8], since the PID gains are fixed, the main disadvantage is that they usually lack in flexibility and capability.

Many researchers attempted to combine conventional PID controllers with fuzzy logic since fuzzy controllers provide reasonable and effective alternatives for conventional controllers. Despite the significant improvement of these fuzzy PID controllers over their classical counterparts, it should be noted that they still have disadvantages. For example, the locations of the peaks of the membership functions are fixed and not adjustable, and the fuzzy control rules are handed-designed rules [9, 10]. Studies on parameter tuning fuzzy PID controllers for nonlinear multivariable systems have been presented [11-13]. In these methods, a multivariable system is decomposed into several subsystems in order to reduce the number of fuzzy rules. The disadvantage is that they neglect the inter-dependency between the variables such that the designed controller cannot work well for some cases.

In this paper, a GA-based approach is presented to determine the membership functions, the fuzzy control rules, and the PID gains in a fuzzy PID controller. Since these components are interdependent, a simultaneous design method is proposed. In the proposed GA-tuning method, based on the concept of multi-objective optimization [14-16], a fitness function is defined in a systematic way such that the centers and the widths of the Gaussian membership functions, the number of fuzzy control rules, and the PID gains can all be chosen as parameters to be determined. In the encoding process, the combination of binary coding and real coding techniques are used to transform a feasible solution into a chromosome. In this manner, the proposed method is fully capable of creating a complete fuzzy PID controller and eliminates the need for human expertise information in the design process. To show the flexibility and capability of the proposed method, a multivariable seesaw problem is used as an example for illustration. From the simulated results, one can find that the designed fuzzy PID controller is more versatile than a conventional one.

Corresponding Author: Chia-Nan Ko is with the Department of Automation Engineering, Nan-Kai University of Technology, 568 Chung Cheng Rd., Tsaotun, Nantou, Taiwan, 54243.
E-mail: t105@nkut.edu.tw
Manuscript received 26 Nov. 2008; revised 15 Dec. 2008; accepted 25 Mar. 2009.

2. Fuzzy PID Controllers

In a classical PID control system, the time domain form of a PID controller is usually expressed as

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \dot{e}(t), \quad (1)$$

where $u(t)$ is the control signal, $e(t)$ is the error signal, and K_P, K_I, and K_D denote the proportional gain, the integral gain, and the derivative gain, respectively.

The block diagram of the proposed fuzzy PID controller is depicted in Figure 1. As shown in the block diagram, the input variables of the fuzzy rules are the error signals and their derivatives, while the output variables are the PID gains. Since a multi-input multi-output (MIMO) system can always be decomposed into a set of multi-input single-output (MISO) system, only the case of MISO systems is considered. Moreover, since the experimental example that will be discussed later has two error signals and one output, the fuzzy PID control rules are expressed as

If e_1 is X_1^i and $\dot{e}_1$ is X_2^j and e_2 is X_3^k and $\dot{e}_2$ is X_4^l, then $K_{P1} = Y_{P1}^{ijkl}, K_{I1} = Y_{I1}^{ijkl}, \cdots, K_{D2} = Y_{D2}^{ijkl}$

for $1 \le i \le n_1$, $1 \le j \le n_2$, $1 \le k \le n_3$, $1 \le l \le n_4$, (2)

where e_1, $\dot{e}_1$, e_2, and $\dot{e}_2$ are the error signals and their derivatives, X_1^i, X_2^j, X_3^k, and X_4^l are the membership functions of e_1, $\dot{e}_1$, e_2, and $\dot{e}_2$, K_{P1}, $K_{I1}, \cdots, K_{D2}$ are the PID gains, Y_{P1}^{ijkl}, $Y_{I1}^{ijkl}, \cdots$, Y_{D2}^{ijkl} are real numbers to be determined, n_1, n_2, n_3, and n_4 denote the numbers of input membership functions, respectively.

The membership functions of an FLC are usually parametric functions such as triangular functions, trapezoidal functions, Gaussian functions, and singletons. Though the proposed method is equally applicable to all these kinds of membership functions, the Gaussian ones are used as the antecedent fuzzy sets in this paper. This means that input membership functions are represented as

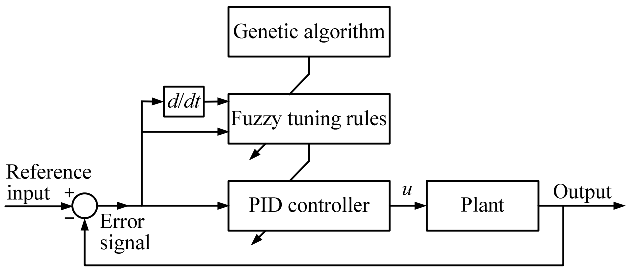


Figure 1. Block diagram for the fuzzy PID controller, in which a genetic algorithm is used to search for the optimal parameters in the fuzzy tuning rules.

$$X_k^{m_k}(x_k) = \begin{cases} \exp\left[-\left(\dfrac{x_k - \rho_k^{m_k}}{\sigma_{kl}^{m_k}}\right)^2\right] & \text{if } x_k \le \rho_k^{m_k} \\ \exp\left[-\left(\dfrac{x_k - \rho_k^{m_k}}{\sigma_{kr}^{m_k}}\right)^2\right] & \text{if } x_k > \rho_k^{m_k} \end{cases}$$

$$\text{for } k = 1, 2, \cdots, 4,$$

$$1 \le m_1 \le n_1, 1 \le m_2 \le n_2, 1 \le m_3 \le n_3, 1 \le m_4 \le n_4, \quad (3)$$

where x_k represents the input linguistic variables, $\rho_k^{m_k}$, $\sigma_{kl}^{m_k}$, and $\sigma_{kr}^{m_k}$ denote the values of the centers, the left widths, and the right widths of the input membership functions, respectively. For the output membership functions, singleton sets are adopted. In the defuzzification process, the center of gravity method [17] is used to determine the PID gains $K_{P1}, K_{I1}, \cdots, K_{D2}$. Taking the values of fuzzy PID gains into (1), one can determine the control signal $u(t)$ as

$$u(t) = \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl} Y_{P1}^{ijkl}}{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl}} e_1(t) + \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl} Y_{I1}^{ijkl}}{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl}} \int e_1(t)dt$$
$$+ \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl} Y_{D1}^{ijkl}}{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl}} \dot{e}_1(t) + \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl} Y_{P2}^{ijkl}}{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl}} e_2(t)$$
$$+ \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl} Y_{I2}^{ijkl}}{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl}} \int e_2(t)dt + \frac{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl} Y_{D2}^{ijkl}}{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\sum_{l=1}^{n_4} \omega_{ijkl}} \dot{e}_2(t),$$

where $\omega_{ijkl} = X_1^i(e_1) \cdot X_2^j(\dot{e}_1) \cdot X_3^k(e_2) \cdot X_4^l(\dot{e}_2)$. (4)

From the above description, one can find that the gains of a fuzzy PID controller are adaptive. However, it is very difficult, if not impossible, to determine the parameters n_1, n_2, n_3, n_4, Y_{P1}^{ijkl}, Y_{I1}^{ijkl}, Y_{D1}^{ijkl}, Y_{P2}^{ijkl}, Y_{I2}^{ijkl}, Y_{D2}^{ijkl}, $\rho_k^{m_k}$, $\sigma_{kl}^{m_k}$, and $\sigma_{kr}^{m_k}$, simultaneously. Therefore, a GA-based searching approach will be proposed since GAs provide a means to search poorly understood, highly complex space.

3. Genetic Algorithms

A. *Chromosome Representations*

One of the key issues of the proposed GA-based method is the way to encode the parameters n_1, n_2, n_3, n_4, Y_{P1}^{ijkl}, Y_{I1}^{ijkl}, Y_{D1}^{ijkl}, Y_{P2}^{ijkl}, Y_{I2}^{ijkl}, Y_{D2}^{ijkl}, $\rho_k^{m_k}$, $\sigma_{kl}^{m_k}$, and $\sigma_{kr}^{m_k}$. In the past, binary coding techniques have been used extensively because they are simple in representation and easy to be implemented. Recently, many practical applications in the industrial engineering world show that real coding techniques provide better performance than binary ones for real parameters [18, 19]. In this paper, since the parameters to be coded include binary numbers and real numbers, a mixed coding method is proposed to try to take the advantages of both kinds of coding techniques. In the proposed coding techniques, n_1, n_2, n_3, and n_4 are encoded as binary numbers and Y_{P1}^{ijkl}, Y_{I1}^{ijkl}, Y_{D1}^{ijkl}, Y_{P2}^{ijkl}, Y_{I2}^{ijkl}, Y_{D2}^{ijkl}, $\rho_k^{m_k}$, $\sigma_{kl}^{m_k}$, and $\sigma_{kr}^{m_k}$ are encoded as real numbers.

Once the mixed coding techniques are used, a chromosome can be represented as

$$\boldsymbol{p} = [p_1, p_2, \cdots, p_{4n_1+4n_2+4n_3+4n_4+6n_1n_2n_3n_4}] = [\boldsymbol{p}_{binary}\ \boldsymbol{p}_{real}], \quad (5)$$

where

$$\boldsymbol{p}_{binary} = [p_1, p_2, \cdots, p_{n_1+n_2+n_3+n_4}], \quad (6)$$

$$\boldsymbol{p}_{real} = [p_{n_1+n_2+n_3+n_4+1}, \cdots, p_{4n_1+4n_2+4n_3+4n_4+6n_1n_2n_3n_4}]. \quad (7)$$

As described above, the values of p_1 to $p_{n_1+n_2+n_3+n_4}$ are used to indicate which ones of the membership functions are activated.

As for the real genes, the values of $p_{n_1+n_2+n_3+n_4+1}$ to $p_{2n_1+2n_2+2n_3+2n_4}$, $p_{2n_1+2n_2+2n_3+2n_4+1}$ to $p_{3n_1+3n_2+3n_3+3n_4}$, and $p_{3n_1+3n_2+3n_3+3n_4+1}$ to $p_{4n_1+4n_2+4n_3+4n_4}$ are used to represent the values of $\rho_k^{m_k}$, $\sigma_{kl}^{m_k}$, and $\sigma_{kr}^{m_k}$, respectively. Furthermore, the values $p_{4n_1+4n_2+4n_3+4n_4+1}$ to $p_{4n_1+4n_2+4n_3+4n_4+6n_1n_2n_3n_4}$ are used to represent the values of Y_{P1}^{ijkl}, Y_{I1}^{ijkl}, Y_{D1}^{ijkl}, Y_{P2}^{ijkl}, Y_{I2}^{ijkl}, Y_{D2}^{ijkl}, for $1 \le i \le n_1$, $1 \le j \le n_2$, $1 \le k \le n_3$, $1 \le l \le n_4$.

B. *Fitness*

In GAs, fitness is the performance index to evaluate the suitability of each chromosome. By the principle of survival of the fittest, a chromosome with higher fitness value has a higher probability of contributing one or more offspring in the next generation. Its value is calculated by a fitness function, which is defined according to the performance requirements of each specific problem. The fitness function is chosen so that its maximum value is the desired value of the quantity to be optimized.

Based on the concept of multi-objective optimization, a systematic way to define the fitness function is proposed in this paper. Assuming that one is interested in M performance criteria, which are denoted by $f_1(\boldsymbol{x})$, $f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})$, respectively, then the fitness function can be defined as

$$fitness = \underset{x\in\Omega}{\text{Maximize}}\ \ f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x}), \quad (8)$$

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ is a variable vector in a real and n-dimensional space, Ω is the feasible solution space, and $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})$ are the M objective functions to be maximized simultaneously.

From the above illustration, one can realize that the fitness function can be defined systematically as shown in (8). However, how to find an optimal solution for this multi-objective minimization problem is not an easy task and several algorithms have been proposed [14-16]. Among these methods, one feasible way is to search from various search directions to find the optimal solution [14]. In this method, the first step is to rewrite the fitness function (8) as

$$fitness = w_1 h_1(\boldsymbol{x}) + w_2 h_2(\boldsymbol{x}) + \cdots + w_M h_M(\boldsymbol{x}), \quad (9)$$

where

$$h_i(\boldsymbol{x}) = \frac{f_i(\boldsymbol{x})}{\max\limits_{y\in\Omega}\{|f_i(\boldsymbol{y})|\}} \quad \text{for } i = 1, 2, \cdots, M, \quad (10)$$

$(w_1, w_2, \cdots, w_M)$ is the weight vector and

$$w_i = \frac{r_i}{r_1 + r_2 + \cdots + r_M} \quad \text{for } i = 1, 2, \cdots, M, \quad (11)$$

in which $r_1, r_2, \cdots, r_M$ are nonnegative random real numbers.

Then N sets of weight vector, which are represented as $(w_{11}, w_{21}, \cdots, w_{M1}), \cdots, (w_{1N}, w_{2N}, \cdots, w_{MN})$, respectively, are generated randomly as described in (11). Corresponding to these weight vectors, M different fitness functions are defined as shown in (9) and N optimal solutions, which are represented as $\boldsymbol{x}_{opt,1}, \boldsymbol{x}_{opt,2}, \cdots, \boldsymbol{x}_{opt,N}$, respectively, are obtained by applying the GA.

How to fuse the solutions $\boldsymbol{x}_{opt,1}, \boldsymbol{x}_{opt,2}, \cdots, \boldsymbol{x}_{opt,N}$ to obtain the optimal solution of the multi-objective problem in (8) was not discussed in [14]. Therefore, a novel method is developed and the basic idea is to determine the average weight vector of the N sets of weight vector by

$$\overline{w}_i = \frac{\sum_{j=1}^{N} w_{ij} \cdot h_i(\boldsymbol{x}_{opt,j})}{\sum_{j=1}^{N} h_i(\boldsymbol{x}_{opt,j})} \quad \text{for } i = 1, 2, \cdots, M. \quad (12)$$

With this new set of weight vector, the corresponding fitness function is defined as

$$fitness = \overline{w}_1 h_1(\boldsymbol{x}) + \overline{w}_2 h_2(\boldsymbol{x}) + \cdots + \overline{w}_M h_M(\boldsymbol{x}), \quad (13)$$

in which the *fitness* acts as the performance criteria in the GA searching procedure to find the optimal solution.

C. *Crossover and Mutation*

Crossover is an effective way of exchanging and recombining genes from the higher fitness individuals. In the proposed GA-based method, a chromosome is represented as $\boldsymbol{p}=[\boldsymbol{p}_{binary}\ \boldsymbol{p}_{real}]$. Consequently, different kinds of crossover operators are applied to $\boldsymbol{p}_{binary}$ and $\boldsymbol{p}_{real}$, respectively. For binary-coded genes $\boldsymbol{p}_{binary}$, one cut-point crossover operator [20] is adopted because of its simplicity. As for real-coded genes $\boldsymbol{p}_{real}$, the convex operator is adopted. The basic concept of convex crossover is stemmed from the convex set theory [21].

Mutation is also a crucial operator of GAs. It alters one or more genes with a mutation probability. In the proposed GA-based method, for the binary-coded chromosomes $\boldsymbol{p}_{binary}$, single-point mutation operator [19] will be employed. For the real-coded chromosome $\boldsymbol{p}_{real}$, the non-uniform mutation is adopted since it has more flexibility [18].

D. *Searching Procedure*

The procedure of the proposed GA-based method is itemized as follows:

Algorithm A:

Step 1: Choose the population size, the maximal generation number, the crossover rate, the mutation rate, and the number of *N* to generate different weight vectors.

Step 2: $j=0$.

Step 3: $j=j+1$.

Step 4: Generate $(w_{1j},\ w_{2j},\cdots,\ w_{Mj})$ as described in (11) and define the corresponding fitness function (9).

Step 5: Produce an initial generation randomly.

Step 6: Evaluate the fitness value for each chromosome of the population.

Step 7: Generate the offspring with the crossover rate and mutation rate in Step 1, in which the ranking mechanism is used for selecting chromosomes.

Step 8: Select the population of the new generation from the parents in the old generation and the offspring in Step 7 according to their corresponding fitness values.

Step 9: Repeat the procedure from Step 6 to Step 8 until the number of generations reaches the maximal generation number. Then determine the corresponding optimal solution $\boldsymbol{x}_{opt,j}$.

Step 10: If $j<N$, then go to Step 3. Otherwise, determine the average weight vector $(\bar{w}_1,\ \bar{w}_2,\cdots,\ \bar{w}_M)$ as described in (12). With this weight vector, define the corresponding fitness function (13) and apply the procedure described in Step 5 through Step 9 to determine the optimal solution.

Step 11: The solution obtained in Step 10 is considered as the optimal solution of the multi-objective problem in (8). End.

4. Application Example

A. *The Seesaw System*

To demonstrate the feasibility of the proposed approach to multivariable systems, the seesaw system shown in Figure 2 is used for illustration [22]. This system is a typical benchmark to evaluate the performance of a designed controller [23]. According to the PID control law, the control voltage is sent via the D/A converter and the power amplifier to the DC motor to make it rotate clockwise or counterclockwise. Moreover, two bumpers are installed at both ends of the beam to prevent the motor from falling off.

In the seesaw system, the input and output state variables are represented by $\tau(t)$, $r(t)$, and $\theta(t)$, which denote the torque of the motor, the position of the motor, and the angle of the beam, respectively. By applying Newton's second and third laws [24], the dynamic equations of the seesaw system can be derived as follows:

$$\begin{aligned}&(m_b+m_r-\frac{J_r}{R^2})\ddot{r}(t)+(m_b+m_r)g\sin\theta(t)\\&-(m_b+m_r)r(t)\dot{\theta}^2(t)+\frac{\tau(t)}{R}=0,\end{aligned} \tag{14}$$

$$\begin{aligned}&[(m_b+m_r)r^2(t)+J_b]\ddot{\theta}(t)+2(m_b+m_r)r(t)\dot{\theta}(t)\dot{r}(t)\\&+(m_b+m_r)gr(t)\cos\theta(t)=0,\end{aligned} \tag{15}$$

where m_b is the mass of the base together with the motor, m_r is the mass of the three rollers, J_r is the polar moment inertia of the three rollers, R is the radius of the roller, g is the gravity acceleration, and J_b is the polar moment inertia of the beam, respectively.

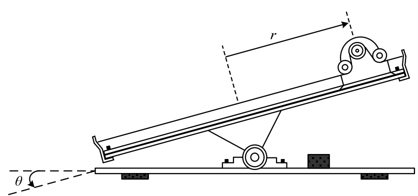


Figure 2. A schematic diagram of the seesaw system, in which r and θ denote the position of the motor and the angle of the beam, respectively.

B. *Stability of the System*

Let $x=(r,\dot{r},\theta,\dot{\theta})^T$ be the state of the system and $y=\theta$ be the output of the system. Then, the seesaw system can be repressed by using the state-space model [23]

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix}=\begin{bmatrix}x_2\\ 0\\ x_4\\ -A(2x_1x_2x_4+gx_1\cos x_3)\end{bmatrix}+\begin{bmatrix}0\\ 1\\ 0\\ 0\end{bmatrix}u_s, \tag{16}$$

$$A=\frac{m_b+m_r}{(m_b+m_r)x_1^2+J_b}. \tag{17}$$

Equation (16) can be expressed as

$$\dot{x}=p(x)+q(x)u_s, \tag{18}$$

where the control u_s denotes the acceleration $\ddot{r}$. The objective of the regulation problem is to determine u_s such that the closed-loop system output y will converge to zero from given initial states [25].

In [25], the exact input-output linearization method uses coordinate transformation to derive a feedback control law to make the closed-loop system in neighborhood of the equilibrium point. Following the usual procedure to examine the seesaw system, one differentiates the output y until the input u_s appears as follows:

$$y=\theta=x_3, \tag{19}$$

$$\dot{y}=x_4, \tag{20}$$

$$\ddot{y}=\dot{x}_4=-A(2x_1x_2x_4+gx_1\cos x_3), \tag{21}$$

$$\begin{aligned}\dddot{y}=\ddot{x}_4&=A^2x_1^2x_2(6x_2x_4+3g\cos x_3)\\ &-A(2x_2^2x_4+gx_2\cos x_3-gx_1x_4\sin x_3)\\ &-2Ax_1x_4u_s.\end{aligned} \tag{22}$$

Equation (22) can be re-expressed as

$$v(x)=b(x)+a(x)u_s, \tag{23}$$

where

$$\begin{aligned}v(x)&=\dddot{y},\\ b(x)&=A^2x_1^2x_2(6x_2x_4+3g\cos x_3)\\ &\quad-A(2x_2^2x_4+gx_2\cos x_3-gx_1x_4\sin x_3),\\ a(x)&=-2Ax_1x_4.\end{aligned}$$

The input u_s can be obtained as

$$u_s=\frac{-b(x)+v(x)}{a(x)}. \tag{24}$$

In (23), $a(x)$ is zero at $x=0$ but is not identically zero in the neighborhood of $x=0$, it is said to have no well defined relative degree at a region. If the region of interest is away from any point such that $x_1x_4=0$, that is $a(x)\neq 0$, a control law of the form (24) to yield a linear input-output system. The value of $v(x)$ can be chosen as to make the error system into an exponentially stable linear system [25]. However, the exact input-output linearization method can only be used in a small neighborhood of the region. Integrate fuzzy dynamic model with linearization method to achieve stabilization in a large region [26, 27].

If fuzzy PID control law is employed, then the input-output relation of the seesaw system is expressed as

$$u_s(t)=K_Pe(t)+K_I\int e(t)dt+K_D\dot{e}(t), \tag{25}$$

where $e(t)=y_d(t)-y(t)$ and $\dot{e}(t)=\dot{y}_d(t)-\dot{y}(t)$. The analogy between a fuzzy controller and a conventional PID controller has been widely studied. In [2], small gain theorem has been adopted to devise the stability conditions. Passivity theorem in [28] has been adopted to analyze the stability of fuzzy PID controllers.

C. *GA-based Fuzzy PID Controllers*

In the seesaw system, the desired value of $r(t)$ and $\theta(t)$ are denoted by r_d and θ_d. If the PID control law is employed, then the input-output relation of the seesaw system is expressed as

$$\begin{aligned}\tau(t)&=K_{P1}e_1(t)+K_{I1}\int e_1(t)dt+K_{D1}\dot{e}_1(t)+K_{P2}e_2(t)\\ &+K_{I2}\int e_2(t)dt+K_{D2}\dot{e}_2(t),\end{aligned} \tag{26}$$

where $e_1(t)=r_d-r(t)$ and $e_2(t)=\theta_d-\theta(t)$ are the error signals, $\dot{e}_1(t)=\dot{r}_d-\dot{r}(t)$ and $\dot{e}_2(t)=\dot{\theta}_d-\dot{\theta}(t)$ are the derivatives of the error signals, K_{P1}, K_{I1}, K_{D1}, K_{P2}, K_{I2}, K_{D2} denote the proportional gains, the integral gains, and the derivative gains that correspond to each error signal, respectively.

In the experiment, the goal is to use the proposed approach to tune the PID gains in (26) such that the DC motor can be driven from one end of the beam to the balance state. When applying Algorithm A, referring to the works [29, 30] the population size, the maximal generation number, the crossover rate, and mutation rate are chosen to be 40, 2000, 0.8, and 0.2, respectively. Moreover, it is assumed that the values of n_1, n_2, n_3, n_4 are all chosen as five, and the singletons of the output linguistic variables are all chosen as real numbers in the range $[-5, 5]$. In this manner, each chromosome has 3830 genes, consisting of 20 binary-coded genes and 3810 real-coded genes.

In designing the fuzzy PID controller, the primary goal is to drive the seesaw system from the given initial state to the desired final state. However, if the number of fuzzy rules is large, then heavy computation burden and huge memory requirement are inevitable. Therefore, the primary goal and the way to reduce the number of fuzzy rules should be taken into account simultaneously in defining the fitness function. This means that two performance criteria are chosen in this paper as follows:

$$f_1 = \frac{1}{\int t[e_1^2(t)+e_2^2(t)]dt}, \quad (27)$$

$$f_2 = \frac{1}{[(1+\sum_{i=1}^{n_1} p_i)\cdot(1+\sum_{j=1}^{n_2} p_j)\cdot(1+\sum_{k=1}^{n_3} p_k)\cdot(1+\sum_{l=1}^{n_4} p_l)]^2}, \quad (28)$$

where p_i, p_j, p_k, and p_l are the binary genes to indicate which ones of the membership functions are activated. Then based on the multi-objective optimization concept described in Section 3.B, the fitness function is defined as

$$fitness = \bar{w}_1 h_1 + \bar{w}_2 h_2, \quad (29)$$

$$\bar{w}_1 = \frac{\sum_{j=1}^{N} w_{1j}\cdot h_1}{\sum_{j=1}^{N} h_1}, \quad \bar{w}_2 = \frac{\sum_{j=1}^{N} w_{2j}\cdot h_2}{\sum_{j=1}^{N} h_2}, \quad (30)$$

where h_1 and h_2 are normalized functions corresponding to f_1 and f_2, respectively.

In the following simulated example, using the definition (24), the fitness value will be used to evaluate the performance of the designed fuzzy PID controller and a higher fitness value will denote a better performance.

D. *Simulated Results*

The values of m_b, m_r, J_r, R, g, and J_b in (14) and (15) can be determined in the laboratory by measurement [22]. Substituting these measured values into (14) and (15), one can obtain

$$\ddot{r}(t) = \frac{-4.753\sin\theta(t) + 0.485 r(t)\dot{\theta}^2(t) - 1.053\times 10^2 \tau(t)}{0.451}, \quad (31)$$

$$\ddot{\theta}(t) = \frac{-0.97 r(t)\dot{\theta}(t)\dot{r}(t) - 4.753 r(t)\cos\theta(t)}{0.485 r^2(t) + 0.161}. \quad (32)$$

The ranges of $r(t)$, $\dot{r}(t)$, $\theta(t)$, and $\dot{\theta}(t)$ are chosen as $[-0.38\text{m}, 0.38\text{m}]$, $[-0.76\text{m/sec}, 0.76\text{m/sec}]$, $[-\pi/12\,\text{rad}, \pi/12\,\text{rad}]$, and $[-\pi/6\,\text{rad/sec}, \pi/6\,\text{rad/sec}]$, respectively. The initial state is $[r,\ \dot{r},\ \theta,\ \dot{\theta}] = [0.38\text{ m},\ 0,\ \pi/12\text{ rad},\ 0]$ and the desired final state is $[r,\ \dot{r},\ \theta,\ \dot{\theta}] = [0,\ 0,\ 0,\ 0]$. Meanwhile, the input torque $\tau(t)$ of the motor is also assumed to be within the range $[-0.2\text{N}\cdot\text{m}, 0.2\text{N}\cdot\text{m}]$ due to the physical limitations of power amplifiers.

Applying Algorithm A with $N = 10$ and the fitness function (29), the plot of the fitness values is obtained as shown in Figure 3. Moreover, the optimal values of $\boldsymbol{p}_{binary}$ and $\boldsymbol{p}_{real}$ can be determined. The former is found to be [01010100011101011010] and it means that only the membership functions X_1^2, X_1^4, X_2^1, X_2^5, X_3^1, X_3^2, X_3^4, X_4^1, X_4^2, and X_4^4 are activated. Meanwhile, this also means that there are 36 ($=2\times2\times3\times3$) fuzzy rules in the fuzzy PID controller. Since the number of fuzzy rules is reduced from 625 ($=5^4$) to 36, the computation burden in implementation of this fuzzy PID controller will also be reduced significantly.

The vector $\boldsymbol{p}_{real}$ contains 3810 real-coded genes such that it is impractical to list the values of all these genes. However, with the optimal $\boldsymbol{p}_{real}$ determined by Algorithm A, the membership functions of e_1, $\dot{e}_1$, e_2, and $\dot{e}_2$ are determined as shown in Figure 4(a) through Figure 2(d), respectively. Meanwhile, the 36 fuzzy rules are listed as shown in Table 1. Corresponding to these fuzzy PID tuning rules, the plots of $r(t)$ and $\theta(t)$ are shown in Figure 5 and Figure 6, respectively. In the same figures, the simulated results with fixed PID gains are also given for comparison, in which the genetic searching algorithm in [19] is adopted to find the optimal values of K_{P1}, K_{I1}, K_{D1}, K_{P2}, K_{I2}, and K_{D2}. By comparing the plots in Figure 5 and Figure 6, one can find easily that the designed fuzzy PID controller has a better performance than a traditional one.

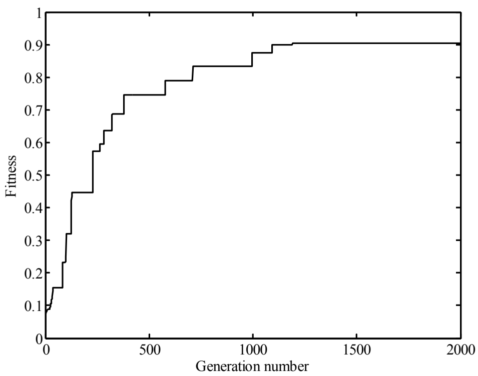


Figure 3. Plot of fitness values in each generation with the fitness function (29).

Table 1. Fuzzy tuning rules for the PID controller in (26), The membership functions X_1^2, X_1^4, X_2^1, X_2^5, X_3^1, X_3^2, X_3^4, X_4^1, X_4^2, X_4^4, are shown in Figure 4(a) through Figure 4(d), respectively.

If e_1 is X_1^2, $\dot{e}_1$ is X_2^1, e_2 is X_3^1, and $\dot{e}_2$ is X_4^1, then K_{P1}= 2.8682, K_{I1}= 0.0287, K_{D1}= 1.2198, K_{P2}= 0.5690, K_{I2}= 0.0087, K_{D2}= -0.2945

If e_1 is X_1^2, $\dot{e}_1$ is X_2^1, e_2 is X_3^1, and $\dot{e}_2$ is X_4^2, then K_{P1}= -1.3757, K_{I1}= -0.0215, K_{D1}= -0.4595, K_{P2}= -1.4672, K_{I2}= -0.0074, K_{D2}= -0.9390

If e_1 is X_1^2, $\dot{e}_1$ is X_2^1, e_2 is X_3^1, and $\dot{e}_2$ is X_4^4, then K_{P1}= -3.7875, K_{I1}= -0.0422, K_{D1}= -0.4358, K_{P2}= 0.0465, K_{I2}= -0.0733, K_{D2}= 0.1890

If e_1 is X_1^2, $\dot{e}_1$ is X_2^1, e_2 is X_3^2, and $\dot{e}_2$ is X_4^1, then K_{P1}= 4.4630, K_{I1}= 0.0749, K_{D1}= 0.9910, K_{P2}= 2.7765, K_{I2}= 0.0583, K_{D2}= 1.1540

If e_1 is X_1^2, $\dot{e}_1$ is X_2^1, e_2 is X_3^2, and $\dot{e}_2$ is X_4^2, then K_{P1}= 2.5952, K_{I1}= 0.0230, K_{D1}= 0.3362, K_{P2}= 0.1646, K_{I2}= 0.0422, K_{D2}= 0.5768

If e_1 is X_1^2, $\dot{e}_1$ is X_2^1, e_2 is X_3^2, and $\dot{e}_2$ is X_4^4, then K_{P1}= 0.3753, K_{I1}= 0.0471, K_{D1}= -0.1506, K_{P2}= -0.9057, K_{I2}= 0.0063, K_{D2}= 0.1585

If e_1 is X_1^2, $\dot{e}_1$ is X_2^1, e_2 is X_3^4, and $\dot{e}_2$ is X_4^1, then K_{P1}= 0.0824, K_{I1}= -0.0652, K_{D1}= -0.5649, K_{P2}= -1.0053, K_{I2}= 0.0195, K_{D2}= -0.7729

If e_1 is X_1^2, $\dot{e}_1$ is X_2^1, e_2 is X_3^4, and $\dot{e}_2$ is X_4^2, then K_{P1}= 2.2057, K_{I1}= 0.0486, K_{D1}= 1.2804, K_{P2}= 2.4163, K_{I2}= 0.0231, K_{D2}= 0.8480

If e_1 is X_1^2, $\dot{e}_1$ is X_2^1, e_2 is X_3^4, and $\dot{e}_2$ is X_4^4, then K_{P1}= -1.8394, K_{I1}= -0.0320, K_{D1}= -1.1105 K_{P2}= -1.2021 K_{I2}= -0.0538, K_{D2}= 0.6695

If e_1 is X_1^2, $\dot{e}_1$ is X_2^5, e_2 is X_3^1, and $\dot{e}_2$ is X_4^1, then K_{P1}= -1.7073, K_{I1}= 0.0139, K_{D1}= -0.1654, K_{P2}= 0.5892, K_{I2}= 0.0577, K_{D2}= -0.6810

If e_1 is X_1^2, $\dot{e}_1$ is X_2^5, e_2 is X_3^1, and $\dot{e}_2$ is X_4^2, then K_{P1}= 0.5786, K_{I1}= 0.0359, K_{D1}= 0.1738, K_{P2}= 1.6417, K_{I2}= 0.0266, K_{D2}= 0.0744

If e_1 is X_1^2, $\dot{e}_1$ is X_2^5, e_2 is X_3^1, and $\dot{e}_2$ is X_4^4, then K_{P1}= -2.5941, K_{I1}= -0.0294, K_{D1}= -1.2703, K_{P2}= -2.4953, K_{I2}= -0.0557, K_{D2}= -0.6018

If e_1 is X_1^2, $\dot{e}_1$ is X_2^5, e_2 is X_3^2, and $\dot{e}_2$ is X_4^1, then K_{P1}= 0.2822, K_{I1}= -0.0001, K_{D1}= -0.1064, K_{P2}= 2.9570, K_{I2}= -0.0132, K_{D2}= 0.3909

If e_1 is X_1^2, $\dot{e}_1$ is X_2^5, e_2 is X_3^2, and $\dot{e}_2$ is X_4^2, then K_{P1}= 0.8558, K_{I1}= 0.0388, K_{D1}= 0.8031, K_{P2}= -0.1191, K_{I2}= 0.0090, K_{D2}= 0.0019

If e_1 is X_1^2, $\dot{e}_1$ is X_2^5, e_2 is X_3^2, and $\dot{e}_2$ is X_4^4, then K_{P1}= -2.2683, K_{I1}= -0.0522, K_{D1}= -1.5002, K_{P2}= -3.3676, K_{I2}= -0.0358, K_{D2}= -0.0197

If e_1 is X_1^2, $\dot{e}_1$ is X_2^5, e_2 is X_3^4, and $\dot{e}_2$ is X_4^1, then K_{P1}= 1.0145, K_{I1}= 0.0532, K_{D1}= 0.2857, K_{P2}= 1.6720, K_{I2}= 0.0363, K_{D2}= 0.7254

If e_1 is X_1^2, $\dot{e}_1$ is X_2^5, e_2 is X_3^4, and $\dot{e}_2$ is X_4^2, then K_{P1}= 1.6103, K_{I1}= -0.0094, K_{D1}= -0.2663, K_{P2}= -0.8959, K_{I2}= 0.0611, K_{D2}= 0.2820

If e_1 is X_1^2, $\dot{e}_1$ is X_2^5, e_2 is X_3^4, and $\dot{e}_2$ is X_4^4, then K_{P1}= 0.7535, K_{I1}= -0.0736, K_{D1}= 0.5630, K_{P2}= -3.1136, K_{I2}= -0.0841, K_{D2}= -1.1673

If e_1 is X_1^4, $\dot{e}_1$ is X_2^1, e_2 is X_3^1, and $\dot{e}_2$ is X_4^1, then K_{P1}= -3.0661, K_{I1}= -0.0242, K_{D1}= -0.5421, K_{P2}= -0.8855, K_{I2}= -0.0603, K_{D2}= -1.1585

If e_1 is X_1^4, $\dot{e}_1$ is X_2^1, e_2 is X_3^1, and $\dot{e}_2$ is X_4^2, then K_{P1}= 0.4679, K_{I1}= 0.0697, K_{D1}= 1.0742, K_{P2}= 2.0409, K_{I2}= 0.0219, K_{D2}= 1.3463

If e_1 is X_1^4, $\dot{e}_1$ is X_2^1, e_2 is X_3^1, and $\dot{e}_2$ is X_4^4, then K_{P1}= -0.7945, K_{I1}= -0.0438, K_{D1}= -1.2045, K_{P2}= -0.0254, K_{I2}= -0.0268, K_{D2}= -1.0095

If e_1 is X_1^4, $\dot{e}_1$ is X_2^1, e_2 is X_3^2, and $\dot{e}_2$ is X_4^1, then K_{P1}= 1.1326, K_{I1}= 0.0236, K_{D1}= 0.7151, K_{P2}= 0.6961, K_{I2}= 0.0614, K_{D2}= 0.6017

If e_1 is X_1^4, $\dot{e}_1$ is X_2^1, e_2 is X_3^2, and $\dot{e}_2$ is X_4^2, then K_{P1}= 4.4419, K_{I1}= 0.0873 K_{D1}= 1.9821, K_{P2}= 4.3846, K_{I2}= 0.0836, K_{D2}= 1.7676

If e_1 is X_1^4, $\dot{e}_1$ is X_2^1, e_2 is X_3^2, and $\dot{e}_2$ is X_4^4, then K_{P1}= -3.9385, K_{I1}= -0.0692, K_{D1}= -0.9888, K_{P2}= -2.3860, K_{I2}= -0.0768, K_{D2}= -1.3306

If e_1 is X_1^4, $\dot{e}_1$ is X_2^1, e_2 is X_3^4, and $\dot{e}_2$ is X_4^1, then K_{P1}= 1.4892, K_{I1}= 0.0621, K_{D1}= -0.0550, K_{P2}= 2.5764, K_{I2}= -0.0232, K_{D2}= -1.1940

If e_1 is X_1^4, $\dot{e}_1$ is X_2^1, e_2 is X_3^4, and $\dot{e}_2$ is X_4^2, then K_{P1}= -0.9930, K_{I1}= -0.0625, K_{D1}= 0.0350, K_{P2}= 0.6840, K_{I2}= -0.0364, K_{D2}= 0.6437

If e_1 is X_1^4, $\dot{e}_1$ is X_2^1, e_2 is X_3^4, and $\dot{e}_2$ is X_4^4, then K_{P1}= 0.4430, K_{I1}= 0.0569, K_{D1}= 1.2228, K_{P2}= 0.8575, K_{I2}= 0.0534, K_{D2}= 0.5220

If e_1 is X_1^4, $\dot{e}_1$ is X_2^5, e_2 is X_3^1, and $\dot{e}_2$ is X_4^1, then K_{P1}= -2.2333, K_{I1}= -0.0644, K_{D1}= -0.9373, K_{P2}= -1.3156, K_{I2}= -0.0527, K_{D2}= -0.6200

If e_1 is X_1^4, $\dot{e}_1$ is X_2^5, e_2 is X_3^1, and $\dot{e}_2$ is X_4^2, then K_{P1}= -1.1252, K_{I1}= -0.0531, K_{D1}= -1.0403, K_{P2}= -1.6370, K_{I2}= -0.0324, K_{D2}= 1.6556

If e_1 is X_1^4, $\dot{e}_1$ is X_2^5, e_2 is X_3^1, and $\dot{e}_2$ is X_4^4, then K_{P1}= -3.9674, K_{I1}= -0.0698, K_{D1}= -1.1934, K_{P2}= -2.0840, K_{I2}= -0.0202, K_{D2}= -1.2586

If e_1 is X_1^4, $\dot{e}_1$ is X_2^5, e_2 is X_3^2, and $\dot{e}_2$ is X_4^1, then K_{P1}= -0.2150, K_{I1}= 0.0379, K_{D1}= 0.9206, K_{P2}= 2.8857, K_{I2}= 0.0380, K_{D2}= 0.3312

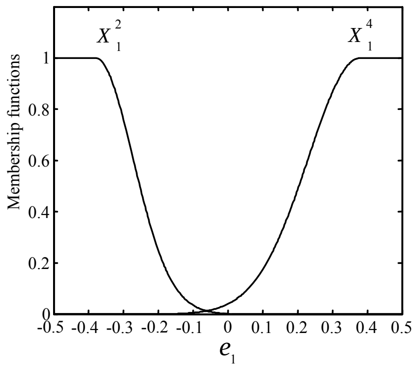
If e_1 is X_1^4, $\dot{e}_1$ is X_2^5, e_2 is X_3^2, and $\dot{e}_2$ is X_4^2, then K_{P1}= 0.9367, K_{I1}= 0.0418, K_{D1}= 0.2171, K_{P2}= -0.5480, K_{I2}= 0.0687, K_{D2}= 0.5382

If e_1 is X_1^4, $\dot{e}_1$ is X_2^5, e_2 is X_3^2, and $\dot{e}_2$ is X_4^4, then K_{P1}= -2.3306, K_{I1}= -0.0287, K_{D1}= -1.1927, K_{P2}= -1.2180, K_{I2}= -0.0198, K_{D2}= -0.3673

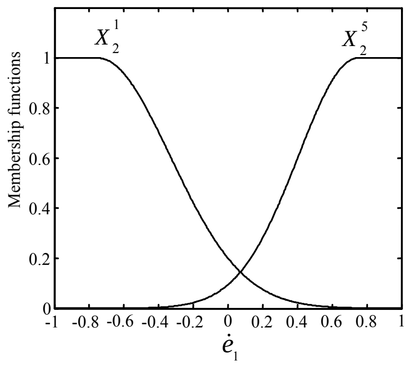
If e_1 is X_1^4, $\dot{e}_1$ is X_2^5, e_2 is X_3^4, and $\dot{e}_2$ is X_4^1, then K_{P1}= -3.1251, K_{I1}= -0.0319, K_{D1}= -1.4141, K_{P2}= -0.9229, K_{I2}= -0.0054, K_{D2}= -0.5983

If e_1 is X_1^4, $\dot{e}_1$ is X_2^5, e_2 is X_3^4, and $\dot{e}_2$ is X_4^2, then K_{P1}= -2.6762, K_{I1}= 0.0175, K_{D1}= -0.6700, K_{P2}= 2.6604, K_{I2}= 0.0231, K_{D2}= 0.4075

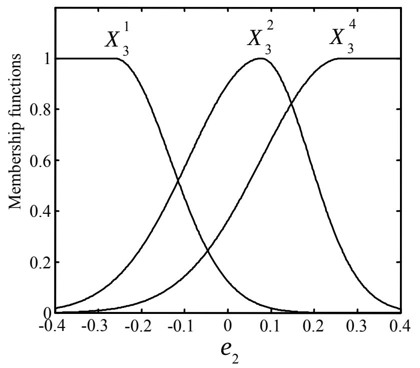
If e_1 is X_1^4, $\dot{e}_1$ is X_2^5, e_2 is X_3^4, and $\dot{e}_2$ is X_4^4, then K_{P1}= -1.3645, K_{I1}= -0.0602, K_{D1}= 0.6010, K_{P2}= 0.8689, K_{I2}= -0.0644, K_{D2}= -1.3954



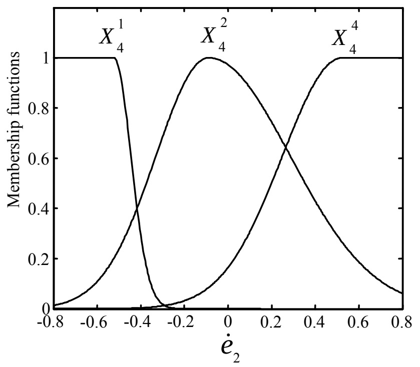
(a) Optimal membership function of $e_1(t)$



(b) Optimal membership function of $\dot{e}_1(t)$



(c) Optimal membership function of $e_2(t)$



(d) Optimal membership function of $\dot{e}_2(t)$

Figure 4. The membership functions of the input linguistic variables which are determined by applying Algorithm A with the fitness defined in (29).

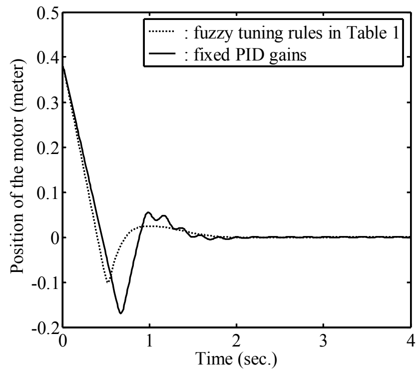


Figure 5. Plots of $r(t)$ for the seesaw system, which are generated by the fuzzy tuning rules and fixed PID gains, respectively.

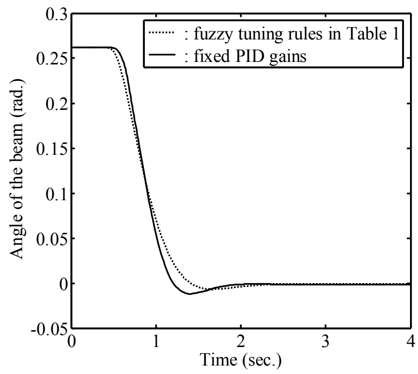


Figure 6. Plots of $\theta(t)$ for the seesaw system, which are generated by the fuzzy tuning rules and fixed PID gains, respectively.

5. Conclusion and Discussion

In this paper, a GA-based approach toward auto-tuning of a fuzzy PID controller is proposed. In the proposed method, the PID gains are expressed in the form of fuzzy rules such that they are not fixed and adaptive. Moreover, comparing with GA-tuned PID controllers presented in the past, the novelty of the proposed method is that all major components of a fuzzy PID controller can be determined simultaneously. Another merit of the proposed method is the way to define the fitness function based on the concept of multi-objective optimization. If one is interested in many performance criteria, then the fitness function can be defined a systematic way.

Actually, with a proper definition of the fitness function as shown in the simulated example, the number of fuzzy rules can also be treated as a parameter to be minimized in the proposed GA-based tuning process. To evaluate the feasibility of the proposed method, a typical benchmark such as the seesaw system is used for test. Though good simulated results are obtained, more effort will be needed before verifying the overall performance of the proposed method. Therefore, the application of the proposed method to more complicated systems is suggested as a topic for further research.

Acknowledgment

This work was supported in part by the National Science Council, Taiwan, R.O.C., under grants NSC NSC96-2628-E-224-007-MY2.

References

[1] S. Bennett, "Development of the PID controller," *IEEE Control Systems Magazine*, vol. 13, pp. 58-65, 1993.

[2] G. Chen, "Conventional and fuzzy PID controller: An overview," *International Journal of Intelligent Control Systems*, vol. 1, pp. 235-246, 1996.

[3] J. G. Ziegler and N. B. Nichols, "Optimum setting for automatic controller," *Transactions ASME*, vol. 64, pp. 759-768, 1942.

[4] M. G. Na, "Auto-tuned PID controller using a model predictive control method for the stream generator water level," *IEEE Transactions on Nuclear Science*, vol. 48, no. 5, pp. 1664-1671, 2001.

[5] A. Karimi, D. Garcia, and R. Longchamp, "PID controller tuning using Bode's integrals," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 6, pp. 812-821, 2003.

[6] S. Chen and Y. M. Cheng, "A structure-specified H_∞ optimal control design for practical applications: a genetic approach," *IEEE Transactions on Mechatronics*, vol. 6, no. 6, pp. 707-718, 1998.

[7] C. L. Lin, H. Y. Jan, and N. C. Shieh, "GA-based multiobjective PID control for a linear brushless DC motor," *IEEE Transactions on Mechatronics*, vol. 8, no. 1, pp. 56-65, 2003.

[8] I. Cervantes, R. Garrido, A. R. Jose, and A. Martinez, "Vision-based PID control of planar robots," *IEEE Transactions on Mechatronics*, vol. 9, no. 1, pp. 132-136, 2004.

[9] D. Misir, H. A. Malki, and G. Chen, "Design and analysis of a fuzzy proportional-integral-derivative controller," *International Journal of Fuzzy Sets Systems*, vol. 79, pp. 297-314, 1996.

[10] J. Casillas, O. Cordon, M. J. Jesus, and F. Herrera, "Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction," *IEEE Transaction on Fuzzy Systems*, vol. 13, no. 1, pp. 13-29, 2005.

[11] C. J. Wu, G. Y. Liu, M. Y. Cheng, and T. L. Lee, "A neural-network-based method for fuzzy parameter tuning of PID controllers," *Journal of the Chinese Institute of Engineers*, vol. 25, no. 3, pp. 265-276, 2002.

[12] Y. L. Sun and M. J. Er, "Hybrid fuzzy control of robotics systems," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 6, pp. 755-765, 2004.

[13] X. F. Wang, "Fuzzy number intuitionistic fuzzy arithmetic aggregation operators," International Journal of Fuzzy systems, vol. 10, no. 2, pp. 104-111, 2008.

[14] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Review*, vol. 28, no. 3, pp. 392-403, 1998.

[15] Y. W. Leung and Y. Wang, "Multi-objective programming using uniform design and genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Review*, vol. 30, no. 3, pp. 293-304, 2000.

[16] S. L. Ho, S. Y. Yang, G. Z. Ni, E. W. C. Lo, and H. C. Wong, "A particle swarm optimization- based method for multiobjective design optimizations," *IEEE Transactions on Magnetic*, vol. 41, no. 5, pp. 1756-1759, 2005.

[17] L. X. Wang, *A Course in Fuzzy Systems and Control*, New Jersey, Prentice-Hall, 1997.

[18] Z. Michalewicz, *Genetic Algorithm + Data Structure = Evolution Programs*, New York: Springer-Verlag, 1996.

[19] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms 2nd edition*, New York: John Wiley & Sons, 2004.

[20] M. Gen and R. Cheng, *Genetic Algorithm and Engineering Design*, New York: John Wiley & Sons, 1997.

[21] M. Bazaraa, J. Jarvis, and H. Sherali, *Linear Programming and Network Flows 2nd edition*, New York: Wiley, 1990.

[22] C. J. Wu, "Quasi Time-optimal PID control of a multivariable system: A seesaw example," *Journal of the Chinese Institute of Engineers*, vol. 22, no. 5, pp. 617-625, 1999.

[23] J. Hauser, S. Sastry, and P. Kokotovic, "Nonlinear control via approximate input-output linearization: the ball and beam example," *IEEE Transaction on Automatic Control*, vol. 37, no. 3, pp. 392-398, 1992.

[24] Y. Guo, D. J. Hill, and Z. P. Jiang, "Global nonlinear control of the ball and beam system," In *Proc. of the 35th Conference on Decision and Control*, pp. 2818-2823, Kobe, Japan, 1996.

[25] S. G. Cao and N. W. Rees, "Identification of dynamic fuzzy model," *Fuzzy Sets and System*, vol. 74, pp. 307-320, 1996.

[26] S. G. Cao, N. W. Rees, and G. Feng, "Quadratic

stability analysis and design of continuous-time fuzzy control systems," *International Journal of Systems Science*, vol. 27, pp. 193-203, 1996.
[27] A. Sabna, *Computational Dynamics*, New York: John Wiley & Sons, 1994.
[28] K. C. Sio and C. K. Lee, "Stability of Fuzzy PID Controllers," I*EEE Transaction on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 28, no. 14, pp. 490-495, 1998.
[29] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 122-128, 1986.
[30] K. F. Man, K. S. Tang, and S. Kwong, "Genetic algorithms: Concepts and applications," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 5, pp. 519-534, 1996.