A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System

Zwe-Lee Gaing, *Member, IEEE*

***Abstract*—In this paper, a novel design method for determining the optimal proportional-integral-derivative (PID) controller parameters of an AVR system using the particle swarm optimization (PSO) algorithm is presented. This paper demonstrated in detail how to employ the PSO method to search efficiently the optimal PID controller parameters of an AVR system. The proposed approach had superior features, including easy implementation, stable convergence characteristic, and good computational efficiency. Fast tuning of optimum PID controller parameters yields high-quality solution. In order to assist estimating the performance of the proposed PSO-PID controller, a new time-domain performance criterion function was also defined. Compared with the genetic algorithm (GA), the proposed method was indeed more efficient and robust in improving the step response of an AVR system.**

***Index Terms*—AVR system, optimal control, particle swarm optimization, PID controller.**

I. Introduction

DURING the past decades, the process control techniques in the industry have made great advances. Numerous control methods such as adaptive control, neural control, and fuzzy control have been studied [1]–[5]. Among them, the best known is the proportional-integral-derivative (PID) controller, which has been widely used in the industry because of its simple structure and robust performance in a wide range of operating conditions. Unfortunately, it has been quite difficult to tune properly the gains of PID controllers because many industrial plants are often burdened with problems such as high order, time delays, and nonlinearities [1]–[6]. Over the years, several heuristic methods have been proposed for the tuning of PID controllers. The first method used the classical tuning rules proposed by Ziegler and Nichols. In general, it is often hard to determine optimal or near optimal PID parameters with the Ziegler-Nichols formula in many industrial plants [1]–[3].

For these reasons, it is highly desirable to increase the capabilities of PID controllers by adding new features. Many artificial intelligence (AI) techniques have been employed to improve the controller performances for a wide range of plants while retaining their basic characteristics. AI techniques such as neural network, fuzzy system, and neural-fuzzy logic have been widely applied to proper tuning of PID controller parameters [1], [2].

Many random search methods, such as genetic algorithm (GA) and simulated annealing (SA) [2]–[9], have recently received much interest for achieving high efficiency and searching global optimal solution in problem space. The GA method is usually faster than the SA method because the GA has parallel search techniques, which emulate natural genetic operations. Due to its high potential for global optimization, GA has received great attention in control systems such as the search of optimal PID controller parameters. Although GAs have widely been applied to many control systems, its natural genetic operations would still result in enormous computational efforts [5], [6]. In order to overcome the disadvantages, the use of real-value representation in the GA is proposed to offer a number of advantages in numerical function optimization over binary encoding because there is no need to convert chromosomes to binary type [3]–[5], [15].

Though the GA methods have been employed successfully to solve complex optimization problems, recent research has identified some deficiencies in GA performance. This degradation in efficiency is apparent in applications with highly *epistatic* objective functions [i.e., where the parameters being optimized are highly correlated (the crossover and mutation operations cannot ensure better fitness of offspring because chromosomes in the population have similar structures and their average fitness is high toward the end of the evolutionary process)] [10], [14]. Moreover, the premature convergence of GA degrades its performance and reduces its search capability [10].

Particle swarm optimization (PSO), first introduced by Kennedy and Eberhart, is one of the modern heuristic algorithms. It was developed through simulation of a simplified social system, and has been found to be robust in solving continuous nonlinear optimization problems [11]–[15]. The PSO technique can generate a high-quality solution within shorter calculation time and stable convergence characteristic than other stochastic methods [14]–[16]. Much research is still in progress for proving the potential of the PSO in solving complex power system operation problems. Researchers including Yoshida *et al.* have presented a PSO for reactive power and voltage control (VVC) considering voltage security assessment. Their method is compared with the reactive tabu system (RTS) and enumeration method on practical power system, and has shown promising results [16]. Naka *et al.* have presented the use of a hybrid PSO method for solving efficiently the practical distribution state estimation problem [17]. Because the PSO method is an excellent optimization methodology and a promising approach for solving the optimal PID controller parameters problem; therefore, this study develops the PSO-PID

Manuscript received November 6, 2002. This work was supported by the National Science Council of Taiwan, R.O.C.

The author is with the Department of Electrical Engineering, Kao-Yuan Institute of Technology, Kaohsiung 821, Taiwan, R.O.C. (e-mail: zlgaing@ms39.hinet.net).

Digital Object Identifier 10.1109/TEC.2003.821821

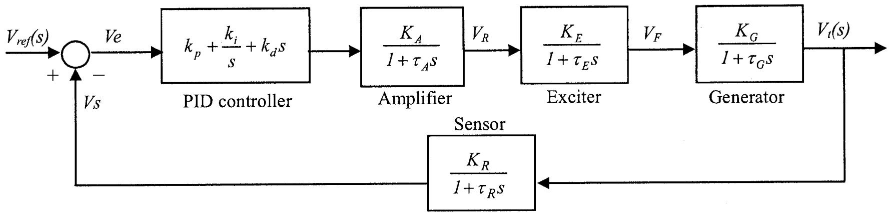

Fig. 1. Block diagram of an AVR system with a PID controller.

controller to search optimal PID parameters. This controller is called the PSO-PID controller.

The integral performance criteria in frequency domain were often used to evaluate the controller performance, but these criteria have their own advantages and disadvantages [5], [6]. In this paper, a simple performance criterion in time domain is proposed for evaluating the performance of a PSO-PID controller that was applied to the complex control system.

The generator excitation system maintains generator voltage and controls the reactive power flow using an automatic voltage regulator (AVR) [18]. The role of an AVR is to hold the terminal voltage magnitude of a synchronous generator at a specified level. Hence, the stability of the AVR system would seriously affect the security of the power system. In this paper, a practical high-order AVR system with a PID controller is adopted to test the performance of the proposed PSO-PID controller.

In this paper, besides demonstrating how to employ the PSO method to obtain the optimal PID controller parameters of an AVR system, many performance estimation schemes are performed to examine whether the proposed method has better performance than the real-value GA method in solving the optimal PID controller parameters.

II. Linearized Model of an AVR System

A. PID Controller

The PID controller is used to improve the dynamic response as well as to reduce or eliminate the steady-state error. The derivative controller adds a finite zero to the open-loop plant transfer function and improves the transient response. The integral controller adds a pole at the origin, thus increasing system type by one and reducing the steady-state error due to a step function to zero. The PID controller transfer function is

$$C(s) = k_p + \frac{k_i}{s} + k_d s. \tag{1}$$

B. Linearized Model of an AVR System [16]

The role of an AVR is to hold the terminal voltage magnitude of a synchronous generator at a specified level. A simple AVR system comprises four main components, namely amplifier, exciter, generator, and sensor. For mathematical modeling and transfer function of the four components, these components must be linearized, which takes into account the major time constant and ignores the saturation or other nonlinearities. The reasonable transfer function of these components may be represented, respectively, as follows [16].

- Amplifier model.

  The amplifier model is represented by a gain K_A and a time constant τ_A; the transfer function is

$$\frac{V_R(s)}{V_e(s)} = \frac{K_A}{1+\tau_A s}. \tag{2}$$

  Typical values of K_A are in the range of 10 to 400. The amplifier time constant is very small ranging from 0.02 to 0.1 s.

- Exciter model.

  The transfer function of a modern exciter may be represented by a gain K_E and a single time constant τ_E

$$\frac{V_F(s)}{V_R(s)} = \frac{K_E}{1+\tau_E s}. \tag{3}$$

  Typical values of K_E are in the range of 10 to 400. The time constant τ_E is in the range of 0.5 to 1.0 s.

- Generator model.

  In the linearized model, the transfer function relating the generator terminal voltage to its field voltage can be represented by a gain K_G and a time constant τ_G

$$\frac{V_t(s)}{V_F(s)} = \frac{K_G}{1+\tau_G s}. \tag{4}$$

  These constants are load dependent, K_G may vary between 0.7 to 1.0, and τ_G. between 1.0 and 2.0 s from full load to no load.

- Sensor model.

  The sensor is modeled by a simple first-order transfer function, given by

$$\frac{V_s(s)}{V_t(s)} = \frac{K_R}{1+\tau_R s}. \tag{5}$$

  τ_R is very small, ranging from of 0.001 to 0.06 s.

C. AVR System With PID Controller

The above models provide an AVR system compensated with a PID controller block diagram, which is shown in Fig. 1.

D. Performance Estimation of PID Controller

In general, the PID controller design method using the integrated absolute error *(IAE)*, or the integral of squared-error *(ISE)*, or the integrated of time-weighted-squared-error *(ITSE)* is often employed in control system design because it can be evaluated analytically in the frequency domain [3]–[6]. The three integral performance criteria in the frequency domain have their own advantages and disadvantages. For example, a disadvantage of the *IAE* and *ISE* criteria is that its minimization can result in a response with relatively small overshoot but a long settling time because the *ISE* performance criterion weights all errors equally independent of time. Although the

ITSE performance criterion can overcome the disadvantage of the *ISE* criterion, the derivation processes of the analytical formula are complex and time-consuming [6]. The *IAE*, *ISE*, and *ITSE* performance criterion formulas are as follows:

$$IAE = \int_0^\infty |r(t) - y(t)|\,dt = \int_0^\infty |e(t)|dt \qquad (6)$$

$$ISE = \int_0^\infty e^2(t)dt \qquad (7)$$

$$ISTE = \int_0^\infty te^2(t)dt. \qquad (8)$$

In this paper, a new performance criterion in the time domain is proposed for evaluating the PID controller. A set of good control parameters k_p, k_i, and k_d can yield a good step response that will result in performance criteria minimization in the time domain. These performance criteria in the time domain include the overshoot M_p, rise time t_r, settling time t_s, and steady-state error E_{ss}. Therefore, a new performance criterion $W(K)$ is defined as follows:

$$\min_{K:\text{stabilizing}} W(K) = (1 - e^{-\beta}) \cdot (M_p + E_{ss}) + e^{-\beta} \cdot (t_s - t_r) \qquad (9)$$

where K is $[k_p, k_i, k_d]$, and β is the weighting factor.

The performance criterion $W(K)$ can satisfy the designer requirements using the weighting factor β value. We can set β to be larger than 0.7 to reduce the overshoot and steady-state error. On the other hand, we can set β to be smaller than 0.7 to reduce the rise time and settling time. In this paper, β is set in the range of 0.8 to 1.5.

III. Overview of Particle Swarm Optimization

A. Features of Particle Swarm Algorithm [10]–[13]

In 1995, Kennedy and Eberhart first introduced the particle swarm optimization (PSO) method. It is one of the optimization techniques and a kind of evolutionary computation technique. The method has been found to be robust in solving problems featuring nonlinearity and nondifferentiability, multiple optima, and high dimensionality through adaptation, which is derived from the social-psychological theory. The features of the method are as follows [14].

- The method is developed from research on swarm such as fish schooling and bird flocking.
- It can be easily implemented, and has stable convergence characteristic with good computational efficiency.

Instead of using evolutionary operators to manipulate the particle (individual), like in other evolutionary computational algorithms, each particle in PSO flies in the search space with velocity which is dynamically adjusted according to its own flying experience and its companions' flying experience. Each particle is treated as a volumeless particle in ***g***-dimensional search space.

Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (evaluating value) it has achieved so far. This value is called pbest. Another best value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the group, is called gbest.

The PSO concept consists of, at each time step, changing the velocity of each particle toward its pbest and gbest locations. Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and gbest locations.

For example, the jth particle is represented as $x_j = (x_{j,1}, x_{j,2}, \ldots, x_{j,g})$ in the ***g***-dimensional space. The best previous position of the jth particle is recorded and represented as $\text{pbest}_j = (\text{pbest}_{j,1}, \text{pbest}_{j,2}, \ldots, \text{pbest}_{j,g})$. The index of best particle among all of the particles in the group is represented by the gbest_g. The rate of the position change (velocity) for particle j is represented as $v_j = (v_{j,1}, v_{j,2}, \ldots, v_{j,g})$. The modified velocity and position of each particle can be calculated using the current velocity and the distance from $\text{pbest}_\text{j,g}$ to gbest_g as shown in the following formulas:

$$\begin{aligned} v_{j,g}^{(t+1)} &= w \cdot v_{j,g}^{(t)} + c_1^* rand()^* (\text{pbest}_{j,g} - x_{j,g}^{(t)}) \\ &\quad + c_2^* Rand()^* (\text{gbest}_g - x_{j,g}^{(t)}) \qquad (10) \\ x_{j,g}^{(t+1)} &= x_{j,g}^{(t)} + v_{j,g}^{(t+1)} \\ j &= 1, 2, \ldots, n \\ g &= 1, 2, \ldots, m \qquad (11) \end{aligned}$$

where

n	number of particles in a group;
m	number of members in a particle;
t	pointer of iterations (generations);
$v_{j,g}^{(t)}$	velocity of particle j at iteration t, $V_g^{\min} \le v_{j,g}^{(t)} \le V_g^{\max}$;
w	inertia weight factor;
$c1$, $c2$	acceleration constant;
$rand()$, $Rand()$	random number between 0 and 1;
$x_{j,g}^{(t)}$	current position of particle j at iteration t;
pbest_j	pbest of particle j;
gbest	gbest of the group.

In the above procedures, the parameter $V^{\max}$ determined the resolution, or fitness, with which regions be searched between the present position and the target position. If $V^{\max}$ is too high, particles might fly past good solutions. If $V^{\max}$ is too small, particles may not explore sufficiently beyond local solutions. In many experiences with PSO, $V^{\max}$ was often set at 10–20% of the dynamic range of the variable on each dimension.

The constants c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle toward pbest and gbest positions. Low values allow particles to roam far from the target regions before being tugged back. On the other hand, high values result in abrupt movement toward, or past, target regions. Hence, the acceleration constants c_1 and c_2 were often set to be 2.0 according to past experiences.

Suitable selection of inertia weight w in (12) provides a balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution. As originally developed, w often decreases linearly from about 0.9 to 0.4 during a run. In general, the inertia weight w is set according to the following equation:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{\text{iter}_{\max}} \times \text{iter} \qquad (12)$$

where $\mathrm{iter}_{\max}$ is the maximum number of iterations (generations), and iter is the current number of iterations.

IV. Implementation of a PSO-PID Controller

In this paper, a PID controller using the PSO algorithm was developed to improve the step transient response of AVR of a generator. It was also called the PSO-PID controller. The PSO algorithm was mainly utilized to determine three optimal controller parameters k_p, k_i, and k_d, such that the controlled system could obtain a good step response output.

A. Individual String Definition

To apply the PSO method for searching the controller parameters, we use the "individual" to replace the "particle" and the "population" to replace the "group" in this paper. We defined three controller parameters k_p, k_i, and k_d, to compose an individual K by $K := [k_p, k_i, k_d]$; hence, there are three members in an individual. These members are assigned as real values. If there are n individuals in a population, then the dimension of a population is $n \times 3$. The matrix representation in a population is as follows.

B. Evaluation Function Definition

In the meantime, we defined the evaluation function f given in (13) as the evaluation value of each individual in population. The evaluation function f is a reciprocal of the performance criterion $W(K)$ as in (9). It implies the smaller $W(K)$ the value of individual K, the higher its evaluation value

$$f = \frac{1}{W(K)}. \tag{13}$$

In order to limit the evaluation value of each individual of the population within a reasonable range, the ***Routh–Hurwitz*** criterion must be employed to test the closed-loop system stability before evaluating the evaluation value of an individual. If the individual satisfies the ***Routh–Hurwitz*** stability test applied to the characteristic equation of the system, then it is a feasible individual and the value of $W(K)$ is small. In the opposite case, the $W(K)$ value of the individual is penalized with a very large positive constant.

C. Proposed PSO-PID Controller

This paper presents a PSO-PID controller for searching the optimal or near optimal controller parameters k_p, k_i, and k_d, with the PSO algorithm. Each individual K contains three members k_p, k_i, and k_d. The matrix representation of the initial population is described in Section IV-A. Its dimension is $n \times 3$.

The searching procedures of the proposed PSO-PID controller were shown as below.

Step 1) Specify the lower and upper bounds of the three controller parameters and initialize randomly the individuals of the population including searching points, velocities, pbests, and gbest.

Step 2) For each initial individual K of the population, employ the ***Routh-Hurwitz*** criterion to test the closed-loop system stability and calculate the values of the four performance criteria in the time domain, namely M_p, E_{ss}, t_r, and t_s.

Step 3) Calculate the evaluation value of each individual in the population using the evaluation function f given by (13).

Step 4) Compare each individual's evaluation value with its pbest. The best evaluation value among the pbest is denoted as gbest.

Step 5) Modify the member velocity v of each individual K according to (14)

$$\begin{aligned} v_{j,g}^{(t+1)} &= w \cdot v_j^{(t)} + c_1^* rand()^* (\mathrm{pbest_{j,g}} - \mathrm{k_{j,g}^{(t)}}) \\ &\quad + c_2^* Rand()^* (\mathrm{gbest_g} - \mathrm{k_{j,g}^{(t)}}) \\ j &= 1, 2 \ldots n, \\ g &= 1, 2 \ldots, 3 \end{aligned} \tag{14}$$

where the value of w is set by (12). When g is 1, $v_{j,1}$ represents the change in velocity of k_p controller parameter. When g is 2, $v_{j,2}$ represents the change in velocity of k_i controller parameter.

Step 6) If $v_{j,g}^{(t+1)} > V_g^{\max}$, then $v_{j,g}^{(t+1)} = V_g^{\max}$
If $v_{j,g}^{(t+1)} < V_g^{\min}$, then $v_{j,g}^{(t+1)} = V_g^{\min}$.

Step 7) Modify the member position of each individual K according to (15)

$$\begin{aligned} k_{j,g}^{(t+1)} &= k_{j,g}^{(t)} + v_{j,g}^{(t+1)}, \\ k_g^{\min} &\le k_{j,g}^{(t+1)} \le k_g^{\max} \end{aligned} \tag{15}$$

where $k_g^{\min}$ and $k_g^{\max}$ represent the lower and upper bounds, respectively, of member g of the individual K. For example, when g is 1, the lower and upper bounds of the k_p controller parameter are $k_p^{\min}$ and $k_p^{\max}$, respectively.

Step 8) If the number of iterations reaches the maximum, then go to **Step 9**. Otherwise, go to **Step 2.**

Step 9) The individual that generates the latest *gbest* is an optimal controller parameter.

V. Dynamic Behaviors Estimation

In order to examine the dynamic behaviors and convergence characteristic of the proposed method, two statistical indexes, namely the mean value (μ) and the standard deviation (σ) of evaluation values of all individuals in the population during the computing processes, were used. The mean value can display the accuracy of the algorithm, and the standard deviation can measure the convergence speed of the algorithm. The formulas for calculating the mean value and the standard deviation (σ) of evaluation values are as follows, respectively:

$$\mu = \frac{\sum_{i=1}^{n} f(K_i)}{n} \tag{16}$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (f(K_i) - \mu)^2} \tag{17}$$

where $f(K_i)$ is the evaluation value of the individual K_i and n is the population size.

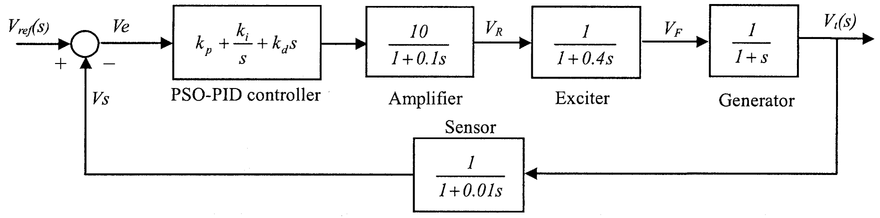

Fig. 2. Block diagram of an AVR system with a PSO-PID controller.

TABLE I
RANGE OF THREE CONTROLLER PARAMETERS

Controller parameters	*Min. value*	*Max. value*
k_p	*0*	*1.50*
k_i	*0*	*1.0*
k_d	*0*	*1.0*

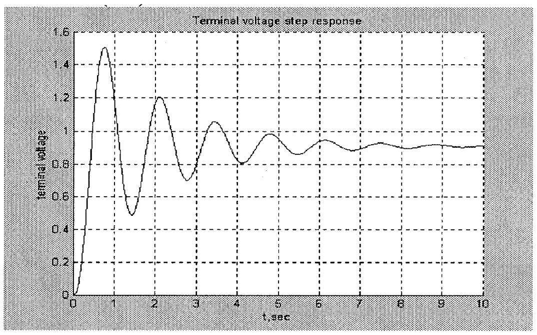

Fig. 3. Terminal voltage step response of an AVR system without PID controller.

VI. Numerical Examples and Results

A. AVR System Parameters

To verify the efficiency of the PSO-PID controller, a practical high-order AVR system was tested. The AVR system has the following parameters:

The block diagram of the AVR system with a PID controller is shown in Fig. 2. The lower and upper bounds of the three controller parameters were as shown in Table I.

Fig. 3 shows the original terminal voltage step response of the AVR system without a PID controller. To simulate this case, we found that $M_p = 50.61\%$, $E_{ss} = 0.0909$, $t_r = 0.2693$ (s), and $t_s = 6.9834$ (s.

B. Performance of the PSO-PID Controller

According to the trials, the following PSO parameters are used for verifying the performance of the PSO-PID controller in searching the PID controller parameters:

- the member of each individual is k_p, k_i, and k_d;
- population size $= 50$;
- inertia weight factor w is set by (12), where $w_{\max} = 0.9$ and $w_{\min} = 0.4$;
- the limit of change in velocity $V_{k_p}^{\max} = k_p^{\max}/2$, $V_{k_i}^{\max} = k_i^{\max}/2$, and $V_{k_d}^{\max} = k_d^{\max}/2$;
- acceleration constant $c_1 = 2$ and $c_2 = 2$.

The PSO-PID controller shown in Fig. 2 then replaced the PID controller; the simulation results that showed the best solution were summarized in Table II. Figs. 4 and 5 showed the convergence characteristics of the PSO-PID controller and terminal voltage step response of the AVR system at different simulation conditions, respectively. As can be seen, through about 50 iterations (50 generations), the PSO method can prompt convergence and obtain good evaluation value. These results show that the PSO-PID controller can search optimal PID controller parameters quickly and efficiently.

TABLE II
BEST SOLUTION USING PSO-PID CONTROLLER WITH THE DIFFERENT β VALUES

β	*Number of Generation*	k_p	k_i	k_d	$M_p(\%)$	E_{ss}	t_s	t_r	*Evaluation value*
1.0	*200*	0.6570	0.5389	0.2458	1.16	0	0.4025	0.2767	1.4583
1.5	*200*	0.6254	0.4577	0.2187	0.44	0	0.4528	0.3070	1.2303

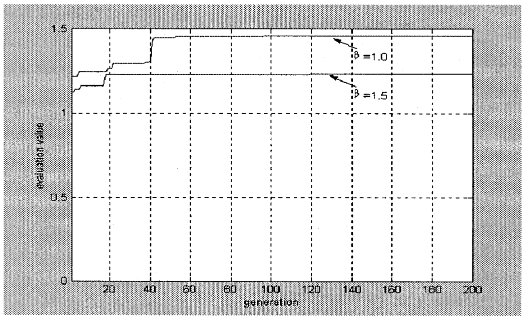

Fig. 4. Convergence tendency of the PSO-PID controller.

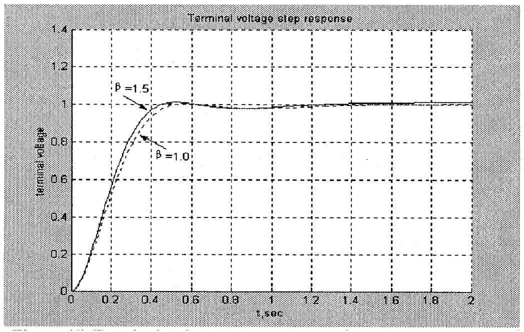

Fig. 5. Terminal voltage step response of an AVR system with the PSO-PID controller.

C. Comparison of Two Proposed Controllers

In order to emphasize the advantages of the proposed PSO-PID controller, we also implemented the GA-PID controller derived from the real-value GA method with the Elitism scheme [5], [6]. We have compared the characteristics of the two controllers using the same evaluation function and individual definition. The following real-value GA parameters have been used:

TABLE III
SUMMARY OF SIX SIMULATION RESULTS

Simulation Example	*β*	*Number of Generations*
Example I	1.0	50
Example II	1.0	100
Example III	1.0	150
Example IV	1.5	50
Example V	1.5	100
Example VI	1.5	150

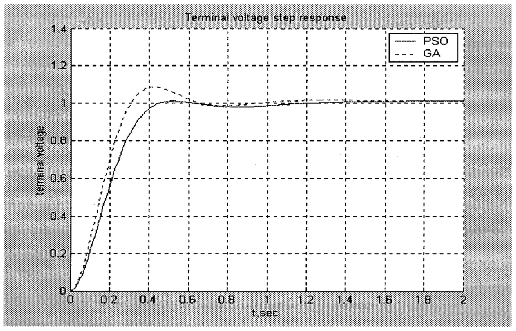

Fig. 6. Terminal voltage step response of an AVR system with different controllers (Example I, $\beta = 1.0$, generations $= 50$).

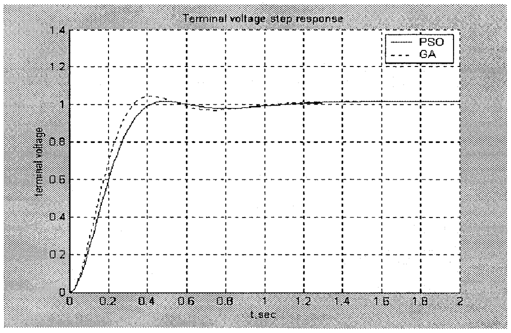

Fig. 7. Terminal voltage step response of an AVR system with different controllers (Example II, $\beta = 1.0$, generations $= 100$).

- the members of each individual are k_p, k_i, and k_d;
- population size $= 50$;
- crossover rate $P_c = 0.6$;
- mute rate $P_m = 0.01$;
- $a = 2$.

Two proposed controllers and their performance evaluation criteria in the time domain were implemented by Matlab and control system toolbox, and executed on a Pentium III 550 personal computer with 256-MB RAM.

1) Terminal Voltage Step Response: There were six simulation examples to evaluate the performance of both the PSO-PID and the GA-PID controllers. In each simulation example, the weighting factor β in the performance criterion and the number of iterations (generations) were set as follows:

The simulation results that showed the best solution were summarized in Table III. As can be seen, both controllers could give good PID controller parameters in each simulation example, providing good terminal voltage step response of the AVR system. Table III also shows the four performance criteria in the time domain of each example. As revealed by the above four performance criteria, the PSO-PID controller has better performance than the GA-PID controller.

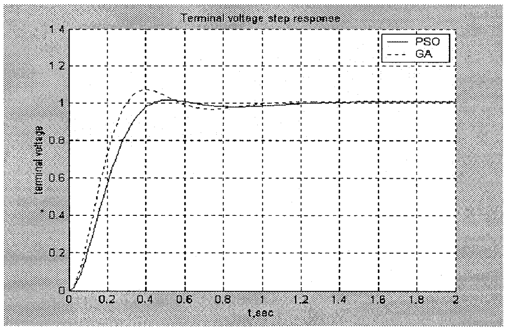

Fig. 8. Terminal voltage step response of an AVR system with different controllers (Example III, $\beta = 1.0$, generations $= 150$).

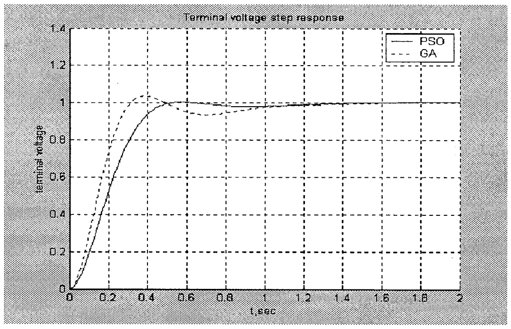

Fig. 9. Terminal voltage step response of an AVR system with different controllers (Example IV, $\beta = 1.5$, generations $= 50$).

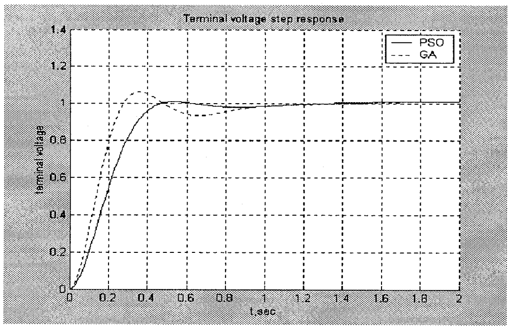

Fig. 10. Terminal voltage step response of an AVR system with different controllers (Example V, $\beta = 1.5$, generations $= 100$).

Figs. 6–11 show the terminal voltage step response of the AVR system of the six simulation examples. As can be seen, the PSO-PID controller could create very perfect step response of the AVR system, indicating that the PSO-PID controller is better than the GA-PID controller.

2) Convergence Characteristic: Under the same conditions, we performed simulations using the two proposed controllers to compare their convergence characteristics. Fig. 12 showed their convergence properties. As can be seen, the PSO-PID controller has better evaluation value than the GA-PID controller. The results showed that the PSO-PID controller could obtain higher quality solution, indicating the drawbacks of GA method mentioned in [10] and [14].

We also performed 100 trials for both proposed controllers with different random number to observe the variation in their evaluation values. In addition, the maximum, minimum, and average evaluation values were obtained by the two methods. The

TABLE IV
COMPARISON OF THE EVALUATION VALUE BETWEEN BOTH METHODS ($\beta = 1.5$, GENERATION $= 100$)

Simulation Example	β	*Number of Generations*	*Type of Controller*	k_p	k_i	k_d	$M_p(\%)$	E_{ss}	t_s	t_r	*Evaluation value*
Example I	*1.0*	*50*	***GA-PID***	0.8861	0.7984	0.3158	8.66	0	0.5980	0.2019	1.2011
			PSO-PID	0.6568	0.5393	0.2458	1.17	0	0.4027	0.2768	1.4581
Example II	*1.0*	*100*	***GA-PID***	0.7722	0.7201	0.3196	4.54	0	0.8645	0.2138	1.1109
			PSO-PID	0.6751	0.5980	0.2630	1.71	0	0.3795	0.2648	1.4596
Example III	*1.0*	*150*	***GA-PID***	0.8663	0.7531	0.3365	7.34	0	0.8519	0.1959	1.0871
			PSO-PID	0.6570	0.5390	0.2458	1.16	0	0.4025	0.2767	1.4583
Example IV	*1.5*	*50*	***GA-PID***	0.7717	0.5930	0.3507	3.62	0	1.0517	0.2003	1.0051
			PSO-PID	0.6271	0.4652	0.2209	0.45	0	0.4498	0.3025	1.2297
Example V	*1.5*	*100*	***GA-PID***	0.8372	0.6973	0.3927	6.17	0	0.9396	0.1859	1.0071
			PSO-PID	0.6477	0.5128	0.2375	0.92	0	0.4168	0.2860	1.2297
Example VI	*1.5*	*150*	***GA-PID***	0.8935	0.6458	0.4014	7.48	0	0.9822	0.1750	0.9851
			PSO-PID	0.6476	0.5216	0.2375	0.91	0	0.4168	0.2861	1.2297

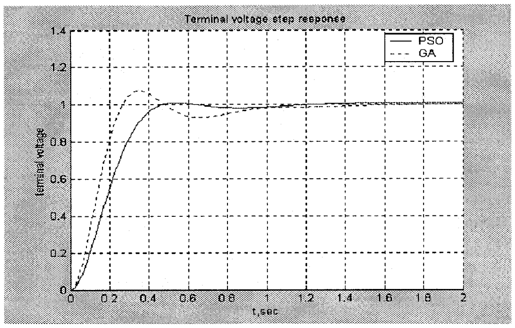

Fig. 11. Terminal voltage step response of an AVR system with different controllers (Example VI, $\beta = 1.5$, generations $= 150$).

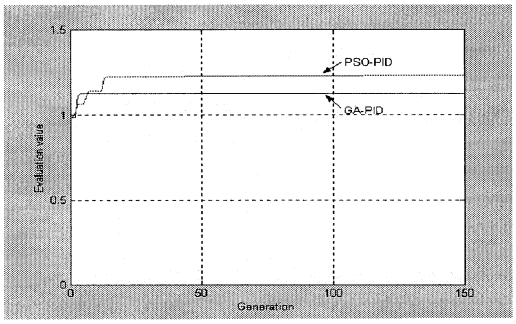

Fig. 12. Convergence tendency of the evaluation value of both methods.

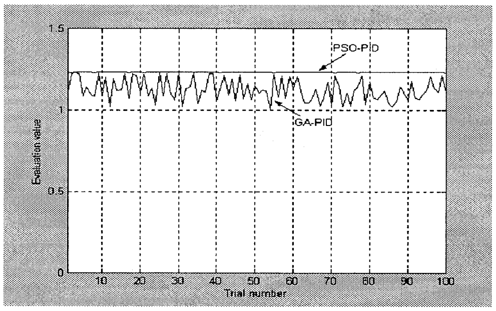

Fig. 13. Comparison of the statistical evaluation values of both methods (100 trials).

results were shown in Fig. 13 and Table IV. As can be seen, the evaluation values of the PSO-PID controller generated fluctuation in a small range ($\Delta E = 0.0030$), thus verifying that the PSO-PID controller has better convergence characteristic.

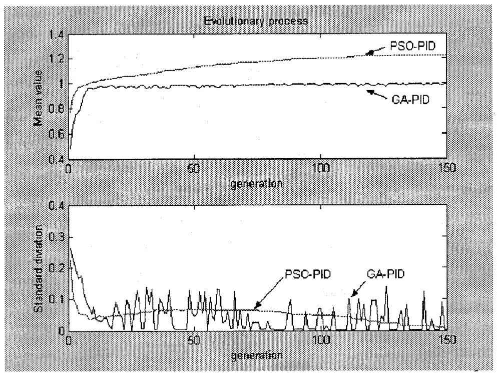

Fig. 14. Convergence tendency of both μ and σ of evaluation values using both methods (Example IV, $\beta = 1.5$, generations $= 150$).

TABLE V
COMPARISON OF COMPUTATION EFFICIENCY OF BOTH METHODS ($\beta = 1.5$)

	Max.	*Min.*	*$\triangle E$ (Max.-Min.)*	*Average*
PSO-PID	1.2301	1.2271	0.0030	1.2291
GA-PID	1.2234	0.9993	0.2241	1.1237

3) Dynamic Convergence Behavior: In addition, during the evolutionary processing of the two proposed methods, after each iteration, the mean value (μ) and the standard deviation (σ) of the evaluation values of all individuals in the population were recorded for observing the dynamic convergence behavior of the individuals in population. Fig. 14 displays the recorded data in Example VI. As seen in the simulation, with the same number of iterations, though both controllers can obtain stable mean evaluation value (μ) under the same evaluation function and simulation conditions, the GA-PID controller brings premature convergence such that the evaluation value and mean value are smaller. Conversely, the PSO-PID controller has better evaluation value and mean value, showing that it can achieve better accuracy. Simultaneously, we can also find that the convergence tendency of the standard deviation (σ) of evaluation values in the PSO-PID controller is much faster than the GA-PID controller, because the latter presented fluctuation resulted from the mutation in the GA method. This can prove that the PSO method has better convergence efficiency in solving the power optimization problems.

4) Computation Efficiency: The comparison of computation efficiency of both methods is shown in Table V. As can be seen, because the PSO method does not perform the selection and crossover operations in evolutionary processes, it can save some computation time compared with the GA method, thus proving that the PSO-PID controller is more efficient than the GA-PID controller.

VII. Discussion and Conclusion

This paper presents a novel design method for determining the PID controller parameters using the PSO method. The proposed method integrates the PSO algorithm with the new time-domain performance criterion into a PSO-PID controller. Through the simulation of a practical AVR system, the results show that the proposed controller can perform an efficient search for the optimal PID controller parameters.

In addition, in order to verify it being superior to the GA method, many performance estimation schemes are performed, such as

- multiple simulation examples for their terminal voltage step responses;
- convergence characteristic of the best evaluation value;
- dynamic convergence behavior of all individuals in population during the evolutionary processing;
- computation efficiency.

It is clear from the results that the proposed PSO method can avoid the shortcoming of premature convergence of GA method and can obtain higher quality solution with better computation efficiency. Therefore, the proposed method has more robust stability and efficiency, and can solve the searching and tuning problems of PID controller parameters more easily and quickly than the GA method.

Different PSO optimization parameters are required for solving different problems in practical application, such as the number of agents (individuals), weight factors $\boldsymbol{w}_{\max}$ and $\boldsymbol{w}_{\min}$, acceleration factors $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, and the limit of change in velocity $V^{\max}$. Hence, how to select suit parameters for the target problem, such as the sensitivity analysis of optimization parameters for finding the best parameters, is one of our future works.

Acknowledgment

The authors acknowledge the technical support from Taiwan Power Company.

References

[1] A. Visioli, "Tuning of PID controllers with fuzzy logic," *Proc. Inst. Elect. Eng. Contr. Theory Applicat.*, vol. 148, no. 1, pp. 1–8, Jan. 2001.

[2] T. L. Seng, M. B. Khalid, and R. Yusof, "Tuning of a neuro-fuzzy controller by genetic algorithm," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 226–236, Apr. 1999.

[3] R. A. Krohling and J. P. Rey, "Design of optimal disturbance rejection PID controllers using genetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 5, pp. 78–82, Feb. 2001.

[4] Y. Mitsukura, T. Yamamoto, and M. Kaneda, "A design of self-tuning PID controllers using a genetic algorithm," in *Proc. Amer. Contr. Conf.*, San Diego, CA, June 1999, pp. 1361–1365.

[5] T. Kawabe and T. Tagami, "A real coded genetic algorithm for matrix inequality design approach of robust PID controller with two degrees of freedom," in *Proc. 12th IEEE Int. Symp. Intell. Contr.*, Istanbul, Turkey, July 1997, pp. 119–124.

[6] R. A. Krohling, H. Jaschek, and J. P. Rey, "Designing PI/PID controller for a motion control system based on genetic algorithm," in *Proc. 12th IEEE Int. Symp. Intell. Contr.*, Istanbul, Turkey, July 1997, pp. 125–130.

[7] D. P. Kwok and F. Sheng, "Genetic algorithm and simulated annealing for optimal robot arm PID control," in *Proc IEEE Conf. Evol. Comput.*, Orlando, FL, 1994, pp. 707–713.

[8] T. Ota and S. Omatu, "Tuning of the PID control gains by GA," in *Proc. IEEE Conf. Emerging Technol. Factory Automation*, Kauai, HI, Nov. 1996, pp. 272–274.

[9] A. H. Jones and P. B. D. Oliveira, "Genetic auto-tuning of PID controllers," in *Proc. Inst. Elect. Eng. Conf. Genetic Algorithm Eng. Syst.: Innovations Applicat.*, Sept. 1995, pp. 141–145.

[10] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 2 ed. New York: IEEE Press, 2000.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, vol. IV, Perth, Australia, 1995, pp. 1942–1948.

[12] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput.*, Anchorage, AK, May 1998, pp. 69–73.

[13] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proc. IEEE Int. Conf. Evol. Comput.*, Washington, DC, July 1999, pp. 1945–1950.

[14] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Proc. IEEE Int. Conf. Evol. Comput.*, Anchorage, AK, May 1998, pp. 611–616.

[15] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proc. IEEE Int. Conf. Evol. Comput.*, Anchorage, AK, May 1998, pp. 84–89.

[16] H. Yoshida, K. Kawata, and Y. Fukuyama, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Trans. Power Syst.*, vol. 15, pp. 1232–1239, Nov. 2000.

[17] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "Practical distribution state estimation using hybrid particle swarm optimization," in *Proc. IEEE Power Eng. Soc. Winter Meeting*, vol. 2, 2001, pp. 815–820.

[18] G. Cheng, *Genetic Algorithms & Engineering Design*. New York: Wiley, 1997.

[19] H. Saadat, *Power System Analysis*. New York: McGraw-Hill, 1999.

Zwe-Lee Gaing (M'02) received the Ph.D. degree from National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C., in 1996.

Currently, he is an Associate Professor of Electrical Engineering at Kao-Yuan Institute of Technology, Kaohsiung, Taiwan. His research interests include the field of artificial intelligence with application to control system.