Tamkang Journal of Science and Engineering, Vol. 12, No. 3, pp. 259–270 (2009)

Optimal PID Controller Design for AVR System

Ching-Chang Wong*, Shih-An Li and Hou-Yi Wang

Department of Electrical Engineering, Tamkang University, Tamsui, Taiwan 251, R.O.C.

Abstract

In this paper, a real-valued genetic algorithm (RGA) and a particle swarm optimization (PSO) algorithm with a new fitness function method are proposed to design a PID controller for the Automatic Voltage Regulator (AVR) system. The proposed fitness function can let the RGA and PSO algorithm search a high-quality solution effectively and improve the transient response of the controlled system. The proposed algorithms are applied in the PID controller design for the AVR system. Some simulation and comparison results are presented. We can see that the proposed RGA and PSO algorithm with this new fitness function can find a PID control parameter set effectively so that the controlled AVR system has a better control performance.

***Key Words*:** PID Controller, Genetic Algorithms, Particle Swarm Optimization, Automatic Voltage Regulator (AVR)

1. Introduction

There are three coefficients: proportional coefficient, differential coefficient, and integral coefficient in the PID controller. By tuning these three parameters (coefficients), the PID controller can provide individualized control requirements. In recent years, many intelligence algorithms are proposed to tuning the PID parameters. Tuning PID parameters by the optimal algorithms such as the Simulated Annealing (SA), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO) algorithm. Chent et al. proposed a method to tune PID parameters by SA [1]. However, it is slow to search the best solution. Kwok and Sheng considered GA and SA for the optimal robot arm PID control [2]. Some simulation results illustrate that not only the speed of operation but also the system response by GA is better than that by SA. Mitsukura et al. [3] and Krohling et al. [4] also used GA to search the optimal PID control parameters and they have nice performance in the simulation results. Genetic algorithms are methods to obtain an optimal solution by applying a theory of biological evolution [5,6]. Genetic algorithms can be found in many applications in biogenetics, computer science, engineering, economics, chemistry, manufacturing, mathematics, physics, and other fields. For example, GAs can be applied to discuss the fuzzy modeling, data classification, and omni-directional robot design problems [7–9]. The PSO algorithm, proposed by Kennedy and Eberhart [10] in 1995, is an another popular optimal algorithm. It was developed through a simulation of a simplified social system and some papers were proposed to improve the PSO algorithm [11–13]. The PSO technique can generate a high-quality solution within a shorter calculation time and have a stable convergence characteristic than other stochastic methods [14–16]. It has many applications in engineering fields. In the PID controller design, the PSO algorithm is applied to search a best PID control parameters [17,18].

Many research papers provided many improvement methods to improve the search performance of the GA and PSO algorithms [19,20]. In this paper, a real-valued GA (RGA) and a PSO algorithm with a new fitness function is proposed to find a PID control parameter set for

*Corresponding author. E-mail: wong@ee.tku.edu.tw

AVR system so that the controlled AVR system has a better control performance than other methods. The rest of this paper is organized as follows: The proposed RGA and PSO algorithm are described in Section II. The optimal PID controller design by the proposed RGA and PSO algorithm for AVR system is described in Section III. Some MATLAB simulation results and some comparison results are shown in Section IV. Finally, some conclusions are made in Section V.

2. Real-Valued GA and PSO Algorithm

In this paper, a real-valued GA (RGA) and a PSO algorithm are proposed to choose an appropriate control parameter set $K = (k_p, k_d, k_i)$ of the PID controller. A defined fitness function will guide these two algorithms to find an appropriate control parameter set to meet the desired objective. They are described as follows:

In the RGA, the control parameter set $K = (k_p, k_d, k_i)$ is viewed as an individual and each parameter value is coded by a real number [21,22]. If there are L individuals in a generation, the procedure of the proposed RGA can be described by the following steps:

Step 1: Initialize RGA by setting the number of individuals (L), the number of generations (N), the crossover probability (P_c), and the mutation probability (P_m). The i-th individual of the population with L individuals in the g-th generation is denoted by

$$K^i(g) = (K_1^i(g), K_2^i(g), K_3^i(g)) \tag{1}$$

where the number of parameters in the parameter set is 3 and $K_j^i(g)$, $i \in \{1, 2, \ldots, L\}$, $j \in \{1, 2, 3\}$, $g \in \{1, 2, \ldots, N\}$ is the j-th parameter of the i-th individual in the g-th generation. Note that $K_j^i(g)$ is a real number in the real-valued GA.

Step 2: Set $g = 1$ for the first generation and randomly generate the initial population with L individuals $pop(1) = \{K^1(1), K^2(1), \ldots, K^L(1)\}$ by

$$\begin{aligned} &K_j^i(1) = K_j^{\min} + (K_j^{\max} - K_j^{\min}) \cdot rand(), \\ &i = 1, 2, \ldots, L,\ j = 1, 2, 3 \end{aligned} \tag{2}$$

where the searching range of the parameter K_j is $[K_j^{\min}, K_j^{\max}]$ (i.e., $K_j \in [K_j^{\min}, K_j^{\max}]$) and $rand(\,)$ is an uniformly distributed random number in $[0,1]$.

Step 3: Calculate the fitness value of each individual in the g-th generation by

$$f^i = fit(K^i(g)),\ i = 1, 2, \ldots, L \tag{3}$$

where $fit(.)$ is the fitness function.

Step 4: Find an index q of the individual with the highest fitness value by

$$q = \arg \max_{i \in \{1,2,\ldots,L\}} f^i \tag{4}$$

and determine f^{best} and K^{best} by

$$f^{best} = f^q = \max_{i \in \{1,2,\ldots,L\}} f^i \tag{5}$$

and

$$K^{best} = K^q \tag{6}$$

where f^{best} is the highest fitness value in the current generation and K^{best} is the individual with the highest fitness value in the current generation.

Step 5: If $g > N$, then go to Step 11. Otherwise, go to Step 6.

Step 6: Reproduce each individual in the reproduction process by

$$n_i = L \cdot P_i,\ i = 1, 2, \ldots, L \tag{7}$$

where n_i is the reproduced number of the i-th individual, L is the number of individuals in a population, and P_i is the reproduce rate of the i-th individual and is determined by

$$P_i = \frac{f^i}{\sum_{i=1}^{L} f^i},\ i = 1, 2. \ldots, L \tag{8}$$

where f^i is the fitness value of i-th individual.

Step 7: Choose two individuals $K^m(g)$ and $K^n(g)$ from the current population ($m, n \in \{1, 2, \ldots, L\}$) to be the parents and generate two new individuals in the crossover process (the crossover probability P_c) by

$$\begin{cases} K_j^m(g) = K_j^m(g) + \sigma_1 (K_j^m(g) - K_j^n(g)) \\ K_j^n(g) = K_j^n(g) - \sigma_1 (K_j^m(g) - K_j^n(g)) \end{cases}, \ j = 1,2,3 \tag{9}$$

where σ_1 is an uniformly distributed random number in [0,1].

Step 8: Generate a new individual in the mutation process (the mutation probability P_m) for each individual by

$$K_j^i(g) = K_j^i(g) + (\sigma_2 - 0.5) \cdot s_j, \ j = 1,2,3 \tag{10}$$

where $s_j = (K_j^{\max} - K_j^{\min})$ is a range value for the searching range $K_j \in [K_j^{\min}, K_j^{\max}]$ of the j-th searching parameter K_j. σ_2 is an uniformly distributed random number in [0,1].

Step 9: Bound each updated parameter K_j^i in its searching range by

$$K_j^i(g) = \begin{cases} K_j^{\max} & \text{if } K_j^i(g) > K_j^{\max} \\ K_j^i(g) & \text{if } K_j^{\min} \le K_j^i(g) \le K_j^{\max} \\ K_j^{\min} & \text{if } K_j^i(g) < K_j^{\min} \end{cases}, \quad i = 1,2,\ldots,L, \ j = 1,2,3 \tag{11}$$

Step 10: Let $g = g + 1$ and go to Step 3.

Step 11: Determine the selected controller by the proposed method based on the obtained parameter set K^{best} with the best fitness f^{best}.

In the PSO algorithm, the control parameter set $K = (k_p, k_d, k_i)$ is viewed as a position $p = (p_1, p_2, p_3)$ of a particle in a 3-dimensional searching space. If there are L particles in a generation, the procedure of the proposed PSO algorithm can be described by the following steps:

Step 1: Initialize the PSO algorithm by setting the number of particles (L), the number of iterations (N), the searching range ($p_j \in [p_j^{\min}, p_j^{\max}]$), the velocity constraint ($v_j \in [v_j^{\min}, v_j^{\max}]$), $\bar{f}^1 = \bar{f}^2 = \ldots = \bar{f}^L = 0$, and $c_1 = c_2 = 2$. The i-th particle of the population with L particles in the g-th iteration is denoted by

$$p^i(g) = (p_1^i(g), p_2^i(g), p_3^i(g)) \tag{12}$$

where the number of parameters is 3, and $p_j^i(g)$, $i \in \{1, 2, \ldots, L\}, j \in \{1, 2, 3\}, g \in \{1, 2, \ldots, N\}$, is the j-th parameter of the i-th particle in the g-th iteration.

Step 2: Set g =1 for the first generation and randomly generate L particles $pop(1) = \{p^1(1), p^2(1), \ldots, p^L(1)\}$ in the initial generation by

$$p_j^i(1) = p_j^{\min} + (p_j^{\max} - p_j^{\min}) \cdot rand(), \quad i = 1,2,\ldots L, \ j = 1,2,3 \tag{13}$$

where the searching range of the parameter P_j is $[p_j^{\min}, p_j^{\max}]$ (i.e., $p_j \in [p_j^{\min}, p_j^{\max}]$) and $rand(\,)$ is an uniformly distributed random number in [0,1].

Step 3: Calculate the fitness value of each particle in the g-th generation by

$$f^i(g) = fit(p^i(g)), \ i = 1,2,\ldots,L \tag{14}$$

where $fit(.)$ is the fitness function.

Step 4: Determine $\bar{f}^i$ and $\bar{p}^i$ for each particle by

$$\bar{f}^i = \begin{cases} f^i(g), & \text{if } \bar{f}^i \le f^i(g) \\ \bar{f}^i, & \text{otherwise} \end{cases}, \ i = 1,2,\ldots,L \tag{15}$$

and

$$\bar{p}^i = \begin{cases} p^i(g), & \text{if } \bar{f}^i \le f^i(g) \\ \bar{p}^i, & \text{otherwise} \end{cases}, \ i = 1,2,\ldots,L \tag{16}$$

where $\bar{p}^i$ is the position vector of the i-th particle with the personal best fitness value $\bar{f}^i$ from the beginning to the current generation.

Step 5: Find an index q of the particle with the highest fitness by

$$q = \arg \max_{i \in \{1,2,\ldots,L\}} \bar{f}^i \tag{17}$$

and determine f^{best} and p^{gbest} by

$$f^{gbest} = \bar{f}^q = \max_{i \in \{1,2,\ldots,L\}} \bar{f}^i \tag{18}$$

and

$$p^{gbest} = \bar{p}^q \tag{19}$$

where p^{gbest} is the position vector of the particle with the global best fitness value f^{best} from the beginning to the current generation.

Step 6: If $g > N$, then go to Step 12. Otherwise, go to Step 7.

Step 7: Update the velocity vector of each particle by

$$\begin{aligned} v_j^i(g+1) &= \omega \cdot v_j^i(g) + c_1 \cdot r_1 \cdot (\bar{p}_j^i - p_j^i(g)) \\ &\quad + c_2 \cdot r_2 \cdot (p_j^{gbest} - p_j^i(g)), \\ i &= 1,2,\ldots,L,\ \ j = 1,2,3 \end{aligned} \tag{20}$$

where $v^i(g) = (v_1^i(g), v_2^i(g), v_3^i(g))$ is the current velocity vector of the i-th particle in the g-th generation. $v^i(g+1) = (v_1^i(g+1),\ v_2^i(g+1),\ v_3^i(g+1))$ is the next velocity vector of the i-th particle in the g+1-th generation. r_1 and r_2 are two uniformly distributed random numbers in [0,1]. ω is a weight value and defined by

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{N} \cdot g \tag{21}$$

where $\omega_{\max}$ and $\omega_{\min}$ are respectively a maximum value and a minimum value of ω. $\omega_{\max} = 0.9$ and $\omega_{\min} = 0.2$ are used in this paper.

Step 8: Check the velocity constraint by

$$v_j^i(g+1) = \begin{cases} v_j^{\max}, & \text{if } v_j^i(g+1) > v_j^{\max} \\ v_j^i(g+1), & \text{if } v_j^{\min} \le v_j^i(g+1) \le v_j^{\max} \\ v_j^{\min}, & \text{if } v_j^i(g+1) < v_j^{\min} \end{cases}, \quad i = 1,2,\ldots,L,\ \ j = 1,2,3 \tag{22}$$

where $v_j^{\max}$ and $v_j^{\min}$ are the maximum velocity and minimum velocity of the j-th parameter, respectively.

Step 9: Update the position vector of each particle by

$$p_j^i(g+1) = p_j^i(g) + v_j^i(g+1), \quad i = 1,2,\ldots,L,\ \ j = 1,2,3 \tag{23}$$

where $p^i(g) = (p_1^i(g), p_2^i(g), p_3^i(g))$ is the current position vector of the i-th particle in the g-th generation. $p^i(g+1) = (p_1^i(g+1), p_2^i(g+1), p_3^i(g+1))$ is the next position vector of the i-th particle in the $(g+1)$-th generation.

Step 10: Bound the updated position vector of each particle in the searching range by

$$p_j^i(g+1) = \begin{cases} p_j^{\max}, & \text{if } p_j^i(g+1) > p_j^{\max} \\ p_j^i(g+1), & \text{if } p_j^{\min} \le p_j^i(g+1) \le p_j^{\max} \\ p_j^{\min}, & \text{if } p_j^i(g+1) < p_j^{\min} \end{cases}, \quad i = 1,2,\ldots,L,\ \ j = 1,2,3 \tag{24}$$

where $p_j^{\max}$ and $p_j^{\min}$ are the maximum value and minimum value of the j-th parameter, respectively.

Step 11: Let $g = g + 1$ and go to Step 3.

Step 12: Determine the selected controller based on the obtained parameter set p^{best} with the best fitness f^{best}.

The RGA and PSO algorithm only require the information of the fitness function value of each parameter set. These two algorithms are applied to choose a good PID control parameter set for AVR system. They are described in the next section.

3. PID Controller Design for AVR System

It is an important matter for the stable electrical power service to develop the automatic voltage regulator (AVR) of the synchronous generator with a high efficiency and a fast response. Until now, the analog PID controller is generally used for the AVR because of its simplicity and low cost. However, these parameters of PID controller are not easy to tune. Gaing [17] proposed a method to search these parameters by using a particle swarm optimization (PSO) algorithm. The AVR system model controlled by the PID controller can be expressed by Figure 1. where v_s is the output voltage of sensor model, v_e is the error voltage between the v_s and reference input voltage v_{ref} (S), v_R is an amplify voltage by amplifier model, v_F is a output voltage by exciter model, and v_t is a output voltage by generator. There are five models: (a) PID Controller Model, (b) Amplifier Model, (c) Exciter Model, (d) Generator Model, and (e) Sensor Model. Their transfer functions are described as follows:

(a) PID Controller Model

The transfer function of PID controller is

$$G_c(s) = k_p + k_d s + \frac{k_i}{s} \tag{25}$$

where k_p, k_d, and k_i are the proportion coefficient, differential coefficient, and integral coefficient, respectively.

(b) Amplifier Model

The transfer function of amplifier model is

$$\frac{V_R(s)}{V_e(s)} = \frac{K_A}{1+\tau_A s} \tag{26}$$

where K_A is a gain and τ_A is a time constant.

(c) Exciter Model

The transfer function of exciter model is

$$\frac{V_F(s)}{V_R(s)} = \frac{K_E}{1+\tau_E s} \tag{27}$$

where K_E is a gain and τ_E is a time constant.

(d) Generator Model

The transfer function of generator model is

$$\frac{V_t(s)}{V_F(s)} = \frac{K_G}{1+\tau_G s} \tag{28}$$

where K_G is a gain and τ_G is a time constant.

(e) Sensor Model

The transfer function of sensor model is

$$\frac{V_S(s)}{V_t(s)} = \frac{K_R}{1+\tau_R s} \tag{29}$$

where K_R is a gain and τ_R is a time constant.

In this paper, the GA and PSO algorithm are applied to search a best PID parameters so that the controlled system has a good control performance. In [17], a perfor-

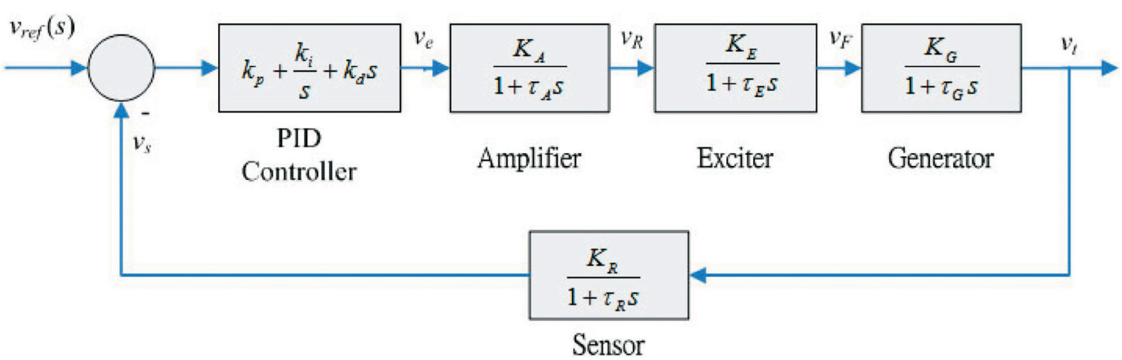


Figure 1. Closed-loop block diagram of AVR system.

mance criterion $W(K)$ is defined by

$$\min_{K:stabilizing} W(K) = (1-e^{-\beta})(M_p + e_{ss}) + e^{-\beta}(t_s - t_r) \quad (30)$$

where $K = (k_p, k_d, k_i)$ is a parameter set of PID controller, β is a weighting factor, M_p, t_r, t_s, and e_{ss} are respectively the overshoot, rise time, settling time, and steady-state error of the performance criteria in the time domain. M_p and e_{ss} are positive values. Moreover, the fitness function is proposed by Gaing [17] and described by

$$f_G(K) = \frac{1}{W(K)} \quad (31)$$

When this fitness function f_G is used in an over-damping system, as shown in Figure 2, the rise time will be too long and the settling time will approach the rise time. It will cause the value of $W(K)$ is too small and a wrong parameter set may be selected by the optimal algorithm. In order to overcome this defect, a modified

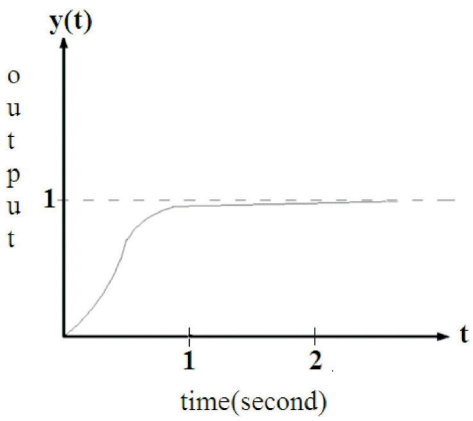


Figure 2. Time response of an over-damping system.

fitness function is defined by

$$f(K) = \frac{1}{W(K) \times ITAE} \quad (32)$$

where $W(K)$ is the performance criterion described by Equation (30) and $ITAE$ is an integral of time multiplied by absolute-error value and it is defined by

$$ITAE = \sum_{i=0}^{end\_time} t_i \left| e_i \right| \quad (33)$$

where $i \in \{0, 1, 2, \ldots, \text{end time}\}$ is an index, t_i is the i-th sampling time, and e_i is the absolute-error value in the i-th sampling time. Equation (32) can increase the efficiency and accuracy of the intelligent algorithm to search a high-quality solution.

4. Simulation Results

To verify the efficiency of the proposed fitness function in the RGA and PSO algorithm, a practical high-order AVR system [17] as shown in Figure 3 is tested. The AVR system has the following parameters. The lower and upper bounds of the three control parameters are shown in Table 1.

The following parameters are used for the real-valued GA (RGA): the population size $L = 50$, the maximum generation number $N = 100$, the maximum iteration number is 50, crossover rate $P_c = 0.9$, mutation rate $P_m = 0.01$. The following parameters are used for the PSO algorithm: the particle swarm population size $L = 50$, maximum generation number $N = 100$, the maximum iteration number is 50, $c_1 = c_2 = 2$. Each parameter set (indi-

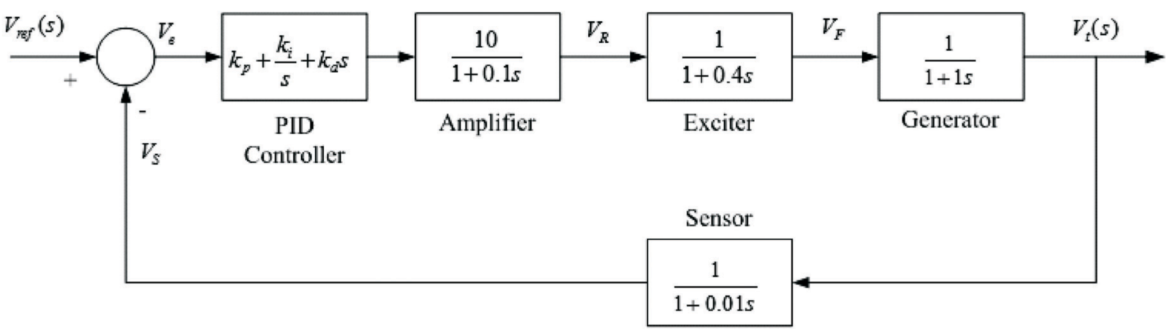


Figure 3. A practical high-order AVR system controlled by a PID controller.

vidual) of the PID controller selected by the optimal algorithm is $K = (k_p, k_d, k_i)$. The RGA and PSO algorithm are used to search an optimal parameter set of the PID controller. The searching range of each parameter is described in Table 1. The simulation time is 2 second and the sampling time is 0.01 second. In the simulation step, one iteration has 100 generations. We run each algorithm with 50 iterations and find the best fitness value, best control parameters that selected by these two algorithms in each iteration. The PID control parameter set selected by the RGA and PSO algorithm in 50 iterations for the AVR system based on the fitness function defined by Equation (31) and Equation (32) with β = 1 are respectively described in Table 2 and Table 3. Comparison of the control performance of the AVR system controlled by different controllers described in Table 2 (β = 1 and $f_G = \frac{1}{W(K)}$) and Table 3 (β = 1 and $f = \frac{1}{W(K) \times ITAE}$) are respectively described in Table 4 and Table 5. The best evaluation value in each iteration for the RGA and PSO algorithm (β = 1 and $f_G = \frac{1}{W(K)}$) and the best output response in 50 iteration are shown in Figure 4. In this paper, the performance evaluation criteria of two controllers PID^{RGA} and PID^{PSO} selected by RGA and PSO

Table 1. Searching range of each parameter

Parameter	Minimal Value	Maximal Value
k_p	0.0001	1.5000
k_d	0.0001	1.0000
k_i	0.0001	1.0000
v_{K_p}	-0.7500	0.7500
v_{K_d}	-0.5000	0.5000
v_{K_i}	-0.5000	0.5000

Table 2. Best control parameters selected by two optimal algorithms in 50 iterations for the AVR system with β = 1 and $f_G = \frac{1}{W(K)}$

Controller	Selected Control Parameters		
	k_p	k_d	k_i
PID^{GA}[17]	0.7722	0.3196	0.7201
PID^{PSO}[17]	0.6751	0.2630	0.5980
PID^{RGA}	0.0222	0.2451	0.2913
PID^{PSO}	0.0001	0.4226	0.3965

Table 3. Best control parameters selected by two optimal algorithms in 50 iterations for the AVR system with β = 1 and $f = \frac{1}{W(K) \times ITAE}$

Controller	Selected Control Parameters		
	k_p	k_d	k_i
PID^{RGA}	0.6311	0.2125	0.4615
PID^{PSO}	0.6443	0.2423	0.4700

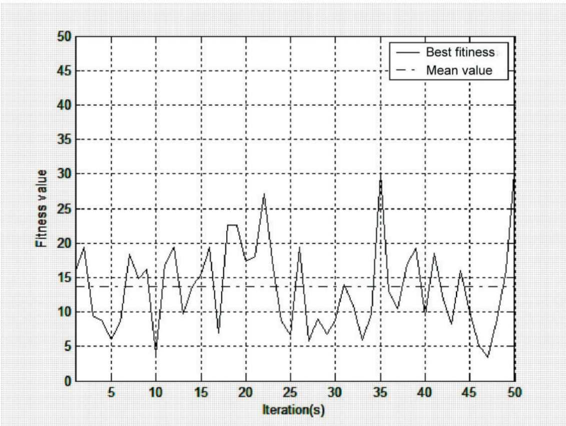
Table 4. Comparison of the control performance of the AVR system controlled by different controllers described in Table 2 (β = 1 and $f_G = \frac{1}{W(K)}$)

Controller	Number of Generation	t_r	t_s	M_P (%)	e_{ss}	$W(K)$	f_G
PID^{GA}[17]	100	0.2138	0.8645	4.54	0	0.9002	1.1109
PID^{PSO}[17]	100	0.2648	0.3795	1.71	0	0.6851	1.4596
PID^{RGA}	100	1.9200	2.0100	0	0	0.0331	30.2031
PID^{PSO}	100	1.9400	2.0100	0	0	0.0258	38.8326

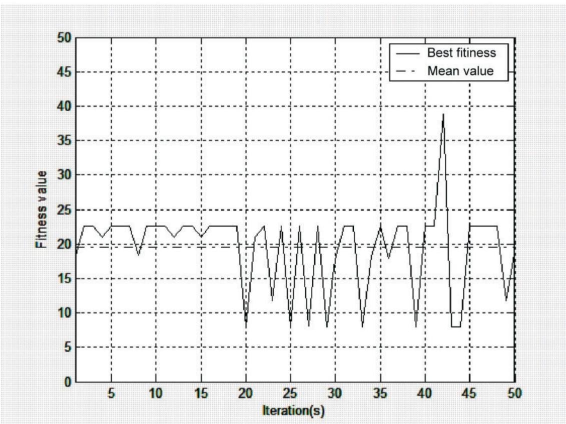
Table 5. Comparison of the control performance of the AVR system controlled by different controllers described in Table 3 (β = 1 and $f = \frac{1}{W(K) \times ITAE}$)

Controller	Number of Generation	t_r	t_s	M_P (%)	e_{ss}	$ITAE$	$W(K)$	f
PID^{RGA}	100	0.3100	0.4300	1.41	0	5.0159	0.0530	3.7583
PID^{PSO}	100	0.2800	0.4000	0	0	4.2040	0.0441	5.3882

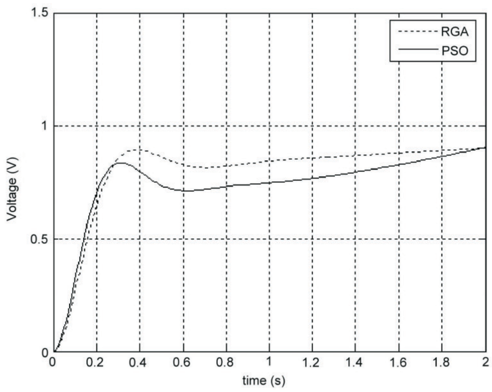
algorithm in the time domain are executed on a Pentium computer. We can find that the fitness values of PID^{RGA} and PID^{PSO} obtained by the proposed algorithms based on the fitness function defined by Equation (31) are better than that of PID^{GA} and PID^{PSO} obtained in [17], but the control performance of PID^{RGA} and PID^{PSO} are not



(a)

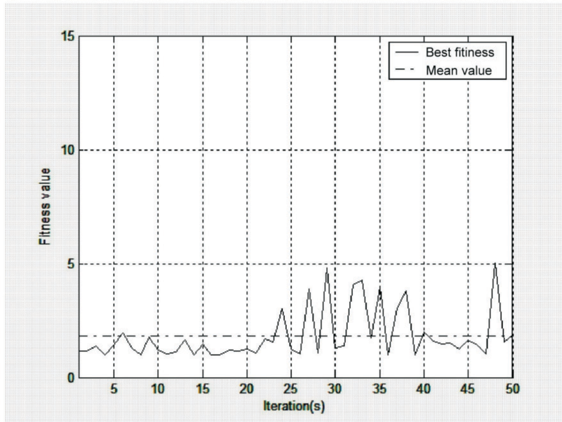


(b)

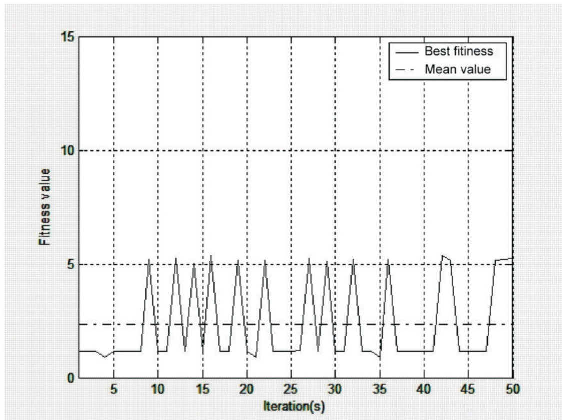


(c)

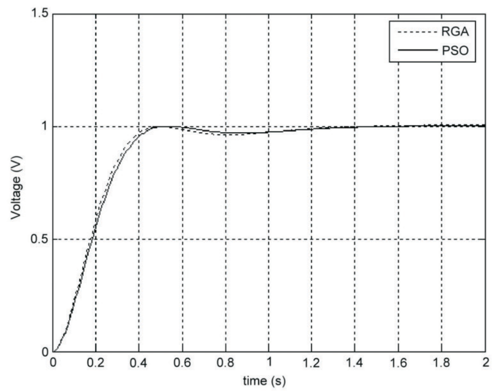
Figure 4. The best evaluation value in each iteration for β = 1 and $f_G = \frac{1}{W(K)}$. (a) RGA, (b) PSO algorithm, (c) Best output response in 50 iterations of two algorithms.



(a)



(b)



(c)

Figure 5. The best evaluation value in each iteration for β = 1 and $f = \frac{1}{W(K) \times ITAE}$. (a) RGA, (b) PSO algorithm, (c) Best output response in 50 iterations of two algorithms.

better than that of PID^{GA} and PID^{PSO} obtained in [17]. This is because the rise time is too large and closed to the settling time. Therefore, the fitness function defined by Equation (31) lets the RGA and PSO algorithm easy to search a bad parameter set with β = 1. The best evaluation value in each iteration for the RGA and PSO algorithm (β = 1 and $f = \frac{1}{W(K) \times ITAE}$) and the best output response in 50 iteration are shown in Figure 5. From the simulation and comparison results, we can find that the new fitness function defined by Equation (32) helps the RGA and PSO algorithm to search a best solution more accurate than the fitness function defined by Equation (31). The PID control parameter set selected by the RGA and PSO algorithm in 50 iterations for the AVR system based on the fitness function defined by Equation (31) and Equation (32) with β = 1.5 are respectively described in Table 6 and Table 7. Comparison of the control performance of the AVR system controlled by different controllers described in Table 6 (β = 1.5 and $f_G = \frac{1}{W(K)}$) and Table 7 (β = 1.5 and $f = \frac{1}{W(K) \times ITAE}$) are respectively described in Table 8 and Table 9. The best evaluation value in each iteration for the RGA and PSO algorithm (β = 1.5 and $f_G = \frac{1}{W(K)}$) and the best output response in 50 iteration are shown in Figure 6. The best evaluation value in each iteration for the RGA and PSO algorithm (β = 1.5 and $f = \frac{1}{W(K) \times ITAE}$) and the best output response in 50 iteration are shown in Figure 7. From the simulation and comparison results, we can see

Table 6. Best control parameters selected by two optimal algorithms in 50 iterations for the AVR system with β = 1.5 and $f_G = \frac{1}{W(K)}$

Controller	Selected Control Parameters		
	k_p	k_d	k_i
PID^{GA}[17]	0.8372	0.3927	0.6973
PID^{PSO}[17]	0.6477	0.2375	0.5128
PID^{RGA}	0.6324	0.2464	0.5047
PID^{PSO}	0.6455	0.2458	0.4513

Table 7. Best control parameters selected by two optimal algorithms in 50 iterations for the AVR system with β = 1.5 and $f = \frac{1}{W(K) \times ITAE}$

Controller	Selected Control Parameters		
	k_p	k_d	k_i
PID^{RGA}	0.6193	0.2228	0.4589
PID^{PSO}	0.6300	0.2276	0.4538

Table 8. Comparison of the control performance of the AVR system controlled by different controllers described in Table 6 (β = 1.5 and $f_G = \frac{1}{W(K)}$)

Controller	Number of Generation	t_r	t_s	M_P (%)	e_{ss}	$W(K)$	f_G
PID^{GA} [17]	100	0.1859	0.9396	6.17	0	0.9930	1.0071
PID^{PSO} [17]	100	0.2860	0.4168	0.92	0	0.8132	1.2297
PID^{RGA}	100	0.2800	0.4000	0	0	0.0268	37.3474
PID^{PSO}	100	0.2800	0.4000	0	0	0.0268	37.3474

Table 9. Comparison of the control performance of the AVR system controlled by different controllers described in Table 7 (β = 1.5 and $f = \frac{1}{W(K) \times ITAE}$)

Controller	Number of Generation	t_r	t_s	M_P (%)	e_{ss}	$ITAE$	$W(K)$	f
PID^{RGA}	100	0.3000	0.4300	0	0	4.1343	0.0290	8.3386
PID^{PSO}	100	0.3000	0.4200	0	0	4.1812	0.0268	8.9322

that the new fitness function can let the RGA and PSO algorithm find a high-quality PID control parameter set effectively so that the controlled AVR system has a better control performance than the other methods.

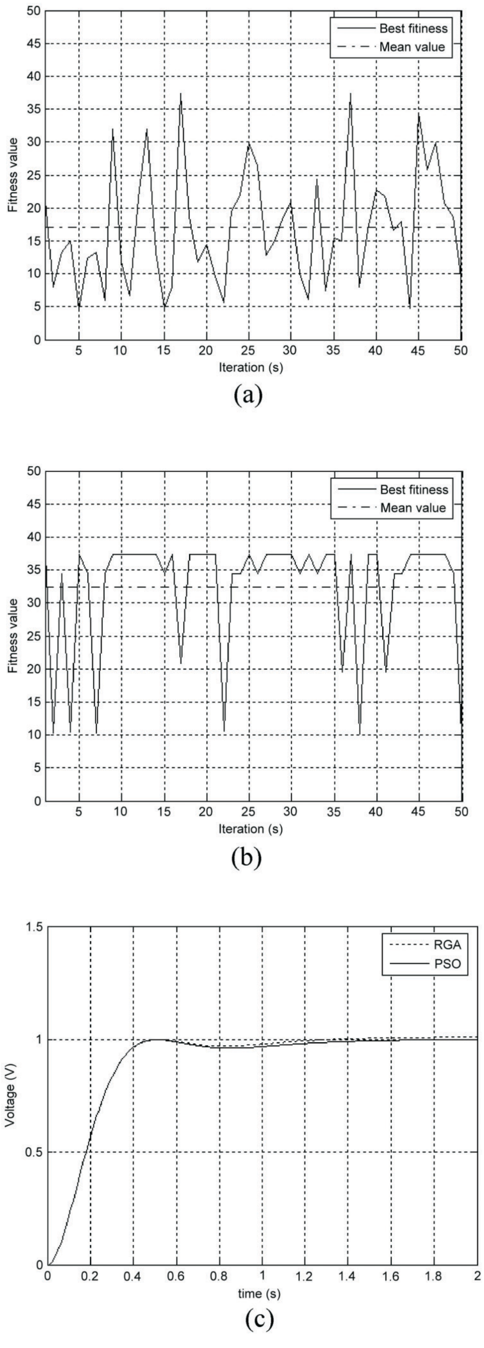


Figure 6. The best evaluation value in each iteration for β = 1.5 and $f_G = \frac{1}{W(K)}$. (a) RGA, (b) PSO algorithm, (c) Best output response in 50 iterations of two algorithms.

5. Conclusion

In this paper, a RGA and a PSO algorithm with a new fitness function are proposed to find a better PID control

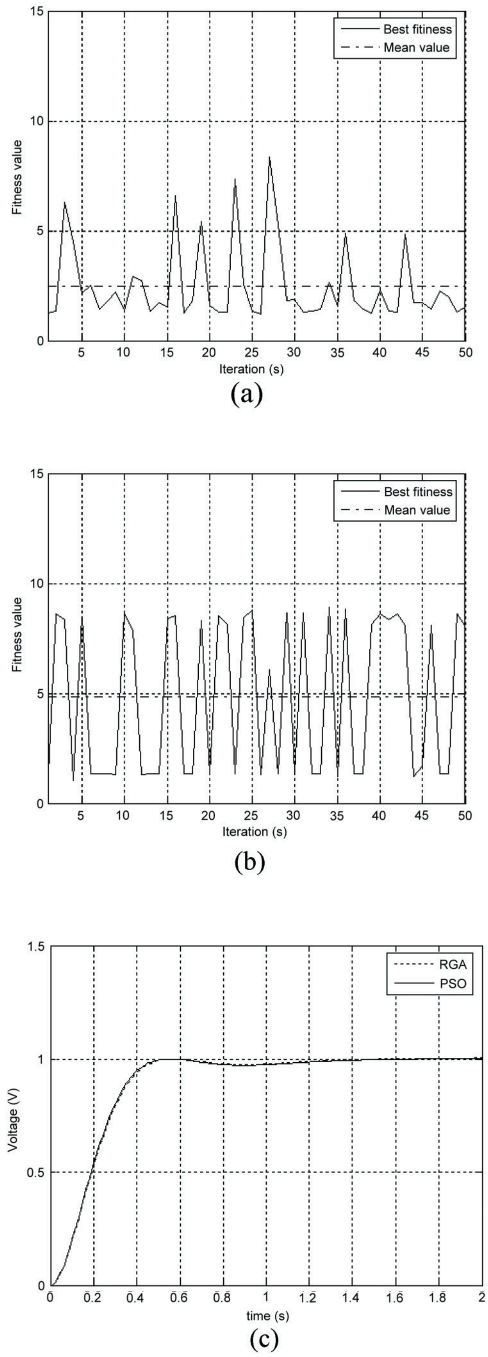


Figure 7. The best evaluation value in each iteration for β = 1.5 and $f = \frac{1}{W(K) \times ITAE}$. (a) RGA, (b) PSO algorithm, (c) Best output response in 50 iterations of two algorithms.

parameter set for AVR system. From the simulation and comparison results, we can see that the proposed fitness function can let the RGA and PSO algorithm find a high-quality PID control parameter set effectively so that the controlled AVR system has a better control performance than the other methods. Moreover, some results illustrate that a good fitness function can help the optimal algorithms to find a high-quality solution effectively.

Acknowledgement

This research was supported in part by the National Science Council of the Republic of China under contract NSC 95-2221-E-032-057-MY3.

References

[1] Chent, S., Istepaniant, R. H., Whidbornet, J. F. and Wu, J., "Adaptive Simulated Annealing for Designing Finite-Precision PID Controller Structures," *IEE Colloquium on Optimisation in Control: Methods and Applications,* pp. 1–3 (1998).

[2] Kwok, D. P. and Sheng, F., "Genetic Algorithm and Simulated Annealing for Optimal Robot Arm PID Control," *IEEE Conference on Evolutionary Computation,* pp. 707–712 (1994).

[3] Mitsukura, Y., Yamamoto, T. and Kaneda, M., "A Genetic Tuning Algorithm of PID Parameters," *Inference on Systems, Man, and Cybernetics,* Vol. 1, pp. 923–928 (1997).

[4] Krohling, R. A., Jaschek, H. and Rey, J. P., "Designing PI/PID Controller for a Motion Control System Based on Genetic Algorithm," *12th IEEE International Symposium on Intelligent Control,* July, pp. 125–130 (1997).

[5] Davis, L., *Handbook of Genetic Algorithms,* Van Nostrand Reinhold, New York (1991).

[6] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addison-Wesley, Reading, Mass (1989).

[7] Wong, C. C. and Chen, C. C., "A GA-Based Method for Constructing Fuzzy Systems Directly from Numerical Data," *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 30, pp. 904–911 (2000).

[8] Wong, C.,C., Lin, B. C. and Chen, C. C., "Fuzzy System Design by a GA-Based Method for Data Classification," *Cybernetics and Systems: An International Journal,* Vol. 33, pp. 253–270 (2002).

[9] Wong, C. C., Lin, B. C., Lee, S. A. and Tsai, C. H., "GA-Based Fuzzy System Design in FPGA for an Omni-Directional Mobile Robot," *Journal of Intelligent & Robotic Systems,* Vol. 44, pp. 327–347 (2005).

[10] Kennedy, J. and Eberhart, R., "Particle Swarm Optimization," *IEEE International Conference on Neural Networks,* pp. 1942–1948 (1995).

[11] Shi, Y. and Eberhart, R., "A Modified Particle Swarm Optimizer," *IEEE Congress on Evolutionary Computation,* May, pp. 69–73 (1998).

[12] Angeline, P. J., "Using Selection to Improve Particle Swarm Optimization," *IEEE Congress on Evolutionary Computation,* May, pp. 84–89 (1998).

[13] Shi, Y. and Eberhart, R. C., "Empirical Study of Particle Swarm Optimization," *IEEE Congress on Evolutionary Computation,* July, pp. 1945–1950 (1999).

[14] Clerc, M. and Kennedy, J., "The Particle Swarm – Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transactions on Evolutionary Computation,* Vol. 6, pp. 58–73 (2002).

[15] Parsopoulous, K. E. and Vrahatis, M. N., "On the Computation of All Global Minimizers through Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation,* Vol. 8, pp. 211–224 (2004).

[16] Yoshida, H., Kawata, K., Fukuyama, Y. and Nakanishi, Y., "A Particle Swarm Optimization for Reactive Power and Voltage Control Considering Voltage Security Assessment," *IEEE Transactions on Power Systems,* Vol. 15, pp. 1232–1239 (2000).

[17] Gaing, Z. L., "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System," *IEEE Transactions on Energy Conversion,* Vol. 19, pp. 384–394 (2004).

[18] Liu, Y., Zhang, J. and Wang, S., "Optimization Design Based on PSO Algorithm for PID Controller," *5th World Congress on Intelligent Control and Automation,* Vol. 3, pp. 2419–2422 (2004).

[19] Eberhart, R. C. and Shi, Y., "Comparison between Genetic Algorithms and Particle Swarm Optimization," *IEEE Congress on Evolutionary Computation,* May,

pp. 611–616 (1998).

[20] Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C. and Wang, L. M., "An Improved GA and a Novel PSO-GA-Based Hybrid Algorithm," *Information Processing Letters,* Vol. 93, pp. 255–261 (2005).

[21] Kawabe, T. and Tagami, T., "A Real Coded Genetic Algorithm for Matrix Inequality Design Approach of Robust PID Controller with Two Degrees of Freedom," *12th IEEE International Symposium on Intelligent Control,* July, pp. 119–124 (1997).

[22] Su, M. C. and Chang, H. T., "A Real-Valued GA-Based Approach to Extracting Fuzzy Rules for System Identification," *International Joint Conference of CFSA/IFIS/SOFT' 95 on Fuzzy Theory and Applications,* pp. 41–46 (1995).

Manuscript Received: Nov. 6, 2007

Accepted: Aug. 1, 2008