Hybrid strategy for parameter estimation and PID tuning

Ling Wang*, Da-Zhong Zheng, De-Xian Huang

Department of Automation, Tsinghua University, Beijing 10084, P. R. China

Abstract

Parameter estimation and PID tuning are two crucial issues in control engineering. Classical methods either require some prior information or depend on some rules, especially they are short of generality and their performances are not satisfied in many engineering fields. Although genetic algorithm and simulated annealing approaches have gained much attention and applications during the past decades, it may cause the premature convergence of genetic algorithm and prohibitive time-consumption required for simulated annealing if executing them alone. In this paper, reasonably combining the parallel structure of genetic algorithm with the controllable jumping property of simulated annealing, a class of effective and general hybrid optimization strategy is proposed for parameter estimation and PID tuning. The proposed strategy is easy to be understood and implemented, and only a little pre-needed information is required. Numerical simulation results demonstrate that the hybrid strategy is of effectiveness, robustness on initial states, and adaptability on models or plants, and comparisons show that the hybrid strategy can achieve performances greatly better than those of pure genetic algorithm and classical methods.

Keywords: Hybrid strategy; parameter estimation; PID tuning

1. Introduction

Parameter estimation and PID tuning are two crucial issues in control engineering, which are of important theoretical value and engineering significance and have been widely studied so far. Traditional estimation methods, such as Least Square Method (LSM) and their generalizations, gradient estimation algorithms, and maximum likelihood algorithms require some prior information and model structure, e.g. time-delay, order etc, which greatly limit their applications, especially in the field of nonlinear systems. Moreover, most classic and improved methods are intrinsically dependent on the gradient information of the error index so as to be prone to be trapped in local optima. It is known that the control performances of PID are completely dependent on PID parameters, but classical tuning methods, such as Ziegler-Nichols method, Cohen-Coon etc (Astrom and Hagglund, 1995), are based on experiments and strongly depend on the plant model and the tuning results are not satisfied in many fields, which leads to the limitation of their applications. In the past decade, genetic algorithm (GA) gained much attention (Michalewicz, 1994) and was widely applied in many engineering fields, including control engineering (Szczerbicka and Becker, 1998). Versek, Urbancic and Filipic (1993) developed a three-stage framework based on genetic algorithms for learning control. Lima and Ruano (2000) proposed neural network models of tuning criteria together with the use of GA to achieve PID autotuning. Li and Shieh (2000) designed a GA-based fuzzy PID controller for non-minimum phase systems. Teo and Marzuki (1999) presented a neuro-fuzzy controller based on neural network with all the parameters tuned by GA. Kristinsson and Dumont (1992) used GA for model identification and then used the model parameters in a certainty equivalence control law based on pole-placement method. Lennon and Passino (1999) used a different method for fitness evaluation and employed a model reference approach. Wu and Yu (2000) proposed a GA based learning algorithm for the identification of a class of fuzzy models. But there still existed two main drawbacks when using pure GA alone, that is, difficult to determine operating parameters and pre-mature convergence.

In this paper, GA and Simulated Annealing (SA) are reasonably combined to construct an effective hybrid strategy (HS) for parameter estimation and PID tuning, which utilizes the population parallel search structure of GA and the controllably probabilistic jumping of SA. With some operations specific designed, the HS can be applied to various kinds of models and controlled plants. The computation procedure is simple and easy to understand and accomplish, and only a little pre-needed information is required. Numerical simulation results based on some classical problems demonstrate that the HS is of effectiveness, robustness on initial states, and adaptability on models or plants, and it can achieve better performances than classical approaches.

The organization of remain contents is as follows. Firstly, the problems to be studied are described, secondly the HS is proposed, then the strategy is implemented in detail for parameter estimation and PID tuning, and some numerical simulations and comparisons are carried out, lastly we end with some conclusions.

2. Problem statement

Generally, the representation style of the estimated model is known in advance, and it is supposed that

* Corresponding author. Email: wangling@mail.tsinghua.edu.cn. This paper was not presented at any IFAC meeting.

the system output can be measured and the ratio of signal/noise should be large enough, as well as the parameters to be estimated should be specified. Usually, a system model can be generally described as follows.

$$y(t) = f(r, \theta) \quad (1)$$

where, $y(t)$ is the system output, r is the system input, $\theta = (\theta_1, \theta_2, \cdots, \theta_k)$ is the parameters to be estimated, f is the model representation which can be expressed by transfer function, state space or ARMA model etc.

Parameter estimation means to obtain the estimated parameters using certain algorithm according to certain error index based on the model output and actual sampling data $y_0(t), t = 1,2,\cdots,n$, with certain model input. In this paper, hybrid strategy will be proposed to estimate the model parameters. The principle can be illustrated by Fig. 1, which will be explained in detail later.

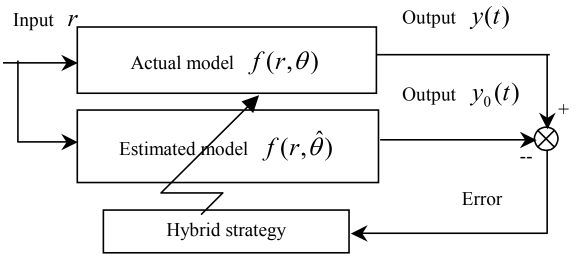

Fig. 1. Illustration of model parameter estimation

PID controllers are well known to engineers for their simple structure, easy implementation, good performances and strong robustness, so that they are widely used in various fields of industry, especially in chemical process industry. More than 90% of the controllers used in real applications are of the PID types. Generally, the formula of conventional PID and its discretized formula can be written as follow.

$$u(t) = K_p[e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d \frac{de(t)}{dt}] \quad (2)$$

$$\Delta u(k) = K_p\{\Delta e(k) + \frac{T_0}{T_i}e(k) + \frac{T_d}{T_0}[\Delta e(k) - \Delta e(k-1)]\} \quad (3)$$

where K_p, T_i and T_d are proportional (P), integral (I) and derivative (D) parameters respectively, T_0 is sampling period, $e(t)$ and $u(t)$ are error variable and plant input respectively. Let $K_i = K_p T_0 / T_i$ and $K_d = K_p T_d / T_0$, then the formula (3) can be rewritten as follows.

$$\Delta u(k) = K_p \Delta e(k) + K_i e(k) + K_d[\Delta e(k) - \Delta e(k-1)] \quad (4)$$

PID tuning means to determine the above three parameters by certain algorithm to achieve the optimal control performances. In this paper, PID will be tuned by hybrid strategy, whose principle can be illustrated by Fig. 2 and interpreted later.

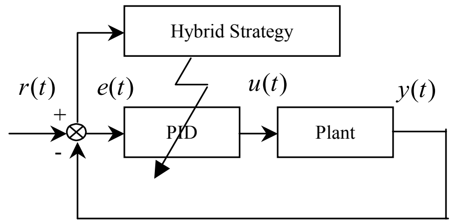

Fig. 2. Illustration of PID tuning

Intrinsically, both parameter estimation and PID tuning are the problems to search optimal parameters according to certain objective functions, which can be regarded as complex functional optimization problems with high dimensions and many local minima. To solve the problems effectively and achieve good performances, in next section a class of HS by combining genetic algorithm and simulated annealing will be proposed.

3. Hybrid optimization strategy

Based on the mechanics of natural selection and genetics, genetic algorithm combines the concept of *survival of the fittest* among solutions with a structured yet randomized information exchange and offspring creation. GA is naturally parallel and is able to exhibit *implicit parallelism*, which does not evaluate and improve a single solution but analyses and modifies a set of solutions simultaneously. There are three basic operators in pure GA, i.e. selection, crossover and mutation. The ability of GA, i.e., operating on many solutions simultaneously and gathering information from all current solutions to direct search, reduces the possibility of being trapped in a local optimum.

Originated from the similarity between statistical mechanics and combinatorial optimization, simulated annealing provides a framework for optimization of properties of very large complex system and can be viewed as an enhanced version of local optimization or iterative improvement algorithms (Kirkpatrick et al, 1983). SA attempts to avoid entrapment in a local optimum by sometimes accepting a move that deteriorates the value of the objective function. With the help of the distribution scheme, SA can provide a reasonable control over the initial temperature and cooling schedule so that it performs effective exploration of solution space and good confidence in the solution quality.

GA is a highly parallel procedure, which contains certain redundancy and historical information of past solutions. However, GA may lose solutions and substructures due to the disruptive effects of genetic operators. In addition, it is not easy to regulate GA's convergence and tune global parameters. Consequently, GA is easy to be premature and results in poor solution (Leung et al, 1997). On the other hand, SA maintains only one solution at a time, whenever they accept a new solution, the old one must be discarded, which often causes low search efficiency. But, SA has the ability to escape from local optima that can be controlled by cooling schedule (Hajek, 1988). Reasonably combining these two approaches from mechanism to structure, it will develop novel hybrid strategy (HS) with more powerful search efficiency. So, utilizing the parallel searching framework of GA and incorporating SA to avoid individual being trapped in local minima with controllable probability, an efficient HS is proposed as follows.

Step1: initialize population, and determine the initial temperature t_0, and set $k = 0$.

Step2: if stop criterion has been satisfied, then output the results; else go on below steps.

Step3: implement selection and crossover operators.

Step4: implement mutation operator.

Step 5: perform simulated annealing for each individual i in parallel mode, then back to Step 2:

Step 5.1: if equilibrium condition has been reached, then decrease temperature $t_{k+1} = update(t_k)$ and set $k = k+1$, and go to step2; otherwise go to step 5.2.

Step 5.2: generate a neighbor solution j from solution i randomly and calculate the difference of the objective values $\Delta c_{ij} = c_j - c_i$;

Step 5.3: if $\min\{1, \exp(-\Delta c_{ij} / t_k)\} > random[0,1]$, then let $i = j$, and update the best solution found so far if possible; else keep the old solution.

It can be seen that during the hybrid search process GA provides a set of initial solutions for SA at each temperature to perform Metropolis sample for each solution until equilibrium condition is reached, and GA uses the solutions found by SA to continue parallel evolution. Temperature is adjusted to control the behavior of SA, i.e., at a high temperature, SA performs a "course" search with high escaping probability from current solution; while at a low temperature, SA performs a "fine" search among the neighbor solutions of current solution. In addition, the optimization operators, such as mutation operator and the new solution generator of SA, can be different or hybrid used to yield a large neighborhood and efficiently explore better solutions among the solution space. Theoretically, Wang and Zheng (1998) analyzed the convergence behavior of such HS and a sufficient condition for global convergence was provided, and the job-shop scheduling was solved (Wang and Zheng, 2001).

Moreover, such HS reserves the generality of GA, SA and can be easily implemented and applied to any optimization problems by suitably modifying the encoding scheme, optimization operators, algorithm criteria and parameters. In next sections, the HS will be designed in detail for parameter estimation and PID tuning.

4. Implementation of the strategy

Encoding scheme: The real value encoding is used, i.e., all the parameters are specified by real values except that the model order is encoded by an integer.

Objective function or fitness function: The Integral of Time multiplied by Absolute Error (ITAE) index, i.e., $\int_{\tau}^{\infty} t|e(t)|dt$ is employed as objective function in PID tuning, which is able to restrain the overshoot and settling-time to certain extent. While $f = 1/[\sqrt{\sum_t (y(t) - y_0(t))^T (y(t) - y_0(t))} + 0.01]$ is used as fitness function in parameter estimation, where $y_0(t)$ is the actual output and $y(t)$ is the output of the estimated system.

Initialization of population and temperature: After generating initial population with P_{size} individuals randomly at the beginning of the procedure, the best and worst individuals with the objective index c_{best} and c_{worst} are determined. Then, at the initial temperature t_0 the probability to accept the worst individual with respect to the best individual is set as $p_r \in (0,1)$, i.e., $p_r = \exp[-(c_{worst} - c_{best}) / t_0]$. Hence, t_0 can be determined by $t_0 = -(c_{worst} - c_{best}) / \ln(p_r)$. Obviously, such process can be implemented easily and the relative performance of the initial population is used, so such method is of handleability and reasonability.

Selection: In parameter estimation, classical proportional selection based on fitness value is applied, i.e., individual i would be selected with probability $f_i / \sum f_j$, where f_i is the fitness value of i. While in PID tuning, rank-based selection is applied, i.e. all the individuals are arranged with decently order according to the objective value firstly, then the kth individual would be selected with probability $2k / [P_{size}(1 + P_{size})]$.

Crossover: Based on real-value-encoding scheme, crossover operator is designed as Equation 5 to generate two new individuals after selection operator. And such procedure is repeated $P_{size}/2$ times (P_{size} is population size) to generate the new population. Then, the top P_{size} solutions with better objective values from the old population and new solutions are reserved for the next optimization operator.

$$\begin{cases} w_1' = a \cdot w_1 + (1-a) \cdot w_2 \\ w_2' = a \cdot w_2 + (1-a) \cdot w_1 \end{cases} \tag{5}$$

where $\alpha \in (0,1)$ is a random variable, w_1 and w_2 are parents, w_1' and w_2' are children.

Mutation, SA state generator: Due to the merit of HS, here mutation rate is set to one to perform a "fine" local neighbor search and all these operators can be conducted by appending random noise for each parameter.

$$w' = w + \eta \cdot \xi \quad (6)$$

where ξ is a random variable subjected to Gaussian distribution N(0,1), η is a scale parameter.

Moreover, during the evolution process the best solution found so far should be updated if possible to avoid the lost of good solution, i.e., "elitist" scheme.

Annealing function: Exponential cooling schedule is used to adjust the temperature, i.e. $t_k = \lambda \cdot t_{k-1}$, where $\lambda \in (0,1)$ is decrease rate.

Equilibrium condition and stop criterion: since theoretical convergence conditions may lead to huge computation and are not practicable, two approximate conditions are simply designed to provide a rather good compromise between quality and search efficiency. The Metropolis sample process is set to L_1 iterations fixed, and if the best solution found so far keeps fixed at L_2 consecutive temperatures, the algorithm will stop.

5. Numerical simulation

5.1 Simulations on parameter estimation

Based on three kinds of models (Jiang and Wang, 2000) described as follows, the performances of the HS are tested and some comparisons with simple GA are carried out.

Model 1 (Transfer function of 2-ordered system with time-delay): parameters to be estimated are k, T_1, T_2 and time-delay τ.

$$y(s)/u(s) = ke^{-\tau s}/(T_1 s^2 + T_2 s + 1) \quad (7)$$

Model 2 (Nonlinear state space model): parameters to be estimated are θ_1, θ_2, θ_3 and θ_4.

$$\begin{cases} \begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} \theta_1 x_1(t) x_2(t) \\ \theta_2 x_1^2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ u(t) \end{bmatrix} \\ y(t) = \theta_3 x_2(t) - \theta_4 x_1^2(t) \\ x_1(0) = x_2(0) = 1 \\ t = 0,1,\ldots,50 \end{cases} \quad (8)$$

Model 3 (Hammerstein Model): parameters to be estimated are a_1, a_2, b_0, b_1 and d.

$$\begin{cases} A(q^{-1}) y(k) = q^{-d} B(q^{-1}) \phi[u(k)] \\ A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} \\ B(q^{-1}) = b_0 + b_1 q^{-1} \\ \phi[u] = \begin{cases} \sqrt{u+1/2} - \sqrt{1/2} & 5 \ge u \ge -1/2 \\ -\sqrt{|u+1/2|} - \sqrt{1/2} & -5 \le u < -1/2 \end{cases} \end{cases} \quad (9)$$

Combining Matlab and C++ simulation environment, and setting P_{size}=20, p_r=0.1, λ=0.85, η=0.1, L_1=30, L_2=20, sampling time T_0=0.1, numerical simulations are carried out on PIII/550 PC and the average results of 20 random simulation are summarized as follows.

Table 1. Estimation results of Model 1

Parameter	k	T_1	T_2	τ
Actual	1	1	2	1
Estimated	1	1	1.9997	1

Table 2. Estimation results of Model 2

Parameter	θ_1	θ_2	θ_3	θ_4
Actual	0.5	0.3	1.8	0.9
Estimated	0.5069	0.3048	1.8095	0.9077
Results of Huang and Wang (1996)	0.4916	0.3014	1.8432	0.9267

Table 3. Estimation results of Model 3

Parameter	a_1	a_2	b_0	b_1	d
Actual	-1.5	0.7	1	0.5	2
Estimated	-1.5004	0.6984	0.9861	0.4516	2
Results of Huang and Wang (1996)	-1.4982	0.6970	1.3654	-0.0371	2

The results demonstrate the effectiveness of the HS, which is competent for the models with different styles and properties, including nonlinear systems. In addition, in the Hammerstein model there are some parameters may affect the system output much less than others, e.g. b_1, which are hard to estimate. The estimation process of pure GA would lead to these parameters far deviate from the true value, but by incorporating SA into GA the hybrid strategy can achieve good results. The system output of actual model and the estimated one are shown on Fig. 3, from which it can be seen that the two curves are so adjacent each other. Compared with the results of Jiang and Wang (2000), the results from hybrid strategy are much better than those from pure GA. The reason is that the hybrid strategy takes the advantages of both GA and SA and improves the potential of global optimization.

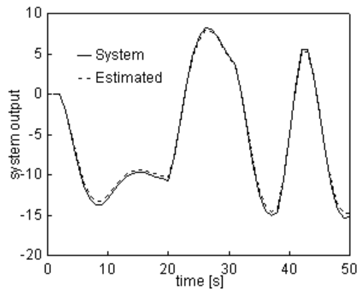

Fig. 3. Output of the actual and estimated Model 3

5.2 Simulations on PID tuning

Firstly based on the controlled plant $e^{-0.5s}/(2s+1)$, the statistical performances of PID controller tuned by hybrid strategy (HS) are investigated, and some comparisons with GA and Z-N methods are carried out. The parameter η is set to 0.6 and the others are the same as before. The statistical results of 20 random simulations are shown in Table 4 (comparatively, the ITAE object value by

Z-N method is 12.6951). The closed loop step output response using PID tuned by the three methods are shown in Fig. 4, and the decreasing curves of objective value are shown in Fig. 5.

Table 4. Average performances gained by the HS and GA

Algorithm	Average ITAE	ITAE Variance	Average overshoot	Average generation
HS	4.1625	1.5974	1.2%	39.80
GA	7.3840	2.3001	18.5%	49.15

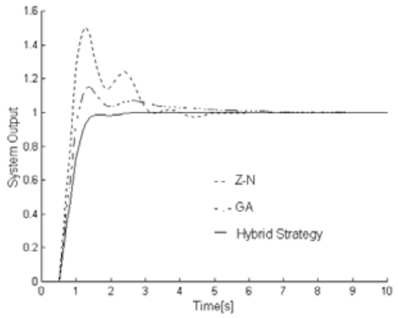

Fig. 4. The closed loop step output response

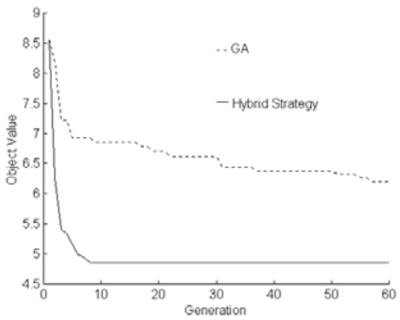

Fig. 5. Decent curves of objective value

To further test the tuning performances of the hybrid strategy, we study the plants as follows. The average performances of 20 random simulations are shown in Table 5, and the corresponding closed loop step output response curves are shown in Fig. 6~10.

$$G_1(s) = 1/(s+1)^2 \quad (10)$$

$$G_2(s) = 1/[s^2 + 2\xi s + 1], \xi = 0.2 \text{ } or \text{ } 0.8 \quad (11)$$

$$G_3(s) = 1/(1+s)^3 \quad (12)$$

$$G_4(s) = e^{-0.1s} /[(1+s)(1+0.5s)(1+0.25s)(1+0.125s)] \quad (13)$$

Table 5. Average control performances of PID

Plant	Overshoot(%)		Average ITAE	
	GA	HS	GA	HS
$G_1(s)$	26.1	4.8	6.83	3.17
$G_2(s), \tau = 0.2$	15.1	3.5	5.18	3.13
$G_2(s), \tau = 0.8$	6.2	2.4	4.86	2.77
$G_3(s)$	23	0	21.29	10.92
$G_4(s)$	20.1	4.3	10.15	6.92

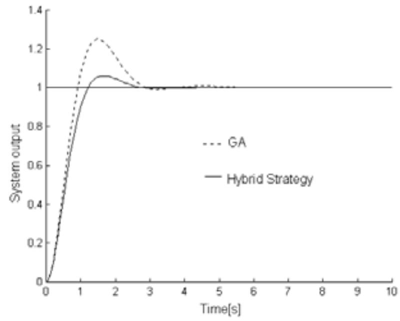

Fig. 6. The closed loop step output response of $G_1(s)$

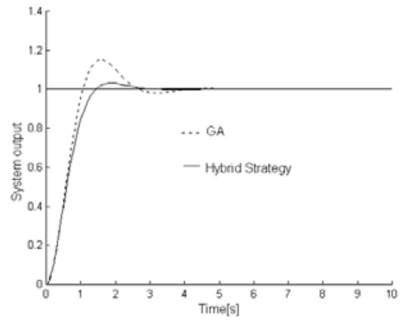

Fig. 7. The closed loop step output response of $G_2(s), \xi = 0.2$

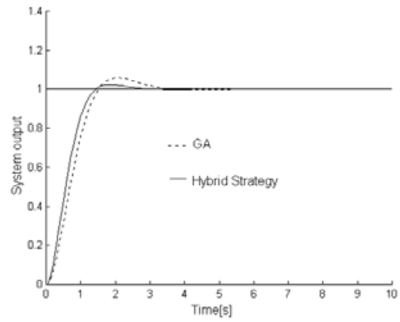

Fig. 8. The closed loop step output response of $G_2(s), \xi = 0.8$

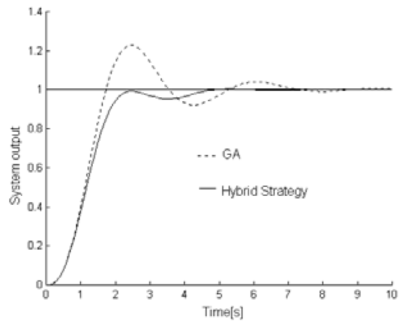

Fig. 9. The closed loop step output response of $G_3(s)$

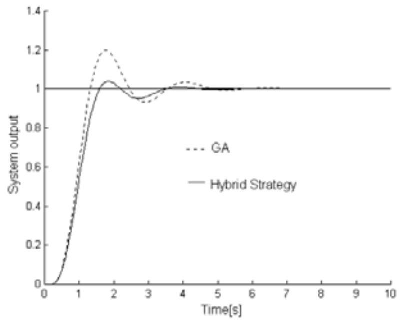

Fig. 10. The closed loop step output response of $G_4(s)$

Lastly, considering the water turbines plant $\frac{1}{1+0.2s}\cdot\frac{1-0.8s}{1+0.4s}\cdot\frac{1}{0.2+0.96s}$ with a non-minimum phase zero and taking η=0.1, T_0=0.04, the closed loop step output response curves using PID tuned by HS, GA and the simplex method (Liu and Mao, 1997) are show in Fig. 11.

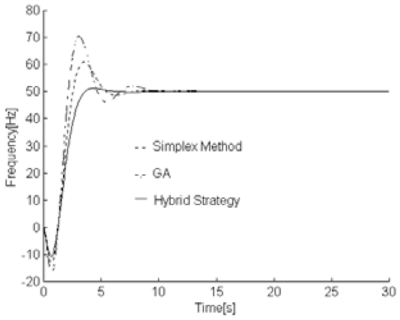

Fig. 11. The closed loop step output response

From the simulation results, it can be concluded that the HS can achieve good optimization performances, such as high quality, rapid speed and robustness on initial values. Secondly the controllers tuned by the HS can achieve better control performances than those of the controllers tuned by GA and Z-N methods, in particular, the overshoot, settling-time and error index are very small. In addition, the HS is independent of plant and control objective. So the HS is well fit for PID tuning.

In brief, the features of the proposed method can be summarized as followed. (1) The computation of parameters is easy and simple. (2) The estimation and tuning procedure is easy to understand and accomplish. (3) Only a little pre-needed information is required. (4) Hybrid global search can achieve very satisfied and better performances than traditional or pure GA methods. (5) The method is general and has a wide range of applications.

6. Conclusion

This paper proposed an effective hybrid strategy by combining SA and GA for parameter estimation and PID tuning. Numerical simulation results demonstrated the effectiveness, robustness on initial states, and adaptability on models or plants. The comparisons showed that the HS could achieve performances greatly better than those obtained by pure GA and traditional methods. The future work is to apply the proposed HS or combining fuzzy logic and neural networks in time-varying systems, some advanced controllers, such as fuzzy controller and neural network controllers, and online estimation and tuning, especially in actual industry environments.

Acknowledgements

This paper was supported by the NSFC under the Grant 60204008, and 973 Program under the Grant 2002CB312200.

References

Astrom, K., Hagglund, T. (1995). *PID controllers: theory, design and tuning*. New York: ISA.

Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13(2): 311-329.

Huang, J., Wu, Y. 1996. On-line system identification based on genetic algorithms. *Information and Control* 25(3): 171-176.

Jiang, B., B. Wang. (2000). Parameter estimation of nonlinear system based on genetic algorithms (in Chinese). *Control Theory and Application*, 17(1): 150-153.

Kirkpatrick, S., Gelatt, C., Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220(4598): 671-680.

Kristinsson, K., Dumont, G. (1992). System identification and control using genetic algorithms. *IEEE Trans. Systems, Man and Cybernetics*, 22(5): 1033-1046.

Lennon, W. K., Passino, K. M. (1999). Genetic adaptive identification and control. *Engineering Applications of Artificial Intelligence*, 12: 185-200.

Leung, Y., Gao, Y., Xu, Z. B. (1997). Degree of population diversity- a perspective on premature convergence in genetic algorithms and its Markov-chain analysis. *IEEE Trans. Neural Networks*, 8(5): 1165-1176.

Li, T. S., Shieh, M. (2000). Design of a GA-based fuzzy PID controller for non-minimum phase systems. *Fuzzy Sets and Systems*, 111: 183-197.

Lima, J., Ruano, A. E. (2000). Neuro-genetic PID autotuning: time invariant case. *Mathematics and Computers in Simulation*, 51: 287-300.

Liu, L., Mao, Z. (1997). Water turbines PID controller based on genetic algorithm. *Automation of Electric Power Systems*, 21(12): 41-43.

Michalewicz, Z. (1994). *Genetic algorithms + data structures = evolution programs*. Berlin: Springer-Verlag.

Szczerbicka, H., Becker, M. (1998). Genetic algorithms: a tool for modeling, simulation, and optimization of complex systems. *Cybernetics and Systems*, 29: 639-659.

Teo, L., Marzuki, B. (1999). Tuning of a neuro-fuzzy controller by genetic algorithm. *IEEE Trans. Systems, Man, and Cybernetics*, 29(2): 226-236.

Varsek, A., Urbancic, T., Filipic, B. (1993). Genetic algorithms in controller design and tuning. *IEEE Trans. Systems, Man and Cybernetics*, 23(5): 1330-1339.

Voda, A. A., Landau, I D. (1995). A method for the auto-calibration of PID controllers. *Automatic*, 31(1): 41-53.

Wang, L., Zheng, D. Z. (1998). Study on a class of GASA hybrid strategy and its convergence behavior. *Control and Decision*, 13(6): 672-675.

Wang, L., Zheng, D. Z. (2001). An effective hybrid strategy for job-shop scheduling problems. *Computers and Operations Research*, 28(6): 595-596.

Wu, B., Yu, X. (2000). Fuzzy modeling and identification with genetic algorithm based learning. *Fuzzy Sets and Systems*, 113: 351-365.