ON-LINE TEMPERATURE CONTROL OF AN OVEN BASED ON GENETIC ALGORITHMS

Z. Y. Yang[1, 2], C. W. Chan[1], M. S. Xue[2], G. J. Luo[2]

[1] *Dept. of Mech. Eng., Univ. of Hong Kong, Pokfulam Road, Hong Kong*
[2] *Dept. of Automation, Univ. of Sci. & Tech. of China, Hefei, P. R. C.*

Abstract: The genetic algorithms are powerful for optimization, and have been successfully applied to controller design. However, most existing works are based on simulations and little works are available in the literature on on-line control applications. In this paper, a special feature in selecting the population of the genes is proposed, such that the genetic based algorithms can be modified for on-line applications. The modified algorithms are applied successfully to control on-line the temperature of an electric oven. Better performance results than the PID and Dynamic Matrix Control are obtained, illustrating the modified genetic algorithms are suitable for on-line applications.

Keywords: genetic algorithms, on-line control, least-squares method, PID control, predictive control

1. INTRODUCTION

Genetic algorithm (GA) was first proposed by John Holland (1975) for solving optimization problems by imitating the evolution process. It is shown to be a robust and global search method, and has been used widely to design controllers for systems that are difficult to analyse and solve analytically (Fleming and Purshouse, 2002). The general procedure is that the controller parameters are encoded and integrated as a GA chromosome, and then adjusted genetically to achieve satisfactory results.

To obtain better results, it is often required that both controller structure and its parameters are optimized simultaneously, which is difficult to achieve by GA. Two approaches: the Hierarchical GA (HGA) (Man *et al*., 1997) and the Genetic Programming (GP) (Koza, 1992) are proposed to tackle this problem. The main difference between HGA and GA is that in HGA, the structure of its chromosome is organised hierarchically. As some genes dominate others in the same chromosome, it is suitable to optimize simultaneously both the structure and the parameters of the controller (Ke, *et al*., 1998; Man, *et al*., 1997; Tang, *et al*., 2000; Zhou, *et al*., 2002).

However, as fixed-length representation is still used in HGA, the search is restricted in a bounded area. By employing the tree-like program representation in GP approach, this limitation is overcome. Since a controller or any other function can be expressed in a more general form by a tree, there is little limitation on its structure, as long the syntax is satisfied. Following this approach, the controller structure and parameters can be obtained simultaneously (Koza, *et al*. 1999; Yu, *et al*. 2000).

Although the GA has been used successfully in controller design, most applications are based on simulated experiments using off-line optimizations. Very few on-line applications involving real processes are available in the literature. There are two main difficulties in applying methods derived from the GA in practice. First, genetic approaches require a large amount of computation, making on-line application difficult (Fleming and Purshouse, 2002). Second, the system model used in genetic approaches to guide the searching is very difficult to establish precisely in practice. As the models based on GA are obtained off-line, any drifting in the model parameters cannot be readily compensated.

In this paper, on-line control based on the GA for the temperature control of a laboratory-scale oven is discussed. Three conventional genetic algorithms: GA, HGA and GP are considered. The GA is used to tune the parameters of PID controller, and is referred

to as GA-PID, while HGA and GP are used to simultaneously optimize both the controller structure and the parameters. To guarantee efficient searching, a new population formation method with special adjustments is proposed in the evolution process. It is shown that better results are obtained from genetic approaches with special adjustments than the PID and Dynamic Matrix Control (DMC).

The paper is organized as follows. In Section 2, a brief review of GA, HGA and GP for controller design are presented. In Section 3, the laboratory-scale oven is presented. Special adjustments in the evolution process for on-line applications are also presented. Experimental results, and comparison of these control algorithms are given in Section 4, and the conclusion is presented in Section 5.

2. GENETIC APPROACHES FOR ON-LINE CONTROLLER DESIGN

2.1 System framework

The traditional methodology is followed to construct the framework of the on-line control system based on genetic approaches. It consists of two main parts, one for modeling and another for searching controller. As shown in Fig. 1, in every sample-control period, the real-time model is identified by an on-line identification method, for example the Recurrent Least-squares method used in this paper, and updated if it is convergent and physically reasonable. Then a genetic approach is called to design a controller for the new model.

Among numerous genetic approaches (Fleming and Purshouse, 2002), three typical ones that represent different levels are employed to design controller separately. In GA-PID, GA that belongs to the parameter-optimization level is used to tune the parameters of PID controller. Then HGA is employed to determine the configuration among some predefined optional controller blocks and optimize their corresponding parameters. Since HGA can optimize the controller structure in a limited space, it is in the semi-structure-optimization level. At last, GP is used to automatically create a controller, both the structure and parameters. Undoubtedly, it belongs to the structure-optimization level.

2.2 Genetic algorithm PID

It is well known that, PID control is the most popular technique in industrial process systems. Many optimization algorithms have been used to tune its parameters for better performance, including GA (Porter and Jones, 1992; Vlachos *et al*, 1999; Wang *et al*, 2002).

In this paper, the discrete PID controller is used:

$$\Delta u(k) = K_P \Delta e(k) + K_I T_S e(k) + \frac{K_D}{T_S} \Delta(\Delta e(k)) \quad (1)$$

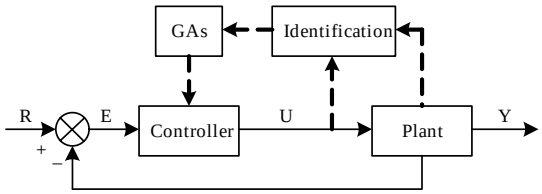

Fig. 1. System framework

where, T_S is the sample-control period, and K_P, K_I and K_D are parameters, which are tuned by GA with float number representation.

Given a candidate controller, the coming control and response sequences can be predicted with on-line model from historical data. In this paper, the quadratic objective is proposed as,

$$Obj = \sum_{l=1+\tau}^{M+\tau} e^2(l) + \alpha \sum_{m=1}^{M} \Delta u^2(m) \quad (2)$$

where, τ, M and α stand for pure delay, prediction horizon and weight. Easily, the fitness function for maximization can be obtained:

$$Fitness = 1/Obj \quad (3)$$

The objective and fitness function above are also used in latter approaches.

2.3 Hierarchical genetic algorithm

As described previously, HGA uses different genetic structure of the chromosome. From the biological knowledge, the genes are arranged in a hierarchical manner. There are two types of genes, control genes and parametric genes. The control genes determine the parametric genes that should be utilized during the evolution process.

The following discrete controllers are used,

$$\begin{aligned} C(z^{-1}) &= \frac{\Delta u(k)}{e(k)} = \frac{M(z^{-1})}{N(z^{-1})} \\ &= g \frac{(z^{-1} + b_1)(z^{-2} + b_2 z^{-1} + b_3)}{(z^{-1} + a_1)(z^{-2} + a_2 z^{-1} + a_3)} \end{aligned} \quad (4)$$

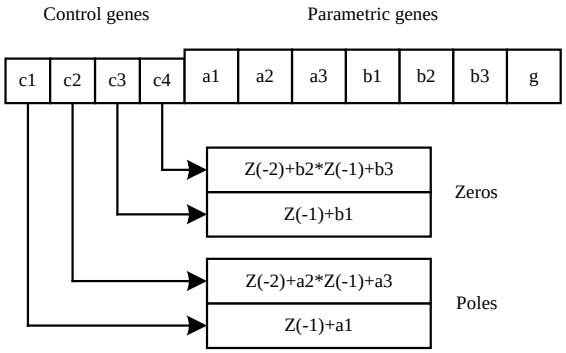

Fig. 2. The structure of the HGA chromosome

It contains a gain coefficient and four blocks, which can be represented by the HGA chromosome shown in Fig. 2. In this representation, symbols “c1” to “c4” are control genes, which determine the activity of corresponding blocks. Symbols “a1” to “a3”, “b1” to “b3” are parametric genes, whose activities are

determined by the corresponding control genes. And "g" is a parametric gene too, but it is always active. The stability of the controller is also guaranteed by limiting the parameters in a feasible domain. For example, the controller:

$$C(z^{-1}) = 2\frac{z^{-1}+1.5}{z^{-2}+4z^{-1}+4}$$

can be represented as the HGA gene string, {0,1,1,0, *,4,4,1.5,*,*,2}, where "0", "1" and symbol "*" stand for inactivity, activity and the value of inactive parametric gene.

Although chromosomes are interpreted hierarchically, both kinds of genes are integrated in the same chromosome from the view of genotype. As they can evolve simultaneously, the structure and the parameters of the controller can be optimized at the same time. Furthermore, its genetic operators are the same as those of GA and do not require any extra modifications. However, the genetic operations that affect the high-level genes can result in changes within the active genes that eventually lead to multiple changes in the lower level genes (Man, *et al*., 1997).

Although HGA is more flexible, its searching space is still bounded. It also limits the solution from HGA to be within its boundaries, even they are much better. Fortunately, this disadvantage is overcome by GP.

2.4 Genetic programming

In GP, the chromosome is represented by a program, which is often in the form of a binary-tree. In general, any function can be considered as a program and expressed by a tree, e.g., $f(x) = x^2 + 2x + 1$ can be described by a tree shown in Fig. 3. The nodes below the line are "terminal nodes", which are variables or constants, and are the input of the function. Other nodes are "functional nodes", which could be any functions. The uppermost node is the root that outputs the final value of entire function.

To accompany this representation, particular genetic operators are designed. The crossover exchanges the sub-trees of the parents, while mutation replaces the old sub-tree with a randomly created one. It's worthy to note that there is no limitation on the size of the individual as long as it satisfies the syntax. From this approach, the structure and parameters of the function can be optimized simultaneously.

If $N(z^{-1})$ is monic, (4) can be rewritten as,

$$\Delta u(k) = A(z^{-1})\Delta u(k-1) + B(z^{-1})e(k) \qquad (5)$$

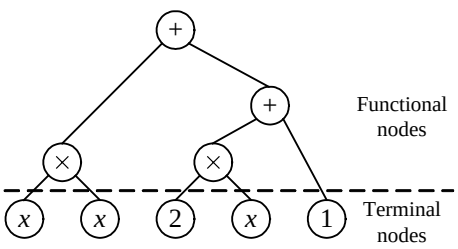

Fig. 3. The tree-like representation

where, $B(z^{-1}) = M(z^{-1})$, $A(z^{-1}) = (1 - N(z^{-1}))z$. Equation (5) can be naturally describes by the tree shown in Fig. 4.

Two parts separately stand for $A(z^{-1})\Delta u(k-1)$ and $B(z^{-1})e(k)$. The crossover is only allowed to perform on the sub-trees of the same type. Without losing generality, we can simply select the functional and the terminal node sets as $\{+, -, \times, 1/x, z^{-1}\}$ and $\{\Delta u(k-1), e(k), C\}$, where $1/x$, z^{-1} and C denote reciprocal operator, backward shift operator and constant respectively. For example, the controller used in 2.3 can be rewritten as:

$$\Delta u(k) = -\Delta u(k-1) - 0.25\Delta u(k-2) + 0.75e(k) + 0.5e(k-1)$$

and further represented as Fig. 5.

The depth-fair crossover is used to enhance searching efficiency (Kessler and Haynes, 1999). Due to the node-bloating phenomenon (Soule and Hechendorn, 2002), the individual size is restricted below 50.

3. BASIC IMPLEMENTATION ISSUES

To investigate the real-time performance of the genetic algorithms described above, they are applied to control the heating and preservation process of an electrical oven. In this section, a brief description of the oven control system is given first.

3.1 Physical features of the plant

This experimental oven has a width of 45cm, a depth of 30cm and a height of 30cm. The temperature inside the oven is measured by a platinum resistance thermometer PT100, which is converted to an electrical signal of 1 to 5V. This output voltage is digitalized and the control is computed on a 233MHz PC. The control computed from the PC is a 4-20mA current signal, which is used to switch on or off the two electrical heaters by Solid State Relay (SSR). The control configuration is shown in Fig. 6.

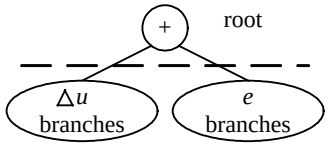

Fig. 4. The controller described in GP form

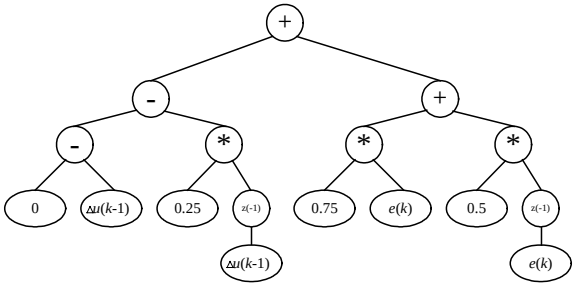

Fig. 5. The GP individual of the controller

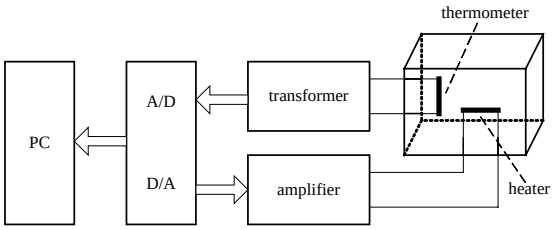

Fig. 6. The physical structure of the computer-controlled system

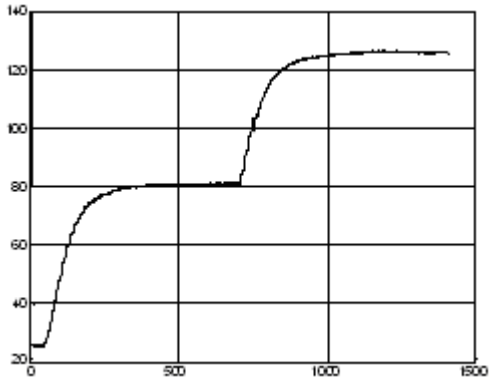

Fig. 7. The piecewise step response. X and Y axes are time and temperature, with units 10 seconds and 1 centigrade separately.

To obtain an approximation of the plant, a step input is applied to the heaters, and the temperature in the oven is shown in Fig. 7, from which the following first order model is obtained,

$$\begin{cases} G(s) = \dfrac{27.8e^{-70s}}{850s+1} & U \in [4,6]mA \quad Y \in [25,81]°C \\ G(s) = \dfrac{22.7e^{-70s}}{700s+1} & U \in [6,8]mA \quad Y \in [81,126]°C \end{cases} \tag{6}$$

Clearly, the time constant is large with a relatively short pure delay, and these parameters are dependant on the operating point.

3.2 Special adjustments for practical applications

As discussed previously, there are two problems in applying genetic algorithms for on-line applications. The first one is the computing load, and the second is the time-varying model. To overcome these problems and to obtain good on-line performance, it is necessary to adjust the evolution process such that it can utilize prior information, and simultaneously maintain the population diversity.

In engineering, it is reasonable to assume that the plant changes continuously under normal conditions. Therefore, the prior information can be utilized to make the search more efficient. On the other hand, to avoid being trapped in a local optimum when environment changes acutely and is in conflict with past information, population diversity must be maintained through out the evolution process. Therefore, a special population formation method is proposed here.

The initial population, which is created at the start of the genetic-based searching program, consists of three parts. The first one consists of the superior individuals obtained in the last time interval. The second one recodes the best individuals in the history of evolution process, and the third one is some randomly created individuals. Therefore, the population starts searching at a relatively high level if the model changes mildly. A new search in a new direction can also be launched, if the model changes acutely. Additionally, large population size, small generation number and selective pressure are used to maintain the diversity.

Further, because of the model uncertainty, over-computation cannot in general achieve better performance, but make the controller less robust. Therefore, the terminal criteria have to design specially. Once the highest fitness by now exceeds a threshold, saying 1.2 times of the highest of last time, which is used in this paper, or the pre-defined generation is exhausted, the search will stop. Fig. 8 illustrates the entire flow, and Fig. 9 gives a simulated comparison between the proposed population formation method and the traditional one in solving an on-line controller design problem for a time variant system.

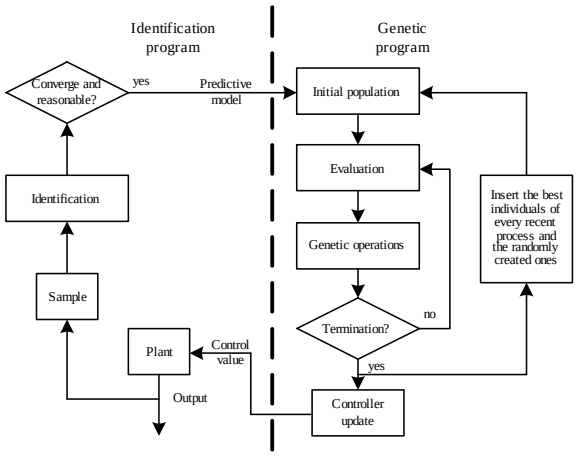

Fig. 8. The system flowchart.

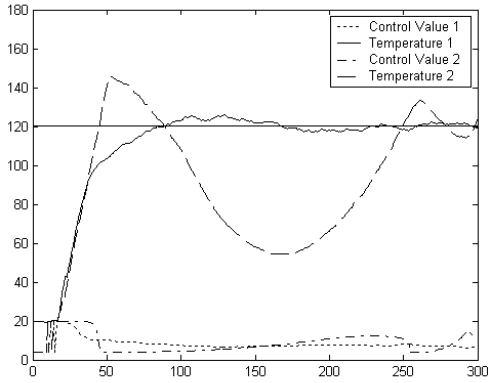

Fig. 9. A simulated comparison between: (1) the proposed population formation method, (2) the traditional one, in solving an on-line controller design problem for a time variant system.

4. EXPERIMENTAL RESULTS

In this experiment, all sampling interval is set at 10 seconds. The results obtained from the three genetic algorithms are shown in Figs. 10 to 12. Normally, all genetic approaches can finish the computation in 2 seconds. For comparison, the results of the traditional PID and DMC are shown in Figs. 13 and 14. DMC is a model predictive control algorithm, which is widely used in industrial processes because of its robustness and simplicity in implementation (Cutler and Perry, 1980). In DMC, the predictive model is obtained first from step response. Based on this model, further optimization of a quadratic function is performed to drive the output to approach a reference trajectory.

Comparing with well tuned PID and DMC, the genetic approaches can achieve better temperature control with smaller settling time, overshoot and undershoot, and zero steady error, as illustrated in Table 1. However, the control changes more frequently and with larger magnitude. The main reason is that the genetic algorithms are stochastic in nature. For this reason, it is difficult to obtain smoother control when the process is time-varying.

Table 1 Comparisons between different methods

	Overshoot	Undershoot	Settling Time (s)
GA-PID	0%	0%	650
HGA	2.2%	2.0%	730
GP	2.3%	2.0%	720
PID	3.8%	2.0%	1160
DMC	1.3%	0%	940

The following common features can be found in the genetic algorithms considered in this paper. The control can be roughly divided into four stages. At the initial stage, the genetic searching program is set to standby manually and the lower boundary of 4mA is used until the model converges and approaches to the real system from the initial settings. During the heating stage, the plant is completely controlled by genetic algorithms. And the upper boundary 20mA is set to speed up the process in all genetic approaches. When the temperature is close to the set point, the control computed from the genetic algorithms decreases rapidly. However, there are larger fluctuations in the control computed from the genetic algorithms, as the stability of genetic search is weakened. Also, all genetic algorithms can eliminate steady errors.

Furthermore, it is interesting to note that the performance of GA-PID is even better than that obtained from HGA and GP, although the latter is somehow superior to GA in the sense of optimization ability. The main reason is that the selected structure of the PID controller is extracted from the domain of expert knowledge and is therefore suitable for process control. This structure is difficult to obtain from numerical computation, and it is, therefore, important to include the domain knowledge into the evolution process to obtain better performance.

5. CONCLUSIONS

A special formation of the population with other adjustments is proposed in this paper for the on-line implementation of GA, HGA and GP. It is shown that these algorithms are implemented successfully for the on-line control of an electrical oven. Better performance than the PID and DMC is obtained from these modified genetic approaches, and in particular, the GA-PID performs much better because it is able to include the domain knowledge into the evolution process. The results indicate that genetic approaches are efficient for on-line control of the electric oven.

REFERENCES

Cutler C. R., T. Perry (1980). Dynamic matrix control: a computer control. *Proc. of the Joint Automatic Control Conference*, San Francisco, WP5-B

Fleming, P. J., R. C. Purshouse (2002). Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, **Vol. 10**, 1223-1241.

Holland, J. (1975). *Adaptation in Natural and Artificial Intelligence*. University of Michigan Press, Michigan.

Ke, J. Y., K. S. Tang, and K. F. Man, et al. (1998). Hierarchical genetic fuzzy controller for a solar power plant. *IEEE International Symposium on Industrial Electronics*, **Vol. 2**, 584-588.

Kessler, M., T. Haynes (1999). Depth-fair crossover in genetic programming. *Proc. of the 1999 ACM Symposium on Applied Computing*, 319-323.

Koza, J. (1992). *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Massachusetts.

Koza, J., M. A. Keane, and J. Yu *et al*. (1999). Automatic synthesis of both the topology and parameters for a robust controller for a non-minimal phase and a three-lag plant by means of genetic programming. *Proc. of the 38th Conference on Decision and Control*, 5292-5300.

Man, K. F., K. S. Tang and S. Kwong, *et al*. (1997). *Genetic algorithms for control and signal processing*. Springer, London.

Porter, B., & A. H. Jones (1992). Genetic tuning of digital PID controllers. *Electronics Letters*, **Vol. 28**, 843–844.

Soule, T., R. B. Hechendorn (2002). An analysis of the causes of code growth in genetic programming. *Genetic Programming and Evolvable machine*, **Vol. 3**, 283-309.

Tang, K. S., K.F. Man, and S. Kwong, *et al*. (1998). Design and optimization of IIR filter structure using hierarchical genetic algorithms. *IEEE Trans. Industrial Electronics*, **Vol. 45**, 481-487.

Tang, K. S., K. F. Man, and R. S. H. Istepanian (2000). Teleoperation controller design using hierarchical genetic algorithm. *Proc. of IEEE International Conference on Industrial Technology*, **Vol. 1**, 707-711.

Vlachos, C., D. Williams, & J. B. Gomm (1999). Genetic approach to decentralised PI controller

tuning for multivariable processes. IEE Proc., Control Theory and Applications, **Vol. 146**, 58–64.

Wang, Y. P., N. R. Watson and H. H. Chong (2002). Modified genetic algorithm approach to design of an optimal PID controller for AC–DC transmission systems. *International Journal of Electric Power & Energy Systems*, **Vol. 24**, 59-69.

Yu, J., M. A. Keane, and J. Koza. (2000). Automatic design of both topology and tuning of a common parameterized controller. Proc. *of IEEE International Symposium on Computer-Aided Control System Design*, 234-242.

Zhou, X., Y. Zhou, D. Gong (2002). Optimization of fuzzy sets of fuzzy control system based on hierarchical genetic algorithms. *IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering*, **Vol. 3**, 1463-1466.

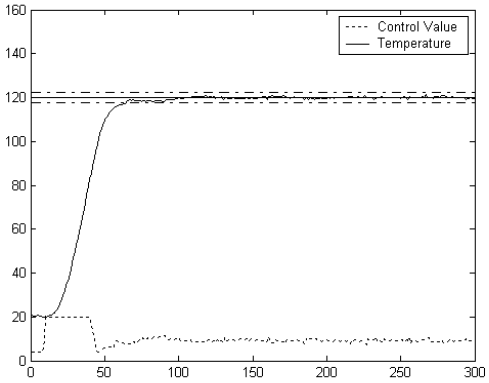

Fig. 10. The performance of GA-PID. The main settings are: float coded, population size 400, selective probability 0.7, crossover probability 1, mutation probability 0.05, generation number 50.

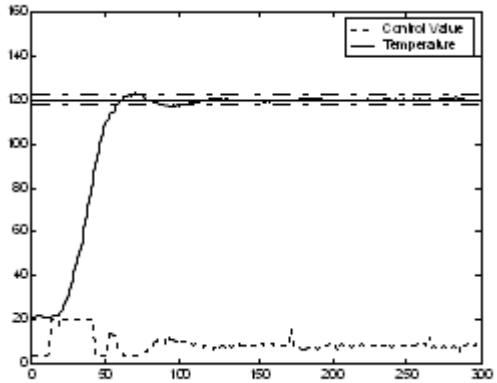

Fig. 11. The performance of HGA. The main settings are: control genes binary coded, parametric genes float coded, population size 400, selective probability 0.7, crossover probability 1, mutation probability 0.05, generation number 50.

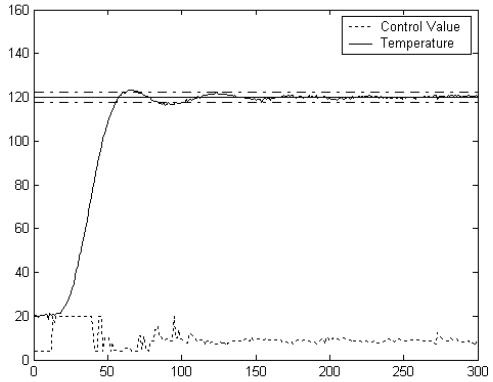

Fig. 12. The performance of GP. The main settings are: maximal individual node number 50, population size 200, depth-fair crossover, selective probability 0.7, crossover probability 1, mutation probability 0.05, generation number 50.

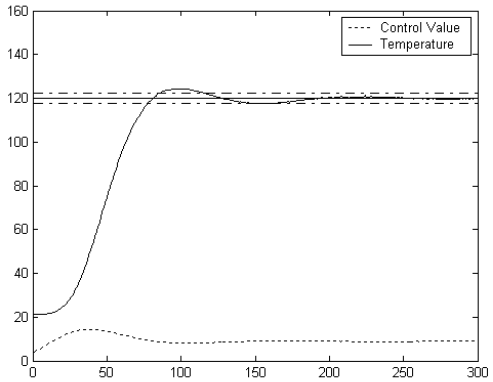

Fig. 13. The performance of PID controller in the form of formula (1), whose parameters are: $K_P = 0.33$, $K_I = 0.5$ and $K_D = 1$.

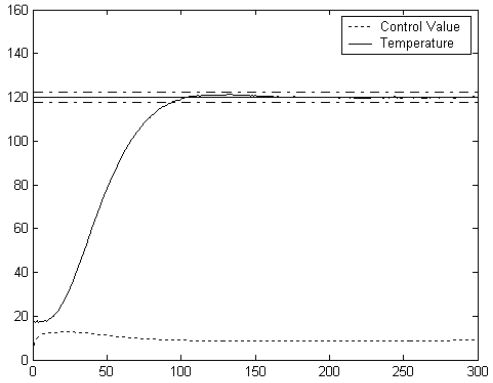

Fig. 14. The performance of DMC. Several critical parameters are: predictive horizon 15, control horizon 2, soft factor 0.975.