2009 Second International Conference on Intelligent Networks and Intelligent Systems

A Nonlinear PID Controller based on an Adaptive Genetic Algorithm

Zhou Ying, Zhang Lei, Wang Gui-li, Yang Qian
Electrical Engineering & Automation
Hebei University of Technology
Tianjin, China
E-mail: warlack@163.com

***Abstract*—the precision of temperature control is a very important factor to fermentation process. However, because the serious integral and time delay exists in the controlled object, the conventional control algorithm could hardly obtain satisfactory quality. Thus, a nonlinear PID controller is designed based on the nonlinear relationship between PID control parameters and error signals. Furthermore, an adaptive genetic algorithm is proposed in this paper in order to overcome the disadvantages in classical genetic algorithm, such as slow early convergence, local optimum, etc. The new adaptive algorithm adjusts the probabilities of crossover and mutation by construct function of the changes of fitness value. Simulation results show that the nonlinear PID controller with the GA for tuning the parameter is very effective. It has good dynamic and static response and strong robustness.**

Keywords-Adaptive genetic algorithm; Nonlinear PID control; large time delay system

I. INTRODUCTION

From 20 century 40's to now, PID (Proportional-Integral-Derivative) is used as the basic control algorithm and adopted for control systems. At present, in the modern industrial production process systems, the traditional PID is still used widely. However, in the field of process control, the conventional control algorithm could hardly obtain satisfactory quality and the control object has become more and more complex. The involved control processes have the features such as strong nonlinearity, strong coupling, uncertainty and long time-delay. In order to solve these problems, with the rapid development of expert systems technology, fuzzy logic control and neural networks, intelligent PID control strategies is proposed by combining intelligent control and PID control formed[1]. For improving the performance of control systems, a non-linear controller was designed based on the nonlinear relationship between PID control parameters and error signals by some scholars [2, 3]. However, these control strategies have limitations because they require prior knowledge about the controlled plant and need the search space of parameter optimization to be continuously differentiable.

In this paper, by analyzing the ideal relationship [4] between PID parameters and error change in transition process, we proposed a method of non-linear controller based on adaptive genetic algorithm. Comparing proportion, integral and differential parameters with error, we proposed a nonlinear function. Through establishing fitness function, adaptive genetic algorithm could adjust crossover and mutation probability timely. It overcomes the disadvantages in classical genetic algorithm, such as slow early convergence and local optimum. And then makes the algorithm have adaptability. The paper shows that the nonlinear PID controller with the GA for tuning the parameter is very effective. It has good dynamic and static response and strong robustness

II. ADAPTIVE GENETIC ALGORITHM

A. Genetic Algorithm

The basic idea of genetic algorithm is a population composed by a group of individuals. Each individual represents one of potential solutions. Every individual is evaluated and then obtained its fitness value, which responded the individual's adapt degree to environment. The main genetic operators have choice, reproduction, crossover and mutation.

The problems of premature convergence and local optimum should be considered. Concrete manifestation is: after crossover, progeny chromosomes produced by whole population can't excel father generation and become homogeneous populations, the crossover could only repeat father generation chromosome. And then the whole optimization process completely depends on mutation. Davis [5] believed: when convergence of the population completely depends on synchronization mutation, in fact, the process has come to the end. According to the defects of genetic algorithm, some improved methods have been proposed [6]. The probabilities of crossover and mutation often are set an experience value or a simple linear function in the general genetic algorithms [7, 8]. It will make the genetic algorithm in the process of evolution become the mechanical operation, can't tack changes in the operation and make adjustment. Therefore, in order to improve optimization speed of system and avoid appearing slow early convergence and local optimum. In this paper, we put forward an adaptive algorithm for regulating the probabilities of crossover and mutation.

B. Evolution Process of Adaptive Genetic Algorithm

When genetic algorithm begins to solve problems, first of all, we must determine the variables and objective function, and then code variables. Adaptive genetic algorithms transform K_p , K_i and K_d into 10-bit binary number as

DOI 10.1109/ICINIS.2009.34

parameters, and the objective function is composed by overshoot σ, rising time t_r and adjusting time t_s.

In population $P(k)$, the number of s is $m = m(s,k)$ at time k, at $k+1$, the number of s after replication is:

$$m(s,k+1) = m(s,k)N\, f(s) \Big/ \sum f_i = m(s,k)\, f(s) \Big/ \bar{f} \tag{1}$$

Where N is the size of population; $\bar{f}$ is the average fitness value of population; $f(s)$ is the average fitness value of model s at time k.

After the choice operation, a change of special model pattern is proportional to the ratio between the average fitness $f(s)$ and average fitness value of population $\bar{f}$. If the model adaptation value is greater than the average population fitness, then the number of model in the next generation will increase.

In the genetic algorithm parameters, crossover probability P_c and mutation probability P_m is the key factor influence genetic algorithm behavior and performance. The lager P_c is the faster new individual generated. On the other hand, if P_c is too big, which might damage genetic model, because individual structure with high degree of adaptability will be destroyed very quickly due to P_c. Else if P_c becomes too small, evolution of process will slow down even come to the end. Thus crossover takes a very important role in the entire algorithm. So this paper designs adaptive crossover probability P_c as:

$$P_c = P_{c1} - \frac{P_{c2}}{1 + e^{P_{c3}(f_{\max} - f(s))}} \tag{2}$$

Where $P_{c1} = 0.6$, $P_{c2} = 0.2$ and $P_{c3} = 0.05$; $f_{\max}$ is the maximum of fitness function; $f(s)$ is the average fitness value of model s at time k. P_{c3} is used to adjust change rate of P_c.

For mutation probability P_m, if too small, it isn't beneficial to produce new individuals; if too big, genetic algorithm becomes a pure random search algorithm. At the beginning of the evolution, the processes should avoid high fitness individual's rapid breeding and appearing early maturity phenomenon; at the end of the evolution, when the individual close to the optimal solution, P_m should not be too big, in order to avoid that individual should not achieve the optimal solution for a long time. So the paper designs an adaptive mutation function as:

$$P_m = P_{m0} - P_{m1} \operatorname{sec} h(P_{m2}(f_{\max} - f(s))) \tag{3}$$

Where $P_{m0} = 0.02$, $P_{m1} = 0.01$ and $P_{m2} = 0.002$. P_{m2} is used to adjust change rate of P_m.

C. *The Design of Fitness Function*

At the beginning of the evolution, in order to prevent system appeared slow early convergence and local optimum. The design of fitness function should avoid a small number of better individual affect the others. Because control purpose is that we want to control error close to zero, then the system get faster response time and smaller overshoot. So the paper designs a fitness function as follow:

$$F = 1 \big/ J \tag{4}$$

$$J = \int_0^{\infty} (x_1 |e(t)| + x_2 u^2(t)) dt + x_3 t_u \tag{5}$$

Where $e(t)$ is system error; $u(t)$ is controller output; t_u is rise time; x_1, x_2, x_3 are weight values.

In order to avoid overshoot, we use function of punishment. If there is an overshoot, the optimum index will be designed as:

$$J = \int_0^{\infty} (x_1 |e(t)| + x_2 u^2(t) + x_4 |ey(t)|) dt + x_3 t_u \tag{6}$$

Where x_3 is weight value, and $x_4 >> x_1$, $ey(t) = y(t) - y(t-1)$. $y(t)$ is controlled object output.

III. NONLINEAR PID CONTROLLER

Mathematical model of standard PID controller:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \tag{7}$$

Mathematical model of nonlinear PID controller:

$$u(t) = K_p(e_p(t)) e_p(t) + K_d(e_p(t), e_v(t)) \frac{de_p(t)}{dt} + K_i(e_p(t)) \int_0^t e_p(t) dt \tag{8}$$

Where $K_p(e_p(t))$, $K_d(e_p(t), e_v(t))$ and $K_i(e_p(t))$ are respectively represent proportional coefficient, integral coefficient and differential coefficient.

For ordinary step response, we can structure nonlinear PID controller gain parameters as follows.

Proportion gain parameter:

$$K_p(e_p(t)) = K_{p1} + K_{p2}(1 - \operatorname{sec} h(K_{p3} e_p(t))) \tag{9}$$

Differential gain parameter:

$$K_d(e_p(t)) = K_{d1} + K_{d2} \big/ (1 + K_{d3} \exp(K_{d4} e_p(t))) \tag{10}$$

Integral gain parameter:

$$K_i(e_p(t)) = K_{i1} \operatorname{sec} h(K_{i2} e_p(t)) \tag{11}$$

Form 9 to 11, $K_{p1}, \cdots, K_{p3}, K_{d1}, \cdots, K_{d4}, K_{i1}, K_{i2}$ are all positive real number, through adjusting K_{p3}, K_{d4}, K_{i2} could adjust the change rate of K_p, K_d, K_i.

During dynamic process, the gain parameters of nonlinear PID controller change with the control error. Each feature of PID controllers according to the nonlinear relationship between the parameters and error carry out structural adjustment. Therefore, anti-interference ability is better than normal linear PID.

IV. COMPUTER SIMULATION

A. *Simulation Object*

Proceed form actual engineering, the design will be applied to fermentation experimental device designed by Shanghai Jiao Tong University. So, we obtained experimental model by neural network:

$$G(s) = \frac{525}{1000000s + 1} e^{-240s}$$

B. Simulation Parameters

Sample size of adaptive genetic algorithm is 30. $x_1 = 0.999, x_2 = 0.001, x_3 = 2.0, x_4 = 100$.

TABLE I. TABLE TYPE STYLES

parameters	K_{p1}	K_{P2}	K_{p3}	K_{d1}	K_{d2}
result	0.8324	3.5013	0.552	48.8088	30.232
parameters	K_{d3}	K_{d4}	K_{i1}	K_{i2}	
result	6.502	0.510	0.000008	0.000005	

Simulation figure as follows:

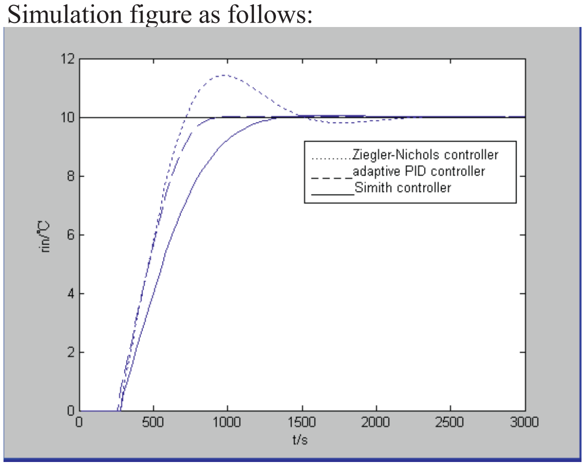

Figure 1. Step response of three control modes.

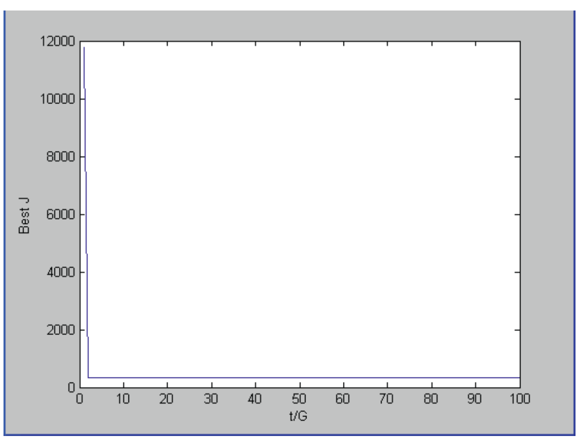

Figure 2. The optimize process of function.

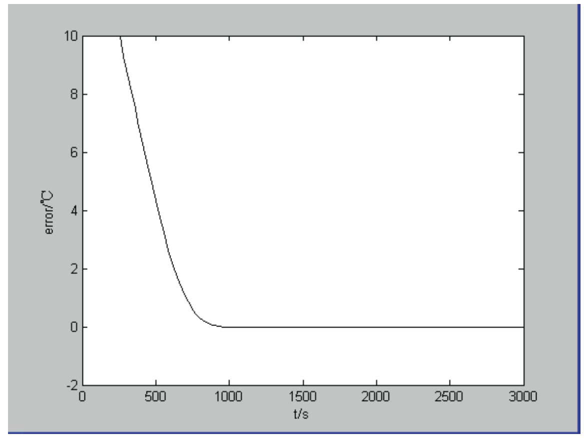

Figure 3. Error change of adaptive PID controller.

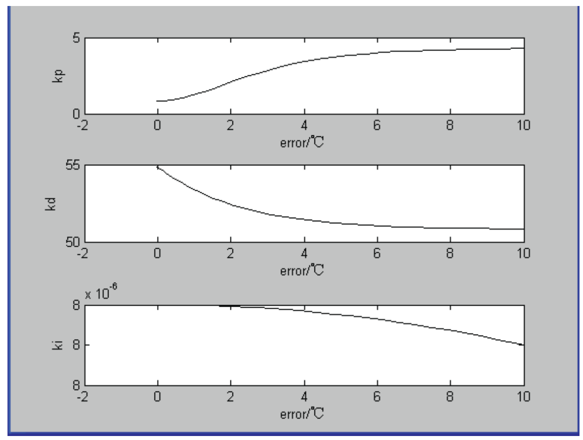

Figure 4. The parameters follow error change.

Simulation results show that a nonlinear controller combined with adaptive algorithm is superior to the other control methods obviously. Nonlinear regulator based on different time response requirement carries out the real-time adjustment. At the same time, adaptive genetic algorithm monitors dynamic errors. Change crossover and mutation probability to make parameters optimization. Therefore, a nonlinear PID controller based on adaptive genetic algorithm has obvious advantages of short response time and lower overshoot.

V. CONCLUSION

In this paper, a nonlinear PID controller which is designed bases on an adaptive genetic algorithm solves the problem of nonlinear PID parameters optimization by using global optimization of adaptive genetic algorithm. The results show that the design has preferable anti-noise ability and overcome the defects of classical genetic algorithm.

REFERENCES

[1] Subbun R, Sanderson A C and Bonissone P P. “Fuzzy logic controlled genetic algorithms versus tuned gentic algorithms: An agile manufacturing application,” Proc of the 1998 IEEE ISIC/CIRA/ISAS Joint Conf. gaitherberg. 1998, pp. 434-440.

[2] Y. X. Su, Dong Sun and B. Y. Duan, “Design of an enhanced nonlinear PID controller,” Mechatronics, 2005, 15(3), pp. 1005-1024.

[3] JIANG Shan-he, JIANG Ju-lang and WU Lei, “A Nonlinear PID Controller Based on Particle Swarm Tuning Algorithm ,” Computer Technology and Development. 2007, 17(4), pp. 71-74.

[4] He Guanghui and Tan Guanzheng, “An Optimal Nonlinear PID Controller Based on Ant Algorithm,” Programmable Controller & Factory Automation, 2007, pp. 99-105.

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

[6] LI Qing, ZHANG Wei and YIN Yi-xin, WANG Zhi-liang, “A new adaptive algorithm for regulating the probabilities of crossover and mutation,” Control and Decision. 2008, 23(1), pp. 79-83.

[7] GAN Shu-chuan and YANG Ping-xian, “PID self-tuning based on fuzzy genetic algorithm,” Journal of North China Electric Power University. 2005, 32(5), pp.43-46.

[8] WANG Yi-qun and LIU Yu, “Fuzzy controller based on adaptive genetic algorithm for a electro-hydraulic bending control system,” Chinese Journal of Construction Machinery. 2006, 4(2), pp.159-163.