Application of Improved Genetic Algorithm in PID Controller Parameters Optimization

Ying Chen, Yong-jie Ma*, Wen-xia Yun
College of Physics and Electronic Engineering, Northwest Normal University,
Anning Road no. 967, Lanzhou, China,0931-7971503
Corresponding author, e-mail: chenying1386685@126.com, 583159210@qq.com

Abstract

The setting and optimization of Proportion Integration Differentiation (PID) parameters have been always the important study topics in the automatic control field. The current optimization design methods are often difficult to consider the system requirements for quickness, reliability and robustness. So a method of PID controller parameters optimization based on Improved Genetic Algorithm(IGA) is presented .Simulations with Matlab have proved that the control performance index based on IGA is better than that of the GA method and Z-N method, and is a method which has good practical value of the PID parameter setting and optimization .

Keywords: *PID controller; Improved Genetic Algorithm; Parameter setting; Optimization*

1. Introduction

Proportion Integration Differentiation (PID) control method is one of the widely used and always the important problem in the automatic control field [1, 2], for it does not need to know the mathematical model of the controlled object, and it has simple algorithm, good robustness and high reliability [3, 4]. But in the actual production, PID parameters setting is difficult, the parameters setting of the conventional PID controller is often not good, and the performance is poor. During the past decades, great attention has been paid to optimization methods for controller and people have been looking for the automatic setting technology for PID controller parameters. For example: Ziegler-Nichols method, indirect optimization method, the gradient method, climb method, etc [5].These methods have good optimization features which can make the system performance improved. But in the control process, there are some drawbacks. They are sensitive to the initial value and easy to fall into local optimal solution. And they can't succeed to find the optimal solution. In fact, the actual industrial production process is nonlinear and time-varying. Application of conventional optimization method is difficult to establish accurate mathematical model, the controller can't achieve ideal control effect [6].

The genetic algorithm (GA) is first proposed by Professor Holland in his book *Adaptation in Natural and Artificial System* [7]. And it has proven useful in solving a variety of search and optimization problems [8]. Many real-world problems require an optimization algorithm that is able to explore multiple optima in their search space. GA is a random search algorithm that simulates natural selection and evolution. It searches through the total solution space and can find the optimal solution globally over a domain. It simply need the adaptive value which is converted from the objective function. Thus there is no special requirement on the controlled object itself. GA is particularly suitable for dealing with complex, non-linear problems. Combined with parameters optimization characteristics of PID controller, improved genetic algorithm is applied to the automatic setting of PID controller parameters, and simulation is made in the Matlab platform at last.

2. Conventional PID Controller

The structure of conventional PID control system as shown in Figure 1.The system is mainly composed of PID controller and controlled object. As a linear controller, it generates the system deviation e(t) which is produced according to the value of r(t) and the actual output value y(t),as equation (1).A control volume is the linear combination of the proportional, integral and

Received October 26, 2012; Revised January 20, 2013; Accepted February 1, 2013

differential operation of deviation, and the control volume work on the object. The law of PID control as equation (2), it's transfer function form as equation (3).

$$e(t)=r(t)-y(t) \tag{1}$$

$$u(t) = K_{\mathrm{p}}\left(e(t) + \frac{1}{T_{\mathrm{i}}}\int_0^t (t)dt + T_{\mathrm{d}}\frac{de(t)}{dt} \right) \tag{2}$$

$$G(S) = \frac{U(S)}{E(S)} = K_{\mathrm{p}}\left(1 + \frac{1}{T_{\mathrm{i}}s} + T_{\mathrm{d}}s \right) \tag{3}$$

K_{p} is proportional coefficient, T_{i} is the integral constant, T_{d} is differential constant. The roles of the individual parameters of the PID controller are described below:

K_{p} is introduced in order to timely reflect the deviation signal of the control system. Once the deviation appears, the proportional link immediately plays a regulatory role, then the deviation of system will rapidly decrease. When K_{p} is large, PID controller may accelerate the speed of adjustment. But when K_{p} is too large, overshoot will appear in the system, and it will reduce the stability of the system.

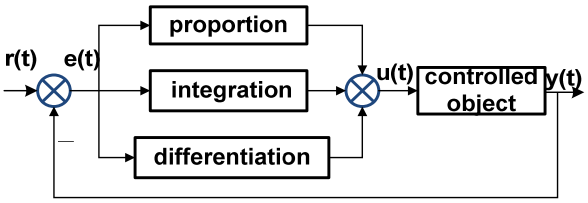

Figure 1. The structure of conventional PID control system

The parameter T_{i} is mainly used to eliminate static error, to improve the degree of no error of the system, to ensure the settings can be tracked with zero steady-state error. The strength of the integral role is depending on integrator time constant T_{i}. When T_{i} is greater, the integral action is weaker, otherwise stronger.

The parameter T_{d} can reflect the trend (the rate of change) of the deviation signal, and introduce an effective correction signal before the deviation signal becomes too large, it can accelerate the operation speed of the system, reduce regulation time.

3. PID Control System Based on Improved Genetic Algorithm

System block diagram of PID controller based on genetic algorithm as shown in Figure 2.r(t) is the system input variable, y(t) is the system output variable, e(t) is deviation, u(t) is the controller output.

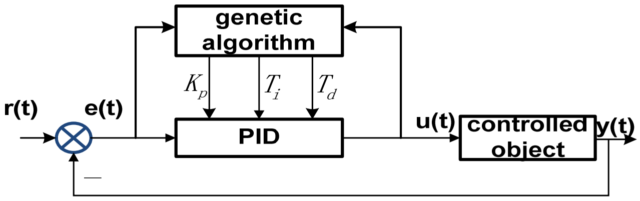

Figure 2. Optimization of PID controller based on Genetic algorithm

3.1. Genetic Algorithm

1) The Selection of The Objective Function and The Fitness Function

The purpose of optimizing the control parameters is to make the control deviation of step response tend to zero, have faster response speed and smaller or even no overshoot. Use the integral of absolute of error as the minimum objective function when select parameters, in order to prevent the control volume to be excessive, adding the square of the input item to the objective function. Select the optimal choice of the following equation(4) as the optimal index when choose parameters[9]:

$$J = \int_0^{\infty} \left(w_1 |e(t)| + w_2 u^2(t) \right) dt + w_3 * t_u \tag{4}$$

e(t) is system error, u(t) is the controller output, t_u for rise time,w_1,w_2,w_3 are the weighting constants. Through setting value of w_1, w_2, w_3 , we adjust the emphasis of the performance. In order to avoid overshoot, we adopt the penalty function. Once produce overshoot, overshoot is used as a optimal index [9], as the equation (5), setting w_4 much larger than w_1, ey(t) = yout(t)-yout(t-1), yout(t) for the output of controlled object. The parameters which make the objective function value minimum are optimization results of PID.PID design goal is to make the system performance index function J minimum, and at the same time, in order to guarantee the robustness, the system must meet the minimum phase and gain margin conditions. Here w_1= 0.999, w_2= 0.001, w_3= 2, w_4=100. The fitness function as the equation (6).

$$J = \int_0^{\infty} \left(w_1 |e(t)| + w_2 u^2(t) + w_4 |ey(t)| \right) dt + w_3 * t_u \tag{5}$$

$$F = \frac{1}{J} \tag{6}$$

2) The Design Steps of PID Control System Based on Genetic Algorithm

Step1 Genetic algorithm makes the problem parameters into chromosomes then simulates evolutionary operation, so the most important issue to be solved is to code of the solution data in the solution space into the genotype string structure data. The multivariable binary encoding method of a variety of problems is adopted to code the three parameters K_p, T_i, T_d. Considered to have a full of search space and search efficiency, each parameter is represented with a 10-bit unsigned binary code, three parameters stringing together constitute a genetic space of individual. Thus the length of each individual is 30, from left to right are K_p, T_i, T_d .

Step2 To determine the size of the population and to initialize. Set the population size 30, the evolution generation 100.

Step3 Decode the individual of the population into optimal parameters for calculating the value of fitness function and evaluating. Decoding is made the binary code string cut off, and then, converted it into the corresponding decimal integer code. In this paper, individual code string with 30 is cut into the code string with10,and then converted them into corresponding decimal integer codes, referred to as yi (i = 1,2,3). [$x_{\min}$ $x_{\max}$] is the range of the variation of decimal x. L is the length of the binary code .Decoding formula as the equation (7).

$$x_i = x_{\min} + \frac{x_{\max} - x_{\min}}{2^{\mathrm{L}} - 1} y_i \text{ (i=1,2,3)} \tag{7}$$

Step4 Do population selection, crossover and mutation operations to produce the next generation population. The crossover probability is 0.6, mutation probability is 0.01.

Step5 Repeat steps 3 and 4 until the parameters of the fitness function satisfy the condition.

3.2. PID Control Based On Improved Genetic Algorithm

1) Generate the Initial Population

The distribution of the initial population seriously affect the performance of the convergence in GA [10].If the performance of the initial population is poor, the convergence of the algorithm will slow, even have no convergence. Initial population is randomly generated in the search space, and this method is used in most of the genetic algorithm design. This paper presents a fast initial population method without affecting other performance of the algorithm, called inter-cell generated. At first, we divide the range of each parameters to be optimized into groups, the total number of groups is the population size, and then in each inter-cell randomly generates an initial individual. Initial individuals will be evenly distributed throughout the solution space, and ensuring that there is a significant gap between the randomly generated individuals. This method ensures that the initial population contains rich modes, enhancing the possibility of the converge to the global optimum [11].

2) Crossover and Mutation Strategy

In the standard genetic algorithm (SGA), crossover and mutation probability are always unchanged, with a certain blindness. In the early time, models focused on individuals with low fitness. If use the low cross-rate and mutation rate, the population is difficult to produce the outstanding individuals. In the late, models concentrated on high fitness individuals, if the crossover and mutation rate are still larger, it's easy to destroy the fine modes and the algorithm will fall into local convergence.

The choose of crossover probability p_c and mutation probability p_m is the key to the behavior and performance of GA. The higher crossover probability is, the faster new individual appear. However, when p_c is excessive, the possibility of the genetic mode damaged is larger, so that structure of individuals with high fitness will soon be destroyed; However, if the p_c value is too small, the search process will become slow, even stagnant. If p_m is too small, it is not prone to the new structure of the individual; p_m value is too large, the genetic algorithm becomes a pure random search algorithm.

In this paper, the crossover and mutation probability change over time [12], to maintain the diversity of the population, at the same time, ensure the convergence of the genetic algorithm. p_c and p_m as equations (8) and (9).

$$p_{\mathrm{c}} = \begin{cases} p_{\mathrm{c}1} - \dfrac{(p_{\mathrm{c}1} - p_{\mathrm{c}2})(f' - f_{\mathrm{avg}})}{f_{\max} - f_{\mathrm{avg}}}, & f' \geq f_{\mathrm{avg}} \\ p_{\mathrm{c}1} & , f' < f_{\mathrm{avg}} \end{cases} \tag{8}$$

$$p_{\mathrm{m}} = \begin{cases} p_{\mathrm{m}1} - \dfrac{(p_{\mathrm{m}1} - p_{\mathrm{m}2})(f' - f_{\mathrm{avg}})}{f_{\max} - f_{\mathrm{avg}}}, & f' \geq f_{\mathrm{avg}} \\ p_{\mathrm{m}1} & , f' < f_{\mathrm{avg}} \end{cases} \tag{9}$$

f_{max} is the maximum individual fitness; f_{avg} is the average individual fitness; f 'for the larger individual fitness between two individuals;p_{c1} =0.9,p_{c2} =0.6,p_{m1} =0.1,p_{m2} =0.01.

3) Selection Policy

Improved genetic algorithm use the following two strategies to select. One is the optimal retention policy, the other is the competition selection strategy.

At first, put the individual with maximum fitness into the progeny group without any changes, preserving the optimal solution of the previous generation group. This method can ensue that algorithm converge to the global optimum with the probability 1 and the best individual does not destructed by the crossover and mutation operations. It is a basic guarantee that group can convergence to the optimal solution of optimization problem.

Then, apart from the optimal solution, we select two individuals randomly from the matching pool, compare their fitness values, reserve the adaptive individual with larger fitness value to progeny.

4) Optimization Process

The flow chart of IGA optimizes PID is shown in Figure 3.

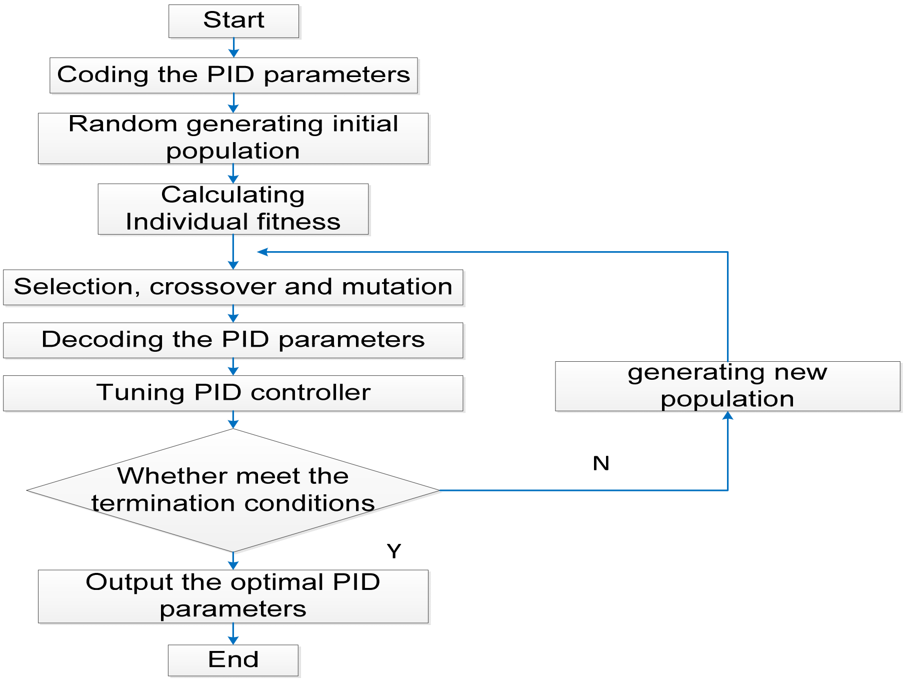
Start
Coding the PID parameters
Random generating initial population
Calculating Individual fitness
Selection, crossover and mutation
Decoding the PID parameters
Tuning PID controller
generating new population
N
Whether meet the termination conditions
Y
Output the optimal PID parameters
End

Figure 3. The flow chart of optimization of PID based on IGA

4. Results and Analysis

The simulation is carried out via Z-N, GA and IGA. The simulation object is as the equation (10). The system sampling time is selected as 1ms when the input is step signal. The experiment is done over 10 times in Matlab, the average of the outcomes is adopted, and the results are as showed in the Figure 4 and Table 1 lists their comparison. The blue line is the step curve of the system based on Z-N, the green line is the step curve of the system based on GA, and the red line is the step curve of the system based on IGA.

$$G(s)=\frac{2}{s^2+1.5s+2}e^{-0.2s} \tag{10}$$

From the Figure 4,we know that method of Z-N is easy to cause shakiness, especially to some time-delay system. The line of GA also has little shakiness, only the line of IGA is smooth. And with the Table1, it was clear that the method based on IGA is faster in the convergence speed, has smaller overshoot, has shorter adjustment time than method based on GA. The response speed and the control effect is improved markedly with the method based on IGA.

Table 1. the effect of the PID controller optimized parameters based on Z-N,GA and IGA

algorithm	K_p	T_i	T_d	evolution generation	adjustment time t_s/s
Z-N	5.0048	2.1603	1.3989		4.21
GA	2.4291	1.1478	1.0869	78	2.42
IGA	2.0534	1.4321	1.1323	57	1.68

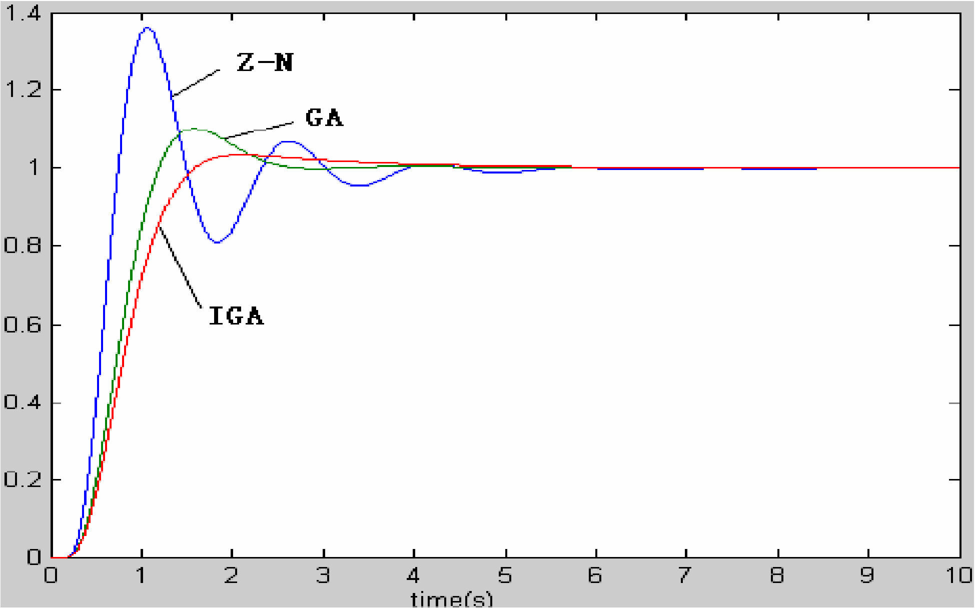

Figure 4. The unit step response curve of PID controller parameter system based on Z-N, GA, IGA

5. Conclusion

In the design of the PID controller, PID parameters setting are a numerous lock and complicated work. Over the past few years, many different techniques have developed to acquire the optimum proportional, integral and derivative control parameters for PID controllers. And it's hard to get the optimal parameters. This paper introduces an improved genetic algorithm (IGA). The purpose of this algorithm is to keep the diversity of population and improve the convergence speed. IGA adopts a new clustering method called inter-cell generated, this method helps the IGA converge to the optimal solution. The used strategy of crossover probability and mutation probability change adaptively over time. The selection strategy also improved, the optimal retention policy and competition selection strategy both adopted, ensuring that the optimal individual may not destroyed. From the above figures and table of simulation experiment, we can see that compared with the method of Z-N and GA, the method based on IGA is the best in the response speed, overshoot, and the control effect is improved markedly. This also illustrates IGA is a method has practical value and good effect to optimize the PID parameters.

Acknowledgement

This work was financially supported by 2011 Provincial Universities of Gansu Fundamental Research Special Fund

References

[1] Bennet S. Development of the PID controllers. *IEEE control Systems Magazine*. 1993; 13(2): 58-65.
[2] Wang W, Zhang JT, Chai TY. A survey of advanced PID parameter tuning methods. *Acta Automatica Sinica*. 2000; 26(3): 347-355.
[3] H Bevrani, T Hiyama, H Bevrani. Robust PID based power system stabilizer: design and real-time implementation International. *Journal of Electrical Power & Energy Systems*. 2011; 33(2): 179–188.
[4] A Khodabakhshian, R Hooshmand. A new PID controller design for automatic generation control of hydro power systems. *International Journal of Electrical Power & Energy Systems*. 2010; 32(5): 375–382.
[5] Fu-zhong He, You-xian Sun. Stabilizing PID Regulator's Parameter Design via Genetic Algorithms. *Control And Decision*. 2000; 15(4): 507-509.

[6] QG Wang, HW Fung, T H Lee. PID Tuning for Improved Performance. *IEEE Trans Control System Tech*. 1999; 7(4): 457-465.
[7] HOLLAND JH. Adaptation in natural and artificial systems: An intro- ductory analysis with applications to biology, control, and artificial intelligence.1st edition. Ann Arbor: The University of Michigan Press. 1975.
[8] Pablo Cortés, Juan Larrañeta, Luis Onieva, Genetic algorithm for controllers in elevator groups: analysis and simulation during lunchpeak traffic. *Applied Soft Computing*. 2004; 4(2) 159–174.
[9] Jin-kun Liu. Advanced PID control and Its MATLAB simulation. Beijing: Publishing House of Electronics Industry. 2002.
[10] Zalzala AM, Fleming PJ. Genetic Algorithms in Engineering Systems. London: The Institution of Electrical Engineers. 1997.
[11] Wei Gao. An Improved Fast-Convergent Genetic Algorithm and Its Performance Study Systems Engineering and Electronics. 2003; 25(11): 1427-1430.
[12] Zi-wu REN, Ye SAN. Improved Adaptive Genetic Algorithm and its Application Research in Parameter Identification. *Journal of System Simulation*. 2006; 18(1): 41-43.