Design for Auto-tuning PID Controller Based on Genetic Algorithms

Liu Fan
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
liuf0009@ntu.edu.sg

Er Meng Joo
School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore
emjer@ntu.edu.sg

***Abstract*—This paper presents a method of designing proportional-integral-derivative (PID) controller based on genetic algorithms (GA). Ziegler-Nichols tuning formula has been used for predicting the range of gain coefficients of GA-PID controller. The use of GA for optimizing the gain coefficients of PID controller has considerably improved the performance of PID controller. Simulation studies on a second-order and a third-order control system demonstrate that the proposed controller provides high performance of dynamic and static characteristics.**

***Keywords*—Ziegler-Nichols tuning formula, fuzzy proportioanl-integral-derivative (PID), genetic algorithm**

I. INTRODUCTION

The most popular controllers used in industrial control processes are proportional-integral-derivative (PID) controller because of their simple structures and robust performance in a wide range of operating conditions [1] [2]. Beginning with Ziegler-Nichols's works [3] [4], much effort has been devoted to developing methods to tuning the parameters of conventional PID controllers [5]-[8]. It is based on the automatic measurement of the ultimate gain and period time, from which the optimal PID and proportional-integral (PI) controller parameters can be computed using Ziegler-Nichols (ZN) tuning formula [4]. However, successful applications of PID controller require the satisfactory tuning of parameters according to the dynamics of the process. At the same time, the conventional PID controller could not satisfy the control requirements of much more complicated systems of today.

Fuzzy logic controller (FLC) makes control decision by its well-known fuzzy IF-THEN rules. In most cases, designing of the FLC is accomplished by trial-and-error method using computer simulations. Due to the popularity of PID controllers in industrial applications, most of the development of fuzzy controllers revolved around fuzzy PID controllers in the past decade [9]-[11]. But there are some difficulties that prevent the designing of fuzzy controllers from being systematic. First, the choice of the overall control structure is a problem faced by many designers. Second, in designing of the FLC, not only structure parameters of the FLC need to be designed, but also the gains of the conventional controller need to be tuned. Because of its complicated cross-effects, analytical tuning algorithms for these parameters are really difficult.

The genetic algorithm is very effective at finding optimal solutions to a variety of problems. By working with a population of solutions, the GA can seek many local minima, and thus increase the likelihood of finding the global minimum. It performs especially well when solving complex problems because it does not impose many limitations of traditional techniques [12]-[15]. Inspired by the earlier work, this paper presents a new approach toward optimal design of a PID controller based on genetic algorithm to calculate the optimal values for the three PID gain coefficients.

This paper is organized as follows. Section 2 reviews the ZN tuning formula in the context of PID and PI auto-tuning. Section 3 introduces auto-tuning method of PID parameters based on GA as an effective controller. Section 4 presents the simulation studies on a second-order and a third-order control system and compares the proposed GA-PID controllers with ZN PID and fuzzy PID controllers with detailed explanations. Section 5 concludes the paper.

II. OVERVIEW OF ZN-PID CONTROLLER

A PID controller is a feedback controller which makes a plant less sensitive to changes in the surrounding environment and small changes in the plant [16]. The Ziegler-Nichols ultimate-cycle or closed-loop tuning has been widely known as a fairly accurate heuristic method to determine good settings of PID and PI controllers for a wide range of common industrial processes [3]. The Ziegler-Nichols tuning formula is based on the empirical knowledge of the ultimate gain k_u and ultimate period time t_u, as shown below.

TABLE I. ZIEGLER-NICHOLS TUNING FORMULA

	PID	***PI***
Proportional gain	$k_p = 0.6k_u$	$k_p = 0.45k_u$
Integral time	$T_i = 0.5t_u$	$T_i = 0.85t_u$
Derivative time	$T_d = 0.125t_u$	

The PID controller is usually implemented as follows:

$$u(t) = K_p(e(t) + \frac{1}{T_i}\int_0^t e(t)dt + T_d \frac{de(t)}{dt}) \tag{1}$$
$$e(t) = y_d(t) - y(t).$$

where $u(t)$ is the controller output, $y(t)$ and $y_d(t)$ are process output and desired output, respectively. The goal of PID controller design is to determine a set of gains: K_p, T_i and T_d, so as to improve the transient response of a system by reducing the overshoot and by shortening the settling time of the system.

III. DESIGN OF GA-PID CONTROLLER

The genetic algorithm is a robust optimization technique based on natural selection. The basic goal of GA is to optimize functions called fitness functions. A possible solution to a specific problem is seen as an individual. A collection of a number of individuals is called a population. The current population reproduces new individuals that are called the new generation. The new individuals of the new generation are supposed to have better performance than the individuals of the previous generation. GA have been successfully implemented in the area of industrial electronics, for instance, parameter and system identification, control robotics, pattern recognition, planning and scheduling and classifier system [14]. For its use in control engineering, GA can be applied to a number of control methodologies for the improvement of the overall system performance.

The structure of the control system with GA-PID controller is shown in Fig. 1. It consists of a conventional PID controller with auto-tuning its gain coefficients based on GA and a control plant.

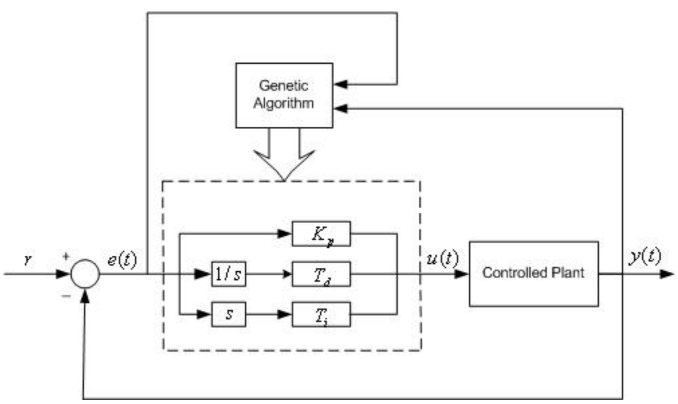

Figure 1. Structure of GA-PID controller

The GA thus consists of three fundamental operators: reproduction, crossover and mutation.

- Reproduction is simply retaining a fit string in the following generation.
- Crossover involves swapping partial strings of random length between two parent strings.
- Mutation involves flipping a random bit in a string.

The GA has the following advantages:

- It is a simple algorithm that is easily understood and implemented.
- The algorithm is robust.
- GA is a non-linear process that could be applied to most industrial processes with good results.
- GA searches a population of points instead of a singly solution. The GA is therefore not easily sidetracked to obtain a local optimal solution instead of a global optimal solution.
- GA does not need information about the system except for the fitness function.

Due to these considerably advantages of GA, we apply it on optimizing gain coefficients of conventional PID controller.

An auto-tuning GA-PID controller can be implemented as follows.

A. Encoding

One binary string consisting of the three PID gain coefficients: K_p, K_i and K_d is used to ensure that the variables are interdependent. Unsigned binary coding is applied.

B. Initialization

The first population is generated at random within boundaries. The boundaries for the PID constants are chosen to ensure that not too many of the generated PID constants lead to an unstable system. In this paper, we use ZN tuning formula to decide the range of PID parameters first. Then based on GA method, we can search the optimal parameters of PID controller.

C. Fitness Function

The fitness of a chromosome is calculated from the integral of the square error (ISE) or other such time integral criteria. Other specifications, such as overshoot, rise time and settling time may also be taken into account [17].

In order to overcome the large energy of the controller, we add the square term of control output $u(t)$ to the fitness function. The fitness function is

$$J = \int_0^\infty (w_1|e(t)| + w_2 u^2(t))dt + w_3 \cdot t_r \tag{2}$$

where w_1, w_2, w_3 are weight coefficients, $u(t)$ is the output of controller, t_r is rise time and $e(t)$ is the system error.

In order to get satisfied transient process and try to suppress overshoot, we revise the fitness function as follows:

$$J = \int_0^\infty (w_1|e(t)| + w_2 u^2(t) + w_4|ey(t)|)dt + w_3 \cdot t_r \quad ey(t) < 0 \tag{3}$$

where w_4 is the weight coefficient, and $w_4 >> w_1$, $ey(t) = y(t) - y(t-1)$, $y(t)$ is the output of the plant. Where there is overshoot, a punishment term $ey(t)$ is added at once.

In this paper the fitness function is chosen as

$$f = 1/(J + 10^{-9}) \quad (4)$$

Moreover, the term 10^{-9} is added to the denominator to avoid the denominator of the fitness function becoming zero.

D. Selection

The standard roulette wheel selection is applied to select individuals in the current population pool. The offspring are produced based on this selection. The selection probability of each individual depends on the fitness value, the bigger the fitness value is, the more offspring the individual reproduce. The number of reproduced made of each individual into the next population pool is calculated as follows:

Number of reproduced =N*relative fitness (5)

where N is the population size.

E. Crossover

Two parent chromosomes are crossed to produce one child. But not all individuals are necessarily used for crossover. The crossover rate is chosen as $P_c = 0.9$. A larger population usually requires a smaller crossover rate. This process primarily combines the best characteristics of the one string with the best characteristics of the other.

F. Mutation

Mutation changes the structure of the string by changing the value of a bit chosen at random. This operator can prevent individuals falling into a local optimum. The optimal mutation rate P_m is suggested in [18] as $P_m = 1/(S_p\sqrt{L})$, where S_p is the size of population and L is the length of individual. A random string with the length L is generated. If the value in a position of this random string is less than or equal to the mutation rate P_m, the gene of the child in the same position will be inverse of the original. In this paper, the mutation rate is chosen as $P_m = 0.032$.

The flowchart of GA-PID controller is shown in Fig. 2. First, all the parameters in simulated program are initialized. Then, a binary string is applied to encode all the three gain coefficients of GA PID. Second, standard genetic algorithm is implemented in order to search the optimal gain coefficients of GA PID. If satisfactory gain coefficients are obtained, parameters will be encoded.

The object of the proposed auto-tuning GA-PID controller is to search the optimal values of the gain coefficients K_p, K_i and K_d on-line in order to obtain a maximum fitness function value and to speed up the convergence of position tracking error and to improve the overall performances of controller.

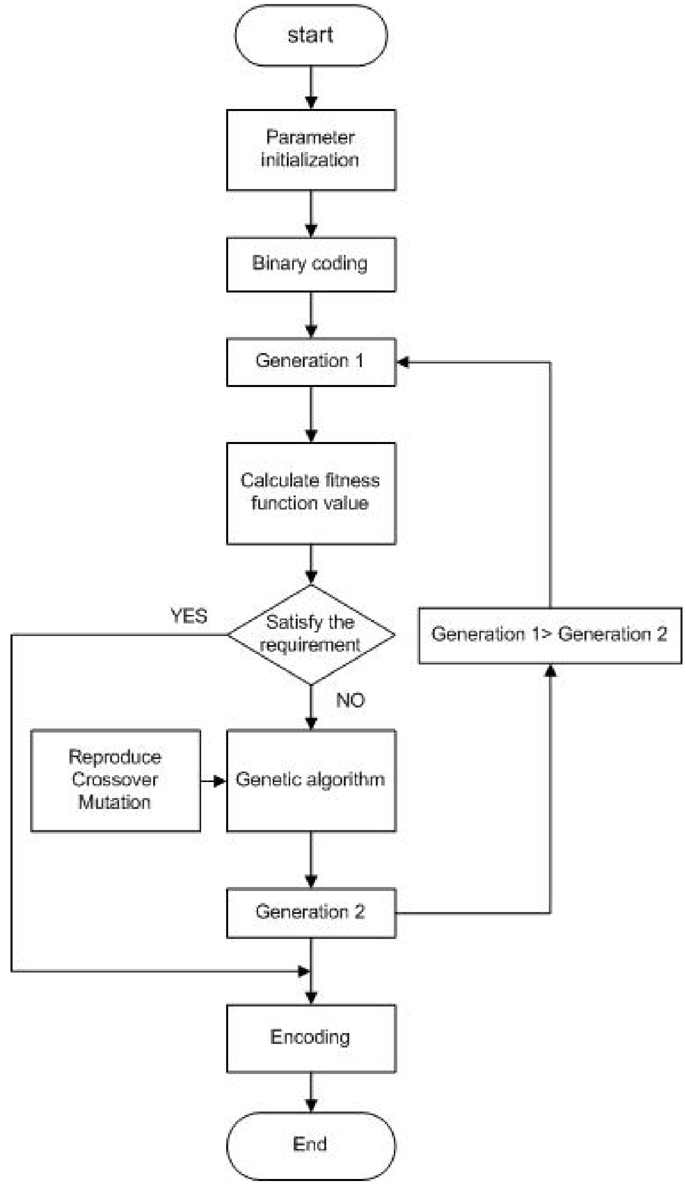

Figure 2. Simulation flowchart of auto-tuning GA-PID controller

IV. SIMULATION EXAMPLES

A. Simulation Plants

Since many industrial processes like temperature control process can be modeled by second-order plant with over-damping, we will use a second-order system to evaluate the effectiveness of the proposed controller. In order to compare with ZN-PID controller and fuzzy PID controller completely, a third-order plant is also used in simulation.

Considering the following second-order and third-order control systems respectively, the second-order plant is:

$$G_1(s) = \frac{200}{s^2 + 50s} \quad (6)$$

The third-order plant is:

$$G_2(s) = \frac{400}{s^3 + 30s^2 + 200s} \quad (7)$$

B. Simulation Parameters

According to the experience, the size of population of GA is often chosen between [20,100], for our examples, we choose 30. The crossover rate and mutation rate are chosen as: $p_c = 0.9, p_m = 0.032$. The number of generation is often chosen between [100,500], we choose $G = 200$. Weight coefficients are $w_1 = 0.998$, $w_2 = 0.001$, $w_3 = 2.0$, and $w_4 = 100$. The sample time is $t_s = 0.001s$.

The parameter range of GA-PID controller is, for the second order system, $K_p \in [0,20], K_i \in [0,1], K_d \in [0,1]$, for the third system, $K_p \in [0,20], K_i \in [0,1], K_d \in [0,5]$.

C. Simulation Results

The proposed algorithm for auto-tuning PID controller has been applied to the second-order process model since many industrial processes can be modeled by second-order plants. And a third-order is also used to evaluate the proposed PID controller.

It is evident from Fig. 3 and Fig. 4 that the proposed controller is better than the ZN-PID controller [4] as well as the fuzzy PID controller [19]. However, due to the nature of GA, we can not guarantee that the results are optimal, but we can be sure that they are at least near optimal. The GA will give close but different results for each new search. PID parameters are shown bellow.

TABLE II. PID PARAMETERS OF PID CONTROLLERS

PID Parameters	Second-order System		Third-order System	
	ZN PID	*GA PID*	*ZN PID*	*GA PID*
K_p	1.4304	18.5962	2.5432	7.7605
K_i	5.7214	0.2400	4.8789	0.9688
K_d	0.0893	0.3883	0.3314	0.6519

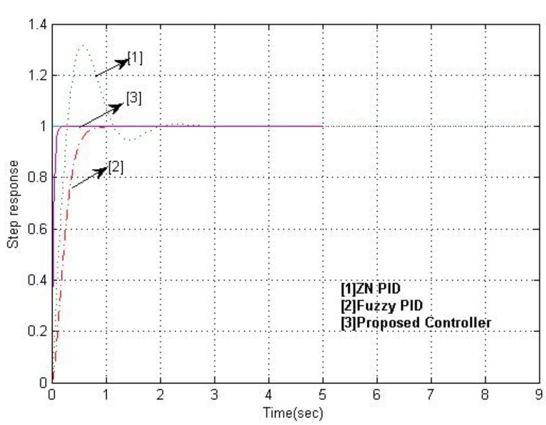

Figure 3. Step response for the second-order process

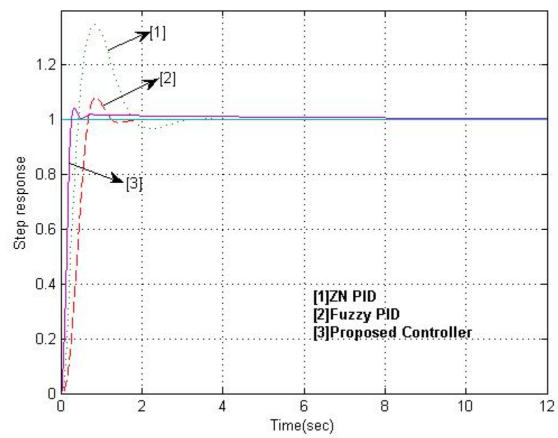

Figure 4. Setp response for the third-order system

V. CONCLUSIONS

In this paper, the designing of auto-tuning PID controller using GA has been presented. The gain coefficients of the GA-PID controller were obtained quickly overcoming the premature convergence in the GA. The performance of the proposed controller was evaluated by simulation studies on a second-order and a third-order control system. Simulation results demonstrate that the proposed controller outperforms ZN-PID and fuzzy PID controllers in terms of dynamic and static characteristics.

REFERENCES

[1] B. C. Kuo, “Automatic Control Systems,” 5th ed. Englewood Cliffs: Prentics-Hall, 1987.

[2] Y. Takashi, M. J. Rabins, and D. M. Auslander, “Control and Dynamic Systems,” Menlo Park, CA: Addison-Wesley, 1970.

[3] J. G. Ziegler and N. B. Nichols, “Optimum settings for automatic controllers,” Trans. ASME, vol. 64, pp. 759-768, 1942.

[4] C. C. Hang, K. J. Aström and W. K. Ho, “Refinements of the Ziegler-Nichols tuning formula,” Proc, Inst. Elect. Eng., pt. D, vol. 138, pp. 111-118, 1991.

[5] M. J. Er and Y. L. Sun, “Hybrid fuzzy proportional-integral plus conventional derivative control of linear and nonlinear system,” IEEE Trans. Ind. Elect., vol. 48, No. 6, 1109-1117, Dec. 2001.

[6] M. Ge, M. S. Chiu and Q. G. Wang, “Robust PID controller design via LMI approach,” Journal of Process Control, vol. 12, no.1, pp. 3-13, 2002.

[7] J. C. Shen and H. K. Chiang, “PID tuning rules for second order systems,” 5th Asian Control Conference, pp. 472-477, 2004.

[8] W. M Qi, W. Y. Cai, Q. L. Ji and Y. C. Cheng, “A design of nonlinear adaptive PID controller based on genetic algorithm,” Chinese Control Conference Proceeding, CCC2006, art. No. 4060394, pp.175-178, 2006.

[9] Z. Y. Zhao, M. Tomizuka and S. Isaka, “Fuzzy gain scheduling of PID controllers,” IEEE Trans. Syst., Man, Cybern., vol. 23, pp. 1392-1398, Sept. 1993.

[10] S. Z. He, S. Tan, F. L. Xu, and P. Z. Wang, “Fuzzy self-tuning of PID controllers,” Fuzzy Sets Syst., vol. 56, pp. 37-46, 1993.

[11] M. J. Er, Y. L. Sun, K.C. Choong and T. Goh, “Design of a hybrid fuzzy proportional-integral plus conventional derivation controller,” in Proc. 3rd Asian Control Conf., pp. 1151-1159.

[12] D. Dasgupta and Z. Michalewicz, “Evolutionary Algorithms in Engineering Applications,” Berlin, Germany: Springer-Verlag, 1997.

[13] V. Maniezzo, “Genetic evolution of topology and weight distribution of neural networks,” IEEE Trans. Neural Netw., vol. 5, no. 1, pp. 39-53, Jan. 1994.

[14] K. F. Man, K. S. Tang, and S. Kwong, “Genetic algorithms: Concepts and applications,” IEEE Trans. Ind. Electron, vol. 43, no. 5, pp. 519-534, May 1996.

[15] X. Yao, “Evolving artificial neural networks,” Proc. IEEE, vol. 87, no. 9, pp. 1423-1447, Sep. 1999.

[16] SHAW, LS. “Industrial process control – third revised and expanded edition,” South Africa, Rand Afrikaans University, Faculty of Engineering, Industrial Electronic Technology Research Group (NIET), 1997.

[17] P.J.van Rensberg, I.S.Shaw, J.D.van Wyk, “Adaptive PID control using a genetic algorithm,” Second International Conference on Knowledge-Based Intelligent Electronic Systems, pp. 133-138, 1998.

[18] J. Hessner and R. Manner and P. Schwefel, “Towards an optimal probability for genetic algorithms,” in Proc. 1st Workshop Parallel Problem Solving From Nayure, Berlin, Germany, 1991, vol. 496, Lecture Notes in Computer Science, pp. 23-31.

[19] D. Misir, H. A. Malki and G. R. Chen, “Design and analysis of a fuzzy proportional-integral-derivative controller,” Fuzzy and Sets Syst., vol. 79, pp. 297-314, 1996.