Journal of Systems Engineering and Electronics

Vol. 19, No. 3, 2008, pp.578–583

Design of PID controller with incomplete derivation based on differential evolution algorithm*

Wu Lianghong[1,2], *Wang Yaonan*[2], *Zhou Shaowu*[1] & *Tan Wen*[1]

1. School of Information and Electric Engineering, Hunan Univ. of Science and Technology, Xiangtan 411201, P. R. China;

2. Coll. of Electric and Information Engineering, Hunan Univ., Changsha 410082, P. R. China

(Received November 5, 2006)

Abstract: To determine the optimal or near optimal parameters of PID controller with incomplete derivation, a novel design method based on differential evolution (DE) algorithm is presented. The controller is called DE-PID controller. To overcome the disadvantages of the integral performance criteria in the frequency domain such as IAE, ISE, and ITSE, a new performance criterion in the time domain is proposed. The optimization procedures employing the DE algorithm to search the optimal or near optimal PID controller parameters of a control system are demonstrated in detail. Three typical control systems are chosen to test and evaluate the adaptation and robustness of the proposed DE-PID controller. The simulation results show that the proposed approach has superior features of easy implementation, stable convergence characteristic, and good computational efficiency. Compared with the ZN, GA, and ASA, the proposed design method is indeed more efficient and robust in improving the step response of a control system.

Keywords: PID controller, incomplete derivation, differential evolution, parameter tuning.

1. Introduction

During the past decades, the control techniques in the industry have made great advances. Numerous control methods such as fuzzy control, neural network control, expert system, and adaptive control have been studied deeply[1]. Among them, the best known is the proportional-integral-derivative (PID) controller, which has been widely used in the industry because of its simple structure and robust performance in a wide range of operating conditions[2]. Unfortunately, it has been quite difficult to tune properly the gains of PID controllers because many industrial plants are often burdened with problems such as high order, time delays, and nonlinearities[3−6]. Over the years, some methods have been proposed for the tuning of PID controllers. The first method that used the classical tuning rules was proposed by Ziegler and Nichols[3]. But the drawback of this method is that the transient response of system has a greater overshoot. In general, it is often difficult to determine optimal or near optimal PID parameters with the Ziegler-Nichols method in many industrial plants[6].

Evolutionary algorithms (EAs) have been received much interests recently and have been applied successfully to solve the problem of optimal PID controller parameters. P. Wang in Ref. [4] used an advanced genetic algorithm to auto-tune classical PID controllers. L. Wang proposed a GASA hybrid strategy for designing a class of PID controller for non-minimum phase systems in Ref. [5]. Particle swarm optimization algorithm was used to optimize PID controller parameters in Refs. [6–7]. In Ref. [8], ant system algorithm was applied to design PID controller with incomplete derivation. Chaotic ant swarm was used to tune the PID parameter in Ref. [9].

Differential evolution (DE), first introduced by R. Storn and K. Price in 1995, is one of the modern

* This project was supported by the National Natural Science Foundation of China (60375001) and the Scientific Research Foundation of Hunan Provincial Education Department (05B016).

heuristic algorithms[10]. The DE algorithm has gradually become more popular and has been used in many practical cases, mainly because it has demonstrated good convergence properties and is principally easy to understand[11]. Because the DE algorithm is an excellent optimization methodology, it may be a promising approach for solving the optimal PID controller parameters problem. In this article, the DE is employed to design the PID controller with incomplete derivation. The controller is called DE-PID controller.

The integral performance criteria in frequency domain were often used to evaluate the performance of the controller, but these criteria have their own advantages and disadvantages[4]. In Ref. [4], a simple performance criterion in time domain was proposed. However, the performance criterion will be invalidated when the system step response has not overshoot. In this article, a new simple performance criterion in time domain is introduced for evaluating the performance of a PID controller.

2. PID controller with incomplete derivation

In general, derivation control can improve the dynamic behavior of a control system, but a pure derivative cannot and should not be implemented, because it will give a very large amplification of measurement noise. To overcome this drawback, a low-pass filter is often added to derivation term. The derivation control with a low-pass filter is called the incomplete derivation control. PID controller with incomplete derivation has better control performances compared with general PID controller, and this PID controller is adopted in this article.

The transfer function of the PID controller with incomplete derivation is expressed as

$$\begin{aligned} U(s) = \left(K_p + \frac{K_p}{T_i s} + \frac{K_p T_d s}{1+T_f s}\right) E(s) = \\ U_p(s) + U_i(s) + U_d(s) \end{aligned} \tag{1}$$

where K_p is the proportional gain, T_i is the integral time constant, T_d is the derivative time constant. In the discrete-time domain, the controller can be described as follows

$$\begin{aligned} u(k) = & u_p(k) + u_i(k) + u_d(k) = \\ & K_p e(k) + K_i \sum_{j=1}^{k} e(j) + u_d(k) \end{aligned} \tag{2}$$

where $K_i = K_p T/T_i$ and T is the sampling period. $u_d(k)$ can be derived as follows. From Eq. (1) we can see that

$$U_d(s) = \frac{K_p T_d s}{1+T_f s} E(s) \tag{3}$$

The differential equation of Eq. (3) is as follows

$$u_d(t) + T_f \frac{\mathrm{d}u_d(t)}{\mathrm{d}t} = K_p T_d \frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{4}$$

In the discrete-time domain, Eq. (4) can be described as

$$\begin{aligned} u_d(k) = & K_d(1-\lambda)[e(k)-e(k-1)] + \\ & \lambda u_d(k-1) \end{aligned} \tag{5}$$

where $\lambda = T_f/(T_f+T) < 1$, $K_d = K_p T_d/T$. Substituting Eq. (5) to (2), then

$$\begin{aligned} u(k) = & K_p e(k) + K_i \sum_{j=0}^{k} e(j) + \\ & K_d(1-\lambda)[e(k)-e(k-1)] + \lambda u_d(k-1) \end{aligned} \tag{6}$$

Equation (6) is the control law of digital PID controller with incomplete derivation. In this article, λ is set 0.9 for minimum phase system and 0.5 for non-minimum phase system.

3. Performance criteria of PID controller

In general, the PID controller design method using the integrated absolute error (IAE), or the integral of squared-error (ISE), or the integrated of time-weighted-squared-error (ITSE) is often employed in the design of control system , because it can be evaluated analytically in the frequency domain[4]. The IAE, ISE, and ITSE performance criterion formulas are as the following.

$$\mathrm{IAE} = \int_0^\infty |r(t)-y(t)|\mathrm{d}t = \int_0^\infty |e(t)|\mathrm{d}t \tag{7}$$

$$\mathrm{ISE} = \int_0^\infty e^2(t)\mathrm{d}t \tag{8}$$

$$\mathrm{ISTE} = \int_0^\infty te^2(t)\mathrm{d}t \tag{9}$$

The three integral performance criteria in the frequency domain have their own advantages and disadvantages. For example a disadvantage of the IAE and ISE criteria is that its minimization can result in a response with relatively small overshoot but a long setting time because the ISE performance criterion weights all errors equally independent of time. Although the ITSE performance criterion can overcome the disadvantage of the ISE criterion, the derivation processes of the analytical formula are complex and time-consuming[4]. To overcome the disadvantages of the integral performance criteria in the frequency domain, a new performance criterion in the time domain was proposed in ref Ref. [4]. The new performance criterion is defined as

$$W(K) = (1 - e^{-\beta}) \cdot (M_p + E_{ss}) + e^{-\beta} \cdot (t_s - t_r) \quad (10)$$

where K is $[K_p, K_i, K_d]$, $\beta \in [0.8, 1.5]$ is the weighting factor, M_p is the overshoot, t_s is the setting time, t_r is the rise time, and the E_{ss} is the steady-state error. However, when M_p=0, even the values of t_r and t_s are large, the difference between t_r and t_s may be very small, accordingly, W will be very small. In this case, the PID controller parameters are not optimal or near optimal, and the performance criterion is invalid. To avoid this drawback, the performance criterion can be redefined as follows.

$$W(K) = (1 - e^{-\beta}) \cdot (M_p + E_{ss}) + e^{-\beta} \cdot (t_s + t_r) \quad (11)$$

After this modification, minimizing W can ensure a set of optimal or near optimal control parameters K_p, K_i and K_d. For non-minimum phase systems, zeros in the right-half plane will result undershoot in the step response of systems, which is harmful in industry control. For the optimum designing of a PID controller, restraining undershoot must be considered in the performance criterion. A new performance criterion in time domain for non-minimum phase systems is also proposed for evaluating the PID controller. The performance criterion is defined as Eq. (12).

$$\begin{aligned} W(K) =& (1 - e^{-\beta}) \cdot (M_p + E_{ss} + |M_u|) + \\ & e^{-\beta} \cdot (t_s + t_r) \end{aligned} \quad (12)$$

where M_u is the undershoot.

4. Differential evolution algorithm

In 1995, R. Storn and K. Price first introduced the differential evolution algorithm. It is one of the optimization techniques and a kind of evolutionary computation technique. The method has been found to be an effective and robust in solving problems with nonlinearity, non-differentiability, multiple optima, and high dimensionality[11]. There are several variants of DE[10]. In this article, we use the DE scheme classified using notation as DE/rand/1/bin strategy[10]. This strategy is the most often used in practice.

A set of D optimization parameters is called an individual. It is represented by a D-dimensional parameter vector. A population consists of NP parameter vectors x_{ij}^G, $i = 1, 2, \ldots, NP, j = 1, 2, \ldots, D$. G denotes one generation. NP is the number of members in a population. It is not changed during the evolution process. The initial population is chosen randomly with uniform distribution in the search space.

DE has three operations: mutation, crossover, and selection. The crucial idea behind DE is a scheme for generating trial vectors. Mutation and crossover are used to generate trial vectors, and selection then determines which of the vectors will survive into the next generation.

4.1 Mutation

For each target vector x_{ij}^G, a mutant vector v is generated according to

$$v_i^{G+1} = x_{r1}^G + F \cdot (x_{r2}^G - x_{r3}^G) \quad (13)$$

The r_1, r_2, and r_3 are randomly chosen indices and $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$. Note that the indices must be different from each other and from the running index i so that NP must be at least four. F is a real number to control the amplification of the difference vector $(x_{r2}^G - x_{r3}^G)$. According to Storn and Price[9], the range of F is in (0,2]. If a component of a mutant vector goes off the search space, then this component is set to bound value.

4.2 Crossover

The target vector is mixed with the mutated vector, using the following scheme, to yield the trial vector u.

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, & rand(j) \leqslant CR \text{ or } j = randn(i) \\ x_{ij}^{G}, & rand(j) > CR \text{ and } j \neq randn(i) \end{cases} \tag{14}$$

where $j = 1, 2, \ldots, D$, $rand(j) \in [0,1]$ is the jth evaluation of a uniform random generator number. $CR \in [0,1]$ is the crossover probability constant, which has to be determined previously by the user. $randn(i) \in (1, 2, \ldots, D)$ is a randomly chosen index which ensures that u_i^{G+1} gets at least one element from v_i^{G+1}. Otherwise, no new parent vector would be produced and the population would not be altered.

4.3 Selection

DE adapts greedy selection strategy. If and only if, the trial vector u_i^{G+1} yields a better fitness function value than x_i^G, then u_i^{G+1} is set to x_i^{G+1}. Otherwise, the old value x_i^G is retained. In this article the minimization optimization is considered. The selection operator is as follows.

$$x_i^{G+1} = \begin{cases} u_i^{G+1}, & f(u_i^{G+1}) < f(x_i^G) \\ x_i^G, & f(u_i^{G+1}) \geqslant f(x_i^G) \end{cases} \tag{15}$$

5. Optimization procedures for PID controller parameters

5.1 Parameters searching space

The parameters searching space of DE is extended on the base of results obtained by Ziegler-Nichols (ZN) method, which not only take use of the reasonable kernel of ZN method but also reduce the parameter searching space. If the optimization result is close to the boundary of the searching space, the searching space should be further extended based on the result. The range of the controller parameters is determined by the following strategies.

$$(1-\alpha)K_p' \leqslant K_p \leqslant (1+\alpha)K_p' \tag{16}$$

$$(1-\alpha)T_i' \leqslant T_i \leqslant (1+\alpha)T_i' \tag{17}$$

$$(1-\alpha)T_d' \leqslant T_d \leqslant (1+\alpha)T_d' \tag{18}$$

where the K_p', T_i', T_d' are the tuning result using ZN method, α is set in the rang of 0 to 1.

5.2 Optimization procedures of DE-PID

The searching procedures of the DE-PID controller with incomplete derivation are shown as below.

Step 1 Specify the number of population NP, difference vector scale factor F, crossover probability constant CR, and the maximum number of generations. Initialize randomly the individuals of the population and the trial vector in the given searching space.

Step 2 Use each individual as the PID controller parameters and calculate the values of the four performance criteria of the system unit step response in the time domain, namely M_p, E_{ss}, t_r and t_s.

Step 3 Calculate the fitness value of each individual in the population using the performance criterion function given by (11) and (12) for non-minimum phase system.

Step 4 Compare the fitness value of each individual and get the best fitness and best individual.

Step 5 Generate a mutant vector according to (13) for each individual.

Step 6 According to (14), do the crossover operation and yield a trial vector.

Step 7 Do the selection operation in terms of (15) and generate a new population.

Step 8 $G = G+1$, return to Step 2 until to the maximum number of generations.

6. Simulation research

Three typical control systems are chosen to evaluate the performance of the proposed DE-PID controller. The transfer functions of the plants are given as follows[8].

Case 1 (Three-order system)

$$G_1(s) = \frac{6\ 068}{s(s^2 + 110s + 6\ 068)} \tag{19}$$

Case 2 (Time-delay system)

$$G_2(s) = \frac{2e^{-s}}{1+5s} \tag{20}$$

Case 3 (Non-minimum phase system)

$$G_3(s) = \frac{1-0.5s}{(1+s)^3} \tag{21}$$

To verify the performance of the proposed DE-PID controller with incomplete derivation, three existing PID controllers, including ZN-PID, GA-PID, and ASA-PID, are compared with the controller. The three PID controllers are introduced in Ref. [8].

The following simulation parameters are used in this article.

· The member of each individual is K_p, T_i, and T_d.

· Population size $NP = 30$.

· Scale factor $F = 0.5$, crossover probability constant $CR = 0.1$.

· The maximum number of generations Maxiteration=400.

· The sampling time $T = 0.01$ s for Case 1, and $T = 0.1$ s for Case 2 and Case 3.

The statistical results of 20 independent runs for each case are shown in Tables 1–3. Figs. 1–3 are the unit step response of Case 1 to Case 3.

Table 1 Statistical results for Case 1

PID controller	ZN-PID	GA-PID	ASA-PID	DE-PID
K_p	33.104	25.765	15.380	15.794
T_i	0.301 8	1.759 5	2.033 6	2.854 7
T_d	0.072 4	0.027 2	0.013 5	0.017 3
M_p/%	17.21	5.23	2.04	2.0
t_r/s	0.06	0.07	0.13	0.11
t_s/s	0.305 2	0.602 5	0.180 2	0.13
E_{ss}	0	0	0	0

Table 2 Statistical results for Case 2

PID controller	ZN-PID	GA-PID	ASA-PID	DE-PID
K_p	2.121 5	1.856 1	1.222 5	1.428 6
T_i	2.624 5	7.511 5	6.018 6	4.835 9
T_d	0.629 9	0.525 9	0.319 9	0.304 0
M_p/%	52.69	11.98	5.65	2.06
t_r/s	2.3	2.5	3.6	2.9
t_s/s	8.368 4	7.726 7	7.509 1	3.4
E_{ss}	0	0	0	0

Table 3 Statistical results for Case 3

PID controller	ZN-PID	GA-PID	ASA-PID	DE-PID
K_p	1.550 5	1.244 5	1.234 4	1.185 9
T_i	3.343 0	2.703 7	2.703 5	2.284 3
T_d	0.802 3	0.995 8	0.990 0	0.862
M_p/%	18.11	6.48	6.19	2.10
M_u/%	10.16	9.16	8.85	8.16
t_r/s	2.8	3.2	3.3	3.7
t_s/s	11.315	5.727 7	5.759 4	4.4
E_{ss}	0	0	0	0

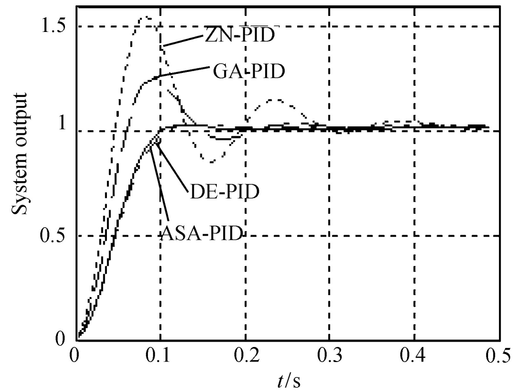

Fig. 1 The unit step response of Case 1

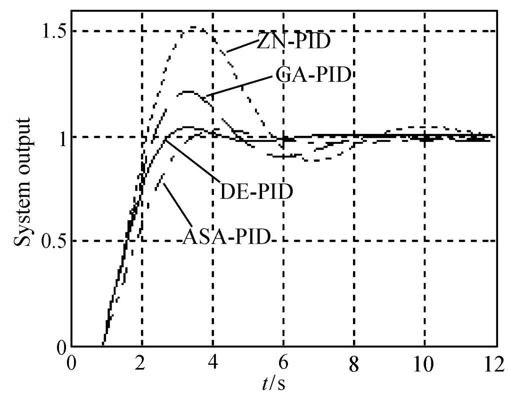

Fig. 2 The unit step response of Case 2

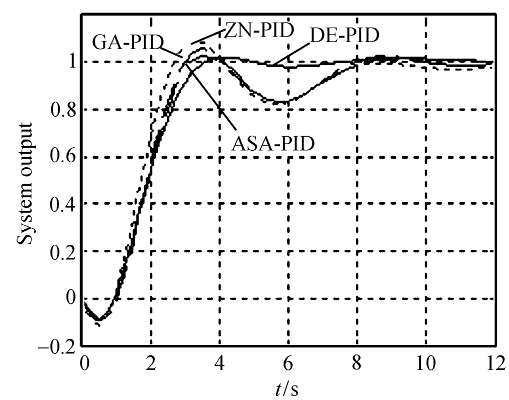

Fig. 3 The unit step response of Case 3

From Tables 1–3, we can see that for each of the three study cases, the performance of the proposed DE-PID controller is much better than that of the ZN-PID controller and GA-PID controller and better than that of ASA-PID controller. As can be seen from the tables and figures, for Case 1 and Case 2, the overshoot M_p and the setting time t_s of the DE-PID controller are better than those of the other three PID controllers, for Case 3, the M_p, t_s and the undershoot M_u of the DE-PID controller are also better than those of the other three PID controllers.

7. Conclusions

This article presents a novel and intelligent design method for tuning the parameters of the PID controller with incomplete derivation using differential evolution algorithm. The proposed method integrates the DE algorithm with the new time-domain performance criterion into a DE-PID controller. Three typical systems are used to evaluate the performance of DE-PID controller. The simulation results show that the DE algorithm can perform an efficient search for the optimal PID controller parameters. Compared with ZN-PID controller, GA-PID controller, and ASA-PID controller, the proposed DE-PID controller has better dynamic performances and robust stability of unit step response. The DE algorithm is easy to be understood and realized and it is very efficient and robust for complex function optimization; therefore, the DE-PID controller can be used in practice engineering widely.

Different DE control parameters are required for solving different practice problems, such as difference vector scale factor F and crossover probability constant CR. Hence, how to select proper parameters for the target problem is an important focus of our future studies.

References

[1] Wang Y N. Intelligent control system (Second Edition). *Changsha*: *Hunan University Press*, 2006.

[2] Wang W, Zhang J T, Chai T Y. A survey of advanced PID parameter tuning methods. *Acta Automatica Sinica*, 2000, 26(3): 347–355.

[3] Ziegler J G, Nichols N B. Optimum setting for automatic controllers. *ASME Trans.*, 1942, 64(8): 759–768.

[4] Wang P, Kwok D P. Auto-tuning of classical PID controllers using an advanced genetic algorithm. *Proc. of IEEE Int. Conf. on Power Electronics and Motion Control, San Diego*, 1992: 1224–1229.

[5] Wang L, Li W F, Zhong D Z. Optimal design of controllers for non-minimum phase systems. *Acta Automatica Sinica*, 2003, 29(1): 135–141.

[6] Gaing Z L. A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Trans. on Energy Conversion*, 2004, 19(2): 384–391.

[7] Wang J S, Wang J C, Wang W. Self-tuning of PID parameters based on particle swarm optimization. *Control and Decision*, 2005, 20(1): 73–76.

[8] Tan G Z, Zeng Q D, Li W B. Design of PID controller with incomplete derivation based on ant system algorithm. *Journal of Control Theory and Applications*, 2004, 2(3): 246–252.

[9] Li L X, Peng H P, Wang X D, et al. PID parameter tuning based on chaotic ant swarm. *Chinese Journal of Scientific Instrument*, 2006, 27(9): 1104–1106.

[10] Storn R, Price K. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. *International Computer Science Institute, Berkley*, 1995.

[11] Wu L H, Wang Y N, Yuan X F, et al. Differential evolution algorithm with adaptive second mutation. *Control and Decision*, 2006, 21(8): 898–902.

Wu Lianghong was born in 1978. He is currently a lecturer at the School of Information and Electric Engineering, Hunan University of Science and Technology, and a Ph. D. candidate of the College of Electric and Information Engineering, Hunan University. His research interests are intelligent control and computation intelligent. E-mail: lhwu©hnust.edu.cn

Wang Yaonan was born in 1957. He is currently a professor and Ph. D. supervisor at Hunan University. His research interests are intelligent control, computation intelligent, and image processing.

Zhou Shaowu was born in 1964. He is currently a full-time professor and M. S. supervisor at Hunan University of Science and Technology. His research interests are nonlinear system control and robust control.