Adaptive Optimal Digital PID Controller

Yanxia Sun[1, a], Zenghui Wang[2,b]

[1]Department of Electrical Engineering, Tshwane University of Technology, Pretoria 0001, South Africa

[2]Department of Electrical and Mining Engineering, University of South Africa, Florida 1710, South Africa

[a]sunyanxia@gmail.com, [b]wangzengh@gmail.com

Keywords: Particle swarm optimization, PID controller, Adaptive control, Parameter tuning.

Abstract. It is necessary to change the parameters of PID controller if the parameters of plants change or there are disturbances. Particle swarm optimization algorithm is a powerful optimization algorithm to find the global optimal values in the problem space. In this paper, the particle swarm optimization algorithm is used to identify the model of the plant and the parameter of digital PID controller online. The model of the plant is identified online according to the absolute error of the real system output and the identified model output. The digital PID parameters are tuned based on the identified model and they are adaptive if the model is changed. Simulations are done to validate the proposed method comparing with the classical PID controller.

Introduction

Proportional-plus-integral-plus-derivative (PID) controller is one of the earliest control method that is used widely in industry because of its easy implementation, robust performance and being simple of the principal of parameters. Many studies suggest that of all the controllers in industrial process control, PID (or PI) controllers are used in 95-97% of the cases [1]. The PID controller involves three separate parameters, and is accordingly sometimes called three-term control: the proportional, the integral and the derivative values, denoted P, I, and D. Simply put, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. The control system using PID controller is illustrated in in Fig. 1 and the transfer function of PID controller is

$$C(s) = K_p + \frac{K_i}{s} + K_d s \,. \tag{1}$$

Here, K_p is the proportional term parameter, K_i is the integral term parameter and K_d is the derivative term parameter. Some applications may require using only one or two terms of PID to provide the appropriate system control. This is achieved by setting the other parameters to zero; and a PID controller will be called a PI, PD, P or I controller in the absence of the respective control terms. Moreover, a digital PID controller is typically necessary as algorithms running in a computer, a microcontroller, an ASIC, or a flexible hardware platform such as FPGA. The bilinear transformation (2) of the PID controller (1) can be used in the digital control system.

$$C_D(z) = K_p\left(1 + \frac{1}{T_i}\frac{T}{2}\frac{z+1}{z-1} + T_d\frac{2}{T}\frac{z-1}{z+1}\right) \tag{2}$$

The tuning methods of PID controller are adjusting the proportional, the integral and the derivative parameters to make the control system stable and achieve good acceptable control performance. [2] For achieving appropriate closed-loop control performance, these three PID parameters need be tuned [1][2]. The tuning methods of PID parameters are classified as traditional and intelligent methods. The traditional methods such as Zigler-Nichols [3] and Cohen Coon [4] are hard to get optimal PID parameters and usually they are difficult to achieve good tuning, i.e. they produce surge and big overshoot, and Cohen Coon is an off line tuning method and only good for first order system.

Recently, some intelligent tuning methods such as genetic algorithm [5-6], particle swarm optimization algorithm [7] and artificial fish swarm algorithm [8] have been proposed. However, the genetic algorithm may be not efficient for solving complex optimization problems, whose parameters are highly correlated [9-10]. Moreover, it is difficult to achieve good control performance using these intelligent tuning methods if there are uncertainties in systems.

In this paper a particle swarm optimization based adaptive digital PID controller is proposed. The parameters are tuning according to the identified model in the discrete time domain. The tuning method and identifying the model of the plant are all based on particle warm optimization.

The rest parts of the paper are organized as follows. Section II gives a brief description of particle swarm optimization. Particle swarm optimization based adaptive digital PID controller is given in section III. Simulations and comparisons are given in section IV to validate the proposed method. Finally, the concluding remarks are given in the last section.

A Brief Introduction of Particle Swarm Optimization

Particle swarm optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart [11] in 1995: it is a population-based optimization technique, inspired by the motion of bird’s flocking, or fish schooling. The particle swarms are social organizations whose overall behavior relies on some sort of communication amongst members, and cooperation. All members obey a set of simple rules that model the communication within the flock, between the flocks and the environment. Each solution is a “bird” in the flock and is referred to as a “particle”. PSO is not largely affected by the size and nonlinearity of the problem, and can converge to the optimal solution in many problems [11-15] where most analytical methods fail to converge. It can, therefore, be effectively applied to different optimization problems.

The standard particle swarm algorithm works by iteratively searching in a region and is concerned with the best previous success of each particle, the best previous success of the particle swarm as a whole, the current position and the velocity of each particle [14]. The particle searches the domain of the problem, according to

$$V_i(t+1) = \omega V_i(t) + c_1 R_1 (P_i - X_i(t)) + c_2 R_2 (P_g - X_i(t)), \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

where $V_i = \left[v_i^1, v_i^2, \cdots v_i^n\right]$ is the velocity of particle i; $X_i = \left[x_i^1, x_i^2, \cdots x_i^n\right]$ represents the position of particle i; P_i represents the best previous position of particle i (indicating the best discoveries or previous experience of particle i); P_g represents the best previous position among all particles (indicating the best discovery or previous experience of the social swarm); ω is the inertia weight that controls the impact of the previous velocity of the particle on its current velocity and is sometimes adaptive. R_1 and R_2 are two random weights whose components r_1^j and r_2^j $(j = 1, 2 \cdots, n,)$ are chosen uniformly within the interval $[0,1]$ which might not guarantee the convergence of the particle trajectory; c_1 and c_2 are the positive constant parameters. Generally the value of each component in V_i should be clamped to the range $[-v_{\max}, v_{\max}]$ to control excessive roaming of particles outside the search space.

The following procedure can be used for implementing the PSO algorithm [14].

1) Initialize the swarm by assigning a random position in the problem hyperspace to each particle.

2) Evaluate the fitness function for each particle.

3) For each individual particle, compare the particle’s fitness value with its best experience p_i. If the current value is better than this value, then set this value as the current particle’s position x_i as p_i.

4) Identify the particle that has the best fitness value. The value of its fitness function is identified as g_{best} and its position as p_g.

5) Update the velocities and positions of all the particles using (1) and (2).

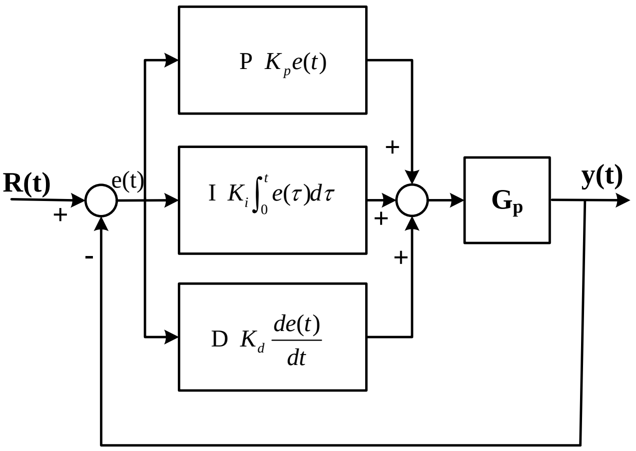
6) Repeat steps 2)–5) until a stopping criterion is met (e.g., maximum number of iterations or a sufficiently good fitness value).

Particle Swarm Optimization Based Adaptive Digital PID Control

Before using PSO to tuning the PID parameters, the fitness function needs to be determined. Here, the digital PID controller design is the focus of this paper and the fitness function is chosen as

$$F = \sum_{k=k_0}^{round(T_0/T)} (k-k_0)\left|e(k)\right|. \tag{3}$$

Here, T_0 is the time period for tuning the PID parameters, T is the sample time, function round(x) rounds the value x to the nearest integer, and k_0 is the tuning time in the discrete domain. The PSO based tuning method is based on the identified discrete model and it is not necessary to get the exact information of the plant. For the delay system, only the order of the system needs be some higher. The general control structure of PID control system is shown in Fig. 1 and the control structure of the digital PID control is changed to Fig. 2. As can be seen from Fig. 1 and Fig. 2, the sampler T and data hold ZOD are added in the control system since the controller is a digital controller.



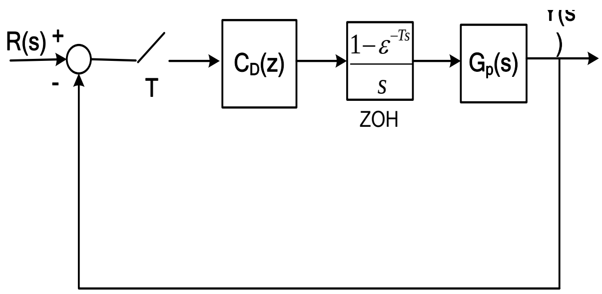


Fig. 1 PID control structure with unit feedback

Fig. 2 Digital PID control structure

To realize the adaptive PID tuning method, the model of the plant is necessary. Here, the model $\hat{G}_p(z)$ is identified using the PSO and its form is

$$Y(z)(1+a_{n_a-1}z^{-1}+\cdots+a_0z^{-n_a}) = U(z)(b_{n_b-1}z^{-1}+\cdots+b_0z^{-n_b}) \tag{4}$$

The corresponding difference equation is

$$y(k) = b_{n_b-1}u(k-1)+\cdots+b_0u(k-n_b)-(a_{n_a-1}y(k-1)+\cdots+a_0y(k-n_a)) \tag{5}$$

The fitness function for identifying the parameters $b_0, b_1, \cdots, b_{n_b-1}, a_0, a_1, \cdots, a_{n_a-1}$ is

$$F_{\text{model}} = \left|y(k)-\hat{y}(k)\right|. \tag{6}$$

The structure of PSO based adaptive digital PID control is shown in Fig. 3.

For the PSO based adaptive digital PID control method, firstly PSO is used to identify the parameters $b_0, b_1, \cdots, b_{n_b-1}, a_0, a_1, \cdots, a_{n_a-1}$ of the model (4) or (5) based on the fitness function (6), and then PSO using the identified model and fitness function (3) optimize the parameters of the PID controller. It should be noted that the orders n_a and n_b should be some large if there is dead time/delay.

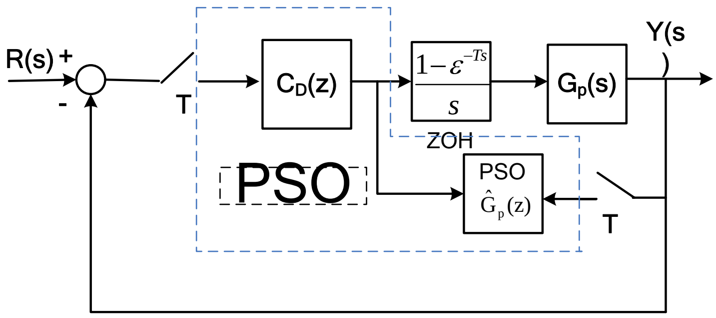


Fig. 3 PSO based adaptive digital PID control

Simulations and Comparison

To validate the proposed method, a fourth-order-plus-dead-time model is used [2].

$$G_p(s) = \frac{K}{(s+1)^4} e^{-0.2s} \quad (7)$$

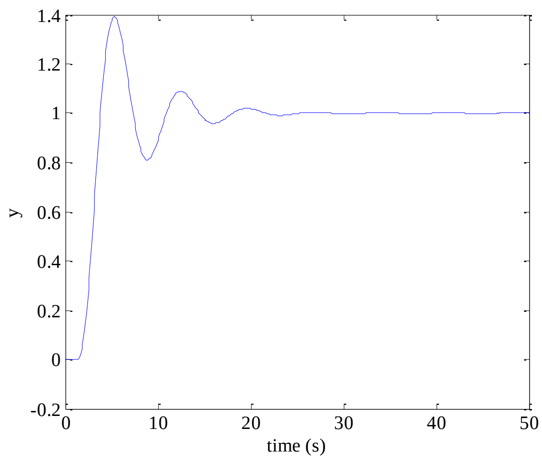
Here K is a parameter of this plant, and (7) is the original system in [2] when K = 1.

By applying the Ziegler-Nichols tuning rules, the classical digital PID controller [2] is

$$C(z) = 2.25(1 + \frac{1}{3.2}\frac{0.1}{2}\frac{z+1}{z-1} + 0.8\frac{2}{0.1}\frac{z-1}{z+1}). \quad (8)$$

The unit step response using the classical digital PID controller is shown in Fig. 4.

When the proposed method is used, the unit step response is shown in Fig. 5. As can be seen from Fig. 4 and Fig. 5, there is no overshoot for the proposed method and the set time is also shorter than the one using the classical digital PID controller.



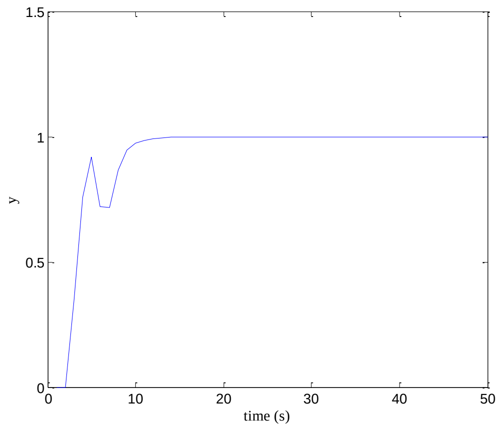
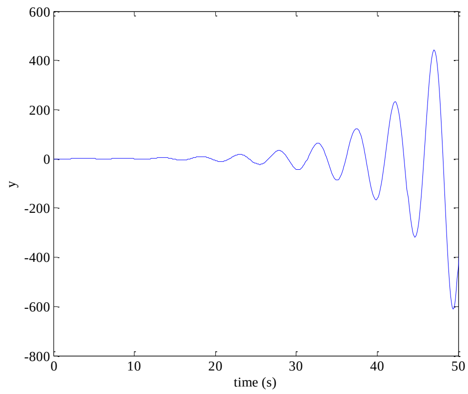


Fig. 4. Step response of the classical PID with K=1 Fig. 5. Step response of PSO based PID with K=1

To check the robustness of the controllers, the parameter K was changed to 3 from 1 and the unit step responses are shown in Fig. 6 and Fig. 7 using the classical PID controller and the proposed PID controller, respectively. As can be seen from Fig. 6 and Fig. 7, the classical PID cannot stabilize this system; however, the proposed method can stabilize the system.



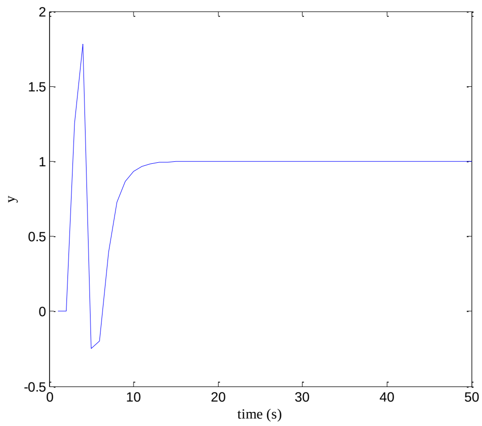
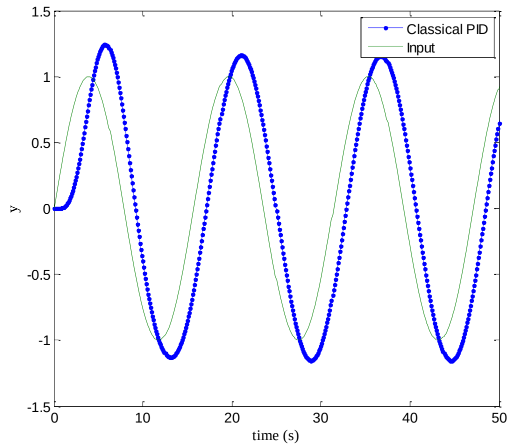


Fig. 6. Step response of the classical PID with K=3 Fig. 7. Step response of PSO based PID with K= 3

As this system includes dead time term, the sine function sin(0.4t) is chosen as the input trajectory to check the control performance. The time responses using the classical PID and the proposed method are shown in Fig. 8 and Fig. 9; and the tracking errors are shown in Fig. 10. As can be seen from Fig. 8, Fig. 9 and Fig. 10, the proposed the method can reduce the effects of time delay as the model considers the time delay.



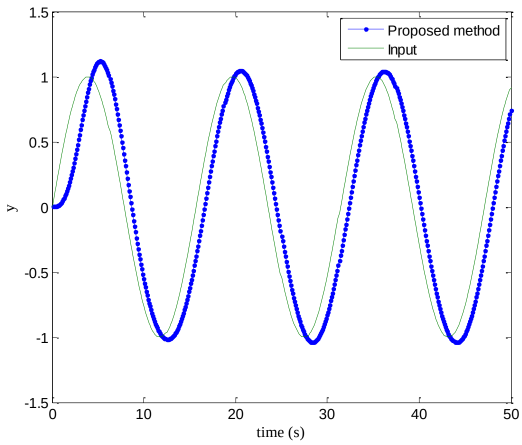


Fig. 8. Time response using the classical PID when K =1 and the input is sin(0.4t)

Fig. 9. Time response using the proposed method when K =1 and the input is sin(0.4t)

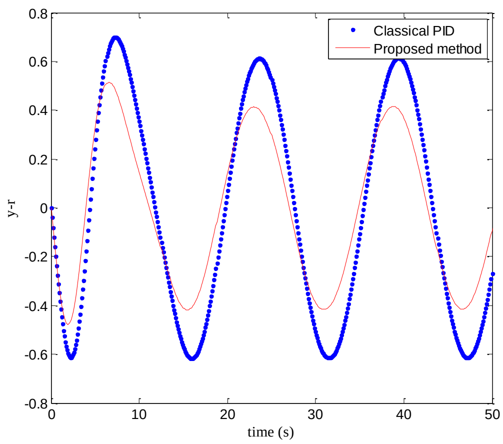


Fig. 10. Time response errors

Summary

This paper proposed a particle swarm optimization based adaptive digital PID control method. The parameter tuning method was based on an identified model and the model for PID tuning is also identified online based on particle swarm optimization. The simulations showed that the proposed method could achieve better control performance if there are uncertainties in the system. Moreover, the proposed method can also reduce the dead time effects. However, this paper did not concern the constrained PID controller based on the particle swarm optimization and this research topic will be our research focus in the future.

Acknowledgment

This work was supported by China/South Africa Research Cooperation Programme (No. 78673 & CS06-L02), South African National Research Foundation Incentive Grant (No. 81705), SDUST Research Fund (No. 2010KYTD101) and Key Scientific Support Program of Qingdao City (No. 11-2-3-51-nsh).

References

[1] H. N. Koivo, “Book review: Practical PID control-Antonio Visioli”, IEEE Transactions on Automatic Control, vol. 53, pp. 2217-2218, 2008.

[2] M. Sam Fadali, Ditital Control Engineering: Analysis and design, Acadamic Press, 2009.

[3] J. G. Zigeler, "Nichols N B. optimization setting for automatic controller," Trans. ASME, Vol. 64, No. 11, pp.756-769, 1942.

[4] G. H. Cohen and G. A. Coon, Theoretical Consideration of Retarded Control, Trans. ASME, vol. 75, pp. 827-834, 1953

[5] H. Han, and A. Luo, "Nonlinear PID controller based on genetic tuning algorithm," Control and Decision, Vol. 20, No.4, pp.448-450, 2005.

[6] J. S. Yang, "PID control for a binary distillation column using a genetic searching algorithm," WSEAS Trans. Systems, Vol. 5, No.4, pp.720726,2006.

[7] T. H. Kim, Maruta, and T. Sugie, "Robust PID controller tuning based on the constrained particle swarm optimization," Automatica, vol. 44, Issue 4, pp. 1104-1110, April 2008.

[8] Y. Luo, J. Zhang, and X. Li, "The optimization of PID controller parameters based on artificial fish swarm algorithm," in: proc. the IEEE International Conf. Automation and Logistics, Jinan, China, pp. 1058-1062, August 2007.

[9] D. B. Fogel, “Evolutionary computation: toward a new philosophy of a machine intelligence”, IEEE Press, New York, 2000.

[10] A. Jalilvand, A. Kimiyahgahalam, A. Ashouri, M. Mahadavi, “Advanced Particle Swarm Optimization-Based PID Controller Parameters Tuning”, 12th IEEE International Multitopic Conference, Karachi , pp.429-435, Dec. 2008

[11] J. Kennedy, R. C. Eberhart, “Particle Swarm Optimisation”, Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942-1948. 1995.

[12] X. Hu, Y. Shi, R. Eberhart, “Recent Advances in Particle Swarm”, Congress on Evolutionary Computation, pp. 90-97. Piscataway, IEEE Service Center, 2004.

[13] C. M Huang, C. J. Huang, M. L. Wang, “A Particle Swarm Optimisation to Identifying the ARMAX Model for Short term Load Forecasting”, IEEE Transactions on Power Systems. 20, pp.1126-1133, 2005.

[14] M. Clerc, Particle Swarm Optimisation. ISTE Publishing Company, 2006.

[15] N. Nedjah, L. D. M. Mourelle, Systems Engineering Using Particle Swarm Optimisation, Nova Science Publishers, 2007.