Information Sciences 179 (2009) 1007–1018

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Self-organizing genetic algorithm based tuning of PID controllers

Zhang Jinhua *, Zhuang Jian, Du Haifeng, Wang Sun'an

School of Mechanical Engineering, Xi'an Jiaotong University, No. 28, Xianning West Road, Xi'an, Shaanxi Province 710049, People's Republic of China

ARTICLE INFO

Article history:
Received 14 July 2007
Received in revised form 17 November 2008
Accepted 19 November 2008

Keywords:
Genetic algorithm
Cyclic mutation
Dominant selection
Self-organizing
PID controller

ABSTRACT

This paper proposes a self-organizing genetic algorithm (SOGA) with good global search properties and a high convergence speed. First, we introduce a new dominant selection operator that enhances the action of the dominant individuals, along with a cyclical mutation operator that periodically varies the mutation probability in accordance with evolution generation found in biological evolutionary processes. Next, the SOGA is constructed using the two operators mentioned above. The results of a nonlinear regression analysis demonstrate that the self-organizing genetic algorithm is able to avoid premature convergence with a higher convergence speed, and also indicate that it possesses self-organization properties. Finally, the new algorithm is used to optimize Proportional Integral Derivative (PID) controller parameters. Our simulation results indicate that a suitable set of PID parameters can be calculated by the proposed SOGA.

1. Introduction

Proportional Integral Derivative (PID) controllers have the advantage of simple structure, good stability, and high reliability. Accordingly, PID controllers are widely used to control system outputs, especially for systems with accurate mathematical models. The key issue for PID controllers is the accurate and efficient tuning of parameters. In practice, controlled systems usually have some features, such as nonlinearity, time-variability, and time delay, which make controller parameter tuning more complex. Moreover, in some cases, system parameters and even system structure can vary with time and environment. As a result, the traditional PID parameter tuning methods are not suitable for these difficult calculations. With the aid of Genetic Algorithms (GAs), Artificial Neural Networks and Fuzzy Logic, many researchers have recently proposed various alternative, intelligent PID controllers [2].

As a popular optimization algorithm, the GA had been widely used to turn PID parameters. Soltoggio [16] proposed an improved GA for tuning controllers in classical first and second order plants with actuator nonlinearities. Chen and Wang [3] used the population-based distribution GA to optimize a PID controller, and found that the search capability of the algorithm was improved by competition among distribution populations in order to reduce the search zone. The PID controllers based on GAs have good performance and have been applied in practice. However, the standard GAs have some disadvantages, such as premature convergence and low convergence speed. In this paper, a self-organizing Genetic Algorithm (SOGA) is proposed. This new algorithm improves the efficiency of global search and was found to avoid premature convergence. Finally, the SOGA is employed to optimize PID parameters.

The organization of this paper is as follows. In Section 2, the structure of the GA-PID controller is discussed. In Section 3, the dominant selection and the cyclic mutation operator are proposed. In Section 4, the self-organizing genetic algorithm is described and its performance is analyzed. Results are presented in Section 5 and conclusions in Section 6.

---

* Corresponding author. Tel.: +86 29 82665712; fax: +86 29 82663872.
E-mail address: jjshua@mail.xjtu.edu.cn (J. Zhang).

0020-0255/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2008.11.038

2. Structure of GA-PID controller

2.1. PID control law

A PID controller is a feedback controller that makes a plant less sensitive to changes in the surrounding environment as well as to small changes in the plant itself. The continuous form of a PID controller, with input e and output u, is given by

$$u(t) = K_\mathrm{p}e(t) + K_\mathrm{i}\int_0^t e(t)\mathrm{d}t + K_\mathrm{d}\frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{1}$$

where $u(t)$ is the controlled output, K_p is the proportional gain, K_i and K_d stand for the integral and derivative gains, respectively.

The discrete velocity-type PID control law is described as

$$\Delta u(k) = K_\mathrm{p}[e(k) - e(k-1)] + K_\mathrm{i}T_\mathrm{s}e(k) + \frac{K_\mathrm{d}}{T_\mathrm{s}}[e(k) - 2e(k-1) + e(k-2)] \tag{2}$$

where K_p, K_i, and K_d are PID turning parameters, and T_s stands for the time constant.

2.2. Fitness function

GAs search for the optimal solution by maximizing a given fitness function, that is, an evaluation function that provides a measure of the quality of the solution to the problem. In the control process, the objective is to minimize the cost function, defined as the Integral Absolute Error (IAE), which determines the performance of any industrial process. At the same time, the square of the controller output is included to avoid exporting a large control value. Thus, the cost function is written as

$$F(t) = \int_0^\infty (\varpi_1|e(t)| + \varpi_2u^2(t))\mathrm{d}t + \varpi_3t_r \tag{3}$$

where $e(t)$ and $u(t)$ are used to represent the system error and the control output at time t, t_r is the rising time, and ω_i ($i = 1,2,3$) are weight coefficients.

To avoid overshooting, a penalty value is adopted in the cost function. That is, once overshooting occurs, the value of overshooting is added to the cost function. Hence, Eq. (3) is rewritten as

$$\begin{aligned}
&\textit{if } ey(t) < 0\\
&\quad F(t) = \int_0^\infty (\varpi_1|e(t)| + \varpi_2u^2(t) + \varpi_4|ey(t)|)\mathrm{d}t + \varpi_3t_r\\
&\textit{else}\\
&\quad F(t) = \int_0^\infty (\varpi_1|e(t)| + \varpi_2u^2(t))\mathrm{d}t + \varpi_3t_r
\end{aligned} \tag{4}$$

where, ω_4 is a coefficient and $\omega_4 \gg \omega_1$, $ey(t) = y(t) - y(t-1)$, and $y(t)$ is the output of the controlled object. The minimization objective function is transformed to be a fitness function:

$$\mathrm{Fitness} = 1/(F(t) + 1) \tag{5}$$

3. Genetic algorithm

A GA is an intelligent optimization technique that relies on the parallelism found in nature, in particular its searching procedures are based on the mechanics of natural selection and genetics. GAs were first conceived in the early 1970s by Holland [7]. GAs are used regularly to solve difficult search, optimization, and machine-learning problems that have previously resisted automated solutions [12]. They can be used to solve difficult problems quickly and reliably. These algorithms are easy to interface with existing simulations and models, and they are easy to hybridize. GAs include three major operators: selection, crossover, and mutation, in addition to four control parameters: population size, selection pressure, crossover and mutation rate. Population-based optimization methods are addressed in [10,13]. This paper is concerned primarily with the selection and mutation operators.

3.1. Selection

Selection is a genetic operator that chooses a chromosome (individual) from the current generation's population to be included in the population of next generation. The selection operator is the algorithmic embodiment of evolutionary biology. Selection demonstrates the phenomenon of "survival of the fittest" and determines the evolutionary trajectory of the GA. Current selection operators include roulette, tournament, top percent, and best selection. The use of a selection operator can-