Neurocomputing 138 (2014) 180–188

Contents lists available at ScienceDirect

Neurocomputing

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

Binary-coded extremal optimization for the design of PID controllers

Guo-Qiang Zeng [a,*], Kang-Di Lu [a], Yu-Xing Dai [a], Zheng-Jiang Zhang [a], Min-Rong Chen [b], Chong-Wei Zheng [a], Di Wu [a], Wen-Wen Peng [a]

[a] Department of Electrical and Electronic Engineering, Wenzhou University, Wenzhou 325035, China
[b] College of Information Engineering, Shenzhen University, Shenzhen 518060, China

ARTICLE INFO

Article history:
Received 10 November 2013
Received in revised form
2 January 2014
Accepted 23 January 2014
Communicated by Zheng-Guang Wu
Available online 24 February 2014

Keywords:
PID controllers
Extremal optimization
PID parameters
Multivariable plant

ABSTRACT

Design of an effective and efficient PID controller to obtain high-quality performances such as high stability and satisfied transient response is of great theoretical and practical significance. This paper presents a novel design method for PID controllers based on the binary-coded extremal optimization algorithm (BCEO). The basic idea behind the proposed method is encoding the PID parameters into a binary string, evaluating the control performance by a more reasonable index than the integral of absolute error (IAE) and the integral of time weighted absolute error (ITAE), updating the solution by the selection based on power-law probability distribution and binary mutation for the selected bad elements. The experimental results on some benchmark instances have shown that the proposed BCEO-based PID design method is simpler, more efficient and effective than the existing popular evolutionary algorithms, such as the adaptive genetic algorithm (AGA), the self-organizing genetic algorithm (SOGA) and probability based binary particle swarm optimization (PBPSO) for single-variable plants. Moreover, the superiority of the BCEO method to AGA and PBPSO is demonstrated by the experimental results on the multivariable benchmark plant.

1. Introduction

It has been widely recognized that Proportional-Integral-Derivative (PID) control is still one of the simplest but most efficient control strategies for many real-world control problems [1–4], although a variety of advancements have been gained in control theories and practices. How to design and tune an effective and efficient single-variable and especially multivariable PID controller to obtain high-quality performances such as high stability and satisfied transient response is of great theoretical and practical significance. This issue has attracted considerable attentions by some researchers using evolutionary algorithms [5,6], such as the genetic algorithm (GA) [7,8], particle swarm optimization (PSO) [9–12], differential evolution (DE) [13,14], and multi-objective optimization algorithms [6,15]. However, the issue of designing and tuning PID controllers efficiently and adaptively is still open. As a consequence, this paper focuses on addressing this issue by adopting another novel optimization algorithm called binary-coded extremal optimization in the attempt to obtain better performances.

Originally inspired by far-from-equilibrium dynamics of self-organized criticality (SOC) [16,17], extremal optimization (EO) [18,19] provides a novel insight into optimization domain because it merely selects against the bad instead of favoring the good randomly or according to a power-law distribution. The basic EO algorithm and its modified versions have been successfully applied to a variety of benchmark and real-world engineering optimization problems, such as graph partitioning [20], graph coloring [21], traveling salesman problem [22,23], maximum satisfiability (MAX-SAT) problem [24–26], heat pipe optimal design [27], and steel production scheduling [28]. The more comprehensive introduction concerning EO is referred to the surveys [29,30]. However, there are only few reported researches concerning the design of PID controllers based on EO. In [31], an improved generalized EO algorithm is proposed for designing two-degree-of-freedom PID regulator. This paper focuses on a generalized design framework based on binary coded EO (BCEO) for PID controllers, especially for more complex multivariable PID controllers. The basic idea behind the BCEO-based PID controller design method is encoding the PID parameters into a binary string, evaluating the control performance by a more reasonable index than the integral of absolute error (IAE) and the integral of time weighted absolute error (ITAE), updating the solution by the selection based on power-law probability distribution and binary mutation for the selected bad elements. Comparing with the existing popular evolutionary algorithms, e.g., adaptive GA (AGA) [32], self-organizing genetic

* Corresponding author.
E-mail addresses: zeng.guoqiang5@gmail.com, zenggq@wzu.edu.cn (G.-Q. Zeng).

http://dx.doi.org/10.1016/j.neucom.2014.01.046
0925-2312

algorithm (SOGA) [8], probability based binary PSO (PBPSO) [11], etc., the proposed BCEO method in this paper is simpler, more efficient and effective. Its superiority is demonstrated by the experimental results on some benchmark single-variable and multivariable instances.

The rest of this paper is organized as follows. Section 2 presents preliminaries concerning on PID controller and EO used in this paper. In Section 3, the BCEO algorithm for the design of PID controllers is proposed. The experimental results on benchmark engineering instances are given and discussed in Section 4. Finally, we give the conclusion and open problems in Section 5.

2. PID controllers and extremal optimization

2.1. PID controllers and its performance index

A standard control system with a PID controller $D(s)$ and controlled plant $G(s)$ is shown in Fig. 1. Let us consider firstly the simplest case, single-input and single-output control system. The transfer function $D(s)$ of a standard single-variable PID controller [1] is generally expressed as the following form:

$$D(s) = K_P\left(1+\frac{1}{T_I s}+T_D s\right) = K_P + K_I\frac{1}{s}+K_D s \tag{1}$$

where T_I and T_D are the integral time constant and the derivative time constant, respectively, K_P, K_I, and K_D are the proportional gain, the integral gain, and the derivative gain, respectively, $K_I = K_P/T_I$ and $K_D = K_P T_D$.

The output $U(s)$ of PID controller is described as follows:

$$U(s) = D(s)E(s) = K_P\left(1+\frac{1}{T_I s}+T_D s\right)E(s) = K_P E(s)+K_I\frac{1}{s}E(s)+K_D sE(s) \tag{2}$$

where $E(s)$ is the transfer function of the system error $e(t)$. Furthermore, the continuous-time form of $U(s)$ is also written as the following equation:

$$u(t) = K_P e(t)+K_I\int_0^t e(t)dt+K_D\frac{de(t)}{dt} \tag{3}$$

The discrete PID controller is described as follows:

$$u(k) = K_P e(k)+K_I T_s\sum_{j=0}^{k} e(j)+\frac{K_D}{T_s}[e(k)-e(k-1)] \tag{4}$$

where T_s is the sampling time.

Then, consider more complex case, an $n \times n$ multivariable plant $G(s)$ [7] in Fig. 1, which is given as follows:

$$G(s) = \begin{bmatrix} g_{11}(s) & \cdots & g_{1n}(s) \\ \vdots & \ddots & \vdots \\ g_{n1}(s) & \ldots & g_{nn}(s) \end{bmatrix} \tag{5}$$

The corresponding $n \times n$ multivariable PID controller is given as follows:

$$D(s) = \begin{bmatrix} d_{11}(s) & \cdots & d_{1n}(s) \\ \vdots & \ddots & \vdots \\ d_{n1}(s) & \ldots & d_{nn}(s) \end{bmatrix} \tag{6}$$

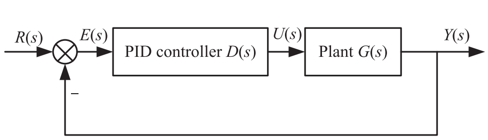

Fig. 1. A control system with PID controller.

where the form of $d_{ij}(s)$ is characterized as the following equation:

$$d_{ij}(s) = K_{Pij}+K_{Iij}\frac{1}{s}+K_{Dij}s, \quad \forall i,j \in \{1,2,\ldots,n\} \tag{7}$$

In most of previous research work, the integral of absolute error (IAE) and the integral of time weighted absolute error (ITAE) are generally used as the indices measuring the performances of PID controllers [1]. However, these above indices are still not sufficient to evaluate the control performances comprehensively [8]. Here, another much more reasonable performance index is presented by considering the following additional factors. The first one is introduction of the square of the controllers' output, i.e., $\int_0^\infty w_2 u^2(t)dt$ in order to avoid exporting a large control value. Secondly, the rising time $w_3 t_u$ is used to evaluate the rapidity of the step response of a control system. The third one $\int_0^\infty w_4|\Delta y(t)|dt$ is added to avoid a large overshoot value.

Definition 1. The objective function (also called fitness) evaluating the control performance of a single-variable PID controller is defined as follows [8]:

$$\min J = \min\begin{cases} \int_0^\infty (w_1|e(t)|)+w_2u^2(t))dt+w_3t_u, & \text{if } \Delta y(t)\ge 0 \\ \int_0^\infty (w_1|e(t)|)(+w_2u^2(t)+w_4|\Delta y(t)|)dt+w_3t_u, & \text{if } \Delta y(t)<0 \end{cases} \tag{8}$$

where $e(t)$ is the system error, $\Delta y(t) = y(t)-y(t-\Delta t)$, $u(t)$ is the control output at the time t, t_u is the rising time, w_1–w_4 are the weight coefficients, and $w_4 \gg w_1$.

Definition 1 can be generalized for evaluating a multivariable PID controller.

Definition 2. The objective function that evaluates the control performance of a multivariable PID controller is defined as follows:

$$\min J = \min\begin{cases} \int_0^\infty (w_1\sum_{i=1}^{n}|e_i(t)|+w_2\sum_i^n u_i^2(t))dt+w_3\sum_{i=1}^{n}t_{ui}, & \text{if } \Delta y_i(t)\ge 0 \\ \int_0^\infty (w_1\sum_{i=1}^{n}|e_i(t)|+w_2\sum_i^n u_i^2(t)+w_4\sum_{i=1}^{n}|\Delta y_i(t)|)dt+w_3\sum_{i=1}^{n}t_{ui}, & \text{if } \Delta y_i(t)<0 \end{cases}. \tag{9}$$

where $e_i(t)$ is the i-th system error, $\Delta y_i(t) = y_i(t)-y_i(t-\Delta t)$, $u_i(t)$ is the i-th control output at the time t, t_{ui} is the rising time of the i-th system output y_i, w_1–w_4 are the weight coefficients, and $w_4 \gg w_1$.

2.2. Extremal optimization

The key operations of EO include the evaluation of global and local fitness, selection of bad variables or elements, mutation and improvement of the selected bad variables or elements. The basic probability-based EO algorithm [18] is described as follows:

(1) Initialize configuration S randomly and set $S_{best}=S$ and $C(S_{best})=C(S)$, where S_{best} is the best solution so far and $C(S_{best})$ is the global fitness of S_{best}.
(2) For the current configuration S,
(a) Evaluate the local fitness λ_i for each variable x_i and rank all the variables according to λ_i, i.e., find a permutation Π_1 of the labels i such that $\lambda_{\Pi_1(1)} \ge \lambda_{\Pi_1(2)} \ge \ldots \ge \lambda_{\Pi_1(n)}$.
(b) Select a rank $\Pi_1(k)$ according to a probability distribution $P(k)$, $1 \le k \le n$ and denote the corresponding variable as x_j.
(c) Generate the new solution S_{new} so that x_j must be according to some mutation rules.
(d) If $C(S_{new}) < C(S_{best})$ then $S_{best}=S_{new}$.
(e) Accept $S=S_{new}$ unconditionally.
(3) Repeat at step (2) as long as desired.
(4) Return S_{best} and $C(S_{best})$.

It is obvious that the probability distributions used for selection of bad variables or elements play critical roles in controlling the performances of the above EO algorithm. Power-law distribution