Computers and Mathematics with Applications 60 (2010) 1088–1104

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Chaos driven evolutionary algorithms for the task of PID control

Donald Davendra [a,*], Ivan Zelinka [b,a], Roman Senkerik [a]

[a] *Tomas Bata University in Zlin, Faculty of Applied Informatics, Department of Informatics and Artificial Intelligence, Nad Stranemi 4511, Zlin 76001, Czech Republic*

[b] *Faculty of Electrical Engineering and Computing Science, Technical University of Ostrava, Tr. 17. Listopadu 15, Ostrava, Czech Republic*

A R T I C L E I N F O

Keywords:
Chaos maps
Differential Evolution
Self-organizing migrating algorithm
PID controller

A B S T R A C T

Chaos driven Differential Evolution algorithm and Self-Organizing Migrating Algorithm are presented in this paper for the task of PID (Proportional–Integral–Derivative) controller optimization. The dissipative chaotic Lozi map is embedded as a number generator inside DE and SOMA in order to avoid local optima stagnation and embed a superior search strategy. Three unique PID controller problems are presented and successfully resolved using these new approaches. The obtained results compare favorably with published results.

1. Introduction

A Proportional–Integral–Derivative (PID) controller is a three-term controller that has a long history in the automatic control field, starting from the beginning of the last century [1]. Owing to its intuitiveness and its relative simplicity, in addition to satisfactory performance which it is able to provide with a wide range of processes, it has become in practice the standard controller in industrial settings.

Evolutionary design and synthesis of PID controllers is a recent manifestation. Evolutionary algorithms (EAs) are considered as a powerful set of tools in the task of any complex optimization. EAs arose with the advent of computers and has now evolved into a very complex set of algorithms. Almost all conceived natural occurring phenomena have been translated into a set of evolutionary heuristics. These heuristics in turn have been applied to the task of PID controller design in its various formats. Chang [2] developed a new variant of Genetic Algorithm (GA) [3] for PID tuning, whereas Yachen and Yueming [4] developed a Simulated Annealing (SA) [5] approach. Dong [6] and Chakraborty [7] during the past few years have compared the performances of Differential Evolution (DE) and Particle Swarm Optimization (PSO) [8] in the task of PID controller design. Further application can be found in [9–11] amongst others.

This research deals with the application of EAs driven by *chaotic maps* in PID controller design. Recent research in chaos driven heuristics has been fueled with the predisposition that unlike stochastic approaches, a chaotic approach is able to bypass local optima stagnation. This one clause is of deep importance to EAs. A chaotic approach generally uses the chaotic map in the place of a random number generator [12]. This causes the heuristic to map unique regions, since the chaotic map iterates to new regions. The onus is then to select a very good chaotic map as the random generator.

Davendra and Zelinka [13] embedded chaotic maps inside DE and compared the performance of canonical and chaos mutated DE in PID controller design. The results confirmed that chaos driven DE performs better than canonical DE over a set of PID problems. This paper builds on the previous work of Davendra and Zelinka [13] with the application of chaos driven Self Organizing Migrating Algorithm (SOMA).

[*] Corresponding author.
E-mail addresses: davendra@fai.utb.cz (D. Davendra), zelinka@fai.utb.cz (I. Zelinka), senkerik@fai.utb.cz (R. Senkerik).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.03.066

This paper is divided into the following parts. Sections 2 and 3 introduces the basic principles of PID specifications and controller tuning. Section 4 describes DE and SOMA. Section 5 gives the mathematical description of the *Lozi Map*. Section 6 gives the experimentation results of the three attempted PID controller tuning problems and finally, in Section 7 the conclusions are presented.

2. PID controller

The PID controller contains three unique parts; proportional, integral and derivative controller [14]. The following sections gives a brief description of the different components.

2.1. Proportional algorithm

The mathematical representation is given as,

$$\frac{mv(s)}{e(s)} = k_c \tag{1}$$

in the Laplace domain or as,

$$mv(t) = mv_{ss} + k_c e(t) \tag{2}$$

in the time domain.

The proportional mode adjusts the output signal in direct proportion to the controller input (which is the error signal, **e**). The adjustable parameter to be specified is the controller gain, k_c. The larger the k_c, the more the controller output will change for a given error [14].

2.2. Proportional integral algorithm

The mathematical representation is given as,

$$\frac{mv(s)}{e(s)} = k_c \left[1 + \frac{1}{T_i s}\right] \tag{3}$$

in the Laplace domain or as,

$$mv(t) = mv_{ss} + k_c \left[e(t) + \frac{1}{T_i}\int e(t)\mathrm{d}t\right] \tag{4}$$

in the time domain.

The additional integral mode corrects for any offset (error) that may occur between the desired value (setpoint) and the process output automatically over time. The adjustable parameter to be specified is the integral time (T_i) of the controller [14].

2.3. Proportional integral derivative algorithm

The mathematical representation is given as,

$$\frac{mv(s)}{e(s)} = k_c \left[1 + \frac{1}{T_i s} + T_D s\right] \tag{5}$$

in the Laplace domain or as,

$$mv(t) = mv_{ss} + k_c \left[e(t) + \frac{1}{T_i}\int e(t)\mathrm{d}t + T_D \frac{\mathrm{d}e(t)}{\mathrm{d}t}\right]. \tag{6}$$

Derivative action anticipates where the process is heading, by looking at the time rate of change of the controlled variable (its derivative). T_D is the 'rate time' and this characterizes the derivative action (with units of minutes).

The parallel PID controller is given as in Eq. (7).

$$mv(s) = k_c e(s) + \frac{1}{T_i s} e(s) + T_D s e(s). \tag{7}$$

A further alternative simplified form is given in Eq. (8).

$$G(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right). \tag{8}$$

The PID form most suitable for analytical calculations is given in Eq. (9).

$$G(s) = k + \frac{k_i}{s} + sk_d. \tag{9}$$