Engineering Applications of Artificial Intelligence 22 (2009) 343–350

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution

Arijit Biswas [a], Swagatam Das [a,*], Ajith Abraham [b], Sambarta Dasgupta [a]

[a] Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata, India
[b] Norwegian University of Science and Technology, Norway

ARTICLE INFO

Article history:
Received 21 April 2008
Accepted 4 June 2008
Available online 2 September 2008

Keywords:
Differential evolution
Fractional calculus
PID controllers
Fractional-order controllers
Evolutionary algorithms

ABSTRACT

Differential evolution (DE) has recently emerged as a simple yet very powerful technique for real parameter optimization. This article describes an application of DE to the design of fractional-order proportional–integral–derivative (FOPID) controllers involving fractional-order integrator and fractional-order differentiator. FOPID controllers' parameters are composed of the proportionality constant, integral constant, derivative constant, derivative order and integral order, and its design is more complex than that of conventional integer-order proportional–integral–derivative (PID) controller. Here the controller synthesis is based on user-specified peak overshoot and rise time and has been formulated as a single objective optimization problem. In order to digitally realize the fractional-order closed-loop transfer function of the designed plant, Tustin operator-based continuous fraction expansion (CFE) scheme was used in this work. Several simulation examples as well as comparisons of DE with two other state-of-the-art optimization techniques (Particle Swarm Optimization and binary Genetic Algorithm) over the same problems demonstrate the superiority of the proposed approach especially for actuating fractional-order plants. The proposed technique may serve as an efficient alternative for the design of next-generation fractional-order controllers.

1. Introduction

Fractional-order dynamic systems and controllers, which are based on fractional-order calculus (Oldham and Spanier, 1974; Lubich, 1986; Miller and Ross, 1993), have been gaining attention in several research communities since the last few years (Oustaloup, 1981; Chengbin and Hori, 2004). In Podlubny (1999b), it was advocated that fractional-order calculus would play a major role in a smart mechatronic system. Podlubny proposed the concept of the fractional-order $PI^{\lambda}D^{\mu}$ controllers and demonstrated the effectiveness of such controllers for actuating the responses of fractional-order systems in 1999. A few recent works in this direction as well as schemes for digital and hardware realizations of such systems can be traced in Chen et al. (2004), Nakagawa and Sorimachi (1992) and Chen et al. (2005). Vinagre et al. (2000) proposed a frequency domain approach based on expected crossover frequency and phase margin for the same controller design. Petras came up with a method based on the pole distribution of the characteristic equation in the complex plane (Petras, 1999). Dorcak et al. (2001) proposed a state-space design approach based on feedback pole placement. The fractional controller can also be synthesized by cascading a proper fractional unit to an integer-order controller (Chengbin and Hori, 2004).

Proportional–integral–derivative (PID) controllers have been used for several decades in industries for process control applications. The reason for their wide popularity lies in the simplicity of design and good performance including low percentage overshoot and small settling time for slow process plants (Astrom and Hagglund, 1995). In fractional-order proportional–integral–derivative (FOPID) controller, I and D operations are usually of fractional order; therefore, besides setting the proportional, derivative and integral constants K_p, T_d, T_i we have two more parameters: the order of fractional integration λ and that of fractional derivative μ. Finding an optimal set of values for K_p, T_i, T_d, λ and μ to meet the user specifications for a given process plant calls for real parameter optimization in five-dimensional hyperspace.

Differential evolution (DE) (Price et al., 2005; Storn and Price, 1997) has recently become quite popular as a simple and efficient scheme for global optimization over continuous spaces. It has reportedly outperformed many types of evolutionary algorithms

* Corresponding author. Tel.: +91 33 2528 2717.
E-mail addresses: arijitbiswas87@gmail.com (A. Biswas), swagatamdas19@yahoo.co.in (S. Das), ajith.abraham@ieee.org (A. Abraham), sambartadg@gmail.com (S. Dasgupta).

0952-1976/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2008.06.003

and search heuristics like PSO when tested over both benchmarks and real-world problems (Vesterstrøm and Thomson, 2004). In this work, a state-of-the-art version of DE has been used for finding the optimal values of K_p, T_i, T_d, λ and μ. The design method focuses on optimum placing of the dominant closed-loop poles and incorporate the constraints thus obtained using DE algorithm. The optimization-based design process has been tested for actuating the response of four process plants of which two are of integer order and two are of fractional order. The performance of the DE-based PI$^{\lambda}$D$^{\mu}$ controller has been compared with two other fractional-order controllers designed with the state-of-the-art versions of two recent swarm intelligence-based techniques well known as the Hierarchical Particle Swarm Optimizer with Time Varying Acceleration Coefficients (HPSO-TVAC) (Ratnaweera and Halgamuge, 2004) and the genetic algorithm (Holland, 1975; Cao et al., 2005). Such comparison reflects the superiority of the proposed method in terms of quality of the final solution, convergence speed and robustness.

The rest of the paper is organized as follows. Section 2 describes the rudiments of fractional calculus and fractional-order control systems. Section 3 provides a brief overview of the DE family of algorithms and describes a recent state-of-the-art version of DE called DE/rand/either–or, which was used, in this specific task. Section 4 demonstrates how the DE can be applied to the PI$^{\lambda}$D$^{\mu}$ controller design problem when formulated as an optimization task. Simulation strategies and experimental results have been presented and discussed in Section 5 and finally the paper is concluded with a discussion on future research issues in Section 6.

2. Fractional-order systems: a brief overview

Fractional calculus is a branch of mathematical analysis that studies the possibility of taking real number power of the differential operator and integration operator. From a purely mathematical point of view, there are several ways to define fractional-order derivatives and integrals. The generalized differ-integrator operator may be put forward as

$$ {}_aD_t^q f(t) = \frac{d^q f(t)}{[d(t-a)]^q} \tag{1}$$

where q represents the real order of the differintegral (an n is used in some literature to denote an integer order), t is the parameter for which the differintegral is taken and a is the lower limit. Unless otherwise stated, the lower limit will be 0 and left out of the notation. Caputo used a popular definition used to compute differintegral in 1960s. The definition for Caputo's fractional derivative of order λ with respect to the variable t and with the starting point $t=0$ goes as follows (Caputo, 1967, 1969):

$$ {}_0D_t^{\lambda} y(t) = \frac{1}{\Gamma(1-\delta)} \int_0^t \frac{y^{(m+1)}(\tau)\,d\tau}{(t-\tau)^{\delta}} \quad (\gamma = m+\delta,\ m \in Z,\ 0<\delta \leqslant 1) \tag{2}$$

where $\Gamma(Z)$ is Euler's gamma function. If $\gamma<0$, then we have a fractional integral of order $-\gamma$ given as

$$ {}_0I_t^{-\gamma} y(t) = {}_0D_t^{\gamma} y(t) = \frac{1}{\Gamma(-\gamma)} \int_0^t \frac{y(\tau)d\tau}{(t-\tau)^{1+\gamma}} \quad (\gamma<0) \tag{3}$$

One distinct advantage of using Caputo's definition is that it only allows for consideration of easily interpretable initial conditions but it is also bounded, which means the derivative of a constant is equal to zero. In time domain, a fractional-order system is governed by an n-term inhomogeneous fractional-order differential equation (FDE):

$$ a_n D^{\beta_n} y(t) + a_{n-1} D^{\beta_{n-1}} y(t) + \cdots + a_1 D^{\beta_1} y(t) + a_0 D^{\beta_0} y(t) = u(t) \tag{4}$$

where $D^{\lambda} \equiv {}_0D_t^{\lambda}$ is the Caputo's fractional derivative of order λ. Converting to frequency domain, the fractional-order transfer function of such a system may be obtained through the Laplace transform function as follows:

$$ G_n(s) = \frac{1}{a_n s^{\beta_n} + a_{n-1} s^{\beta_{n-1}} + \cdots + a_1 s^{\beta_1} + a_0 s^{\beta_0}} \tag{5}$$

where β_k $(k = 0,\ 1,\ \ldots,\ n)$ is an arbitrary real number, $\beta_n > \beta_{n-1} > \cdots > \beta_1 > \beta_0 > 0$ and a_k $(k = 0,\ 1,\ \ldots,\ n)$ is an arbitrary constant. Finally, we would like to mention here that the Laplace transform of the fractional derivative might be given as

$$ \int_0^{\infty} e^{-st} D^{\gamma} y(t)\,dt = s^{\gamma} Y(s) - \sum_{k=0}^{m} s^{\gamma-k-1} y^{(k)}(\theta) \tag{6}$$

For $\gamma<0$ (i.e., for the case of a fractional integral) the sum in the right-hand side must be omitted.

3. The DE algorithm and its modification

Like any other evolutionary algorithm, DE starts with a population of NP D-dimensional parameter vectors representing the candidate solutions. We shall denote subsequent generations in DE by $G = 0, 1, \ldots, G_{\max}$. Since the parameter vectors are likely to be changed over different generations, we may adopt the following notation for representing the ith vector of the population at the current generation as

$$ \vec{X}_{i,G} = [x_{1,i,G},\ x_{2,i,G},\ x_{3,i,G},\ \ldots,\ x_{D,i,G}] \tag{7}$$

The initial population (at $G = 0$) should better cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum bounds: $\vec{X}_{\min} = \{x_{1,\min},\ x_{2,\min},\ \ldots,\ x_{D,\min}\}$ and $\vec{X}_{\max} = \{x_{1,\max},\ x_{2,\max},\ \ldots,\ x_{D,\max}\}$. Hence we may initialize the jth component of the ith vector as

$$ x_{j,i,0} = x_{j,\min} + \mathrm{rand}_j(0,1)(x_{j,\max} - x_{j,\min}) \tag{8}$$

where $\mathrm{rand}_j(0,1)$ is the jth instantiation of a uniformly distributed random number lying between 0 and 1. Following steps are taken next: mutation, crossover and selection, which are explained below.

3.1. Mutation

After initialization, DE creates a *donor* vector $\vec{V}_{i,G}$ corresponding to each population member or *target* vector $\vec{X}_{i,G}$ in the current generation through mutation. It is the method of creating this donor vector, which differentiates between the various DE schemes. For example, five most frequently referred mutation strategies implemented in the public-domain DE codes available online at http://www.icsi.berkeley.edu/~storn/code.html are listed below:

$$ \text{“DE/rand/1”}:\ \vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F(\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) \tag{9}$$

$$ \text{“DE/best/1”}:\ \vec{V}_{i,G} = \vec{X}_{\text{best},G} + F(\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \tag{10}$$

$$ \begin{aligned} \text{“DE/target-to-best/1”}:\ \vec{V}_{i,G} = {} & \vec{X}_{i,G} \\ & + F(\vec{X}_{\text{best},G} - \vec{X}_{i,G}) \\ & + F(\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) \end{aligned} \tag{11}$$