Control Engineering Practice 17 (2009) 1380–1387

Contents lists available at ScienceDirect

Control Engineering Practice

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

Design of a fractional order PID controller for an AVR using particle swarm optimization

Majid Zamani [a], Masoud Karimi-Ghartemani [b,*], Nasser Sadati [b], Mostafa Parniani [b]

[a] Department of Electrical Engineering, University of California at Los Angeles, USA
[b] Department of Electrical Engineering, Sharif University of Technology, P.O. Box 11365-9363, Tehran, Iran

ARTICLE INFO

Article history:
Received 21 October 2007
Accepted 21 July 2009
Available online 19 August 2009

Keywords:
AVR
Optimal control
Particle swarm optimization
Fractional order PID controller
Robustness improvement

ABSTRACT

Application of fractional order PID (FOPID) controller to an automatic voltage regulator (AVR) is presented and studied in this paper. An FOPID is a PID whose derivative and integral orders are fractional numbers rather than integers. Design stage of such a controller consists of determining five parameters. This paper employs particle swarm optimization (PSO) algorithm to carry out the aforementioned design procedure. PSO is an advanced search procedure that has proved to have very high efficiency. A novel cost function is defined to facilitate the control strategy over both the time-domain and the frequency-domain specifications. Comparisons are made with a PID controller and it is shown that the proposed FOPID controller can highly improve the system robustness with respect to model uncertainties.

1. Introduction

Fractional calculus extends the ordinary calculus by extending the ordinary differential equations to fractional order differential equations, i.e. those having non-integer orders of derivatives and integrals. Such equations can provide linear models and transfer functions for some infinite-dimensional physical systems (Canat & Faucher, 2003). On the other hand, fractional order controllers may be employed to achieve feedback control objectives for such systems. Modeling and control topics using the concept of fractional order systems have been recently attracting more attentions because the advancements in computation power allow simulation and implementation of such systems with adequate precision.

Fractional order PID (FOPID) controller is a convenient fractional order structure that has been employed for control purposes (Ferreira & Machado, 2003; Monje, Vinagre, Feliu, & Chen, 2008; Podlubny, 1999). An FOPID is characterized by five parameters: the proportional gain, the integrating gain, the derivative gain, the integrating order and the derivative order. Different design methods have been reported including pole distribution (Petras, 1999), frequency domain approach (Vinagre, Podlubny, Dorcak, & Feliu, 2000), state-space design (Dorcak, Petras, Kostial, & Terpak, 2001) and two-stage or hybrid approach (Chengbin & Hori, 2004) which uses conventional (integer order) controller's design method and then improves performance of the designed control system by adding proper fractional order controller. An alternative design method is presented in this paper based on particle swarm optimization (PSO) algorithm and employment of a novel cost function which offers flexible control over time domain and frequency domain specifications.

PSO is an evolutionary-type global optimization algorithm (Kennedy & Eberhart, 1995; Liang, Qin, Suganthan, & Baskar, 2006) which is different from well-known similar algorithms in that no operators, inspired by evolutionary procedures, are applied to the population to generate new promising solutions. PSO has already been used to determine optimal solution to several power engineering problems such as reactive power and voltage control (Yoshida, Kawata, & Fukuyama, 2000) and state estimation (Naka, Genji, Yura, & Fukuyama, 2001). This algorithm is employed here to design an FOPID controller for an automatic voltage regulator (AVR) problem. Being the main controller of an excitation system, AVR maintains the voltage of a synchronous generator at a specific level. The proposed controller is simulated within various scenarios and its performance is compared with those of an optimally designed PID controller. The results conclude that the FOPID control is able to significantly improve robustness of the system with respect to system uncertainties.

The paper is organized as follows. Sections 2 and 3 overview the concepts of fractional calculus and PSO algorithm, respectively. Design of the proposed FOPID controller for an AVR using PSO algorithm is described in Section 4. Section 5 is devoted to computer simulation of the proposed controller and its comparison with a PID controller. Section 6 concludes the paper.

* Corresponding author. Tel.: +98 21 66165936; fax: +98 21 66023261.
E-mail addresses: zamani@ee.ucla.edu (M. Zamani), karimig@sharif.edu (M. Karimi-Ghartemani), sadati@sharif.edu (N. Sadati), parniani@sharif.edu (M. Parniani).

0967-0661/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2009.07.005

2. Fractional calculus

Fractional calculus is a generalization of the ordinary calculus. The chief idea is to develop a functioning operator D, associated to an order ν not limited to integer numbers, that generalizes the ordinary concepts of derivative (for a positive ν) and integral (for a negative ν) (Valerio & Sa Da Costa, 2006).

There are different definitions for fractional derivatives. The most usual definition is introduced by Riemann and Liouville (Oldham & Spanier, 1974) that generalizes the following definitions corresponding to integer orders:

$$_0D_x^{-n}f(x) = \int_c^x \frac{(x-t)^{n-1}}{(n-1)!} f(t)\,dt, \quad n \in \mathbb{N} \tag{1}$$

The generalized definition of D becomes $_cD_x^{\nu}f(x)$. The Laplace transform of D pursues the renowned rule $L[_0D_x^{\nu}] = s^{\nu}F(s)$ for zero initial conditions. This means that, if zero initial conditions are assumed, the systems with dynamic behavior described by differential equations including fractional derivatives give rise to transfer functions with fractional orders of s. More details are provided in Miller and Ross (1993) and Samko, Kilbas, and Marichev (1993).

The most common way of using, in both simulations and hardware implementations, of transfer functions including fractional orders of s is to approximate them with usual (integer order) transfer functions. To perfectly approximate a fractional transfer function, an integer transfer function would have to involve an infinite number of poles and zeroes. Nonetheless, it is possible to obtain logical approximations with a finite number of zeroes and poles.

One of the well-known approximations is caused by Oustaloup who uses the recursive distribution of poles and zeroes. The approaching transfer function is given by Oustaloup (1991):

$$s^{\nu} \approx k \prod_{n=1}^{N} \frac{1+(s/\omega_{z,n})}{1+(s/\omega_{p,n})} \tag{2}$$

The approximation is legitimate in the frequency range $[\omega_l, \omega_h]$. Gain k is also regulated so that both sides of (2) shall have unit gain at 1 rad/s. The number of poles and zeros (N) is chosen in advance, and the desired performance of the approximation strongly depends on: low values cause simpler approximations, but may cause ripples in both gain and phase behaviors. Such ripples can be functionally removed by increasing N, but the approximation will become computationally heavier. Frequencies of poles and zeroes in (2) are given by

$$\omega_{z,1} = \omega_l\sqrt{\eta} \tag{3}$$

$$\omega_{p,n} = \omega_{z,n}\varepsilon, \quad n = 1, \ldots, N \tag{4}$$

$$\omega_{z,n+1} = \omega_{p,n}\eta, \quad n = 1, \ldots, N-1 \tag{5}$$

$$\varepsilon = (\omega_h/\omega_l)^{\nu/N} \tag{6}$$

$$\eta = (\omega_h/\omega_l)^{(1-\nu)/N} \tag{7}$$

The case $\nu<0$ can be handled by inverting (2). For $|\nu|>1$, the approximation becomes dissatisfactory. So it is common to separate the fractional orders of s as follows:

$$s^{\nu} = s^n s^{\delta}, \quad \nu = n+\delta, \quad n \in \mathbb{Z}, \quad \delta \in [0,1] \tag{8}$$

and only the second term, i.e. s^{δ}, needs to be approximated.

If a discrete transfer function approximation is sought, the above approximation in (2) may be discretized (Vinagre, Podlubny, Hernandez, & Feliu, 2000). There are also methods that directly provide discrete approximations (Lubich, 1986). Besides, electric circuits which can serve as exact fractional integrators and differentiators have also been reported in Oldham and Spanier (1974) and Oldham and Zoski (1983).

3. Particle swarm optimization (PSO)

PSO is a population-based evolutionary algorithm that was developed from research on swarm such as fish schooling and bird flocking (Kennedy & Eberhart, 1995). It has become one of the most powerful methods for solving optimization problems. The method is proved to be robust in solving problems featuring nonlinearity and non-differentiability, multiple optima, and high dimensionality. The advantages of the PSO are its relative simplicity and stable convergence characteristic with good computational efficiency.

The PSO consists of a swarm of particles moving in a D dimensional search space where a certain quality measure and fitness are being optimized. Each particle has a position represented by a position vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and a velocity represented by a velocity vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, which is clamped to a maximum velocity $V_{max} = (v_{max1}, v_{max2}, \ldots, v_{maxD})$. Each particle remembers its own best position so far in a vector $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$, where i is the index of that particle. The best position vector among all the neighbors of a particle is then stored in the particle as a vector $P_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$. The modified velocity and position of each particle can be manipulated according to the following equations:

$$v_{id}^{(t+1)} = wv_{id}^{(t)} + c_1r_1(p_{id} - x_{id}^{(t)}) + c_2r_2(p_{gd} - x_{id}^{(t)}) \tag{9}$$

$$x_{id}^{(t+1)} = x_{id}^{(t)} + v_{id}^{(t+1)}, \quad d = 1, \ldots, D \tag{10}$$

where w can be expressed by the inertia weights approach (Shi & Eberhart, 1998) and often decreases linearly from w_{max} (of about 0.9) to w_{min} (of about 0.4) during a run. In general, the inertia weight w is set according to the following equation:

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}}.iter \tag{11}$$

where $iter_{max}$ represents the maximum number of iterations, and $iter$ is the number of current iteration or generation. Also c_1 and c_2 are the acceleration constants which influence the convergence speed of each particle and are often set to 2.0 according to the past experiences (Eberhart & Shi, 2001). Moreover r_1 and r_2 are random numbers in the range of [0, 1], respectively. If V_{max} is too small, then the particles may not explore sufficiently beyond local solutions. In many experiences with PSO, V_{max} is often set to the maximum dynamic range of the variables on each dimension, $v_{dmax} = x_{dmax}$.

4. AVR design using FOPID controller

4.1. FOPID controller

The differential equation of a fractional order $PI^{\lambda}D^{\mu}$ controller is described by

$$u(t) = k_Pe(t) + k_ID_t^{-\lambda}e(t) + k_DD_t^{\delta}e(t) \tag{12}$$

The continuous transfer function of FOPID is obtained through Laplace transform and is given by

$$G_c(s) = k_P + k_Is^{-\lambda} + k_Ds^{\delta} \tag{13}$$

Design of an FOPID controller involves design of three parameters k_P, k_I, k_D, and two orders λ, δ which are not necessarily integer. The fractional order controller generalizes the conventional integer