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.

© 2009 Elsevier Ltd. All rights reserved.

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 controller 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.
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 \( \nu \) not limited to integer numbers, that generalizes the ordinary concepts of derivative (for a positive \( \nu \)) and integral (for a negative \( \nu \)) \cite{Valerio2006}.

There are different definitions for fractional derivatives. The most usual definition is introduced by Riemann and Liouville \cite{Oldham1974} that generalizes the following definitions corresponding to integer orders:

\[
\begin{align*}
0^D_{-}\nu f(x) & = \int_{c}^{x} \frac{(x-t)^{\nu-1}}{(n-1)!} f(t) \, dt, \quad n \in \mathbb{N} \\
\end{align*}
\]

The generalized definition of \( D \) becomes \( 0^D_{-}\nu f(x) \). The Laplace transform of \( D \) pursues the renowned rule \( \mathcal{L}(D^\nu f) = s^\nu \mathcal{L}(f) \) 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 \( \nu \). Nonetheless, it is important to note that transfer function, an integer transfer function would have to involve an infinite number of poles and zeros \( (\nu \text{ order}) \) transfer functions. To perfectly approximate a fractional approximation will become computationally heavier. Frequencies chosen in advance, and the desired performance of the approximations, but may cause ripples in both gain and phase behaviors.

The chief idea is to develop a functioning operator \( D \), associated to an order \( \nu \) not limited to integer numbers, that generalizes the ordinary concepts of derivative (for a positive \( \nu \)) and integral (for a negative \( \nu \)) \cite{Valerio2006}.

The most common way of using, in both simulations and hardware implementations, of transfer functions including fractional derivatives 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 zeros. Nonetheless, it is possible to obtain logical approximations with a finite number of zeros and poles.

One of the well-known approximations is caused by Oustaloup who uses the recursive distribution of poles and zeroes. The approximating transfer function is given by Oustaloup \cite{Oustaloup1991}:

\[
\begin{align*}
\tilde{s} & = k \prod_{n=1}^{N} \frac{1 + (s/\omega_n)}{1 + (s/\omega_n)} \\
\end{align*}
\]

The approximation is legitimate in the frequency range \([\omega_1, \omega_N]\). 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

\[
\begin{align*}
\omega_{2n} &= \omega_1 \sqrt{n} \\
\omega_{2n+1} &= \omega_1 \sqrt{n}, \quad n = 1, \ldots, N \\
\end{align*}
\]

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

\[
\begin{align*}
\tilde{s} & = s^\delta \tilde{s}, \quad v = n + \delta, \quad n \in \mathbb{Z}, \quad \delta \in [0, 1] \\
\end{align*}
\]

and only the second term, i.e. \( \tilde{s}^\delta \), needs to be approximated.

If a discrete transfer function approximation is sought, the above approximation in (2) may be discretized \cite{Vinagre2000}. There are also methods that directly provide discrete approximations \cite{Lubich1986}. Besides, electric circuits which can serve as exact fractional integrators and differentiators have also been reported in Oldham and Spanier \cite{Oldham1974} and Oldham and Zoski \cite{Oldham1983}.

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 \cite{Kennedy1995}. 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 \( X = (x_1, x_2, \ldots, x_D) \) and a velocity represented by a velocity vector \( V = (v_1, v_2, \ldots, v_D) \), which is clamped to a maximum velocity \( V_{\text{max}} = (V_{\text{max}1}, V_{\text{max}2}, \ldots, V_{\text{max}D}) \). Each particle remembers its own best position so far in a vector \( P_i = (p_1, p_2, \ldots, p_D) \), 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:

\[
\begin{align*}
V_i^{(t+1)} & = w V_i^{(t)} + c_1 r_1 (P_i^{(t)} - x_i^{(t)}) + c_2 r_2 (P_g^{(t)} - x_i^{(t)}) \\
x_i^{(t+1)} & = x_i^{(t)} + V_i^{(t+1)} , \quad d = 1, \ldots, D \\
\end{align*}
\]

where \( w \) can be expressed by the inertia weights approach \cite{Shi1998} and often decreases linearly from \( V_{\text{max}} \) (of about 0.9) to \( V_{\text{min}} \) (of about 0.4) during a run. In general, the inertia weight \( w \) is set according to the following equation:

\[
W = V_{\text{max}} - \frac{V_{\text{max}} - V_{\text{min}}}{\text{iter}_{\text{max}}} \times \text{iter} \\
\]

where \( \text{iter}_{\text{max}} \) represents the maximum number of iterations, and \( \text{iter} \) 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 \cite{Eberhart2001}. Moreover \( r_1 \) and \( r_2 \) are random numbers in the range of \([0, 1]\), respectively. If \( V_{\text{max}} \) is too small, then the particles may not explore sufficiently beyond local solutions. In many experiences with PSO, \( V_{\text{max}} \) is often set to the maximum dynamic range of the variables on each dimension, \( V_{\text{dmax}} = x_{\text{dmax}} \).

4. AVR design using FOPID controller

4.1. FOPID controller

The differential equation of a fractional order \( PI^\delta D^\nu \) controller is described by

\[
u(t) = k_0 e(t) + k_D \frac{d}{dt} \nu(t) + k_D^\delta e(t) \\
\]

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

\[
G_c(s) = k_P + k_I s^{-\delta} + k_D s^\delta \\
\]

Design of an FOPID controller involves design of three parameters \( k_P, k_I, k_D \), and two orders \( \delta, \delta \) which are not necessarily integer. The fractional order controller generalizes the conventional integer
order PID controller. This expansion can provide more flexibility in achieving control objectives.

4.2. Linearized model of excitation system

Automatic voltage regulator (AVR) is the central controller within the excitation system that maintains the terminal voltage of a synchronous generator at a specified level. Depending on the method of supplying DC power, different types of excitation systems exist (IEEE standard definition for excitation systems for synchronous machines (ansi), 1986). As an example, Fig. 1 shows the schematic diagram of an alternator supplied controlled-rectifier excitation system (IEEE standard definition for excitation systems for synchronous machines (ansi), 1986). The DC regulator holds constant generator field voltage and is commonly referred to as manual control. It is primarily for testing, start-up and to cater to situations where the AC regulator is faulty. Closed loop voltage control is carried out through the AC regulator. In addition to the AVR, this loop comprises five main components, namely amplifier, exciter, excitation voltage limiters, generator and measurement and filtering (Fig. 2). To analyze dynamic performance of AVR, transfer functions of these components are represented as follows (Gaing, 2004; Kundur, 1994).

Amplifier model:

\[
\frac{V_R(s)}{V_C(s)} = \frac{k_A}{1 + \tau_A s}
\]  

Typical values of \(k_A\) are in the range of 10–400. The amplifier time-constant often ranges from 0.02 to 0.1 s.

Exciter model:

\[
\frac{V_F(s)}{V_R(s)} = \frac{k_E}{1 + \tau_E s}
\]

Typical values of \(k_E\) are in the range of 0.8–1 and the time-constant \(\tau_E\) for an AC exciter is in the range of 0.5–1.0 s.

Generator model: The generator model used in this paper is the well-known third order simplified model shown in Fig. 3. Values of the parameters are selected as \(K_1 = 1.591, K_2 = 1.5, K_3 = 0.333, K_4 = 1.8, K_5 = -0.12, K_6 = 0.3, T_3 = 1.91, H = 3, K_0 = 0, \omega_0 = 377\) (Kundur, 1994).

Measurement model: The voltage measurement block, including PT, rectifier and filter, is often modeled with a single time constant

\[
\frac{V_L(s)}{V_T(s)} = \frac{k_R}{1 + \tau_R s}
\]

\(\tau_R\) ranges over 0.001–0.06 s.

Excitation voltage limiter: AVR and exciter output voltages are limited by windup and non-windup limiters (Kundur, 1994). Also, dedicated overexcitation and underexcitation limiters are employed to assure safe operation of the generator. The purpose of the overexcitation limiter (OXL) is to protect the generator from overheating due to prolonged field overcurrent. OXL has delayed operation and, thus, needs not to be modeled for AVR dynamic performance analysis. Underexcitation limiter (UXL) operates instantaneously to prevent angle instability.

Block diagram of the AVR compensated with an FOPID controller is shown in Fig. 2. In this figure, the combined effects of these limiters are represented by the upper and lower limits set to three times of the nominal value of the field voltage.

4.3. Performance criterion

There are several performance criteria for design of controllers such as integral of absolute error (IAE), the integral of squared-error (ISE) or integral of time-weighted-squared-error (ITSE) (Krohling & Rey, 2001). A disadvantage of the IAE and ISE criteria is that they may result in a response with a relatively small overshoot but a long settling time because they weigh all errors uniformly over time. Although the ITSE performance criterion can overcome this drawback, but it cannot ensure to have a desirable stability margin (Krohling & Rey, 2001). The IAE, ISE, and ITSE...
This performance criterion includes the overshoot frequency domain is proposed for evaluating the FOPID controller. In this paper, a new performance criterion in the time domain and PM factor terms that the significance of each is determined by a weight expense of degrading other features. Selections are order to attain the desired specification. For the current study, the integral is evaluated up to and integral of squared-input must be computed numerically and (20) to prevent large values of the field current. The error is negligible for t>T. Therefore, the proposed performance criterion J(k) is defined as

\[
J(k) = \int_0^T (w_1 e(t) + w_2 u^2(t)) dt + \frac{w_7}{\text{PM}} + \frac{w_8}{\text{GM}}
\]

where \( k \) is \([k_p, k_i, k_d, \lambda, \delta]\). The field voltage \( V_f \) is used for \( u(t) \) in (20) to prevent large values of the field current. The error \( e(t) \) is the difference between the reference voltage and the output voltage.

The proposed performance criterion (20) comprises eight terms that the significance of each is determined by a weight factor \( w_i \). It is up to the user to set the weight factors properly in order to attain the desired specification. For the current study, selections are \( w_1 = 0.1, w_2 = 1, w_3 = 1, w_4 = 1000, w_5 = 1, w_6 = 50000 \) and \( w_8 = 30000 \). An increase in \( w_i \) will result in some improvement in the corresponding feature at the expense of degrading other features.

### 4.4. Design of FOPID using PSO

The PSO algorithm is utilized to design the controller parameters \( k_p, k_i, k_d, \lambda \) and \( \delta \) such that the controlled system exhibits desired response and robust stability as evaluated by the proposed performance criterion. Details of the design process are relatively similar to the one reported in Gaing (2004) for a regular PID controller. The five controller parameters \( k_p, k_i, k_d, \lambda \) and \( \delta \) compose a particle \( k = [k_p, k_i, k_d, \lambda, \delta] \). The five members are assigned as real values. If there are \( n \) particles in a population (group), then the dimension of that population is \( n \times 5 \). To ensure the stability of closed loop, the cost function is penalized with a penalty function \( P(k) \) given as

\[
P(k) = \begin{cases} 
P_1 & \text{if } k \text{ is unstable} \\
0 & \text{else} 
\end{cases}
\]

If the particle \( k \) does not satisfy the stability of closed loop, then \( k \) is an unstable particle and its cost function is penalized with a very large positive constant \( P_1 \). Consequently, \( k \) does not survive in the evolutionary process. Otherwise, the particle \( k \) is not penalized. Therefore, the total evaluation of a particle \( k \), can be obtained as

\[
F(k) = J(k) + P(k)
\]
A practical high-order AVR is used to verify the efficiency of the proposed FOPID controller. The system parameters are $k_A = 10$, $\tau_A = 0.1$, $k_E = 1$, $\tau_E = 0.5$, $V_{f0} = 1$, $K_1 = 1.591$, $K_2 = 1.5$, $K_3 = 0.333$, $K_4 = 1.8$, $K_5 = -0.12$, $K_6 = 0.3$, $T_3 = 1.91$, $H = 3$, $K_0 = 0$, $\omega_0 = 377$, $K_R = 1$ and $\tau_R = 0.06$.

5.2. Details of FOPID design using PSO for the AVR

Inspired from practical requirements, the lower bounds of the five controller parameters are zero and their upper bounds are set to $k_{p_{\text{max}}} = 1.5$, $k_{i_{\text{max}}} = k_{d_{\text{max}}} = 1$ and $\lambda_{\text{max}} = \delta_{\text{max}} = 2$. The following parameters are used for carrying out the FOPID design using PSO:

- The members of each particle are $k_p$, $k_i$, $k_d$, $\lambda$ and $\delta$.
- Population size = 30.
- Inertia weight factor $w$ is set as (11) where $w_{\text{max}} = 0.9$ and $w_{\text{min}} = 0.4$.
- The limit of change in velocity is set to maximum dynamic range of the variables on each dimension.
- Acceleration constants $c_1 = 2$ and $c_2 = 2$.
- Maximum iteration is set to 1500.
- $\omega_0$ and $\omega_{\text{sh}}$ in (5)–(9) are set to $10^{-5}$ and $10^5$ rad/s, respectively.
- The order of approximation in (2) is set to $n = 3$.
- $T$ in (20) is set to 20 s.
- $P_i$ in (21) is set $10^{10}$.

5.3. First test: basic performance

Ten trials are performed for the proposed controller. The best solution is summarized in the first row of Table 1. The designed FOPID controller is of order six as its fractional derivative and integral have been approximated by $N = 3$ in (2). The Hankel minimum degree approximation (MDA) without balancing (Safonov, Chiang, & Limebeer, 1990) can be employed to reduce its order to four

$$\hat{G}_c(s) = \frac{1.111e005s^4 + 1.268e005s^3 + 1.405e007s^2 + 2.281e006s + 955}{s^4 + 4.258e005s^3 + 8.407e007s^2 + 4.322e004s}$$

(26)

The Bode diagrams of the original and reduced order controllers are shown in Fig. 5, depicting very close behaviors in the frequency range of interest. Thick lines in Figs. 6 and 7 show the terminal voltage step response and Bode diagram of the AVR with the FOPID controller, respectively. Note that the simulations are performed based on the reduced-order implementation of (26) for the FOPID controller.

In order to emphasize advantages of the proposed FOPID controller, a PID controller is also designed using the same method. The same evaluation function is used and the best solution is summarized in the third row of Table 1. The terminal voltage step response and Bode diagram of the AVR with the PID controller are also shown in Figs. 6 and 7 with dash lines for comparison. The FOPID does not offer improved performance as far as transient time characteristics of the response are concerned. However, the FOPID type AVR has higher robust stability than the PID type AVR as is observed from Table 1 and also attested by the subsequent simulations.

5.4. Second test: robustness

It can be observed from Fig. 7 that the phase-angle of the transfer function with FOPID controller is flat around the gain crossover frequency:

$$\frac{d}{d\omega} \arg(G_c(j\omega)G(j\omega))|_{\omega_{\text{crossover}}} \cong 0$$

(27)
This means that the system with FOPID controller is more robust, with respect to gain changes, in comparison with the one with using PID controller (Valerio & Sa Da Costa, 2006). This occurs due to the presence of phase margin in performance criterion (20). As shown in Fig. 7, it is not possible to create such a flatness using the PID type AVR.

To examine numerically the robustness of the FOPID controller with respect to parameter uncertainties, the following simulation is performed. Assume that instead of \( K_1 = 1.59 \), and due to the change in loading condition, the actual value is \( K_1 = 1 \). The terminal voltage step responses of the AVR with previously designed PID and FOPID controllers are shown in Fig. 8. The PID-controlled system exhibits large oscillations with an overshoot of about 62% compared to the 20% overshoot of the FOPID.

Now assume another uncertainty in the exciter model as
\[
\frac{V_F(s)}{V_e(s)} = \frac{1}{0.5 + 0.5} \tag{28}
\]

The step responses with both generator and exciter uncertainties are shown in Fig. 9. The PID-controlled system exhibits large oscillations with an overshoot of about 88% as opposed to the 20% overshoot of the FOPID-controlled system.

6. Conclusion

This paper presents a design method for determining the FOPID controller parameters using the PSO algorithm. The proposed method involves a new time-domain and frequency-domain performance criterion. Application of the method to a practical AVR shows that the proposed algorithm can perform an
efficient search for the optimal FOPID controller parameters. Furthermore, it can be concluded from the above simulations that the proposed FOPID controller has more robust stability and performance characteristics than the PID controller applied to the AVR.

References


