Design of fractional order PID controller for DC motor using evolutionary optimization techniques

ASHU AHUJA*, SANJEEV KUMAR AGGARWAL [1]
Electrical Engineering Department,
M. M. Engineering College, Mullana, Ambala, Haryana,
INDIA
ashu.valecha.ahuja@gmail.com, vasusanjeev@yahoo.co.in
[1] The corresponding author

Abstract: - In this paper, a comparison of various Evolutionary Optimization (EO) techniques has been performed for controller tuning (PID and FOPID) for DC motor speed control applications. Three techniques have been considered, which are: genetic algorithm (GA), particle swarm optimization (PSO), and differential evolution (DE). A fixed structure PID and FOPID controller has been considered. Usually in literature, comparison has been carried out based on time domain performance indices; whereas, in this work, mixed sensitivity H_∞ method is considered as the fitness function for the comparison of various EO techniques in order to assess the robustness of the designed system. It has been shown that GA optimized FOPID controller has better robustness with respect to model uncertainties.

Key-words: Controllers, differential evolution, DC motor, genetic algorithm, mixed sensitivity function, particle swarm optimization

1 Introduction

The proportional–integral–derivative (PID) controller is the most widely used controller for industrial applications due to its simple design and ability to provide effective control [1]. But, many real-world physical systems need fractional calculus as they are modelled by fractional-order differential equations, i.e., equations involving non-integer order integrals and derivatives such as dynamic systems with materials having memory and hereditary effects [2], chaotic systems [3], automatic voltage regulator (AVR) [4], permanent magnet synchronous machine (PMSM) AC servo system [5]. The emergence of 'fractance' and memristors [6] and progress in the synthesis of real non-integer differentiator has led to the development of fractional-order controllers, including fractional-order PID controllers (FOPID) [1]. The main advantage of using fractional-order controllers for a linear control system is that the time and frequency responses can be obtained in the form of functions other than that of exponential type and better performance as compared to integer-order controllers can be obtained. In a PID controller, the derivative and the integral order are of integer nature; whereas, in FOPID, the derivative and integral order are of fractional nature rather than integer [2]. The key challenge of designing a FOPID controller is to determine the two key parameters λ (integral order) and μ (derivative order) apart from usual tuning parameters of PID. Both λ and μ are in fraction, which increases the robustness of the system and gives an optimal control [7-9].

DC motor is widely used for industrial applications due to its easy controllability and high performance [10-11]. In DC motor speed control, robustness is the primary requirement of design followed by the performance of the system. Therefore, many controllers are designed to ensure the disturbance rejection and robustness. While many control structures have been proposed in literature for DC motor speed control, such as adaptive variable structure control [12], differential feedback control [13], fuzzy logic DC motor speed control [14]; yet to incorporate robustness in the controller design, H_∞ is the most popular optimal control technique [15]. A design procedure which incorporates loop shaping methods to obtain performance and robust stability trade off and a particular H_∞ optimization problem to guarantee closed loop stability has been proposed in [16]. Moreover, adaptive neural method [17], robust controller [18] and robust adaptive discrete variable structure control scheme for speed control of DC motor [12] have also been introduced. A multi objective formulation has also been proposed [19].

Numerous controllers have been developed in literature for DC motor speed control namely: PID controller, fuzzy logic controller, neural network [20], fuzzy-neural network, fuzzy-genetic algorithm [21], fuzzy-ant colony, fuzzy-swarm [14], fuzzy-sliding mode control [22]. Many techniques have been used to tune the parameter of these controllers such as particle swarm optimization (PSO) [23-24], augmented Lagrangian particle swarm optimization [25], improved differential evolution [7], and FOPID controllers tuned by Ziegler-Nichols type rules [26]. Among these techniques evolutionary optimization (EO) techniques have emerged as popular controller design techniques. However, a comparison of various evolutionary optimization (EO) techniques like genetic algorithm (GA), Particle Swarm Optimization (PSO) and Differential Evolution (DE) for FOPID tuning still needs to be explored and has been investigated in this work.

In this paper, GA, PSO and DE have been employed to design an FOPID controller for DC motor speed control. The proposed FOPID controller is simulated with six tuning parameters $(k_p, k_i, k_D, \tau_D, \lambda, \mu)$ and its performance is compared with optimally tuned PID controller (k_p, k_i, k_D, τ_D). It has been concluded that with respect to system uncertainties, robustness is significantly improved by FOPID controller. In literature, many authors have explored the FOPID controller tuning using time domain performance indices such as IAE, ISE, ITAE and ITSE [27-28], having constraints on gain crossover frequency, phase margin, robustness to variation in the gain of the plant, robustness to high frequency noise and good output disturbance rejection [29] or minimizing the characteristics equation [30]; but, issues of stability and robustness with mixed sensitivity function involving FOPID controller have not been adequately addressed.

The paper is presented as follows. Sections 2 and 3 present the overview of the concepts of fractional calculus and EO techniques, respectively. Design of controllers for DC motor is described in Section 4. Section 5 is devoted to tuning of the proposed FOPID and PID controllers using various EO techniques like GA, PSO and DE. Section 6 consists of comparison of FOPID and PID controllers. Section 7 concludes the paper.

2 Fractional Calculus

Fractional calculus may be explained as the extension of the concept of a derivative operator from integer order 'n' to arbitrary order 'v' where v may be a real value or a complex value or may be a complex valued function $v = v(x,t)$:

$$\frac{d^n}{dx^n} \rightarrow \frac{d^v}{dx^v}$$

The most commonly referred definition of fractional derivatives and fractional integrals was given by G. F. B. Riemann and J. Liouville, according to which, the fractional integral of order $'v'(v>0)$ for a function $f(t)$ is given by:

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

with the condition that $f(x)$ and ${}_0D_x^{-n} f(x)$ are causal functions. For initial conditions to be zero, the Laplace transform of D is given by $L[{}_0D_x^v] = s^v F(s)$ i.e., for zero initial conditions, the system whose dynamic behavior described by differential equations having fractional derivatives results in transfer functions with fractional orders of s [31-33]. To simulate fractional order of s in MATLAB, this is to be approximated by usual integer order transfer function having an infinite number of poles and zeroes. It is also possible to logically approximate it with a finite number of poles and zeros. Oustaloup proposed a method of approximation of a function of the form [34]:

$$H(s) = s^{\mu}, \mu \in R^{+} \tag{2}$$

by a rational function:

$$H(s) = C\prod_{k=-N}^{N} \frac{1+ s/\omega^k}{1+ s/\omega'^k} \tag{3}$$

and by applying the following synthesis formulas:

$$\omega_0' = {}^{-0.5}_{\alpha}\omega_n ; \omega_0 = {}^{0.5}_{\alpha}\omega_u ; \frac{\omega_{k+1}}{\omega_k} = \frac{\omega_{k+1}}{\omega^k} = \alpha\eta > 1 \tag{4}$$

$$\frac{\omega'_{k+1}}{\omega^k} = \eta > 0; \frac{\omega^k}{\omega'_k} = \alpha > 0; N = \frac{\log\left(\frac{\omega_N}{\omega_0}\right)}{\log(\alpha\eta)}; \mu = \frac{\log\alpha}{\log(\alpha\eta)} \tag{5}$$

where, ω_u is geometrical mean of the unit gain frequency and the central frequency of a band of frequencies distributed around it. That is,

$$\omega_u = \sqrt{\omega_h \omega_b} \tag{6}$$

where, ω_h and ω_b are the high and low transitional frequencies.

3 Evolutionary Optimization (EO) Techniques

Various EO algorithms used in this paper are described as follows:

3.1 Genetic Algorithm (GA)

GA is a global search technique developed by J. Holland in 1970's. It is used to find true or approximate solutions to complex optimization problems. This technique is inspired by evolutionary biology such as inheritance, mutation, selection and crossover (also called recombination). It is implemented as a simulation problem in which a population of abstract representation (called chromosome or the genotype or the genome) of candidate solutions (called individual, creatures or phenotypes) moves towards better solution by iteratively applying a set of stochastic operators [35-37].

3.2 Particle Swarm Optimization (PSO)
The PSO was developed by J. Kenedy and R. Eberhart in 1995. It is an evolutionary optimization technique based on the movement and intelligence of swarm that move around in a search space looking for the best solution. Each particle is treated as a point in N-dimensional space which adjusts its flight according to its own flying experience as well as the flying experience of the other particles. Each particle tries to improve its position using its current position, its current velocity, the distance between its current position and its best position '*pbest*' and the distance between its current position and global best position '*gbest*'. It is an evolutionary technique similar to GA; but, it has a faster convergence [38-40].

3.3 Differential Evolution (DE)
DE was introduced by Storn and Price in 1996. It is a stochastic, population based optimization algorithm like GA. But one big difference is that DE is developed to optimize real parameters or real valued functions that are non-differentiable, non-continuous, non-linear, noisy, flat, multi dimensional or have many local minima. As a result, the idea of mutation and crossover are substantially different in both the techniques. DE has better convergence to global optimum, more accurate and reduced number of simulations in comparison to other optimization techniques [41-42].

4 Controller design for DC motor speed control
A block diagram of speed control system involving DC motor and a controller is shown in Figure 1. Here, input is the reference speed, controller is PID or FOPID, output is speed $\omega(s)$ and disturbance is due to measuring channel, environment or any kind of noise.

4.1 PID controller
PID control with its three term functionality offers the simplest solution to many real world control problems. A PID controller with four tuning parameters is usually selected:

$$K(s) = k_p + \frac{k_i}{s} + \frac{k_D s}{\tau_D s + 1} \tag{7}$$

Tuning parameters of the controller are: $p = (k_p, k_i, k_D, \tau_D)$

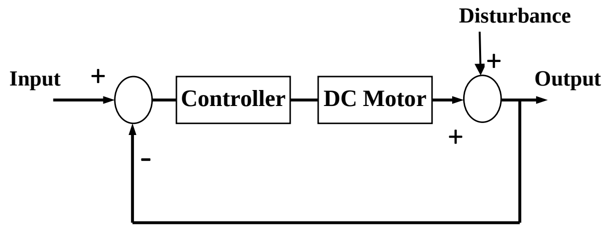

Figure 1. Controller structure for DC motor

4.2 FOPID controller
The differential equation of a fractional order $PI^{\lambda}D^{\mu}$ controller can be described as:

$$u(t) = k_p e(t) + k_i D_t^{-\lambda} e(t) + k_D D_t^{\mu} e(t) \tag{8}$$

and its transfer function can be given as:

$$K(s) = k_p + k_i s^{-\lambda} + k_D s^{\mu} \tag{9}$$

FOPID controller involves selection of five parameters: three parameters (same as PID) and two fractional parameters λ, μ. More flexibility is added in accomplishing control objective by this expansion. A drawback with derivative action is the noise amplification, which can be attenuated by replacing the term $k_D s^{\mu}$ by $\frac{k_D s^{\mu}}{\tau_D s + 1}$ (insertion of pole in the vicinity of the zero of the derivative action). The transfer function of FOPID controller becomes:

$$K(s) = k_p + k_i s^{-\lambda} + \frac{k_D s^{\mu}}{\tau_D s + 1} \tag{10}$$

So, there are six parameters to tune now: $p = (k_p, k_i, k_D, \tau_D, \lambda, \mu)$

4.3 DC motor model
DC machines are characterized by their versatility. By means of various combinations of shunt, series and separately excited field windings, they can be designed to display a wide variety of volt-ampere or speed-torque characteristics for both dynamic and steady state operation and are frequently used in many applications requiring a wide range of motor speeds and a precise output motor control. The schematic diagram of a typical DC motor and its model are shown in Figure 2 and 3 respectively.

From Figure 3, the transfer function from the input voltage, $V(s)$, to the output velocity, $\omega(s)$ and to the output angle, $\theta(s)$ can be written as:

$$\frac{\omega(s)}{V(s)} = \frac{K}{(Ls + R)(Js + B) + K^2} \tag{11}$$

$$\frac{\theta(s)}{V(s)} = \frac{K}{s[(Ls+R)(Js+B)+K^2]} \quad (12)$$

where, K is the emf (Nm/A), L is inductance (henry), R is electrical resistance (ohm), J is the moment of inertia of the rotor (kg.m^2/s^2), and B is the damping ratio of the mechanical system.

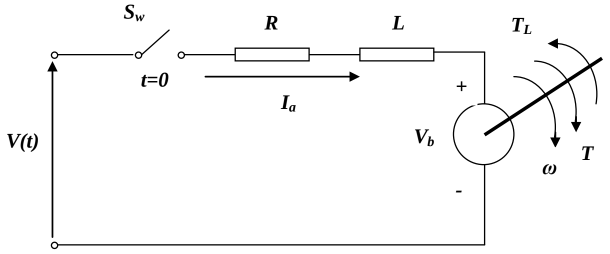

Figure 2. Schematic diagram of a DC Motor

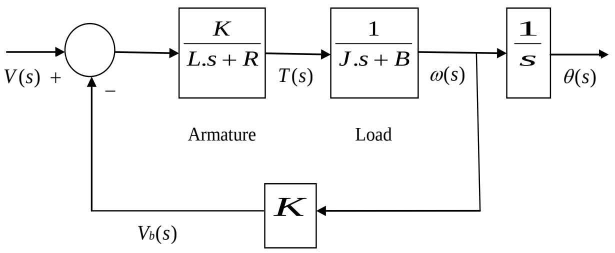

Figure 3. Block diagram of DC motor

4.4 Fitness function in the proposed technique

The fitness function is based on the concept of robust mixed-sensitivity control which is the infinity norm of weighted sensitivity and complimentary sensitivity function [10]. In this control method, the multiplicative weights are taken to formulate the uncertainty which is generated due to changes in the parameters of the DC motor. In this paper, W_1 is the performance weighting function, which is specified for the disturbance rejection of the system to limit the magnitude of the sensitivity function, and W_2 is the robustness weighting function and is specified for the uncertainty in the plant to limit the magnitude of the complementary sensitivity function. This technique, called loop shaping technique, is widely used for selecting the weight functions for the synthesis of the controller. If S is denoted as the plant sensitivity function (transfer function between output and disturbance) and T as the plant's complementary sensitivity function (transfer function between output and input), the cost function can be written in terms of infinity norm as:

$$J = \left\| \begin{matrix} W_1 S \\ W_2 T \end{matrix} \right\|_\infty < 1 \quad (13)$$

In Figure 1, if the DC motor is denoted as plant 'P' and controller as 'K' then S and T can be expressed as:

$$S = (1+PK)^{-1} \quad (14)$$

$$T = PK(1+PK)^{-1} \quad (15)$$

This cost function in (13) is solved as a minimization problem to solve the optimal parameters of the controller.

5 Design Example

The parameters of FOPID and PID controllers have been designed using EO techniques with the objective function given by equation (13). The simulation has been done using MatLab V7.2. The parameters of the armature controlled DC motor are given in Table 1.

Table 1. Parameters of considered DC motor

Motor Parameter	Value
K	0.1 (Nm/A)
L	0.5 (H)
R	2 (ohm)
J	0.02 (kg.m^2/s^2)
B	0.2 (N-m.s/rad)

Thus, the transfer function between speed and voltage of the DC motor by taking above parameters can be written as

$$\frac{\omega(s)}{V(s)} = \frac{0.1}{0.001s^2 + 0.14s + 0.41} \quad (16)$$

The synthesis procedure of H_∞ controller can be done only by selecting proper weight functions. The selection purely depends on the plant model. There are no hard and fast rules for selecting the performance weight function and the robustness weighting function. An iterative work with assumed initial values is usually conducted to find out the weight functions W_1 and W_2. The guidelines for selection of these weighting functions have been explained in [43]. Inverse of W_1 should reflect the desired shape of the sensitivity function (S); whereas, inverse of W_2 should exhibit the shape of the complementary sensitivity function (T). The parameters (coefficients and constants) of W_1 are adjusted such that the singular value curve of S remains below the singular value curve of inverse of W_1. Similarly, the parameters of W_2 are adjusted such that the singular value curve of T remains below the singular value curve of inverse of W_2.

In this work, four cases of different weighting functions have been considered in order to make detailed analysis. The frequency dependent weighting functions considered are:

$W1 = \frac{p_1 s^2 + p_2 s + p_3}{q_1 s^2 + q_2 s + q_3}$ and

$$W_2 = \frac{0.2619s^2 + 5.649s + 19.06}{s^2 + 26.28s + 106.7} \quad (17)$$

In Case 1, W_1 *and* W_2 have been taken from [10]. In Case 2, W_2 has been taken from [10] and W_1 has been tuned using EO technique. In Case 3, W_2 is taken from [10] and denominator of W_1 is tuned using EO technique. In Case 4, W_2 is taken from [10] and numerator of W_1 is tuned using EO technique.

The tuning of controller parameters in Case 1 is explained below:

5.1 Tuning using GA (Case 1)

In Case 1, the EO algorithm (GA, PSO, DE) aims to find optimal value of FOPID controller $[K_p\ K_i\ K_d\ \tau_d\ \lambda\ \mu]$ to minimize the objective function as given in (13). The initial value, lower and upper bound of solution variables are set at [92.38, 198.93, 7.24, 0.0006, 0, 0]; [10, 100, 1, 0.0001, 0, 0] and [1000, 1000, 100, 0.1, 1, 1] respectively. The upper and lower bounds of solution variables have been selected after performing a number of experiments. It was observed that if the range of the bounds is increased then the search space increases and this may lead to non-convergence of the optimization algorithm. On the other hand, if the range is reduced then sufficient optimization is not achieved.

For the proposed simulation, the size of population is taken as 20 and number of generations is equal to 100. The mutation rate is chosen to be 0.05. The GA gets converged in 56 generations with the optimal solution, [291.4001, 672.8005, 99.9982, 0.0832, 0.9969, 1.0000], which on substitution in (10) provide following controller *K(s)*.

$$K(s) = 291.4001 + \frac{672.8005}{s^{0.9969}} + \frac{99.9982s^{1.0000}}{0.0832s + 1}$$

According to the theory of robust mixed sensitivity control, the fitness function chosen should be less than 1and the evaluated infinity norm is 0.5361. Therefore, designed system is robust.

The same GA specifications are used to tune PID controller parameters $[k_p, k_i, k_D, \tau_D]$. The initial value, lower and upper bound of solution variable are set at [95.38, 200.93, 10.24, 0.0009], [1, 10, 1, 0.0001] and [100, 1000, 100, 0.1] respectively. The GA converges in 54 generations with the optimal solution, [29.4099, 825.4733, 9.5151, 0.0158]; which on substitution in (7) provide following controller *K(s)*:

$$K(s) = 29.4099 + \frac{825.4733}{s} + \frac{9.5151s}{0.0158s + 1}$$

The system obtained by the estimated controller is robust as the fitness function obtained is 0.7221, which is less than 1. Figure 4 shows the sensitivity function of the plant using PID and FOPID controller. It is shown that with FOPID, disturbance gets attenuated in larger frequency range. Figure 5 shows that the noise attenuation is more with the FOPID controller. Figure 6 gives the step response of the plant with PID and FOPID controller. A step disturbance is introduced at t=3.5 sec. and it is observed that steady state value is obtained far earlier with FOPID than PID controller.

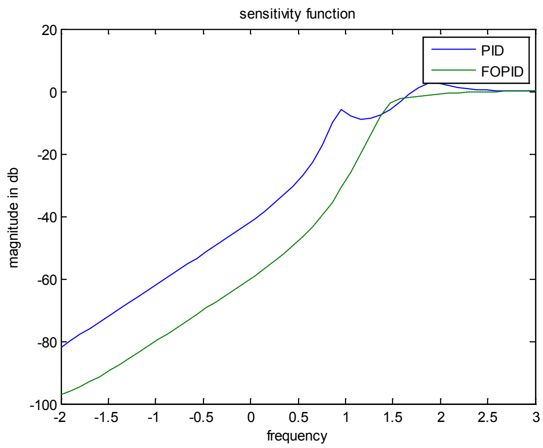

Figure 4. Sensitivity function of DC motor with PID and FOPID controller using GA

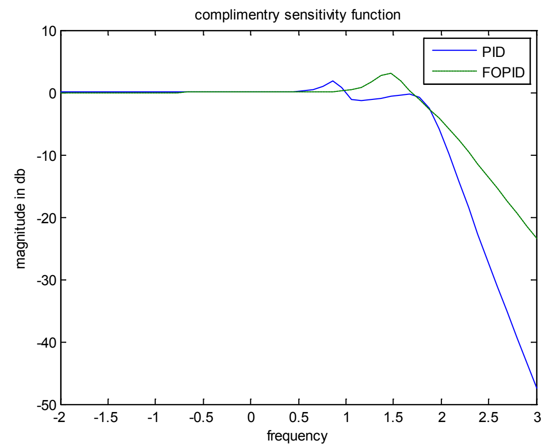

Figure 5. Complimentary sensitivity function of DC motor with PID and FOPID controller using GA

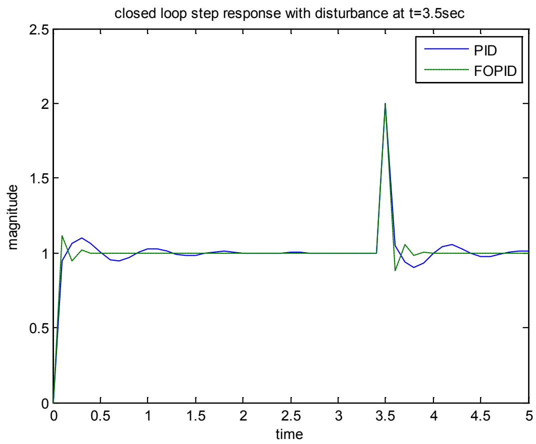

Figure 6. Closed loop step response with PID and FOPID controller using GA

5.2 Tuning using PSO (Case 1)

Initialization parameters used for PSO are: population size = 25, maximum number of iterations = 2500, minimum and maximum velocities are 0 and 2, cognitive acceleration coefficient 'C_1' = 2.1, social acceleration coefficient 'C_2' = 1.3, minimum inertia weight = 0.6 and maximum = 0.9. The PSO converges in 34 iterations with the optimal solution of the FOPID controller, [241.4382, 354.8308, 90.8583, 0.0867, 0.9954, 0.9996], which on substitution in (10) provides the following controller *K(s)*.

$$K(s) = 241.4382 + \frac{354.8308}{s^{0.9954}} + \frac{90.8583 s^{0.9996}}{0.0867 s + 1}$$

The infinity norm obtained by the evaluated controller is 0.5025, which is less than 1. Consequently, the system is robust.

The same PSO specifications are used to tune PID controller parameters. The initial value, lower and upper bound of solution variable are set at as that in case of GA. The PSO converges in 60 iterations with the optimal solution, [72.1678, 202.3394, 4.2014, 0.0001], which on substitution in (7) provides the following controller *K(s)*:

$$K(s) = 72.1678 + \frac{202.3394}{s} + \frac{4.2014 s}{0.0001 s + 1}$$

The system obtained by the estimated controller is also robust as the fitness function obtained is 0.5008. Figure 7 shows the sensitivity function of the plant using PID and FOPID controller. It is shown that with FOPID, disturbance is getting attenuated in larger frequency range. Figure 8 shows that the noise attenuation is more with FOPID controller. Figure 9 gives the step response of the plant with PID and FOPID controller.

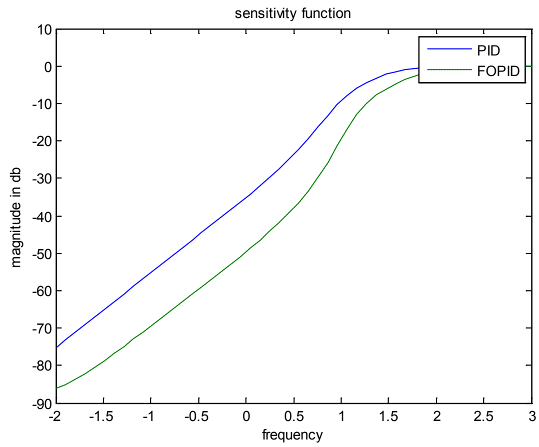

Figure 7. Sensitivity function of DC motor with PID and FOPID controller using PSO

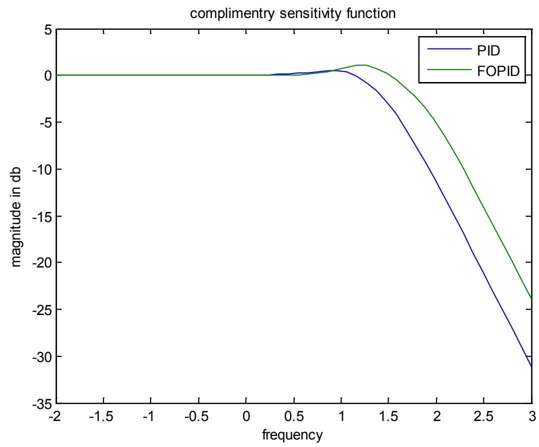

Figure 8. Complimentary sensitivity function of DC motor with PID and FOPID controller using PSO

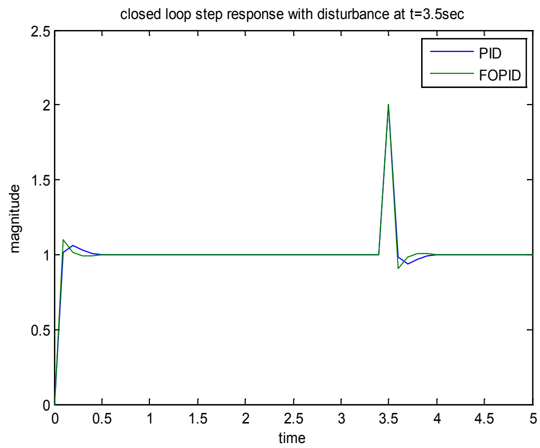

Figure 9. Closed loop step response with PID and FOPID controller using PSO

5.3 Tuning using DE (Case 1)

Initialization parameters used for DE are: value to reach (VTR) = 0.5 (DE will stop its minimization if either the best parameter vector has found a value <= "VTR" or number of iterations reached its maximum value). The other parameters are set as follows: count of population members = 15; maximum number of iteration= 200; step size= 0.8, crossover probability= 0.8. The same initial values, lower and upper bounds for initial values, are taken as in earlier two optimization techniques.

The DE in 200 iterations converges with the optimal solution for FOPID, [125.8399, 82.2894, 52.4687, 0.1972, 1.1986, 1.0188]; which on substitution in (10) provides the following controller *K(s)*:

$$K(s) = 125.8399 + \frac{82.2894}{s^{1.1986}} + \frac{52.4687s^{1.10188}}{0.1972s+1}$$

Here, infinity norm comes out to be 0.5009, therefore, the system is robust. The same DE specifications are used to tune PID controller parameters. The initial value, lower and upper bound of initial values of solution variable are set at as that in case of GA and PSO. The DE converges in 200 iterations with the optimal solution, [65.5010, 985.1620, 12.1832, 0.00001]; which on substitution in (7) provide the following controller *K(s)*:

$$K(s) = 65.5010 + \frac{985.1620}{s} + \frac{12.1832s}{0.00001s+1}$$

Here, infinity norm comes out to be 0.5005; therefore, the system is robust. Figure 10 shows the sensitivity function of the plant using PID and FOPID controller. It shows that with FOPID disturbance is getting attenuated in larger frequency range. Figure 11 shows that the noise attenuation is more with the FOPID controller. Figure 12 gives the step response of the plant with PID and FOPID controller.

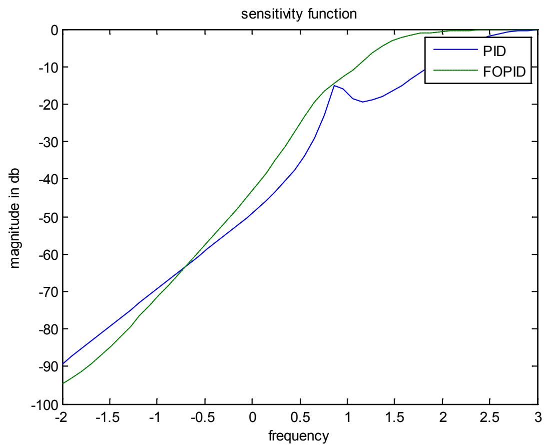

Figure 10. Sensitivity function of DC motor with PID and FOPID controller using DE

The similar exercise was carried out in Case 2, Case 3, and Case 4 as well; where, in addition to the controller parameters, coefficients of weighting function W_1 were also optimized by various optimization techniques. The optimized parameters values of FOPID and PID controllers are given in Table 2-5 for Case 1, Case 2, Case 3 and Case 4 respectively. It can be observed that optimized parameters values for both the controllers using GA and PSO are close to each other as compared to that obtained using DE.

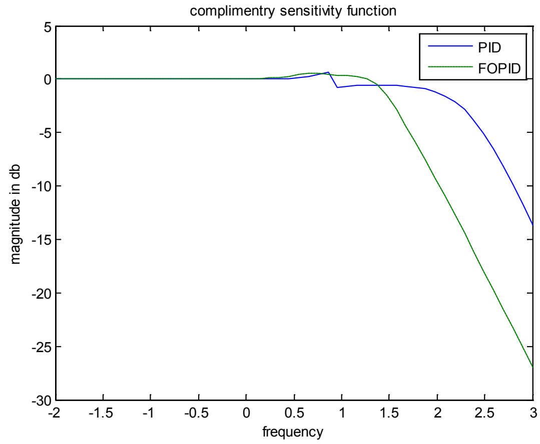

Figure 11. Complimentary sensitivity function of DC motor with PID and FOPID controller using DE

Table 2. Optimized parameters values for Case 1

Optimized Parameters	Value of Optimized parameters and fitness function(f) with FOPID controller					
	GA		PSO		DE	
	FOPID	PID	FOPID	PID	FOPID	PID
k_p	291.40	24.41	241.44	72.168	125.84	65.501
ki	672.80	825.47	354.83	202.34	82.289	985.16
k_D	99.998	9.5151	90.858	4.2014	52.469	12.183
τ_D	0.0832	0.0158	0.0867	0.0001	0.1972	0.00001
λ	0.9969	*	0.9954	*	1.1986	*
μ	1.000	*	0.9996	*	1.0188	*
f	0.5361	0.7221	0.5025	0.5008	0.5009	0.5005
Assumed parameters: p_1=0, p_2=0.5, p_3=10, q_1=0, q_2=1 and q_3=0.001; * Not applicable						

Table 3. Optimized parameters values for Case 2

Optimized Parameters	Value of Optimized parameters and fitness function(f) with FOPID controller					
	GA		PSO		DE	
	FOPID	PID	FOPID	PID	FOPID	PID
k_p	424.54	22.825	142.10	38.952	90.853	65.523
ki	100.25	187.43	100.00	100.02	916.28	195.71
k_D	99.767	17.102	99.997	5.1307	515.78	5.0192
τ_D	0.0819	0.0013	0.0234	0.0001	1.0335	0.0360
λ	0.9969	*	0.9975	*	0.6728	*
μ	0.9884	*	1	*	0.7762	*
p_1	0.1541	0.0248	0.0185	0.0359	0.3096	0.1684
p_2	0.0592	0.8515	0.646	0.2402	7.5188	1.1233
p_3	1.2424	0.8518	2.5366	0.5821	9.0237	3.7505
q_2	0.3793	02666	0.4205	9.8754	8.9039	8.9857
q_3	0.5499	0.1304	0.4666	0.1684	3.7960	9.4919
f	0.3009	0.2190	0.2287	0.1789	0.3549	0.3734
Assumed parameters: q_1=1 * Not applicable						

Table 4. Optimized parameters values for Case 3

Optimized Parameters	Value of Optimized parameters and fitness function(f) with FOPID controller					
	GA		PSO		DE	
	FOPID	PID	FOPID	PID	FOPID	PID
k_p	96.269	34.571	10.160	42.999	363.87	24.969
ki	134.65	65.504	100.13	73.691	185.86	100.12
k_D	80.457	3.2674	45.898	4.8694	985.13	9.7017
τ_D	0.4902	0.0137	0.0577	0.0005	0.8464	0.8607
λ	0.9702	*	0.1871	*	0.7679	*
μ	0.9902	*	1	*	0.7478	*
q_2	0.8547	9.9966	0.8765	9.9892	1.6567	4.4505
q_3	0.8135	9.3605	0.8846	9.9352	0.8048	7.6576
f	0.5880	0.1892	0.5721	0.1854	0.3664	0.4173
Assumed parameters: p_1=0, p_2=0.5, p_3=10, q_1=0 * Not applicable						

Table 5. Optimized parameters values for Case 4

Optimized Parameters	Value of Optimized parameters and fitness function(f) with FOPID controller					
	GA		PSO		DE	
	FOPID	PID	FOPID	PID	FOPID	PID
k_p	62.919	34.571	67.025	42.999	180.40	24.969
ki	280.51	65.504	100.08	73.691	1736.6	100.12
k_D	98.186	3.2674	99.936	4.8694	254.00	9.7017
τ_D	0.7983	0.0137	0.1569	0.0005	0.9000	0.8607
λ	0.5699	*	0.7288	*	0.9000	*
μ	0.9922	*	0.9994	*	0.9000	*
p_2	0.2343	9.9966	0.2235	9.9892	0.2000	4.4505
p_3	0.4242	9.3605	0.3451	9.9352	5.3000	7.6576
f	0.3009	0.1892	0.2317	0.1854	0.3660	0.4173
Assumed parameters: p_1=0, q_1=0, q_2=1 and q_3=0.001 * Not applicable						

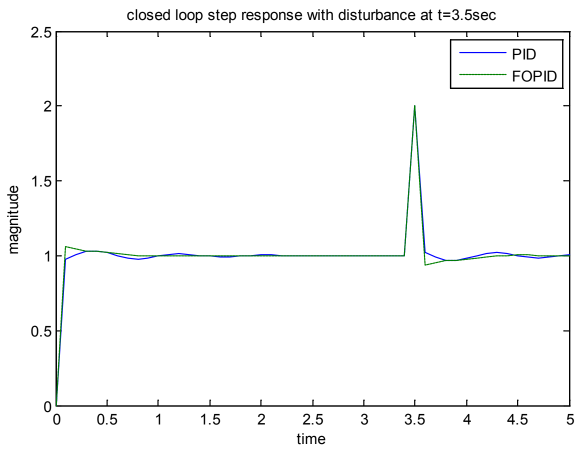

Figure 12. Closed loop step response with PID and FOPID controller using DE

6 Comparison of PID and FOPID controlled process

The comparison drawn between the two controllers has been shown in Table 6 to Table 8 for GA, PSO and DE respectively for all the four cases.

From Table 6-8, it can be observed that the rise time, settling time and peak overshoot are reduced (improved) in case of a FOPID controller than that in the case of a PID controller; but, in case of PSO and GA; overshoot is almost negligible when PID controller is applied. It can also be observed that percentage reduction in performance characteristics is more in case of GA. The variation in the value of the fitness function during iterative process of GA has been shown in Figure 13 and Figure 14 for PID and FOPID respectively. The variation in the value of the fitness function during iterative process of PSO has been shown in Figure 15 and Figure 16 in case of PSO for PID and FOPID respectively. The variation in the value of the fitness function during iterative process of DE has been shown in Figure 17 and Figure 18 for PID and FOPID respectively. It is observed that though PSO requires less number of iterations than the other techniques but GA converges more sharply than the other techniques i.e. variation in the fitness function with successive iteration is very less in GA in comparison to other optimization techniques. Further, it can also be observed that there are many large spikes in fitness function of PID controller in Figure 15 (PSO) and Figure 17 (DE); whereas; in case of Figure 13 (GA) the convergence is smooth. This may be due to the characteristic of the process undertaken by different algorithms for searching the search space.

7 Conclusion

In this paper, a comparison of various evolutionary optimization techniques for the designing of PID and FOPID controllers for DC motor speed control has been investigated. The comparison involves frequency domain performance criterion. In order to examine the effect of cost function, different cases are examined by taking different coefficients of performance weighting function unknown. The performance of PID and FOPID is compared using GA, PSO and DE for different cases and it is observer that in each case, FOPID has better performance characteristics in terms of rise time, settling time and overshoot. Considering all the results, it is also observed that though PSO requires less number of iterations than the other techniques but variation in the fitness function with successive iteration is very less in GA in comparison to other optimization techniques.

Table 6. Comparison of PID and FOPID controller Using GA

Parameters	PID controlled process				FOPID controlled process			
Cases	Case1	Case2	Case3	Case4	Case1	Case2	Case3	Case4
Rise time (s)	0.057	0.221	0.198	0.261	0.0176	0.0192	0.0387	0.030
Settling time (s)	0.982	2.35	0.898	0.539	0.156	0.092	0.475	0.312
Peak Overshoot (%)	18.07	4.92	0	0	17.3	13.6	9.86	12.2
Steady state error	0	0	0	0	0	0	0	0

Table 7. Comparison of PID and FOPID controller using PSO

Parameters	PID controlled process				FOPID controlled process			
Cases	Case1	Case2	Case3	Case4	Case1	Case2	Case3	Case4
Rise time (s)	0.045	0.182	0.176	0.206	0.038	0.031	0.017	0.041
Settling time (s)	0.0775	0.543	0.907	0.557	0.377	0.277	0.445	0.308
Peak Overshoot (%)	0.881	0	0	0	15.4	3.82	6.81	3.78
Steady state error	0	0	0	0	0	0	0	0

Table 8. Comparison of PID and FOPID controller using DE

Parameters	PID controlled process				FOPID controlled process			
Cases	Case1	Case2	Case3	Case4	Case1	Case2	Case3	Case4
Rise time (s)	0.0197	0.0421	0.0937	0.057	0.0703	0.0085	0.0051	0.0129
Settling time (s)	0.081	0.239	0.545	1.78	0.76	0.0831	0.021	0.282
Peak Overshoot (%)	4.52	16.6	22.6	32.1	4.67	7.04	8.62	15.7
Steady state error	0	0	0	0	0	0	0	0

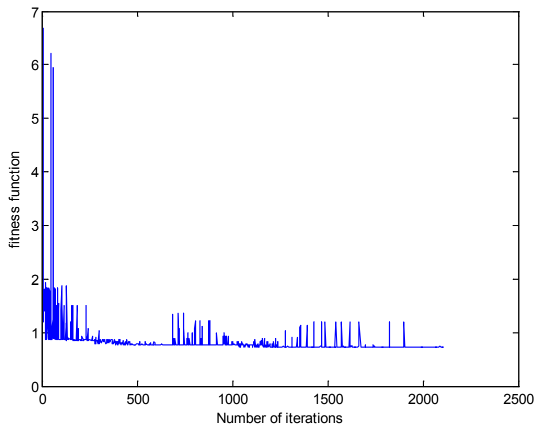

Figure 13. Fitness function using GA for PID controller

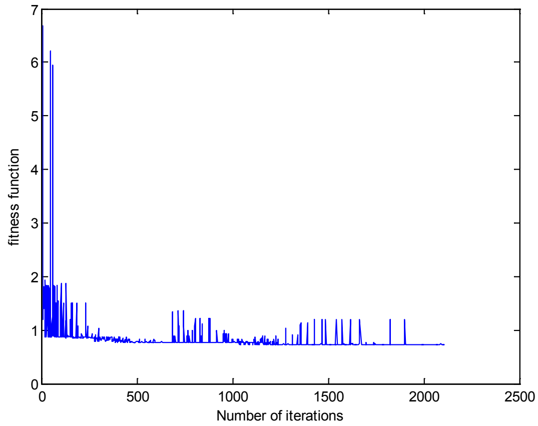

Figure 14. Fitness function using GA for FOPID controller

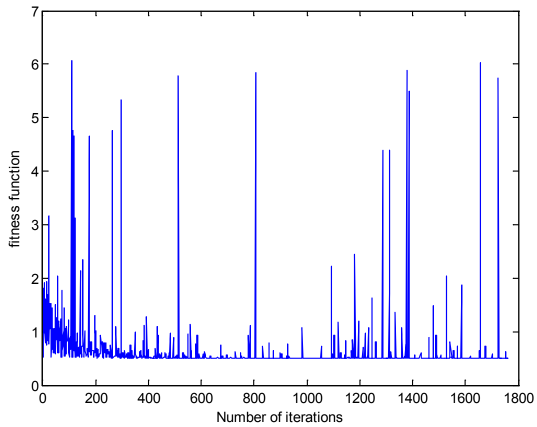

Figure 15. Fitness function using PSO for PID controller

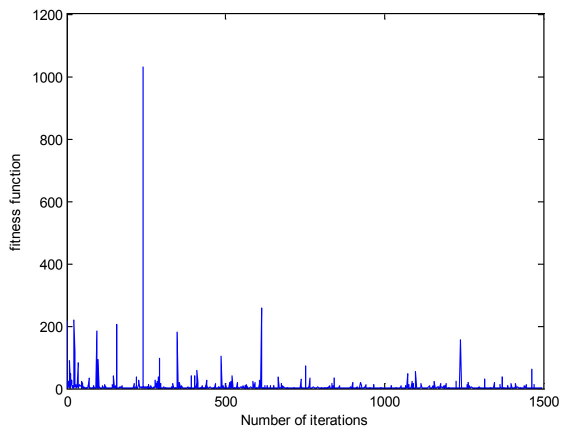

Figure 16. Fitness function using PSO for FOPID controller

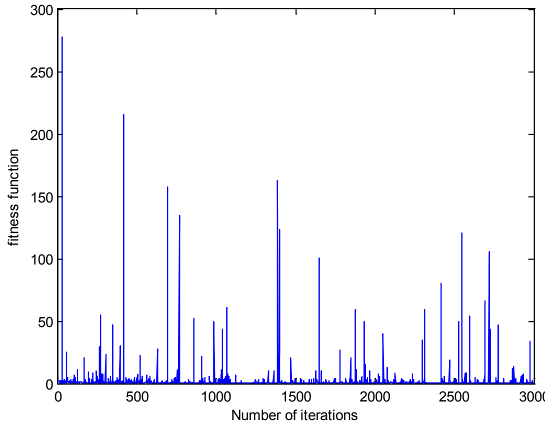

Figure 17. Fitness function using DE for PID controller

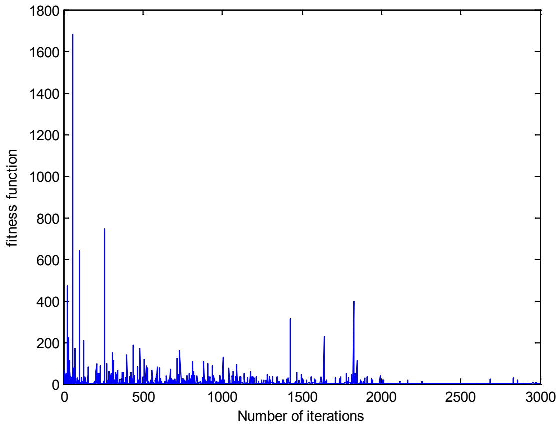

Figure 18. Fitness function using DE for FOPID controller

References:

[1] K. J. Astrom and T. Hagglund, *PID Controller: Theory, Design and Tuning*, Instrument Society of America, 2nd edition, 1995.

[2] C. Hwang, J. Leu, and S. Tsay, "A note on time-domain simulation of feedback fractional-order systems", *IEEE Transactions on Automatic Control*, vol. 47, no. 4, April 2002, pp. 625-631.

[3] I. Petras, *Fractional Order Non Linear Systems: Modeling, Analysis and Simulation*, Higher Education Press, Beijing and Springer-Verlag Berlin Heidelberg, 2011.

[4] M. Zamani, M. Ghartemani, N. Sadati and M. Parniani, "Design of a fractional order PID controller for an AVR using particle swarm optimization," *Control Engineering Practice*, vol. 17, 2009, pp. 1380–1387.

[5] Y. Luo, Y. Q. Chen, H. Ahn and Y. G. Pi, "Fractional order robust control for cogging effect compensation in PMSM position servo systems: Stability analysis and experiments," *Control Engineering Practice*, vol. 18, 2010, pp. 1022–1036.

[6] M. Nakagava and K. Sorimachi, "Basic characteristics of a fractance device," *IEICE Transactions on Fundamental Electronics*, vol. E75-A, no. 12, 1992, pp. 1814–1818.

[7] A. Biswas, S. Das, A. Abraham and S. Dasgupta, "Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution," *Journal of Engineering Applications of Artificial Intelligence*, vol. 22, 2009, pp. 343–350.

[8] J. Cao and B. Cao, "Design of fractional order controller based on particle swarm optimization," *International Journal of Control Automation*, vol. 4, no. 6, 2006, pp. 775-781.

[9] T. V. Moghaddam, N. Bigdeli and K. Afshar, "Tuning a fractional order PID controller with lead compensator in frequency domain," *International journal of electrical and electronics engineering*, vol. 5, no.3, 2011, pp. 139-144.

[10] U. Chaiya and S. Kaitwanidvilai, "Fixed structure robust DC motor speed control," *Proceeding of the International Multi-conference of Engineering and Computer Scientists*, vol. II, March 2009, pp. 18-20.

[11] J. P. Jiang, S. C. and P. K. Sinha, "Optimal feedback control of direct current motors, " *IEEE Transactions on Industrial Electronics*, vol. 31, no. 4, Aug. 1990, pp. 269-274.

[12] A. A. El-Samahy, "Speed control of DC motor using adaptive variable structure control," *IEEE 31st Annual Power Electronics Specialists Conference*, 2000, pp. 1118- 1123.

[13] S. A. Nyabundi, G. Qi Haman and J. Munda, "DC motor control via high order differential feedback control," AFRICON 2009, *AFRICON'09*, 23-25 Sep. 2009, pp. 1-5.

[14] B. Allaoua, A. Abderrahmani, B. Gasbaoui and A. Nasri, "The efficiency of particle swarm optimization applied on fuzzy logic DC motor speed control," *Serbian Journal of Electrical Engineering*, vol. 5, no. 2, Nov. 2008, pp. 247-262.

[15] L. Renquan, L. Shukui, X. Lin and X. Anke, "Robust H_{∞} optimal speed control of linear DC motor," *Proceeding of the 27th Chinese Control conference*, July 2008, pp. 743-747.

[16] D. McFarlane and K. Glover, "A loop shaping design procedure using H_{∞} synthesis," *IEEE Transactions on Automatic Control*, vol. 37, June 1992, pp. 759-767.

[17] M. D. Minkova, D. Minkov, J. L. Rodgerson and R. G. Harley, "Adaptive neural speed controller of a DC motor," *Electric Power*

System Research, vol. 47, March 1998, pp. 123-132.

[18] P. Van den Braembussche, J. Swevers and H. van Brussel, "Design and experimental validation of robust controllers for machine tool drives with linear motor," *Mechatronics*, vol. 11, pp. 545-562, 2001.

[19] T. Ray, "Constrained robust optimal design using a multi objective evolutionary algorithm," *Proceedings of the 2002 Congress on Evolutionary Computation*, *CEC 2002*, 12-17 May 2002, pp. 419-424.

[20] G. M. Rao and B. V. Sanker Ram, "A neural network based speed control of DC motor," *International Journal of Recent Trends in Engineering*, vol. 2, no. 6, Nov. 2009, pp. 121-124.

[21] B. Allaoua, A. Abderrahmani, B. Gasbaoui and A. Nasri, "Intelligent controller design for DC motor speed control based on fuzzy logic genetic algorithms optimization," *Leonardo Journal of Sciences*, Dec. 2008, pp. 90-102.

[22] M. Fallahi and S. Azadi, "Robust control of DC motor using fuzzy sliding mode control with PID compensator," *Proceedings of the International Multi-conference of Engineers and Computer Scientists*, vol. II, March 2009.

[23] M. Nasir, H. Nezamabadi-Pour and M. Maghfoori, "A PSO based optimum design of PID controller for a linear brushless DC motor," *World Academy of Science, Engineering and Technology*, Jan. 2007, pp. 211-215.

[24] Adel A. A. El-Gammal and Adel A. El-Samahy, "A modified design of PID controller for DC motor drives using particle swarm optimization," International conference on Power Engineering, Energy and Electrical Drives 2009, *POWERENG'09*, March 2009, pp. 419-424.

[25] T. Kim, I. Maruta and T. Sugie, "Robust PID controller tuning based on the constrained particle swarm optimization", *Automatica*, vol. 44, No.4, Aug. 2007, pp. 1104-1110.

[26] D.Valerio and J.Sa Da Costa, "Tuning of fractional PID controllers with Ziegler–Nichols type rules," *Signal Processing*, vol. 86, no. 10, 2006, pp. 2771–2784.

[27] Z. Bingul and O. Karahan, "Fractional PID controllers tuned by evolutionary algorithms for robot trajectory control," *Turkish Journal of Electric Engineering and Computer Sciences*, Vol.20, No.Sup.1, 2012, pp. 1123-1136.

[28] F. Padula and A. Visioli, "Tuning rule for optimal PID and fractional order PID controller," *Journal of Process Control*, Vol. 21,issue 1, Jan 2011, pp. 69-81.

[29] C. A. Monje, B. M. Vinagre, V. Feilu and Y. Q. Chen, "Tuning and auto tuning of fractional order controller for industry applications," *Control Engineering Practice*, vol. 16, no. 7, 2008, pp. 798-812.

[30] D. Maiti, S. Biswas and A. Konar, "Design of fractional order PID controller using particle swarm optimization technique," *2nd international conference on recent trends in information systems (ReTIS-08)*, 2008.

[31] K. B. Oldham and J. Spanier, *The Fractional Calculus*, New York: Academia, 1974.

[32] K. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*, New York: Wiley, 1993.

[33] S. G. Samko, A. A. Kilbas and O. I. Marichev, *Fractional Integrals and Derivatives*, Gordon and Breach Science Publication, 1993.

[34] Gao, Zhe, Liao, Xiaozhong, "Improved Oustaloup approximation of fractional order operator using adaptive chaotic particle swarm optimization," *Journal of System Engineering Electronics*, vol. 23, issue 1, Feb. 2012, pp. 145-153.

[35] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Asia: Pearson Education, 2001.

[36] H. G. Beyer and K. Deb, "On self-adaptive features in real-parameter evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 3, 2001, pp. 250-270.

[37] K. Deb, A. Anand, and D. Joshi, "A computationally efficient evolutionary algorithm for real-parameter optimization," *Evolutionary Computation*, vol. 10, no. 4, 2002, pp. 371-395.

[38] M. Clerc, *Particle Swarm Optimization*, ISTE Ltd., 2006.

[39] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization and intelligence: Advances and applications," *Information Sciences*, 2010.

[40] R. C. Eberhart and Y. Shi, "Particle Swarm Optimization: Developments, Applications and Resources", *Proceedings of the 2001 Congress on Evolutionary Computation*, *CEC 2001*, vol. 1, pp. 81-86.

[41] R. Storn and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," *Journal*

of Global Optimization, vol.11, 1997, pp. 341-359.

[42] J. Brest, S. Greiner, M. Mernik and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, Dec. 2006, pp. 646-657.

[43] John E. Bibel and D. Stephen Malyevac, "Guidelines for the selection of Weighting functions for H-infinity control," Technical Report, Naval Surface Warfare Center, Virginia, Jan 1992. Available at: www.dtic.mil/dtic/tr/fulltext/u2/a251781.pdf.