TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 12, December 2014, pp. 8140 ~ 8151
DOI: 10.11591/telkomnika.v12i12.6470

Design of Fractional Order PID Controller for DC Motor using Genetic Algorithm

Ashu Ahuja*[1], Bhawna Tandon[2]
Electrical Engineering Department, Maharishi Markandeshwar Engineering College, Mullana, India.
*Corresponding author, e-mail: ashu.valecha.ahuja@gmail.com[1], bhawna _aneja@rediffmail.com[2]

Abstract

Design of fractional order PID (FOPID) controller for DC motor is proposed in this paper. A FOPID ($PI^{\lambda}D^{\mu}$) is a PID controller whose derivative and integral orders are fractional numbers rather than integers. Design stage of such controller consists of determining six parameters – proportional constant (K_p), integral constant (K_i), derivative constant (K_d), filter time constant (τ_d), integral order (λ) and derivative order (μ). The proposed approach poses the problem as designing a DC motor speed controller on the concept of fixed structure robust controller and mixed sensitivity H_∞ method. The uncertainty caused by the parameter changes of motor resistance, motor inductance and load are formulated as multiplicative uncertainty weight, which are used in the objective function in the design. Genetic Algorithm (GA) and Simulated Annealing (SA) are employed to carry out the aforementioned design procedure. Comparisons are made with a PID with derivative first order filter controller and it is shown that the proposed FOPID controller can highly improve the system robustness with respect to model uncertainties. The comparison of PID and FOPID controllers is also been done on the basis of Time Domain Performance index i.e. ISE (Integral of Square Error).

Keywords: *fractional order controller, mixed sensitivity, simulated annealing (SA), algorithm, genetic algorithm, PID controller.*

1. Introduction

Proportional–integral–derivative (PID) controller is the most widely used controller structure in industrial applications [1]. Its structural simplicity and ability to solve many practical control problems have contributed to this wide acceptance. In PID controller the derivative and the integral order are in integer. Fractional order PID (FOPID) is a special kind of PID controller whose derivative and integral order are fractional rather than integer. The key challenge of designing FOPID controller is to determine the two key parameters λ (integral order) and μ (derivative order) apart from the usual tuning parameters of PID using different tuning methods. Both λ and μ are in fraction which increases the robustness of the system and gives an optimal control [2-6]. This paper proposes a novel tuning method for tuning λ and μ of FOPID using genetic algorithms [7-9].

The speed of DC motor can be adjusted to a great extent as to provide controllability easy and high performance [10, 11]. Duncan McFarlane [12] in 1992 introduced a design procedure which incorporate loop shaping methods to obtain performance and robust stability trade off and a particular H_∞ optimization problem to guarantee closed loop stability. M. D. Minkova [13] in 1998 applied adaptive neural method and A. A. El-Samahy [14] in 2000 described robust adaptive discrete variable structure control scheme for speed control of DC motor. In DC motor speed control, many engineers attempt to design a robust controller to ensure both the stability and the performance of the system under the perturbed conditions. One of the most popular techniques is H_∞ optimal control [10, 15, 21] in which the uncertainty and performance can be incorporated into the controller design. A multi objective formulation [16] is introduced by Tapabrata Ray in 2002. The controllers of the speed that are conceived for goal to control the speed of DC motor are numerous: PID Controller, Fuzzy Logic Controller; or the combination between them [17]: Augmented Lagrangian Particle Swarm Optimization [20], Linear Matrix Inequality [21], Fuzzy-Swarm [22], Fuzzy-Neural Networks, Fuzzy-Genetic Algorithm [23], Fuzzy- Ants Colony, Fuzzy-Sliding mode control [24], Particle Swarm Optimization [18, 25], Neural Network [26].

Received July 15, 2014; Revised October 14, 2014; Accepted November 12, 2014

In this paper, Genetic Algorithm is employed to design an FOPID controller for DC motor speed control. The proposed controller is simulated with six tuning parameters (K_p, K_i, K_d, τ_d, λ, μ) and its performance is compared with those of an optimally designed PID controller with four tuning parameters (K_p, K_i, K_d, τ_d). 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, 3 and 4 overview the concepts of fractional calculus, Genetic Algorithm and Simulated Annealing (SA) Algorithm respectively. Design of the proposed FOPID controller for DC motor using GA and SA, taking both Mixed Sensitivity and Integral of Square error (ISE) as cost functions is described in Section 5. Section 6 is devoted to computer simulation of the proposed controller and its comparison with a PID controller. Section 7 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 v not limited to integer numbers, that generalizes the ordinary concepts of derivative (for a positive v) and integral (for a negative v) [27]. There are different definitions for fractional derivatives. The most usual definition is introduced by Riemann and Liouville [28] that generalizes the following definitions corresponding to integer orders:

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

The generalized definition of D becomes ${}_cD_x^{\,v} f(x)$. The Laplace transform of D pursues the renowned rule $L[{}_cD_x^{\,v} f(x)] = S^v F(s)$ for zero initial conditions. This means that, if zero initial conditions are assumed, the systems with dynamic behaviour described by differential equations including fractional derivatives give rise to transfer functions with fractional orders of s. More details are provided in [29] and [30].

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 there cursive distribution of poles and zeroes [3]. The method is based on the approximation of a function of the form:

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

By a rational function:

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

Using the following set of synthesis formulas,

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

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

$$\mu = \frac{\log \alpha}{\log(\alpha\eta)}$$

With ω_u being the unit gain frequency and the central frequency of a band of frequencies geometrically distributed around it. That is:

$$\omega_u = \sqrt{\omega_h \omega_b} \tag{5}$$

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

3. Genetic Algorithm

The genetic algorithm (GA) [31] is an optimization technique that performs a parallel, stochastic and directed search to evolve the fittest (best) solution. Different from conventional optimization methods, GA employs the principles of evolution, natural selection and genetics, as inspired by natural biological systems, in a computer algorithm to simulate evolution.

Three main operators comprising GAs are: reproduction, crossover, and mutation.

Reproduction: - Evolution is, in effect, a method of searching among an enormous number of possibilities for solutions. For the analysis and control of DC motor, PID controller parameters selection is used for reproduction. A string is permitted reproduction based on fitness for productivity, where productivity of an individual is defined as the value of a string's non-negative objective function.

Crossover: - The crossover operator exchanges genetic information between strings. There are a number of commonly used crossover operators: such as blend crossover (BLX), simulated binary crossover (SBX), unimodal normal distribution crossover (UNDX) and simplex crossover (SPX) and parent centric recombination operator (PCX) [32]. In the present paper PCX operator has been used because this particular operator assigns more probability keeping an offspring closer to the parents than away from parents.

Mutation: - Real coded mutation (RCM) operator [33] has been used to protect the irrecoverable or premature loss of important notions. Since continuous variables are coded directly, RCM is flexible in nature. PCX and RCM operator have been used in conjunction and attain search power similar to the individual methodologies, yet the overall algorithm performs better than binary-coded GAs.

4. Simulated Annealing Algorithm

Simulated Annealing (SA) is motivated by an analogy to annealing in solids. The idea of SA comes from a paper published by Metropolis etc al in 1953 (Metropolis, 1953). The algorithm in this paper simulated the cooling of material in a heat bath. This is a process known as annealing.

If a solid is heated past melting point and then cool it, the structural properties of the solid depend on the rate of cooling. If the liquid is cooled slowly enough, large crystals will be formed. However, if the liquid is cooled quickly (quenched) the crystals will contain imperfections. Metropolis's algorithm simulated the material as a system of particles. The algorithm simulates the cooling process by gradually lowering the temperature of the system until it converges to a steady, frozen state. In 1982; Kirkpatrick et al (Kirkpatrick, 1983) took the idea of the Metropolis algorithm and applied it to optimisation problems. The idea is to use simulated annealing to search for feasible solutions and converge to an optimal solution.

5. DC Motor Design using FOPID

5.1. FOPID Controller

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

$$u(t) = k_p e(t) + k_i D_t^{-\lambda} e(t) + k_D D_t^{\mu} e(t) \tag{6}$$

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

$$G_c(s) = k_p + k_i s^{-\lambda} + k_D s^{\mu} \quad (7)$$

Design of an FOPID controller involves design of three parameters k_p, k_i, k_D, and two orders λ, μ 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. A drawback with derivative action is that an ideal derivative has very high gain for high frequency signals. This means that high frequency measurement noise will generate large variations of the control signal. The effect of measurement noise be reduced by replacing the term $k_D s^{\mu}$ by $\frac{k_D s^{\mu}}{\tau_D s + 1}$.
Therefore, the transfer function of FOPID controller is:

$$G_c(s) = k_p + k_i s^{-\lambda} + \frac{k_D s^{\mu}}{\tau_D s + 1} \quad (8)$$

So, there are six tuning parameters to tune now.

$$p = [k_p, k_i, k_D, \tau_D, \lambda, \mu] \quad (9)$$

5.2. PID Controller

The Proportional-Integral-Derivative (PID) controller [34-35] is the most common form of feedback and also a requisite element of early governors. PID control with its three term functionality covering both transient and steady-states response, offers the simplest and the most efficient solution to many real world control problems. Because of the above advantages PID controller with four tuning parameters are selected:

$$K(s) = k_p + k_i s + \frac{k_D s}{\tau_D s + 1} \quad (10)$$

Tuning parameters of the controller are:

$$p = [k_p, k_i, k_D, \tau_D] \quad (11)$$

5.3. 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. Because of the ease by with which they can be controlled, systems of DC machines have been frequently used in many applications requiring a wide range of motor speeds and a precise output motor control. The schematic diagram of a typical armature controlled DC motor is shown in Figure 1.

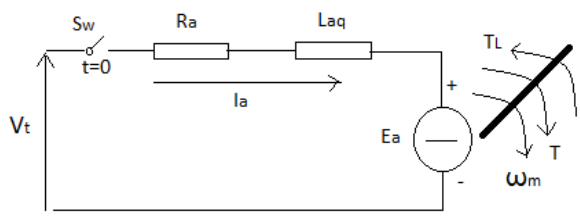

Figure 1. Schematic diagram of armature controlled DC Motor

A well known model of armature controlled DC motor for a speed control system is shown in Figure 2 and its transfer function is represented by Equation (1) and (2) .

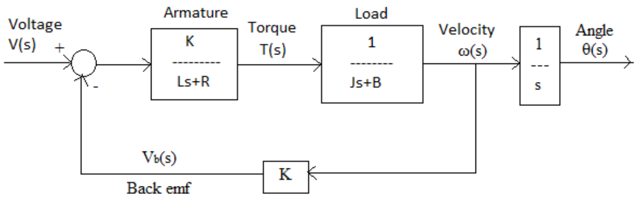

Figure 2. Block diagram of armature controlled DC motor

From the above diagram, the transfer function from the input voltage, V(s), to the output velocity, ω(s) and to the output angle, θ(s) can be written with:

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

Where *J* (kg.m^2/s^2) is the moment of inertia of the rotor, *B* is the damping ratio of the mechanical system, *R* (ohm) is electrical resistance, *L* (H) is electrical inductance, and *K* (Nm/A) is the electromotive force constant.

5.4. Cost Functions in Proposed Technique

The comparison of PID and FOPID controllers is done on the basis of Frequency Domain and Time Domain Performance Indices i.e. Mixed Sensitivity and Integral of Square Error respectively. The cost function in the design is the infinity norm based on the concept of robust mixed-sensitivity control is given by equation (14). In the mixed-sensitivity method, firstly, the weighting function of the plant's perturbation and/or performance must be specified. In this paper, *W*2 is specified for the uncertainty weight of the plant and *W*1 is specified for the disturbance attenuation of the system. The cost function [36-37], can be written as:

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

Where *T* is the plant's complementary sensitivity function and S is the plant sensitivity function. Assume that the plant is denoted as *P.* The controller is denoted as *K* and the system is the unity negative feedback control. The sensitivity and complementary sensitivity function can be expressed as:

$$\begin{aligned} S &= (1+PK)^{-1} \\ T &= PK(1+PK)^{-1} \end{aligned} \quad (14)$$

The cost function in Equation (13) is based on frequency domain specifications.

In Controller design methods, the most common Time Domain Performance criteria are integrated absolute error (IAE), the integrated of time weight square error (ITSE), integrated of squared error (ISE) and Mean Square Error (MSE). These four integral performance criteria have their own advantages and disadvantages.

ISE is proposed as cost function for designing and comparison of PID and FOPID controllers.

$$ISE = \int_0^t e^2(t)\,dt \quad (15)$$

Where e is the steady state error in the step response of the system.

6. Design Example

The parameters of FOPID and PID controllers have been designed using GA with objective functions given by Equation (13) and (15) and simulation has been done using MATLAB. The parameters of the DC motor are given in Table 1.

Table 1. Parameters of DC motor

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

Thus, the transfer function of the DC motor can be written as:

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

6.1. Controller design using Mixed Sensitivity

The synthesis procedure of H$_\infty$ 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 functions. An iteration work with assumed initial values is conducted to find out the weight functions. The frequency dependent weighting functions in case (i) are:

$$W_1 = \frac{0.5s + 10}{s + 0.001} \quad \text{and}$$

$$W_2 = \frac{0.2619\,s^2 + 5.649\,s + 19.06}{s^2 + 26.28\,s + 106.7}$$

In case (ii), W_2 is taken as above and W_1 is tuned using optimization. In case (iii), W_2 is taken as above and denominator of W_1 is tuned using optimization. In case (iv), W_2 is taken as above and numerator of W_1 is tuned using optimization.The size of population of GA is often chosen between [20,100]. For the proposed simulation, the size of population is taken as 20. The number of generation is often chosen between [100,500]. For the proposed case, number of generations is equal to 100. The mutation rate is chosen to be 0.05. The weight co-efficient w1, w2 and w3 are 0.988, 0.001 and 3.0 respectively.

The GA algorithm aims to find optimal value of $p = [k_p, k_i, k_D, \tau_D, \lambda, \mu]$ to minimize the objective function given by (14). The initial value, lower and upper bound of solution variable 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 GA converges with the optimal solution, [670.2301 482.4493 99.9443 0.0673 0.9994 0.9717], which on substitution to Equation (9) provide following controller *K(s)*.

$$K(s) = 670.2301 + \frac{482.4493}{s^{0.9994}} + \frac{99.9443 s^{0.9717}}{0.0673 s + 1}$$

The infinity norm obtained by the evaluated controller is 0.6288 which is less than 1. Consequently, since this norm is less than 1, then the system is robust according to the concept of Mixed Sensitivity robust control. The same GA specifications are used to tune PID controller parameters $p = [k_p, k_i, k_D, \tau_D]$. 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 with the optimal solution, [29.4099, 825.4733, 9.5151, 0.0158], which on substitution to Equation (11) provide following controller *K(s)*.

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

The infinity norm obtained by the evaluated controller is 0.7221 which is less than 1. Consequently, since this norm is less than 1, then the system is robust according to the concept of mixed sensitivity robust control.

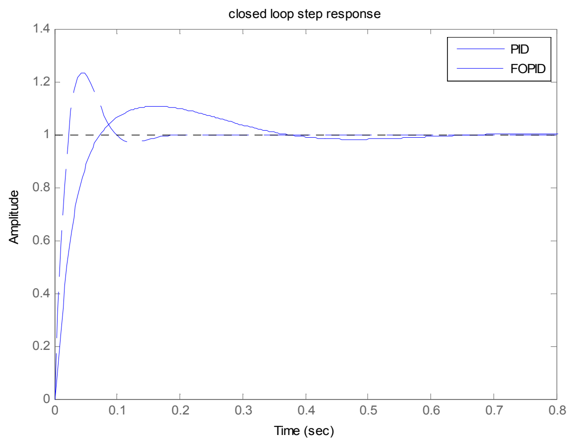

Figure 3. Closed loop step response with PID and FOPID controllers taking Mixed Sensitivity as an objective function

The comparison drawn between the two controllers is explained in the Table 2:

Table 2. Comparison of PID and FOPID controller

Parameters	PID controlled process				FOPID controlled process			
Cases	Case1	Case2	Case3	Case4	Case1	Case2	Case3	Case4
Rise time	.053	.063	.056	.063	.017	.022	.027	.013
Settling time	0.32	0.52	0.9	1.4	0.094	0.149	0.4	0.3
Steady state error	0	0	0	0	0	0	0	0
Peak overshoot	15.4%	13.9%	12.3%	11.2%	16.3%	17.2%	19.4%	19.7%

From the table above, the rise time reduction and the settling time reduction is calculated and is written in the following tables:

Table 3. Rise time reduction

Case 1	67.9%
Case 2	65.2%
Case 3	51.78%
Case 4	79.3%

Table 4. Settling time reduction

Case 1	70.62%
Case2	71.34%
Case 3	55.5%
Case 4	78.57%

Table 3 and 4 shows that the rise time and settling time are being reduced in case of a FOPID controller than in the case of a PID controller.

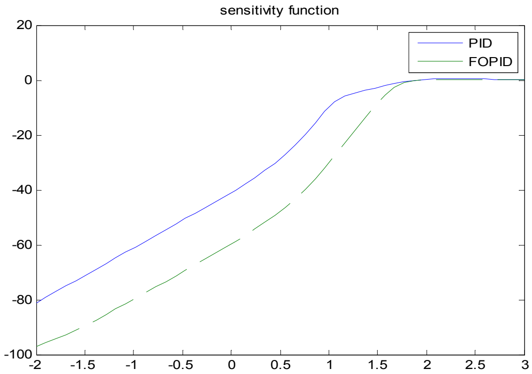

Figure 4. Sensitivity function of DC motor with PID and FOPID controller taking Mixed Sensitivity as an objective function

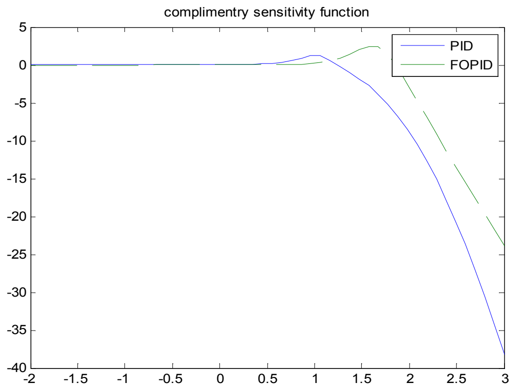

Figure 5. Complimentary Sensitivity function of DC motor with PID and FOPID controller taking Mixed Sensitivity as an objective function

Figure 4 and 5 shows Sensitivity and Complimentary Sensitivity functions for the DC motor with both PID and FOPID controllers using GA.

6.2. Controller Design using Integral of Square Error

The parameters of GA in controller design using ISE as an objective function are taken same as in controller design using Mixed Sensitivity as an objective function. The GA algorithm aims to find optimal value of $p=[k_p,k_i,k_D,\tau_D,\lambda,\mu]$ to minimize the objective function given by (16). The initial value, lower and upper bound of solution variable 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 GA converges with the optimal solution, [162.8495 253.9430 90.7647 0.0351 0.9965 0.9758], which on substitution to Equation (9) provide following controller *K(s)*.

$$K(s) = 162.8495 + \frac{253.9430}{s^{0.9965}} + \frac{90.7647\ s^{0.9758}}{0.0351\ s + 1}$$

The infinity norm obtained by the evaluated controller is 0.5034 which is less than 1. Consequently, since this norm is less than 1, then the system is robust according to the concept of Integral of Square Error. The same GA specifications are used to tune PID controller parameters $p=[k_p,k_i,k_D,\tau_D]$. 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 with the optimal solution, [23.5354 578.0759 10.4080 0.0176], which on substitution to Equation (11) provide following controller *K(s)*.

$$K(s) = 23.5354 + \frac{578.0759}{s} + \frac{10.4080\ s}{0.0176\ s+1}$$

The infinity norm obtained by the evaluated controller is 0.7438 which is less than 1. Consequently, since this norm is less than 1, then the system is robust according to the concept of Integral of Square Error.

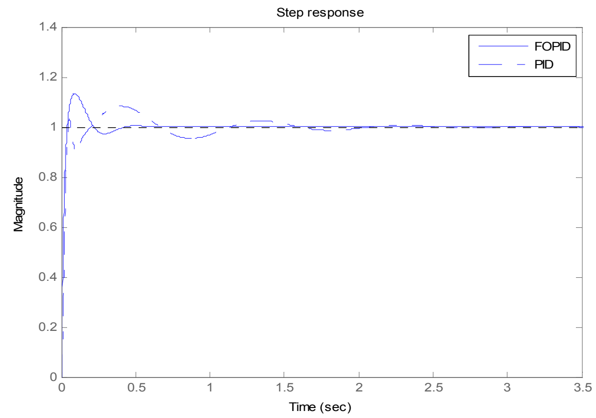

Figure 6. Closed loop step response with PID and FOPID controller taking Integral of Square Error as an objective function

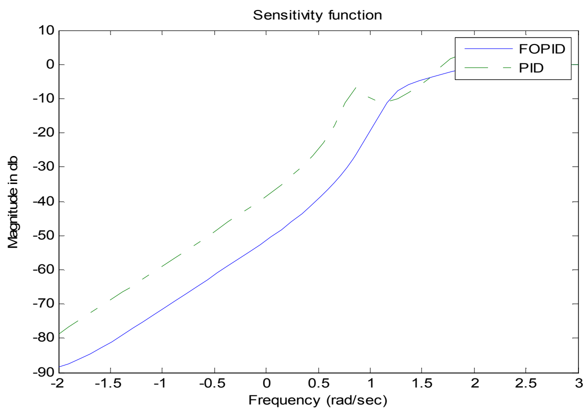

Figure 7. Sensitivity functions of DC motor with PID and FOPID controllers taking Integral of Square Error as an objective function

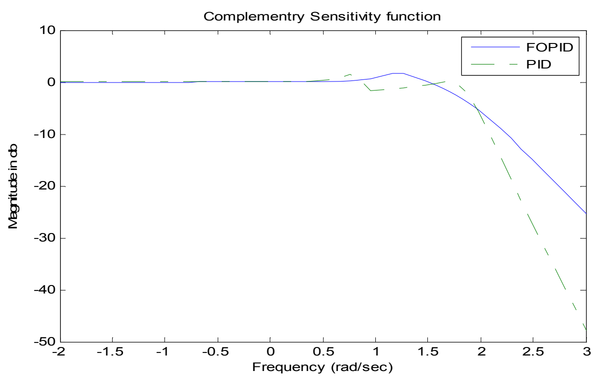

Figure 8. Complimentary Sensitivity function of DC motor with PID and FOPID controllers taking Integral of Square Error as an objective function

As concluded from the Figure 6 that the Rise time for FOPID controller is reduced by 1.39% and Settling time for FOPID is reduced by 76% as compared to PID controller, when Integral of Square error is taken as the objective function. Further comparison is done for FOPID controller, when the parameters are tuned using Genetic Algorithm (GA) and Simulated Annealing (SA) Algorithm. The objective function to be minimized is given by (14). The initial value, lower and upper bound of solution variable 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.

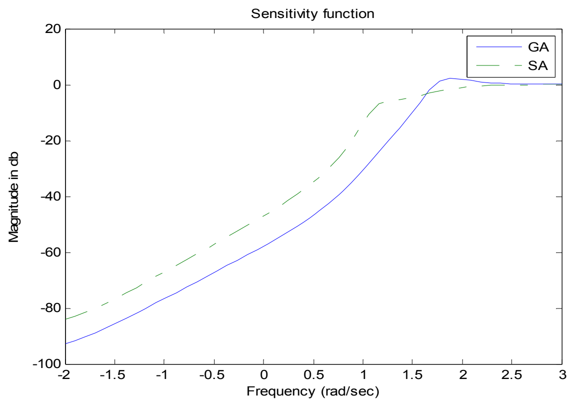

Figure 9. Sensitivity function of DC motor with FOPID controller using GA and SA

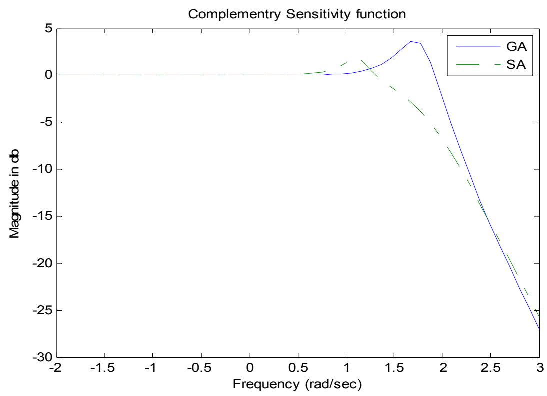

Figure 10. Sensitivity function of DC motor with FOPID controller using GA and SA

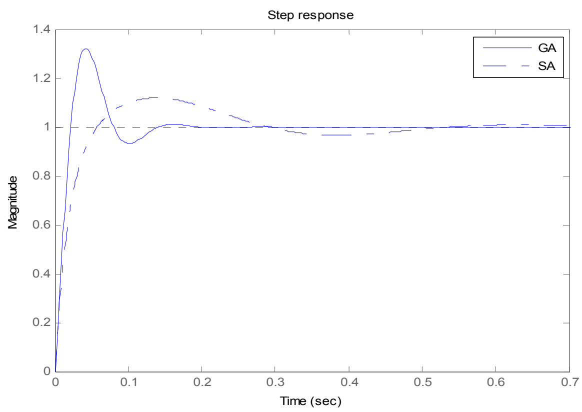

Figure 11. Closed loop step response for FOPID controller using GA and SA

The comparison drawn for FOPID controller using GA and SA as optimization techniques is explained in the Table 5:

Table 5. Comparison of GA and SA

Parameters	Using GA	Using SA
Rise time	0.0166 sec	0.038sec
Settling time	0.129 sec	0.463 sec
Steady state error	0	0
Peak overshoot	32.2%	11.8%

From the above table, it is concluded that the rise time and settling time are being reduced in case of a FOPID controller, when tuned with GA as compared to the FOPID controller when tuned using SA.

7. Conclusion

This paper presents a design method for determining the FOPID controller parameters using the Genetic Algorithm for both Mixed Sensitivity and Integral of Square Error as cost functions. The proposed method involves both Frequency domain and Time domain performance criterion. Application of the methods to DC motor speed control 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 DC motor, for both Frequency domain and Time domain performance criterion. Also, it is concluded that FOPID controller gives better closed loop response, when tuned using GA as compared to FOPID controller when tuned using SA.

References

[1] KJ Astrom, Tore Hagglund, *PID Controller: Theory, Design and Tuning*, Instrument Society of America, 2nd edition, 1995.
[2] Arijit Biswas, Swagatam Das, Ajith Abraham, Sambarta Dasgupta. Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution. *Engineering Applications of Artificial Intelligence*. 2009; 22: 343–350.
[3] Majid Zamani, Masoud Karimi-Ghartemani, Nasser Sadati, Mostafa Parniani. Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Engineering Practice*. 2009; 17: 1380–1387.
[4] Jun-Yi Cao, Bing-Gang Cao. Design of Fractional Order Controller Based on Particle Swarm Optimization. *International Journal of Control, Automation, and Systems*. 2006; 4(6): 775-781.
[5] Ying Luo, Yang Quan Chen, Hyo-Sung Ahn, You Guo Pi. Fractional order robust control for cogging effect compensation in PMSM position servo systems: Stability analysis and experiments. *Control Engineering Practice*. 2010; 18: 1022–1036.
[6] Tahmine V Moghaddam, N Bigdeli, K Afshar. Tuning a Fractional Order PID Controller with Lead Compensator in Frequency Domain. *World Academy of Science, Engineering and Technology*. 2011; 75.
[7] DE Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Asia: Pearson Education, 2001.
[8] HG Beyer, K Deb. On self-adaptive features in real-parameter evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*. 2001; 5(3): 250-270.
[9] K Deb, A Anand, D Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation Journal*. 2002; 10(4): 371-395.
[10] Ukrit Chaiya, Somyot Kaitwanidvilai. Fixed structure robust DC motor speed control. *Proceeding of the international multi conference of Engineering and Computer Scientist*. 2009; II: 18-20.
[11] Jing Ping Jiang, Shen Chen, Pardip K Sinha. Optimal feedback control of Direct- Current motors. *IEEE transaction on Industrial Electronics*. 1990; 31(4): 269-274.
[12] Duncan McFarlane, Keith Glover. A loop shaping design procedure using H_∞ synthesis. *IEEE transaction on Automatic Control*. 1992; 37: 759-767.
[13] MD Minkova, D Minkov, JL Rodgerson, RG Harley. Adaptive neural speed controller of a DC motor. *Department of Electrical Engineering, University of Natal, Durban*. 1998; 47: 123-132.
[14] A El-Samahy. Speed control of DC motor using adaptive variable structure control. *IEEE* 31st annual Power Electronics specialists conference. 2000; 1118- 1123.
[15] P Van den Braembussche, J Swevers, H van Brussel. Design and experimental validation of robust controllers for machine tool drives with linear motor. *Mechatronics*. 2001; 11: 545-562.

[16] Tapabrata Ray. Constrained robust optimal design using a multi objective evolutionary algorithm. *Tamasek Laboratories, National University of Singapore*. 2002; 419-424.
[17] SA Nyabundi, G Qi, Haman, J Munda. DC motor control via high order differential feedback control. *proceeding in Tehwane University of Technology, Nairobi*. 2005; 1-5.
[18] Mehdi Nasir, Hossien Nezamabadi-Pour, Malihe Maghfoori. A PSO based optimum design of PID controller for a linear brushless DC motor. *World Academy of Science, Engineering and Technology*. 2007; 211-215.
[19] Wai Phyo Aung. Analysis on modeling and Simulink of DC motor and its driving system used for wheeled mobile robot. *World academy of Sciences, Engineering and Technology*. 2007; 32: 299-306.
[20] Tae-Hyoung Kim, Ichiron Maruta, Toshiharu Sugie. Robust PID controller tuning based on the constrained Particle Swarm Optimization. *Department of Sciences, Kyoto University*. 2007; 44: 1104-1110.
[21] Lu Renquan, Li Shukui, Xue Lin, Xue Anke. Robust H_∞ optimal speed control of linear DC motor. *Proceeding of the 27th Chinese Control conference*. 2008: 743-747.
[22] Boumediene Allaoua, Abdessalam Abderrahmani, Brahim Gasbaoui, Abdelfatah Nasri. The efficiency of Particle Swarm Optimization applied on Fuzzy Logic DC motor speed control. *Serbian journal of Electrical Engineering*. 2008; 5(2): 247-262.
[23] Boumedience Allaoua, et al. Intelligent controller design for DC motor speed control based on Fuzzy logic Genetic Algorithms optimization. *Leonardo Journal of Sciences*. 2008; 90-102.
[24] M Fallahi, S Azadi. Robust control of DC motor using Fuzzy sliding mode control with PID compensator. *Proceeding of the international multi conference of Engineers and Computer Scientist*. 2009; II.
[25] Adel AA El-Gammal, Adel A El-Samahy. A modified design of PID controller for DC motor drives using Particle Swarm Optimization. *Energy research centre, University of Trinidad and Tobag*. 2009; 419-424.
[26] G Madhusudhana Rao, BV Sanker Ram. A neural network based speed control of DC motor. *International journal of Recent Trends in Engineering*. 2009; 2(6): 121-124.
[27] Valerio D, Sa Da Costa J. Tuning of fractional pid controllers with Ziegler–Nichols type rules. *Signal Processing Journal*. 2006; 86(10): 2771–2784.
[28] Oldham KB, Spanier J. *The fractional calculus*, New York: Academia, 1974.
[29] Miller KS, Ross B. *An introduction to the fractional calculus and fractional differential equations*, NewYork: Wiley, 1993.
[30] Samko SG, Kilbas AA, Marichev OI. *Fractional integrals and derivatives*, Yverdon: Gordon and Breach, 1993.
[31] DE Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Asia: Pearson Education, 2001.
[32] HG Beyer, K Deb. On self-adaptive features in real-parameter evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*. 2001; 5(3): 250-270.
[33] K Deb, A Anand, D Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation Journal*. 2002; 10(4): 371-395.
[34] Zhang Xin, Dang Jianwu, Liu Min. Parameter Optimization of PID Controller Based on PSO for ulti-leaf Collimator. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(10): 6127-6133.
[35] Wang Yeqin. Direct Drive Electro-hydraulic Servo Control System Design with Self-Tuning Fuzzy PID Controller. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(6): 3374-3382.
[36] S Padhee. A Gautam, Y Singh, G Kaur. A novel evolutionary tuning method for fractional order PID controller, *Int. J. Soft Computing & Engg*. 2011; 1(3).
[37] R Singhal, S Padhee, G Kaur. Design of fractional order PID controller for speed control of a DC motor, *Int. J. Scientific & Research Publications*. 2012; 2(6).