# Fractional Order PID Controller Design for Speed Control of Chopper Fed DC Motor Drive Using Artificial Bee Colony Algorithm

Anguluri Rajasekhar[1], Shantanu Das[2], and Ajith Abraham[3,4]

[1]Department of Electrical Engineering, National Institute of Technology-Warangal, 506004, India
[2]Reactor Control Division, Bhabha Atomic Research Center, Mumbai-400085, India
[3]IT4Innovations, VSB-Technical University of Ostrava, Ostrava, Czech Republic
[4]Machine Intelligence Research Labs (MIR Labs), WA, USA
E-mails: rajasekhar.anguluri@ieee.org, shantanu@barc.gov.in, ajith.abraham@ieee.org

***Abstract*— This article deals with an interesting application of Fractional Order (FO) Proportional Integral Derivative (PID) Controller for speed regulation in a DC Motor Drive. The design of five interdependent Fractional Order controller parameters has been formulated as an optimization problem based on minimization of set point error and controller output. The task of optimization was carried out using Artificial Bee Colony (ABC) algorithm. A comparative study has also been made to highlight the advantage of using a Fractional order PID controller over conventional PID control scheme for speed regulation of application considered. Extensive simulation results are provided to validate the effectiveness of the proposed approach.**

***Keywords—DC Drive; PID; FOPID; ABC; control; stability.***

## I. INTRODUCTION

Although extensive research has been done in designing high performance motor drives, industrial applications are demanding more robust and higher performance drives. To match the criteria of industrial applications, a high performance drive system should maintain dynamic speed command tracking and load regulating response. Among various motor available in the market Direct Current (DC) motor provide excellent control of speed for acceleration and deceleration. The main advantage of using DC motor in drive application is that, power supply is directly fed to field of motor which allows for a precision in voltage control, and which in turns finds useful in speed and torque control applications. These motors are also capable of providing starting and torques for loads up to 400% than rated [1, 2].

Due to their simplicity, ease of implementation, reliability and low cost, DC motor drives are widely used in industrial applications. They cover wide range of applications including electric traction, golf carts, quarry and mining applications etc. DC motor can be considered as Single Input and Single Output (SISO) system having speed-torque characteristics well-suited with most mechanical loads. This property makes DC motors controllable over wide range of speed by providing good adjustment schemes to terminal voltage. These exemplary features of DC motors made them a good choice for advanced control algorithm and also speed control concept of these motors can be extendable to other types of motor as well [1].

In this application, we considered an armature voltage controlled scheme. Out of various closed loop controller designs available till date, Proportional Integral Derivative (PID) based control scheme is widely preferred in many industrial applications because of their simple structure and ease in realization. Further, PID based speed control scheme has many advantages like less settling time, fast control and low cost [3]. Recent studies revealed a new extension to PID controller with the help of integrations and differentiations based on Fractional Calculus [14] and it is termed as fractional order (FO) PID controller or $PID^{\alpha}$ [4]. FOPID controllers are based on the concept of fractional order derivatives (integrals) and recent literature has shown an exponential growth of their applications and also these controllers are shown to be more robust and can outperform normal PID controllers if they can be designed effectively [5,6].

In a FOPID controller, apart from the proportional ($K_P$), Integral ($K_I$) and derivative ($K_D$) constants, there are two more constants i.e, order of derivative ($\mu$) and order of integral ($\lambda$). Hence, designing an optimum FOPID controller requires fine tuning of parametric gains $\{K_P, K_I, K_D, \lambda, \mu\}$, which in return calls for real parameter optimization in five-dimensional hyperspace. To carry out this optimization task, we chose a recently evolved swarm intelligent based Artificial Bee Colony (ABC) Algorithm. Since its inception, ABC has shown remarkable performances on wide variety of optimization problems and a comprehensive view of applications of ABC can be found in [7]. The main advantage of ABC over other swarm intelligent methods is its ease in implementation, followed by a well-organized exploitation and exploration phases. These characteristics enabled ABC to be a superior contender among various evolutionary or swarm algorithms. In our current research, ABC has been chosen as optimization algorithm for finding the optimal parametric gains of FOPID controller. The design method focuses on minimization of time domain based objective function. In parallel we also designed optimal PID controller and analysis was made for both PID and FOPID controllers in terms of time domain indices and also via frequency domain stability.

The rest of paper is organized in the following way. Section 2 deals with mathematical modeling of DC motor drive followed by fractional order controller design in Section 3.

Rudiments of Artificial Bee Colony algorithm are presented in Section 4. A detailed explanation of experimental results and their analysis has been made in Section 5. Conclusions and future work details are presented in Section 6.

## II. MATHEMATICAL MODEL OF DC MOTOR DRIVE SYSTEM

In this paper, an armature voltage controlled DC motor has been considered. The basic idea of this type of DC motor speed control is that the output speed can be altered by controlling armature voltage for speed below and up to rated speed (under constant field current). To have good speed regulation characteristics, closed loop speed control is preferred.

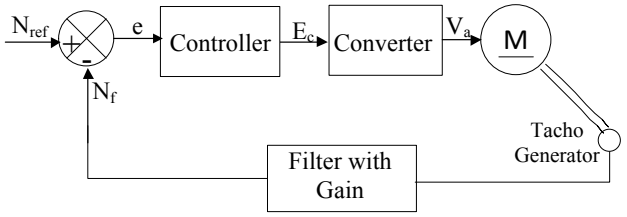


**Figure 1.** Block Diagram of Closed Loop Speed Control of DC Motor Drive

The basic block diagram for DC motor drive speed control is shown in Figure 1. The resultant of reference speed and feedback i.e., error signal is fed as input to speed controller. The output of controller, i.e., control voltage $E_c$ controls the operation of duty cycle of converter. As a result converter provides required $V_a$ to bring motor back to the desired speed. The output speed of motor is measured with the help of Tacho-Generator. Since the Tacho voltage obtained will not be in perfect DC form (include few ripples) a filter with a gain is provided in the feedback path of speed control loop [8].

### *A. Modeling of Separately Excited DC Motor*

A separately excited DC Motor mainly consists of field winding and armature winding with an independent supply. Field windings are used to excite the flux [2, 8]. A separately excited DC motor is excited by a field current $I_f$ and as a consequence an armature current $I_a$ flows in the circuit. As a result motor develops a back EMF and a torque to balance the load torque at a particular speed level.

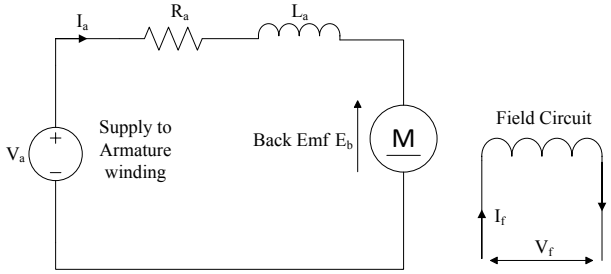


**Figure 2.** Equivalent Circuit of Separately Excited DC Motor

Applying Kirchhoff's Voltage Law (KVL) to the circuit equivalent in Figure 2 will lead to armature Eqn (1) and the equivalent Torque is given by Eqn (2)

$$V_a = I_a R_a + L_a \frac{dI_a}{dt} + E_b \tag{1}$$

$$T_d = J\frac{d\omega}{dt} + B\omega + T_L \tag{2}$$

where $V_a$= armature voltage (Volts); $E_b$= Motor back Emf (Volts); $I_a$= armature current (Amps); $R_a$= armature resistance (Ω); $L_a$= armature inductance (H); $T_L$= load torque (N-m); $T_d$= developed torque ($T_d$); J = Moment of Inertia (Kg/m$^2$); B = friction coefficient of motor; ω= angular velocity (rad/sec).

Assuming negligible friction in motor (B=0) Eqn (2) will be reduced to Eqn (3). Further denoting Φ as field flux and K as Back Emf constant, corresponding equations of Back Emf and torque developed can be obtained.

$$T_d = J\frac{d\omega}{dt} + T_L \tag{3}$$

$$E_b = K\Phi\omega \tag{4}$$

$$T_d = K\Phi I_a \tag{5}$$

With the help of above equations and by applying Laplace Transform to Eqn (1) the following equations are obtained.

$$I_a(s) = \frac{V_a - E_b}{R_a + L_a s} = \frac{V_a - K\Phi\omega}{R_a\left(1 + L_a / R_a s\right)} \tag{6}$$

$$\omega(s) = \frac{T_d - T_L}{Js} = \frac{K\Phi I_a - T_L}{Js} \tag{7}$$

where armature Time constant $T_a = L_a/R_a$. The equivalent model of DC motor is shown in Fig 3[+]. After performing block reduction the resultant transfer function will be of following form

$$\frac{\omega(s)}{V_a(s)} = \frac{\dfrac{K\Phi / R_a}{Js(1+sT_a)}}{1 + \dfrac{K^2\Phi^2}{Js(1+sT_a)}} = \frac{1/K\Phi}{sT_m(1+sT_a)+1} \tag{8}$$

Assuming $T_m = JR_a/(K\phi)^2$ as Electromechanical Time constant. Eqn (8) can be further reduced (replacing KΦ by $K_m$ and also $T_L$=0)

$$\frac{\omega(s)}{V_a(s)} = \frac{1/K_m}{(1+sT_m)(1+sT_a)} \tag{9}$$

Here $T_m$ and $T_a$ are the Electromechanical and electrical time constants of the above system transfer function (T/F), which are accountable for the response of system.

### *B. Current Control Loop*

Due to Electromechanical time constant motor will consume some to speed up [8]. On the other hand speed controller used will be acting very fast. Initially speed feedback is zero, and this results in maximum converter voltage $V_a$. Eventually a large amount of current flow because of zero back EMF. This in course of time may exceed the motor maximum current limit and can damage the motor windings. Hence there is a requirement to control current in motor armature. This problem can be eliminated if closed loop current control scheme can be implemented, in which current controller will take care of motor rated current limit.

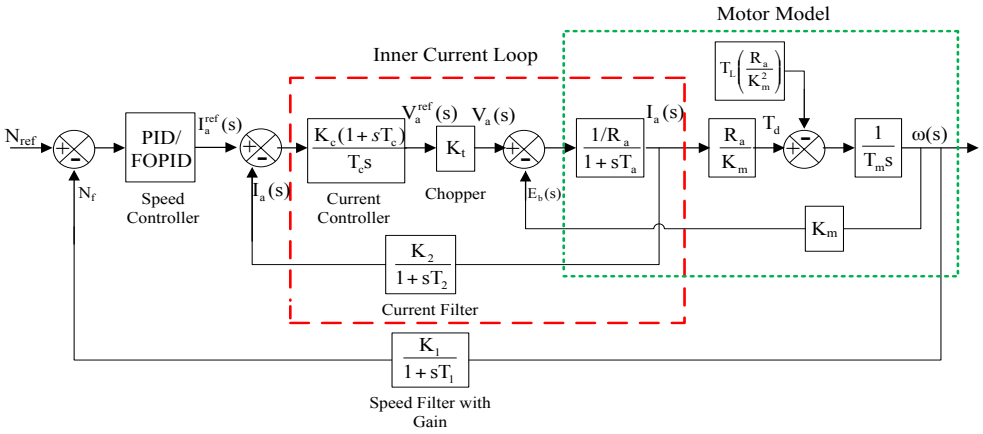


**Figure 3**. A Complete block diagram representation of Speed Control in a DC Motor Drive

In this approach a chopper is considered as a converter. It is a static power electronic component, which converts fixed DC input voltage to a variable DC output. Choppers are more efficient as they involve one stage conversion. Chopper works on the principle of pulse width modulation and it involves no time delay in its operation. Hence, it can be represented by a simple constant gain $K_t$. Detailed explanation of DC-Choppers can be obtained through [9].

As there will be more amount of current flow during starting of motor, design of current controller for extreme condition i.e., for zero back EMF would be optimum. The equivalent model of Current Control Loop has been depicted in Figure 3. The parameter $T_c$ of Current Controller in Figure 3 has to be chosen such that largest time constant in transfer function (T/F) should be canceled. This results in faster response. Hence by assuming $T_c=T_a$, the equivalent T/F of current controller loop can reduced to following form.

$$\frac{I_a(s)}{I_a^{ref}(s)} = \frac{K_0(1+sT_2)}{s(1+sT_2)+K_0K_2} \qquad (10)$$

Where $K_0=(K_cK_t)/(R_aT_a)$; $K_c$= Current controller gain constant; $K_2$= current loop filter lag; $T_2$= current loop filter lag; As the zero in obtained T/F may results in overshoot, care should be taken to cancel its effect. As the current loop time constant is much higher than filter time constant and following some assumptions [8] Eqn (10) can be further reduced to

$$\frac{I_a(s)}{I_a^{ref}(s)} = \frac{1/K_2}{1+s2T_2} \qquad (11)$$

### C. Speed Control Loop

Figure 3 presents a complete layout of closed loop speed control of DC Motor drive. Based on the drive specifications, controller has to be designed to achieve required speed with in specified constraints. As good control scheme involves analysis of time domain parameters and frequency domain stability, with help of above discussions, Figure 3 can be further simplified to Figure 4.

In this approach we considered a FOPID and also PID controller as speed controller for the DC motor drive. A reference signal is given as input and the integral of the error obtained is used in tuning the controllers.

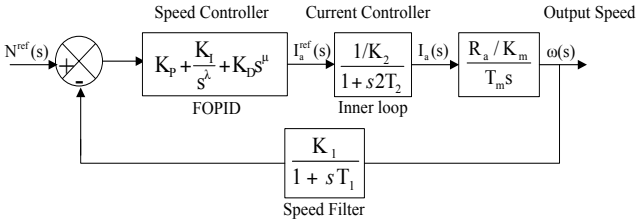


**Figure 4.** Speed Control of DC Motor Drive: A simplified model

## III. Optimal Design of PID/ PI$^{\lambda}$D$^{\mu}$ Controller

### A. Fractional Order Controller: A brief

The idea of a FOPID or PI$^{\lambda}$D$^{\mu}$ controller derives its origin from the concept of fractional order differentiation and integration [10]. Though popular definitions of fractional derivative like Grunwald-Letnikov and Riemann Loville definitions are prevalent, in terms of fractional order systems Caputo definition is widely preferred [5]. This definition of fractional derivative is used to derive fractional order transfer function models from fractional order differential equations with zero initial conditions. According to Caputo's definition the $\alpha^{th}$ order derivative of a function *f(t)* with respect to time is given by following equation.

$$D^{\alpha}f(t) = \frac{1}{\Gamma(m-\alpha)}\int_0^t \frac{D^m f(t)}{(t-\tau)^{\alpha+1-m}}d\tau$$

$$\alpha \in \mathbb{R}^+,\ m \in \mathbb{Z}^+,\ m-1 \le \alpha < m, \qquad (12)$$

Laplace transformation of Caputo's derivative results in "s" domain representation of Eqn (12) and is provided in Eqn (13)

$$\int_0^{\infty} e^{-st}D^{\alpha}f(t)dt = s^{\alpha}F(s) - \sum_{k=0}^{m-1} s^{\alpha-k-1}D^k f(0) \qquad (13)$$

where $\Gamma(\alpha) = \int_0^t e^{-t}t^{\alpha-1}\,dt$ is the Gamma function

$F(s) = \int_0^{\infty} e^{-st}f(t)dt$ is Laplace transform of $f(t)$

With an assumption of zero initial conditions the time domain operator $D^{\alpha}$ can be simply represented in frequency domain as $s^{\alpha}$. A negative sign in the order of derivative ($-\alpha$)

indicates a fractional integral operation. Hence the FOPID controller is a sum of fractional operators along with controller gains. The transfer function representation of a FOPID controller is given in Eqn (14)

$$C(s) = K_P + \frac{K_I}{s^{\lambda}} + K_D s^{\mu} \tag{14}$$

This typical controller consists of three controller gains $\{K_P, K_I, K_D\}$ and two more fractional order operators $\{\lambda,\mu\}$. For Instance, if $\lambda$=1 and $\mu$=1 Eqn (14) reduces to classical controller in parallel structure. In order to implement a controller of form Eqn(14) Oustaloup's band limited frequency domain rational approximation technique is used in the present paper and also in most of FO control literatures [11].

### *B. Digital Realization of Fractional Orders*

The rationale behind the choice of frequency domain rational approximation of FOPID controller is that it can be easily implemented in real hardware using higher order analog or digital filters, corresponding to each fractional order differentiation or integration in FOPID controller. The infinite dimensional nature of fractional order differentiator and integrator in FOPID controller structure creates hardware implementation issues in industrial application of FOPID controllers. However, recent research results demonstrated that band-limited implementation of FOPID controllers using higher order rational transfer function approximation of the integro-differential operators give satisfactory performance in industrial applications [12]. Oustaloup's recursive approximation, which has been implemented to realize fractional integro-differential operators in frequency domain, is given by the following equations.

$$s^{\alpha} = K \prod_{k=-N}^{N} \frac{s+\omega_k^{'}}{s+\omega_k} \tag{15}$$

Here the poles, zeros and gain of the filter can be recursively evaluated as:

$$\omega_k = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{k+n+\frac{1}{2}(1+\alpha)}{2N+1}} \tag{16}$$

$$\omega_k^{'} = \omega_b \left(\frac{\omega_h}{\omega_b}\right)^{\frac{k+n+\frac{1}{2}(1-\alpha)}{2N+1}} \tag{17}$$

where $K = \omega_h^q$. In above equation set $\alpha$ is the order of the differ-integration, *(2N+1)* is the order of the filter and $(\omega_b,\omega_h)$is the expected fitting range. In the current study, 5th order Oustaloup's recursive approximation is done for the integro-differential operators within a frequency band of the constant phase elements (CPEs) as $\omega \in \{10^{-2},10^{2}\}$ rad/sec.

### *C. Problem Formulation*

PID/PI$^{\lambda}$D$^{\mu}$ controller parameters are tuned in optimal fashion such that drive gives optimal performance. For tuning of controllers, we considered two objective functions i.e., Integral Time Squared Error (ITSE) criterion and weighted sum of ITSE & Integral Squared Controller output (ISCO) criterion. The optimal parameters of PID/PI$^{\lambda}$D$^{\mu}$ controller are obtained by minimizing these objective functions via an optimization algorithm. Equations (18-19) represents the mathematical formulation of these objective functions.

$$J_1 = ITSE = \int_0^{\infty} t \cdot e^2(t)\,dt \tag{18}$$

$$J_2 = ITSE + ISCO = \int_0^{\infty} \left[ w_1 \cdot t \cdot e^2(t) + w_2 \cdot u^2(t) \right] dt \tag{19}$$

$J_1$ refers to ITSE which tries to minimize the overshoot & settling time [5]. The higher powers in time and error penalizes the output more at later stages and results in very fast rise and settling time. But for a sudden change in set-point this kind of criteria gives very high value of controller output, resulting in actuator saturation and integral wind up [5]. To overcome this ITSE is enriched with ISCO term ($J_2$), which takes the care of aforementioned problem. The weights $\{w_1,w_2\}$ balances the impact between control error and control action and both have been chosen to be same for present study (to have same penalty).

## IV. ARTIFICIAL BEE COLONY ALGORITHM

Artificial Bee Colony (ABC) Algorithm is a stochastic based optimization algorithm, inspired by the foraging behavior of honey bees. ABC was first proposed by Karaboga [13] for optimization of multivariable and multi-modal continuous functions. Similar to rest of swarm intelligent algorithms ABC algorithm consists of two phases i.e., exploitation phase taken care by *employed & onlooker bee* and exploration phase taken care by *scout bee* [7, 13]. In ABC algorithm, each solution corresponding to the problem is denoted as food source and is represented by a *D*-dimensional real-valued vector; on other hand fitness of solution corresponds to the nectar amount of associated food source. As that of rest of stochastic optimization process ABC follows proceeds iteratively. The algorithm begins by initializing all employed bees with randomly generated food sources (solutions). The position of $i^{th}$ food source that corresponds to a solution in D-dimensional hyper space can be represented as $X_i = \left[x_{i1}, x_{i2}, .... x_{iD}\right]$ and it can be generated by following equation.

$$x_{ij} = lb_j + rand(0,1) \times (ub_j - lb_j) \tag{20}$$

Here, *i=1,2,3…,FS; j=1,2,3…,D; FS* is the number of food sources (equivalent to half to total number of bees) and *D* is the number of variables to be optimized; *rand* is a random number in the range (0, 1); $ub_j$ and $lb_j$ corresponds to upper and lower bounds of the $j^{th}$ dimension respectively. Initially, an employed bee tries to exploit in the vicinity of random food source associated to it and updates its step based on Eqn (21)

$$x_{new} = x_{ij} + R \times (x_{ij} - x_{kj}) \tag{21}$$

where $k$ is a randomly chosen index and $k \in (1,2,...,FS)$ such that $k \notin i$. $R$ is a uniformly distributed random number in the range of [-1, 1]. As soon as $x_{new}$ has been generated, a greedy mechanism is applied between $x_{new}$ and its corresponding previous entity $x_{ij}$ via fitness value. If the obtained new fitness value is better than the fitness value

achieved so far, then the bee moves to this new food source discarding the old source. Once the algorithm completes employed bee phase, bees share their information about food sources to *onlooker* bees. An onlooker bee selects a particular food source $X_i$ based on the probability $P_i$ defined as

$$P_i = fit_i \Big/ \sum_{k=1}^{FS} fit_k \qquad (22)$$

$fit_i$ corresponds to fitness value of $i^{th}$ food source and as the chosen problem is a minimization problem fitness is calculated according to following equation.

$$fit_i = \frac{1}{(1+J(X))} \qquad (23)$$

In this context $J(X)$ represents either $J_1$ or $J_2$ objective functions. Based on above probability relation with respect to food source profitability onlooker tries to exploit a food source making use of Eqn (21) and a greedy mechanism similar to employed bee phase is performed. The above two phases i.e., employed bee and onlooker bee phases are performed in round robin fashion. In the due course of iterative process, it may happen that a food source cannot be improved after $N$ number of trials and this ultimately leads to delay in optimization process or leads to poor convergence. To eliminate this, an exploration scheme has been incorporated via scout bee. Each bee will search for a better food source for a certain number of cycles (*limit*), and if the fitness value doesn't improve then that particular bee becomes a Scout. Food source corresponding to that scout bee is abandoned and is initialized to random food source. This process is continued till the termination criterion is reached.

## V. EXPERIMENTAL STUDIES

**Table 1. DC Motor Drive Specifications [8]**

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Power | 300 KW | Base speed | 500 rpm |
| Max Rated Voltage ($V_a$) | 460 V | Max Rated Current (Ia) | 690 Amps |
| Inductance ($L_a$) | 0.7026 mH | Current Limit | 1200 Amps |
| Moment of Inertia (J) | 84 Kg-m$^2$ | Back EMF Constant ($K_m$) | 8.5 volt-sec/rad |
| Armature Resistance ($R_a$) | 0.02342 Ω | Current Feedback ($T_2$) | 3.5 milli-sec |
| Tachometer Constant ($T_1$) =25 mili-sec | | | |

**Calculation of other Constants**

1. $T_m = J*Ra/(K_m)^2 = 27.55$ milli-sec
2. $T_c = T_a = La/Ra = 30$ milli-sec
3. $K_t$= Max Rated Voltage/10 = 46
4. $K_2$= 10/Current Limit = 0.0083
5. $K_c = T_a*R_a/(2*K_2*K_t*T_2) = 0.2618$

**Algorithmic Parameters**

1. No of Bees ($N_B$):= 20;
2. No of Food Sources (FS)= $N_B/2$;
3. *Limit* (scout trails) = 20;
4. No of iterations:= 100;
5. Run time = 25 independent runs
6. $0 \le K_P, K_I, K_D \le 5;\ 0 \le \lambda, \mu \le 1;$

*A. ITSE ($J_1$) based design of PID/PI$^\lambda$D$^\mu$ Controller*

$G_{PID}(s) = 2 + \frac{0.033}{s} + 1.9312s$ ;

$G_{FOPID}(s) = 2 + \frac{1.9762}{s^{0.1207}} + 1.9139s^{0.4837}$ ;

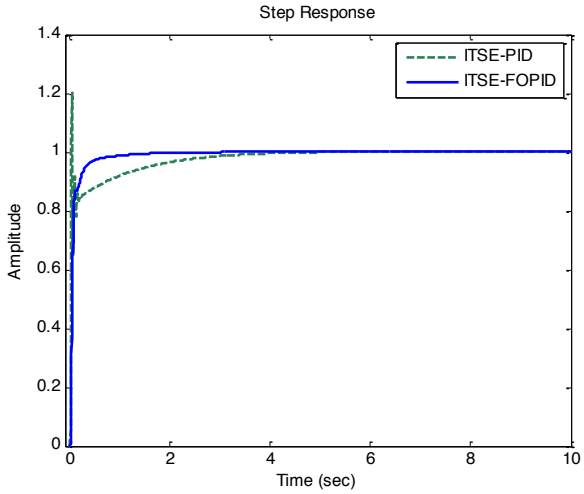


**Figure 5a.** Step response of ITSE tuned PID/FOPID Controller

| Parameters ($J_1$) | PID Controller | FO-PID controller |
|---|---|---|
| Rise Time | **0.015477 sec** | 0.177703 sec |
| Settling Time | 2.2178095 sec | **0.657447 sec** |
| Overshoot | 19.94017 % | **0 %** |
| $J_1$ mean (std) | 0.000411757 (1.18466e-011) | **5.58664e-005 (4.49174e-013)** |
| Gain Margin | -Inf dB | **26.5 dB** |
| Phase Margin | 66.1 deg | **85 deg** |

**Table 2.** Summary of open/close loop response for ITSE tuned Controllers

Figure 5a shows the performance of PID and FOPID controller for a DC motor drive. Fig 5a clearly depicts that the *PI$^\lambda$D$^\mu$* controller outperformed PID controller in terms of overshoot (%) and also in settling time. From Fig 5b it was evident that the margins of FOPID tuned system are better than PID tuned system. Due to the presence of fractional elements, the frequency response maintained flat phases [14] which results in iso-damping nature of the system.

Now to further analyze the performance of optimally designed controllers we increased DC forward path gain up to 60 % and observed the corresponding time responses and frequency responses. From Figures 5c-5f and Table 4 it is clear that FOPID controller provided optimum performance in terms of Overshoot and settling time. Only rise time remained to be good for PID controller and rest of transient response is very poor. FOPID controller continued to show robustness for wide range of gains. From Table 2 and Figure 5g it is further evident that objective value of $J_1$ is less for ITSE-FOPID.

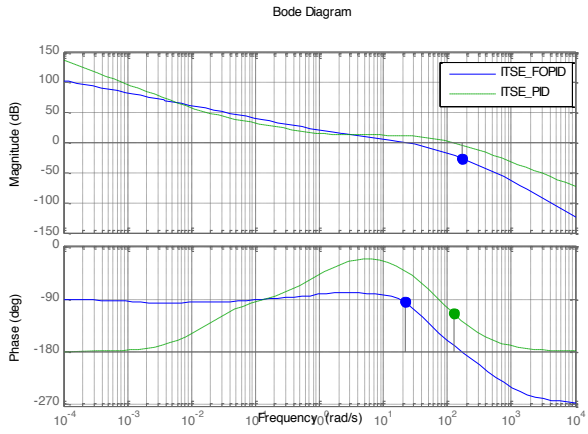


**Figure 5b.** Bode plot of ITSE tuned PID/FOPID Controller

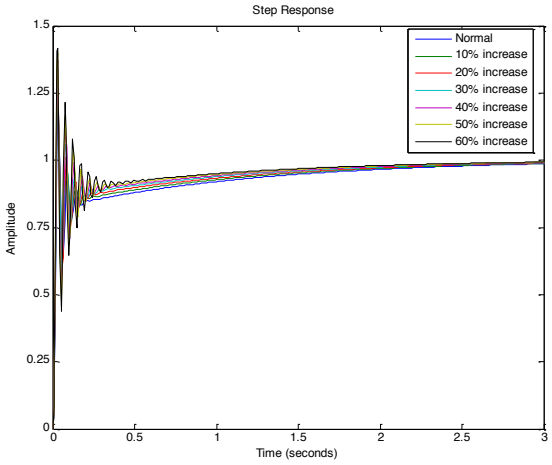


**Figure 5c.** Step responses of ITSE-PID up to 60% increase in gain

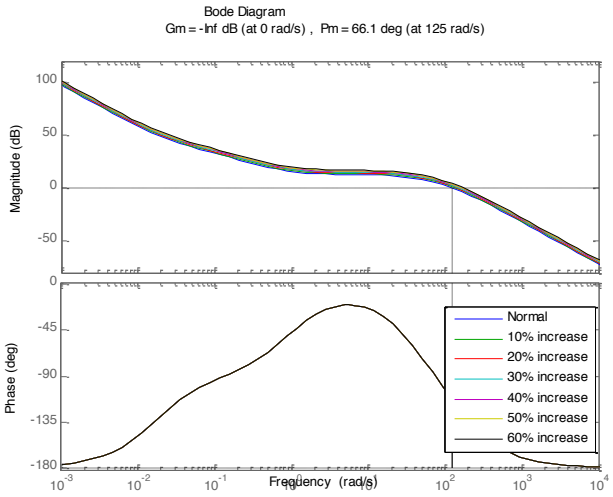


**Figure 5d.** Bode Plot for ITSE-PID up to 60% increase in gain

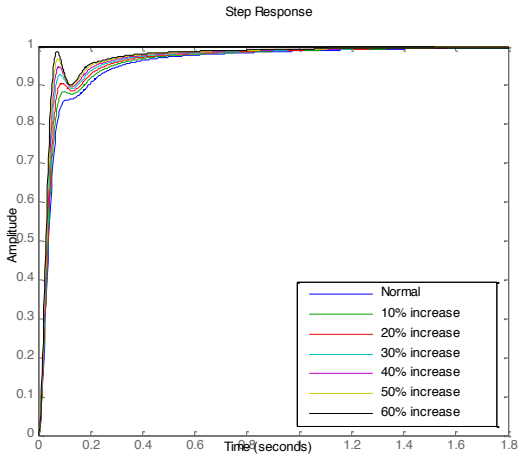


**Figure 5e.** Step responses of ITSE-FOPID up to 60% increase in gain

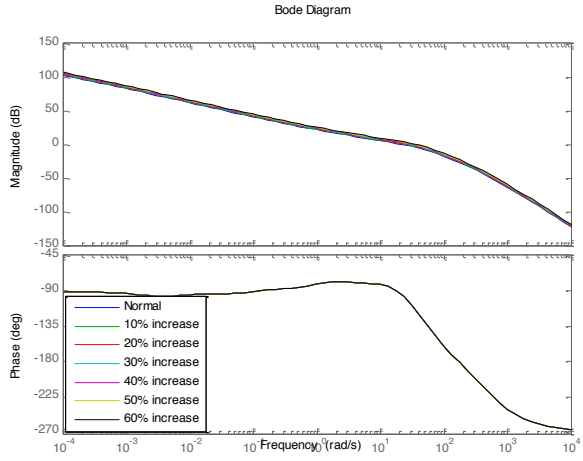


**Figure 5f.** Bode plot of ITSE-FOPID up to 60% increase in gain

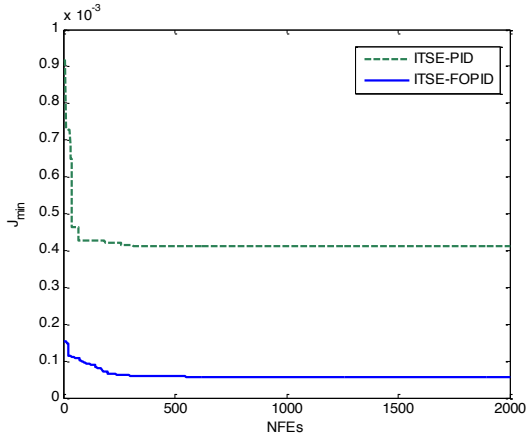


**Figure 5g.** Convergence of ITSE ($J_1$) towards optimum

*B. ITSE + ISCO ($J_1$)* based design of PID/PI$^{\lambda}$D$^{\mu}$ Controller

$$G_{PID}(s) = 0.7689 + \frac{0.0028}{s} + 0.007s;$$

$$G_{FOPID}(s) = 0.3404 + \frac{0.1815}{s^{0.1243}} + 0.2516s^{0.0301};$$

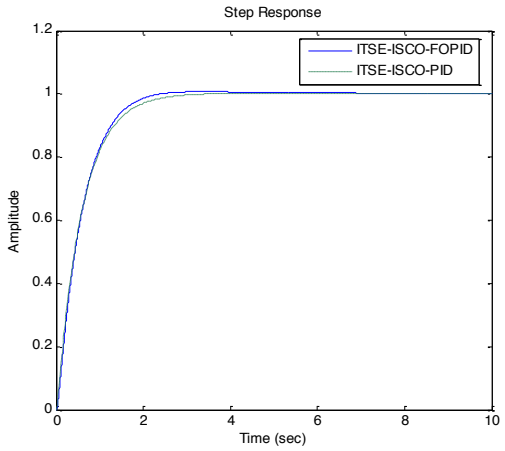


**Figure 6a.** Step response of ITSE+ISCO tuned PID/FOPID Controller

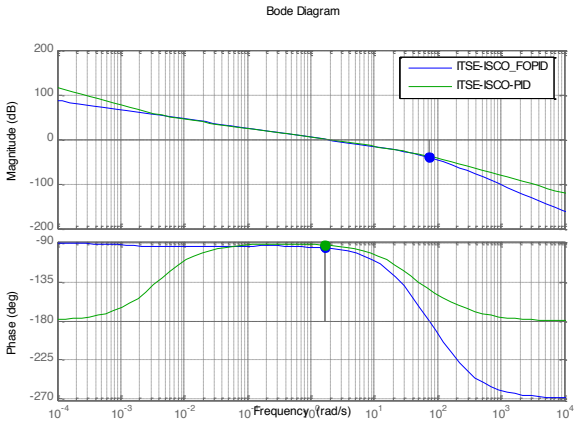


**Figure 6b.** Bode plot for ITSE+ISCO tuned PID/FOPID Controller

This section deals with controllers designed with respect to $J_2$ objective function criterion. Figure 6a reveals that FOPID and PID controller gave similar response and further FOPID controller resulted in good speed tracking. From bode plot (Figure 6b) it is clear that FOPID maintained flat phase, and the margins of FOPID remained to be good in this case also. From Table 3 and Figure 6g it is clear that objective value of $J_2$ is less for FOPID case.

Similar to the former case we also observed the responses for spread in gain. From the Figure 6c, 6e it is evident that dead beat response for PID and FOPID almost remained same. Even for increase in gain both controllers continued to give flat phases (which can be the advantage of this adding ISCO term to $J_1$). Yet there was some superiority of FOPID which can be construed from Table 4. These observations reveal that to get good response characteristics of a plant considered, apart from optimum controller structure, a well posed objective function followed by a robust optimization algorithm should also be carefully chosen.

## VI. CONCLUSIONS

This article presents a novel application of PI$^{\lambda}$D$^{\mu}$ based design of speed controller for DC Motor drive using ABC algorithm as an optimization algorithm for controller tuning. Integral error based objective functions are considered and the results are interpreted in terms of time and frequency domain.

**Table 3:** Summary of open/closed loop responses of $J_2$ tuned Controllers

| Parameters | PID Controller | FO-PID |
|---|---|---|
| Rise Time | 1.257195 sec | **1.168332 sec** |
| Settling Time | 2.276998 sec | **1.882497 sec** |
| Overshoot | **0 %** | **0 %** |
| $J_2$ mean (std) | 0.00905363 (4.32729e-005) | **00898336 (4.33435e-007)** |
| Gain Margin | -Inf dB | **41 dB** |
| Phase Margin | 87.7 deg | **84.4 deg** |

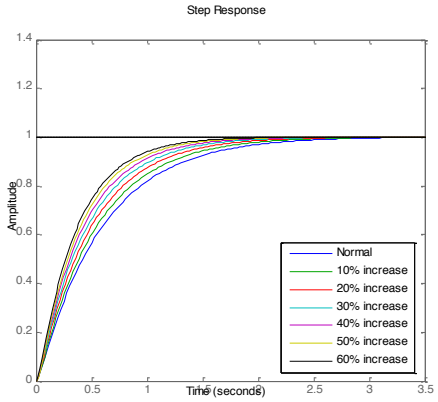


**Figure 6c.** Step responses of ITSE+ISCO-PID up to 60% rise in gain

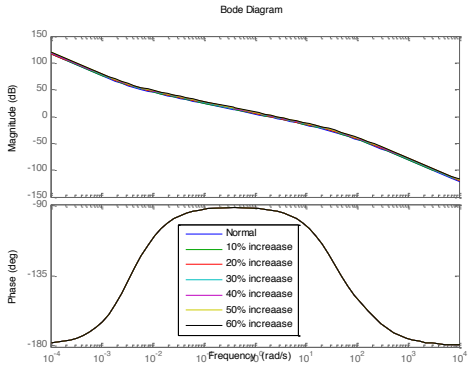


**Figure 6d.** Bode Plot for ITSE+ISCO-PID up to 60% increase in gain

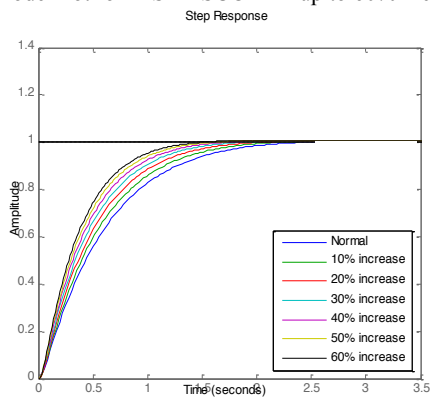


**Figure 6e.** Step responses of ITSE+ISCO-FOPID up to 60% rise in gain

**Table 4:** Summary of Time indices of ITSE Tuned PID/FOPID Controller for increase in DC Motor gain

| Dc Gain Increase ($J_1$) | Rise Time (PID) | Rise Time (FOPID) | Overshoot (%) (PID) | Overshoot (%) (FOPID) | Settling Time (PID) | Settling Time (FOPID) |
|---|---|---|---|---|---|---|
| 10% increase | **0.0152 sec** | 0.1615 sec | 25.52 | **0** | 2.1754 sec | **0.5904 sec** |
| 20% increase | **0.0150 sec** | 0.0671 sec | 30.10 | **0** | 2.1411 sec | **0.5341 sec** |
| 30% increase | **0.0144 sec** | 0.0550 sec | 33.79 | **0** | 2.1049 sec | **0.4871 sec** |
| 40% increase | **0.0089 sec** | 0.0485 sec | 37.02 | **0** | 2.0672 sec | **0.4478 sec** |
| 50% increase | **0.0088 sec** | 0.0440 sec | 39.62 | **0** | 2.0223 sec | **0.4148 sec** |
| 60% increase | **0.0087 sec** | 0.0405 sec | 41.51 | **0** | 1.9775 sec | **0.3869 sec** |

**Table 5:** Summary of Time indices of ITSE+ISCO Tuned PID/FOPID Controller for increase in DC Motor gain

| Dc Gain Increase ($J_2$) | Rise Time (PID) | Rise Time (FOPID) | Overshoot (%) (PID) | Overshoot (%) (FOPID) | Settling Time (PID) | Settling Time (FOPID) |
|---|---|---|---|---|---|---|
| 10% increase | 1.1319 sec | **1.1061 sec** | 0 | 0 | 2.0045 sec | **1.7355 sec** |
| 20% increase | 1.0341 sec | **0.9704 sec** | 0 | 0 | 1.8350 sec | **1.5912 sec** |
| 30% increase | 0.9510 sec | **0.8933 sec** | 0 | 0 | 1.6911 sec | **1.4687 sec** |
| 40% increase | 0.8801 sec | **0.8270 sec** | 0 | 0 | 1.5673 sec | **1.3633 sec** |
| 50% increase | 0.8184 sec | **0.7693 sec** | 0 | 0 | 1.4597 sec | **1.2715 sec** |
| 60% increase | 0.7644 sec | **0.7186 sec** | 0 | 0 | 1.3654 sec | **1.1908 sec** |

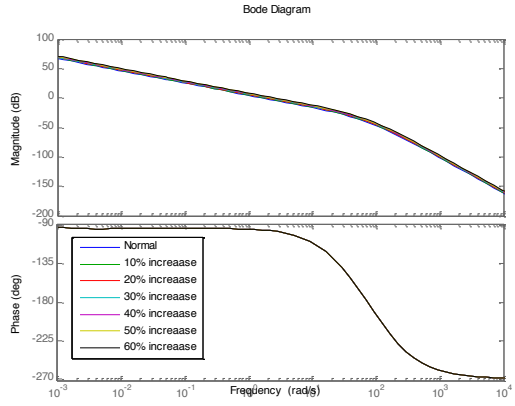


**Figure 6f.** Bode plot of ITSE+ISCO-FOPID up to 60% increase in gain

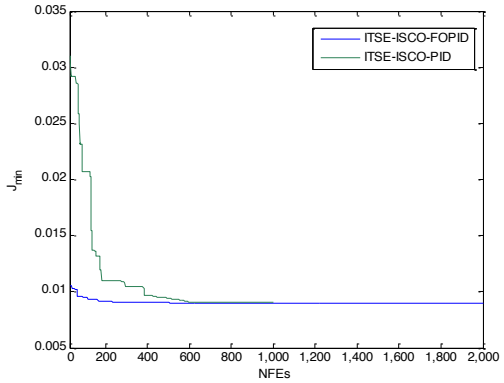


**Figure 6g.** Convergence of ITSE+ISCO ($J_2$)

From the results it is very clear that FOPID based controller outshined PID controller for both the objective functions. Though PID controller gave an satisfactory response in 2nd case, it is observed that its design is highly dependent on objective function and on other hand regard less of type of objective function FOPID gave good responses.

Our future research will include development of FOPID controllers for AC motor drive controls and for multivariate systems. To get even more promising results it would be better to design the controller using multi-objective optimization approach.

## ACKNOWLEDGEMENTS

This research work is supported in the framework of the IT4 Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 by operational programme 'Research and Development for Innovations' funded by the Structural Funds of the European Union and state budget of the Czech Republic, EU.

## REFERENCES

[1] G.K. Dhubey, Fundamentals of Electrical Drives. New Delhi, Narosa Publishing Home, 2009.
[2] V. Subrahmanyam, Electric Drives: Concepts and Applications, New Delhi, Tata McGraw Hill, 2011.
[3] T. Haggalund, and K.J. Astrom, Automatic Tuing of PID Controllers, The Control Handbook, CRC Press, 2010.
[4] I. Podlubny, "Fractional-order systems and PI-D- Controllers, IEEE Trans Autom Control, **44**(1), pp. 208-214, 1999.
[5] I. Pan and S. Das, Intelligent Fractional Order Systems and Control, Studies in Computational Intelligence, Springer, 2013.
[6] C.A. Monje, Y. Chen, B.M. Vinagre, D. Xue and V. Feliu-Battle, Fractional-order Systems and Controls: Fundamentals and Applications, Advances in Inudstrial Control, Springer, 2010.
[7] D. Karaboga, and B. Akay, "A Comprehensive surevy; artificial bee colony algorithm and applications", Artificial Intelligence Review, Springer 2012.
[8] K. Gopakumar, "Power Electronics and Electrical Drives, Video Lectures 1-25", Center for Electronics and Technoloy, IISC, Bangalore.
[9] P.S. Bimbhra, Power Electronics, Khanna Publishers, 2006.
[10] K.B. Oldham and J. Spanier, The Fracitonal Calculus, 1974.
[11] D. Valerio and Sa da Costa, "Introduction to Single-input, Single-ouput fractional control," IET Control Theory Appl **5(8),** pp. 1033-1057, 2011.
[12] M.O. Effe, "Fractional order systems in industrial automation-a survey", IEEE Tran Ind Inform **7**(4), pp. 582-591, 2011.
[13] D. Karaboga, "An Idea based on Honey Bee swarms for Numerical Optimization" Techincal Report-TR06, Eciyes University, Engineering Faculty, Computer Engg Dept.
[14] S. Das, Functional Fractional Calculus, Springer, 2011.