SYSTEMS & CONTROL | RESEARCH ARTICLE

On comparison of tuning method of FOPID controller for controlling field controlled DC servo motor

Prashant Kumar¹, Shantanu Chatterjee*, Devnath Shah¹, U. K. Saha² and Saibal Chatterjee³

Abstract: In this paper, the Simulated Annealing (SA) optimization method is introduced to estimate the parameters of Fractional Order Proportional-Integral-Differential (FOPID) controller for controlling a field-controlled DC servo motor. The parameters are evaluated for considering a transfer function of controlling a field-controlled DC servo motor and its performance is observed for unit step response. Also, the parameters of FOPID controller are calculated using the traditional techniques such as Genetic Algorithm (GA) and frequency domain design method based on constrained optimization for the sake of comparison of performance with respect to simulated annealing method. A detail study on all the specified techniques is discussed in this paper.

Subjects: Automation Control; Control Engineering; Systems & Controls

Keywords: FOPID; fractional calculus; frequency domain design; genetic algorithm; simulated annealing

1. Introduction

In the recent years, modern control theories have made great advances. Control techniques including unified feedback control, optimal control, predictive control, neural network control, fuzzy logic control and so on, have been developed significantly. Nevertheless, the PID control technique has still been widely used in many industrial applications such as process control, motor drives, flight control, etc. More than 90% of the control loops in industry are PID based control loop (Åström, & Hägglund, 1988). This is mainly because of PID controller possesses robust performance to fulfill the

ABOUT THE AUTHOR

Shantanu Chatterjee is currently working as an assistant professor in the Department of Electrical Engineering at NIT, Arunachal Pradesh, India. Before joining NIT Arunachal Pradesh he has worked at CSIR-CMERI, India as Sr. Research Fellow for 1 year. He has more than 4 years of experience in research and teaching field. His research interest includes PID Controller tunings, Active Power Filters, Smart Grid & Anti Islanding, and Power Electronics Applications in Renewable Energy System. He has papers published in Elsevier and different IEEE conferences.

© 2017 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.
global change of industry process, simple structure to be easily understood by engineers, and easiness to design and implement.

Lately, there are increasing interests to enhance the performance of PID controller by using the idea of fractional calculus, which deals with fractional or non-integer order of derivative and integral. Fractional order PID controller proposed by Podlubny is a generalization of the classical PID controller using the fractional calculus (Podlubny, 1999). Its transfer function is described by Equation (1). A Fractional Order Proportional-Integral-Differential (FOPID) controller is characterized by five parameters, i.e. the Proportional gain ($K_p$), the integral gain ($K_i$), the derivative gain ($K_d$), integral order ($\lambda$) and derivative order ($\mu$).

$$G_c(s) = K_p + K_i s^{-\lambda} + K_d s^{\mu}$$

(1)

Many research results show that FOPID controller has better performance and robustness than classical PID controller. So that, the FOPID controller parameter tuning is a critical and major issue (Cominos & Munro, 2002). In the literature, many formulations have been documented to design FOPID controller. These approaches can be classified into two types: analytic methods (Bouafoura & Braiek, 2010; Edet & Katebi, 2016) and heuristic methods (Maione, Punzi, & Li, 2013). In the analytic type, the parameters of FOPID controller are tuned by reducing a nonlinear objective function depending on the specifications given by the designers (Edet & Katebi, 2016).

2. Classical PID vs. fractional order PID controller

Although the classical PID controller works well regarding the fulfillment of the design specification. But with the change of parameters value and aging the output tends to deteriorate and deviate. The classical PID Controller do not have the capability to cope with the changing parameter (Monje, Vinagre, Feliu, & Chen, 2008). With this the output tends to deviate from its nominal value, which force the system towards instability or a region of low efficiency. Moreover the classical PID controllers have only three parameter to tune which limit the tuning strategy. The integral constant and the derivative constant in classical PID controller are related to each other (generally regarded as “$a$” in literature) (Chen, Hu, & Moore, 2003). Therefore the classical PID controller have very little scope for tuning. As a consequent the robustness of a control system cannot be increased with the help of this existing parameter and this limits, the use of classical PID controller in the future prospects.

The limitation of a classical PID controller can be overcome by the introduction of additional tuning parameter that is the order of integrator and differentiator (Vinagre, Podlubny, Dorcak, & Feliu, 2000). Naturally, besides setting proportional gain $K_p$, integral gain $K_i$ and derivative gain $K_d$, we have two more parameter $\lambda$ and $\mu$, which widens the scope of control system design (Li, Yang, Peng, & Wang, 2006). The fractional order controller generalizes the integer order PID controller design and expands it from point to plane (Vu & Lee, 2013). This expansion adds more flexibility to control system design and we can control our real world process more accurately (Maiti, Biswas, & Konar, 2008). This is shown in Figure 1.

With the help of fractional order PID controller it is possible to design a controller to ensure that the closed loop system is robust to gain variation and step response exhibits an iso-damping property (phase-bode plot of the system is flat at the gain crossover frequency) as a result system gets more robustness to gain variation (Chen et al., 2003). The width of flat phase region can be adjusted with the help of fractional order phase shaper (Chen, Moore, Vinagre, & Podlubny, 2004) resulting into a more robust system. $PI^D\mu$ controllers are less sensitive to parameter variation.

As most of the real world process can be efficiently modeled with the help of fractional order differential equation rather than integer ones (Podlubny, 1999). Therefore fractional order controller is naturally suitable for these fractional order models to imply a better control.
3. Fractional calculus and approximation of fractional order transfer function

The concept of FO controller means, controllers can be explained by fractional-order differential equations. Of the various definitions of fractional derivatives, the Grunwald–Letnikov (Miller, 1993) and Riemann–Liouville (Eckert, Kupper, & Hohmann, 2014) definitions are the most widely used. These definitions are required for the realization of discrete control algorithms.

\[
D_t^\alpha f(t) = \begin{cases} 
\frac{1}{\Gamma(-\nu)} \int_a^t (t-x)^{\nu-1} f(x) dx & \nu < 0 \\
\frac{d^n}{dt^n} D_t^{\nu-n} f(t) & a = n \in \mathbb{N} \\
\frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f(x)}{(t-x)^{\alpha-n+1}} dx & 0 \leq n - 1 < \alpha < n
\end{cases}
\]  

(2)

Here, \(\tau(\cdot)\) is the Euler’s gamma function (Davis, 1959) given by Equation (3).

\[
\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt, x > 0
\]  

(3)

With special case when \(x = n\)

\[
\Gamma(n) = (n-1)!
\]

The Laplace transformation of Reimann–Liouville definition (Munkhammar, 2004) is discussed in Equations (5) and (6).

\[
L\{D_t^{-\alpha} f(t)\} = s^{-\alpha} F(s)
\]  

(4)

\[
L\{D_t^\alpha f(t)\} = s^\alpha F(s) - \sum_{k=0}^{n-1} s^k \left[ D_t^{-k-1} f(t) \right]_{t=0}
\]  

(5)

where

\((n-1 \leq \alpha \leq n)\)
These two equations represent the Laplace transformation of fractional order systems. In practice, fractional order differentiators or integrators cannot exactly be achieved or implemented in real-time applications because they represent an infinite dimensional linear filter (Padhee, Gautam, Singh, & Kaur, 2011). A band-limit FOC (Fractional order control) implementation is important in practice i.e. finite-dimensional approximation of FOC in a proper range of frequencies. There are several approximation methods that can directly give the approximate s-transfer function of fractional order differentiator and integrator. The well-known approximation is given by Oustaloup (2000). Oustaloup approximation makes use of a recursive distribution of zeros and poles. The approximating transfer function is given in Equation (7).

\[ s^n \approx k \prod_{n=1}^{N} \frac{1 + (s/\omega_{z,n})}{1 + (s/\omega_{p,n})} \]  

The approximation is valid in the frequency range \([\omega_{z,1}, \omega_{z,N}]\). Gain \(k\) is adjusted so that both sides of Equation (7) shall have unit gain at 1 rad/sec. The number of zeros and poles \(N\) is chosen beforehand. The better performance of approximation strongly depends on \(N\); low values of \(N\) lead to simple approximations and can cause the appearance of ripple in both gain and phase behaviors while highest values of \(N\) will be computationally heavier though such a ripple can be removed. Frequencies of zeros and poles in Equation (7) are given as

\[ \omega_{z,1} = \omega_{l} \sqrt{n} \]  
\[ \omega_{p,n} = \omega_{z,n} \varepsilon, \quad n = 1, 2, 3, \ldots, N \]  
\[ \omega_{z,n+1} = \omega_{p,n} \sqrt{n} \quad n = 1, 2, 3, \ldots, N-1 \]  
\[ \varepsilon = (\omega_{p}/\omega_{l}) \]  
\[ \eta = (\omega_{p}/\omega_{l})^{(1-n)/N} \]  

The case \(\alpha < 0\) can be handled by inverting Equation (7). So, the approximation becomes unsatisfactory; for that reason, it is usual to split fractional powers of \(s\) like this \(|\alpha| < 1\)

\[ s^\alpha = s^n s^\delta \]  
where \(\alpha = n + \delta, n \in \mathbb{Z}, \delta \in [0, 1]\).

As a result, only the latter term has to be approximated.

4. Transfer function of field controller DC servo motor

Field controlled DC servo motor arrangement is mainly used in small servo application (Alshehhi, 2014). We have used it as the main system to be controlled. Mathematical model is built in Laplace domain and the transfer function between angular position and field voltage neglecting disturbance torque comes out to be

\[ G(s) = \frac{\theta(s)}{V_f(s)} = \frac{K_m/BR_f}{s(\tau_s s + 1)(\tau_l s + 1)} \]  

where \(\tau_s = L_f/R_f\) and \(\tau_l = J/b\)

where \(L_f\) is field inductance, \(R_f\) is field resistance, \(J\) is motor inertia constant, \(B\) is speed damping coefficient and \(K_m\) is motor torque constant. Applying the parameters given in Appendix A in Equation (13) we get the following equation.

\[ G(s) = \frac{\theta(s)}{V_f(s)} = \frac{1.91}{s^3 + 21s^2 + 20s} \]
Here output is taken as the instantaneous position of motor and input is taken as field voltage. The equation shows that the system has a dynamics of third order. The system is analyzed by closed loop step response after that the characteristic of uncompensated is determined by time and frequency domain specification (Singhal, Padhee, & Kaur, 2012) then, the system is compensated by designing appropriate controller. The controller is tuned by different tuning algorithm and the performance is compared for relative dominance of different tuning strategy.

5. Analysis of uncompensated system
The system is modeled in MATLAB, SISO design tool (Granda, 2002) is used to determine the characteristic of uncompensated system (Hanselman & Littlefield, 1995). The graph of closed loop step response is given in Figure 2.

The step response shows that the response is an over damped response having rise time of 20.7 sec, settling time of 37.6 sec, zero steady state error and no overshoot. This much rise time and settling time made the system very sluggish and slow. Therefore the system needs compensation so that the characteristics should be in tolerable limit.

6. Controller topology and tuning strategies
It is desired to design a controller to meet the following design specification: Rise time less than 1 sec, settling time less than 1.5 sec, zero steady state error and percent overshoot less than 10%. For achieving the desired specification PID and FOPID are tuned by different tuning algorithm. The controller is put in series with the plant with unity feedback. The overall structure of controlled system is given in Figure 3:

The PID controller is first tuned with classical integer order PID controller with the help of genetic algorithm based optimization approach to find the maximum performance that can be obtained by classical PID controller (Kushwah & Patra, 2014). Then we switched to the fractional order PID controller firstly tuned by genetic algorithm based minimization then simulated annealing based minimization (Bandyopadhyay, Saha, Maulik, & Deb, 2008) and finally with the frequency domain design based on constrained minimization in order to obtain the better result and finally compare all the result by generating closed loop step response.

For tuning the controller with the help of genetic algorithm and simulated annealing method, firstly an optimization problem is formulated which requires generating an objective function. The
objective function is formulated in Section 6.1. Tuning of FOPID by frequency domain design based on constrained minimization is dealt in Section 6.2.

### 6.1. Time domain design

Here firstly optimization a problem is formulated and then minimize the objective function using different search technique.

#### 6.1.1. Formulation of objective function by dominant pole placement method of controller design

Our approach is based on the root locus method (dominant pole method) of designing PID controllers. As in the traditional root locus method, the peak overshoot $M_p$ and rise time $t_{rise}$ are specified, then from these specifications damping ratio $\zeta$ and the undamped natural frequency $\omega_n$ are calculated according to Equations (15) and (16).

$$\zeta = -\frac{\ln(M_p)}{\sqrt{\left(\ln(M_p)\right)^2 + \pi^2}}$$  \hspace{1cm} (15)

$$\pi - \tan^{-1}\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)$$

$$\omega_n = \frac{t_{rise}}{\sqrt{1-\zeta^2}}$$  \hspace{1cm} (16)

$$\rho_1, 2 = -\zeta \omega_n \pm j\omega_n \sqrt{1-\zeta^2} = -a \pm jb$$  \hspace{1cm} (17)

where $a = \zeta \omega_n$, $b = \omega_n \sqrt{1-\zeta^2}$.

Using these computed values of $\zeta$ and $\omega_n$, desired positions of the dominant poles $\rho_1, 2$ of the closed loop system are determined as:

Let $G_p(s)$ be the transfer function of the process to be controlled, $G_c(s)$ the transfer function of the controller to be designed and $H(s)$ the transfer function of the feedback path. Then, the closed loop transfer function of the controlled system is given as:

$$T(s) = \frac{G(s)}{1 + G(s)H(s)}$$  \hspace{1cm} (18)

$$G(s) = G_c(s)G_p(s)$$

$G(s)$ being the forward path transfer function. Assuming unity feedback, then $H(s) = 1$. In this case, the characteristic equation becomes $1 + G(s) = 0$,

$$1 + G_c(s)G_p(s) = 0$$  \hspace{1cm} (19)

$$1 + G(s)\frac{P(s)}{Q(s)} = 0$$  \hspace{1cm} (20)

$$Q(s) + G_c(s)P(s) = 0$$  \hspace{1cm} (21)
where \( P(s) \) and \( Q(s) \) are respectively the numerator and denominator polynomials of \( G(s) \), \( P(s) \) and \( Q(s) \) have no common factor. As \( p_1, p_2 \) is poles of the closed loop system, each of them must be a root of the characteristic equation and hence must satisfy Equation (21). Thus, putting \( s = p_1 = -a + jb \) in Equation (21), we get

\[
Q(p_1) + G_c(p_1)P(p_1) = 0 \tag{22}
\]

\[
Q(-a + jb) + (K_p + K_i(-a + jb)^{-1} + K_d(-a + jb)^s)P(-a + jb) = 0 \tag{23}
\]

Equation (23) is a complex equation in five unknowns, namely \( K_p, K_i, K_d, \lambda, \mu \) and our problem of designing a controller that makes the closed loop system dominant poles to be coincide with \( p_1, p_2 \) now reduces to determining the set of values of \([K_p, K_i, K_d, \lambda, \mu]\) for which Equation (23) holds good. But as the number of unknowns exceed the number of equations, there exists an infinite number of solution sets and the equations can not be unambiguously solved by traditional methods. This necessitates the application of random global search techniques, which, in turn, requires the formulation of a suitable objective function.

Let \( R = \text{Real part} \) and \( I = \text{Imaginary part} \) of the L.H.S. of Equation (23) and

\[
P = \tan^{-1} \frac{I}{R} \tag{24}
\]

we define

\[
F(K_p, K_i, K_d, \lambda, \mu) = |R| + |I| + |P| \tag{25}
\]

as our objective function. Clearly, \( f_0 \), in general, and \( f = 0 \) if and only if \( R = 0 \) and \( I = 0 \) and \( P = 0 \), i.e. if and only if Equation (23) is satisfied. So, we now employ different search technique to scour the five-dimensional search space and find the optimal solution set \([K_p, K_i, K_d, \lambda, \mu]\) for which \( f = f_{\text{min}} = 0 \). This method of finding the parameter of PID and FOPID is widely used in control engineering.

6.1.2. Genetic algorithm based tuning of PID controller

Genetic algorithm (Goldberg & Holland, 1988) belongs to the group of optimization methods called as non- traditional optimization methods. GA tries to imitate natural genetics and natural selection. The main philosophy behind GA is survival of the fittest. As a result GA is used primarily for maximization problems in optimization. GA don’t suffer from the basic setback of traditional optimization methods such as getting stuck in local minima. This is because GA works on the principle of natural genetics, which incorporates large number of randomness. There are basically three operators of genetic algorithm, namely reproduction, crossover and mutation. These operators are used till, the genetic algorithm finds the optimal values.

For searching good set of value of \( K_p, K_i \& K_d \) by genetic algorithm firstly an optimization problem is formulated using Equation (25) and then minimize the fitness function by genetic algorithm.

The process we want to control has the transfer function given by Equation (13). Controller is designed such that closed loop system exhibits maximum over shoot of 10% and rise time less than 1 second. From Equations (15) and (16) the value of damping ratio and undamped natural frequency comes out to be 0.35 and 2 rad/sec. The dominant pole for the closed loop system should lie at \( P_1, P_2 = -0.7 + j1.87 \) and \( -0.7 - j1.87 \). From Equation (19), for \( P_1 = -0.7 + j1.87 \), the characteristics equation can be written as:

\[
\left( 1 + \frac{1.91}{s^3 + 21s^2 + 20s} \right) \left( K_p + \frac{K_i}{s} + K_d s \right) = 0 \tag{26}
\]
which is a complex equation, after separating real and imaginary part

\[ R = 1.91K_p - 0.3354K_i - 1.337K_d - 70.1444 \]  \hspace{1cm} (28)

\[ I = 3.5717K_d - 0.9K_i - 21.3683 \]  \hspace{1cm} (29)

\[ P = \tan^{-1}\left(\frac{3.5717K_d - 0.9K_i - 21.3683}{1.91K_p - 0.3354K_i - 1.337K_d - 70.1444}\right) \]  \hspace{1cm} (30)

We formulate the fitness function as \( F = |R| + |I| + |P| \) and use genetic algorithm to search for good set of value of \( K_p, K_i, K_d \). For optimization Matlab optimization toolbox is used, the parameter of genetic algorithm is set to as given in Table 1.

After performing the minimization of objective function in MATLAB the value of function converge to 0.2891 after 70 generation. The value of \( K_p, K_i, K_d \) comes out to be 199.506, 450.55 and 119.513 respectively. The transfer function of PID controller is given as

\[ G_c(s) = \frac{199.506 + 450.55}{s} + 119.513s \]  \hspace{1cm} (31)

Now we plot the time responses for unit step input in Figure 4 for: uncontrolled system, system controlled by PID controller tuned by genetic algorithm based minimization. A step disturbance is also introduced in the system to check the system robustness with respect to disturbance signal.

The graph show significant improvement in closed step response comparing to uncompensated system. From Table 2 it is clear that there is significant improvement in step response in term of rise time, settling time, peak over shoot. But the specification of overshoot less than 10% is still not achieved. As genetic algorithm has a very high tendency towards global optimum, genetic algorithm gives the limit of performance that can be obtained from PID controller. For obtaining better result we need extra parameter so that the desired specification can be met.

From Table 2 it is clear that there is significant improvement in step response in term of rise time, settling time, peak over shoot. But the specification of overshoot less than 10% is still not achieved. As genetic algorithm has a very high tendency towards global optimum, genetic algorithm gives the limit of performance that can be obtained from PID controller. For obtaining better result we need extra parameter so that the desired specification can be met.

For this we switch to the fractional order PID controller which is the generalization of PID controller using the concept of fractional calculus. Podlubny shows the dominance of FOPID over classical PID controller in term of fulfillment of various performance indices as it has two more parameter to tune.

| Table 1. Parameter of GA PID controller |
|----------------------------------------|
| **Parameter**                          | **Value**                          |
| Lower bound \([K_p, K_i, K_d]\)       | \([1 1 1]\)                        |
| Upper bound \([K_p, K_i, K_d]\)       | \([1000 500 500]\)                 |
| Population size                        | 50                                 |
| Crossover                              | Scattered                           |
| Crossover fraction                     | 0.8                                |
| Mutation fraction                      | 0.011                              |
| Stopping criteria (iterations)         | 70                                 |
| Selection                              | Stochastic uniform                  |
6.1.3. Genetic algorithm based tuning of FOPID controller

In fractional order PID controller, apart from the three tuning parameters \((K_p, K_i, \text{ and } K_d)\) there are two additional tuning parameters that is \(\lambda \& \mu\). The range of \(\lambda \) and \(\mu\) can be anything, but in our problem we restrict it in between 0 and 2. Again the controller is designed such that such that closed loop system exhibits maximum over shoot of 10% and rise time less than 1 second. From Equations (15) and (16) the dominant pole for the closed loop system should lie at \(P_1, 2 = -0.7 + j1.87\) and \(-0.7 - j1.87\).

From Equation (19) the characteristic equation is given as-

\[
1 + \left(\frac{1.91}{s^3 + 21s^2 + 20s}\right)\left(K_p + \frac{K_i}{s} + K_ds^\mu\right) = 0
\]  
(32)

Solving the equation at dominant pole

\[
1 + \frac{1.91}{(-0.7 + j1.87)^3 + 21(-0.7 + j1.87)^2 + 20(-0.7 + j1.87)} \cdot \left(K_p + \frac{K_i}{(-0.7 + j1.87)^\mu}\right) + K_d(-0.7 + j1.87)^\mu = 0
\]  
(33)

This is a complex equation. Separating real and imaginary parts

\[
R = 1.91K_p + 1.91K_d2^{-\mu}\cos(1.93\lambda) + 1.91K_d2^{\mu}\cos(1.93\mu) - 7.01444
\]  
(34)

\[
I = 1.91K_d2^\mu\sin(1.93\mu) - 1.91K_d2^{-\mu}\sin(1.93\lambda) - 21.3683
\]  
(35)

\[
\rho = \tan^{-1}\left(\frac{1.91K_d2^\mu\sin(1.93\mu) - 1.91K_d2^{-\mu}\sin(1.93\lambda) - 21.3683}{1.91K_p + 1.91K_d2^{-\mu}\cos(1.93\lambda) + 1.91K_d2^{\mu}\cos(1.93\mu) - 7.01444}\right)
\]  
(36)
Again the fitness function is given as \( F = |R| + |I| + |P| \) and we use genetic algorithm to search for good set of value of \( K_p, K_i, K_d, \mu \) and \( \lambda \). The parameter of genetic algorithm is set as given in Table 3.

After optimization, the function converge to 0.06028 after 100 generation and the value of \( K_p, K_i, K_d, \mu \) and \( \lambda \) comes out to be 132, 8, 200, 0.835 and 1.3 respectively. The transfer function of FOPID controller is given as

\[
G(s) = 132 + \frac{8}{s^{0.835}} + 200s^{1.3}
\]  

After optimization, the function converge to 0.06028 after 100 generation and the value of \( K_p, K_i, K_d, \mu \) and \( \lambda \) comes out to be 132, 8, 200, 0.835 and 1.3 respectively. The transfer function of FOPID controller is given as

\[
G(s) = 132 + \frac{8}{s^{0.835}} + 200s^{1.3}
\]  

Now we plot the time responses for unit step input in Figure 5 for FOPID controller tuned on the basis of genetic algorithm and compare the result with the classical integer order PID controller. A step disturbance is also introduced in the system to check the system robustness with respect to noise signal.

The graph show significant improvement in closed step response comparing to integer order PID controller. Various step response characteristics are compared in Tables 4 and 5.

From Tables 4 and 5 it is clear that there is significant improvement in step response in term of rise time, settling time, peak over shoot and also in characteristic related to step disturbance. The response shows a very high robustness for step disturbance, it only show a under shoot of 0.6% which is very low as compared to integer order PID controller. The settling time for disturbance is 0 sec as the system response always remain in tolerable limit, this is due to the extra robustness obtained due to the two more degree of freedom of FOPID controller.

In the next section we use simulated annealing to search the parameter of controller.

**6.1.4. Simulated annealing based tuning for FOPID controller**

Annealing is the process of slow cooling of metal. Simulated Annealing is point to point stochastic guided random search global optimization technique which mimics the process of annealing. This optimization method was first proposed by Kiprpatrick, Gellet and Vecchi in 1983. Simulated annealing (SA) (Yachen & Yueqing, 2003) is used to solve unconstrained and bound-constrained optimization problems. When a metal is at high temperature the atoms present within the metals having high energy state move randomly in a high speed.

But as the metal start cooling the atoms losses its energy and ultimately orients itself in a crystalline state having lowest energy. But if the cooling process is very fast then the atoms might reach to semicrystalline state having energy higher than the crystalline state. This is equivalent to getting trapped in a local minima or premature convergence. But if the cooling rate is slow and controlled then this phenomena can be avoided. So cooling rate here is a very important parameter. This

| Table 3. Parameter of GA FOPID |
|--------------------------|-----------------|
| Parameter                | Value           |
| Lower bound \( (K_p, K_i, K_d, \mu, \lambda) \) | \( [1, 1, 1, 0, 0] \) |
| Upper bound \( (K_p, K_i, K_d, \mu, \lambda) \) | \( [1000, 500, 500, 2, 2] \) |
| Population size          | 60              |
| Crossover                | Scattered       |
| Crossover fraction       | 0.8             |
| Mutation                 | 0.01            |
| Stopping criteria        | 400             |
| Selection                | Stochastic uniform |
cooling rate is controlled according to the Boltzmann Probability distribution which can be given by Equation (38).

\[ P(E) = e^{-\frac{\Delta E}{kT}} \]  

(38)

where \( P(E) \) is the probability of energy distribution, \( k \) is the Boltzmann constant, \( T \) is the temperature and \( \Delta E \) is the change in energy. From Equation (38) we can easily conclude that if \( T \) increases there is a uniform probability of the atoms to be at any energy state, but if \( T \) decreases the probability of the atoms to be at a higher energy state reduces. At each of the iterations of the simulated annealing algorithm, a new point is randomly generated. The distance of the new point from the current point, or the extent of the search, is based on a probability distribution with a scale proportional to the temperature.

The algorithm not only accepts all new points that lower the objective function \( F \) (in case of minimizing problem), but also, accepts certain points that raise the objective function with a probability according to the Boltzmann probability distribution. As this algorithm searches the sample space very rigorously it becomes very robust and is used in solving very complex, noisy and discrete problems. For optimization we take \( k = 1 \), \( \Delta E \) is equivalent to the change in objective function and \( T \) is the control parameter.

Figure 5. Comparison of PID and FOPID tuned by GA.

Table 4. Response characteristic for step input

| Response specification | PID tuned by genetic algorithm | FOPID tuned by genetic algorithm |
|------------------------|-------------------------------|---------------------------------|
| Rise time              | 0.815 sec                     | 0.06 sec                        |
| Peak overshoot         | 16.85%                        | 5.4%                            |
| Settling time          | 7.24 sec                      | 1.1 sec                         |
| Steady state error     | 0.00                          | 0.00                            |

Table 5. Response characteristic for step input

| Response specification | PID tuned by GA | FOPID tuned by GA |
|------------------------|-----------------|-------------------|
| Peak undershoot        | 3%              | 0.6%              |
| Settling time          | 3 sec           | 0.0 sec           |
6.1.5. Algorithm of simulated annealing (SA)

Step 1: Select an Initial Point \( x(0) \) and a stopping criterion. Start with a very high value of temperature (T) and set the counter \( t \) on Zero i.e \( t = 0 \).

Step 2: Calculate the neighboring point \( x(t + 1) \).

Step 3: Find \( \Delta E(\Delta E = E(t + 1) - E(t)) \). If \( \Delta E \) is less then Zero then accept it and increment the counter. But if the \( \Delta E \) is greater the zero then unlike the conventional algorithm ,instead of straight forward rejecting it create a random number \( Y \) in between zero and one and if \( Y \) is less than or equal to \( e^{-\Delta E/kT} \) increment the \( t \) to \( (t+1) \). Otherwise go to step 2.

Step 4: If the magnitude of the difference between two points is less then the stopping criteria and the temperature (T) is small, stop the process. Otherwise reduce the temperature and repeat the step 2.

In this case also we use the same equation as Equation (19) as the fitness function and optimize the result using simulated annealing and find the parameter of FOPID that is the value of \( K_p, K_i, K_d, \lambda \) and \( \mu \). The parameter of simulated annealing is set as given in Table 6. The lower and upper bounds of FOPID controller parameter integral order (\( \lambda \)) and derivative order (\( \mu \)) are \( 0 \leq \lambda \leq 2.0 \) and \( 0 \leq \mu \leq 2.0 \).

After optimization, the function converge to 0.0204 after 8569 iteration and the value of \( K_p, K_i, K_d, \lambda \) and \( \mu \) comes out to be 103.1981, 103.5293, 48.0151, 0.6262 and 1.2737 respectively.

Transfer function of FOPID controller is given as:

\[
G_c(s) = 103.1981 + \frac{103.5293}{s^{0.6262}} + 48.0151s^{1.2737}
\]

(39)

Table 6. Parameter of SA for FOPID

| Parameter                  | Value                                      |
|----------------------------|--------------------------------------------|
| Initial value              | [148, 39.42, 47.4, 0.229, 0.813]           |
| Lower bound \([K_p, K_i, K_d, \lambda, \mu]\) | [1, 1, 1, 0, 0]                           |
| Upper bound \([K_p, K_i, K_d, \lambda, \mu]\)| [1000, 500, 500, 2, 2]                  |
| Max iteration              | 15,550                                     |
| Function tolerance         | 0.000001                                   |
| Annealing function         | 0.8                                         |
| Re-annealing interval      | 100                                         |

Again we plot the time responses for unit step input in Figure 6 for FOPID controller tuned on the basis of simulated annealing and compare the result with FOPID controller tuned by genetic algorithm.

Time response characteristics of FOPID controller tuned by genetic algorithm and simulated annealing are shown in Tables 7 and 8 from which comparison can be made between them.

From the curve it is observed the response obtained from simulated annealing based tuning approaches is inferior as compared to the FOPID based on genetic algorithm. Instead the response is far better as compared to integer order PID controller tuned based on genetic algorithm. Now we switch to the next tuning method which is frequency domain design based on constrained minimization.
6.2. Frequency domain design based on constrained minimization for FOPID controller

It is an optimization based frequency domain tuning method proposed by Monje et al. (2008). This takes two extra specifications on maximum value of magnitude of sensitivity and complementary sensitivity function along with the specification presented in minimization (Monje et al., 2008) frequency domain design of FOPID controller.

6.2.1. Design specifications and compensation problem

(1) No steady-state error: Properly implemented a fractional integrator of order \( k + \lambda, k, N, 0 < \lambda < 1 \), is, for steady error cancellation as efficient as integer order integrator of order \( k + 1 \).

(2) Phase margin \((\phi_m)\) and gain crossover frequency \((\omega_c)\) specifications: The conditions must be fulfilled

\[ |G_c(j\omega_c)G_p(j\omega_c)|_{dB} = 0 \text{ dB} \]

\[ \text{Arg}(G_c(j\omega_c)G_p(j\omega_c)) = -\pi + \phi_m \]  

(40)
where $G_c(s)$ is controller and $G_p(s)$ is the plant transfer function and $\omega_{cg}$ is the gain cross over frequency. With this condition the phase is forced to be flat at $\omega_{cg}$ and so, to be almost constant within an interval around $\omega_{cg}$. It means that the system is more robust to gain changes and the overshoot of the response is almost constant within the interval.

(3) Robustness to variations in the gain of the plant: To this respect, the next constraint must be fulfilled

$$\left( \frac{d(\text{Arg}(G_c(j\omega)G_p(j\omega)))}{d\omega} \right)_{\omega=\omega_{cg}} = 0$$

(41)

With this condition the phase is forced to be flat at $\omega_{cg}$ and so, to be almost constant within an interval around $\omega_{cg}$. It means that the system is more robust to gain changes and the overshoot of the response is almost constant within the interval.

(4) Robustness to high frequency noise: Complementary sensitivity function $T(j\omega)$ indicates the robustness to measurement noise and other un-modulated system dynamics. To have good time response $T(j\omega)$ should have small value at higher frequencies. For this it must be fulfilled the condition as in Equation (43):

$$T(j\omega) = \frac{G_c(j\omega)G_p(j\omega)}{1 + G_c(j\omega)G_p(j\omega)} \leq AdB$$

(42)

and $\forall \omega \geq \omega_1 \text{ rad/sec}$. Where $A$ is the desired noise attenuation for frequencies $\omega > \omega_1 \text{ rad/sec}$.

(5) To ensure a good output disturbance rejection: Sensitivity function $S(j\omega)$ indicates the ability of the system to suppress load disturbances and achieve good set-point tracking. To have good time response under disturbed condition the $S(j\omega)$ should have small value at low frequency. For this it must fulfill the condition:

$$S(j\omega) = \frac{1}{1 + G_c(j\omega)G_p(j\omega)} \leq BdB$$

(43)

and $\forall \omega \leq \omega_2 \text{ rad/sec}$ with $B$ the desired value of the sensitivity function for frequencies $\omega < \omega_2 \text{ rad/sec}$. A set of five of these six specifications can be met by the closed-loop system, since the fractional controller $C(s)$ has five parameters to tune. In this case, the specifications considered are those in Equations (39)–(44), ensuring a robust performance of the controlled system to gain changes and noise and a relative stability and bandwidth specifications. The condition of no steady-state error is fulfilled just with the introduction of the fractional integrator properly implemented, as commented before.

6.2.2. The problem of nonlinear minimization

From the specifications above, a set of five nonlinear equations with five unknown parameters $[K_p, K_d, K_i, \lambda, \mu]$ is obtained. The complexity of this set of nonlinear equations is very significant. The function used for this purpose is called fmincon (), which finds the constrained minimum of a function of several variables (Coleman & Zhang, 2011). In our case, the specification in Equation (38) is taken as the main function to minimize, and the rest of specifications are taken as constraints for the minimization, all of them subjected to the optimization parameters defined within the function fmincon(·) (Tepljakov, Petlenkov, & Belikov, 2011).

The transfer function of field controlled DC servo motor we want to control is given by Equation (13). The controller is designed to fulfill the following design specification in frequency domain.
(1) Gain crossover frequency, \( \omega_{cg} = 1 \text{ rad/sec} \)
(2) Phase margin, \( \phi_m = 0.73267 \text{ rad} = 42 \text{ degree} \)
(3) Robustness to the gain variation must be fulfilled
(4) \(|T(j\omega)|_{db} \leq -20 \text{dB}, \forall \omega \geq \omega_t = 10 \text{ rad/sec} \)
(5) \(|S(j\omega)|_{db} \leq -20 \text{dB}, \forall \omega \leq \omega_s = 0.1 \text{ rad/sec} \)

Now using the design specification and the condition given in Equations (39)–(44), we formulate the nonlinear equations as in Equation (45):

Putting \( \omega_{cg} = 1 \text{ rad/sec} \) in Equation (38)

\[
20 \log \left\{ 0.06744 \left( \left( K_p + K_i \cos \frac{\lambda \pi}{2} + K_d \cos \frac{\mu \pi}{2} \right)^2 + \left( K_d \cos \mu \frac{\pi}{2} - K_i \sin \lambda \frac{\pi}{2} \right)^2 \right) \right\} = 0 \tag{44}
\]

Putting the value of phase margin \( \phi_m = 0.73267 \text{ rad} \) and \( \omega_{cg} = 1 \text{ rad/sec} \) in Equation (41)

\[
0.002 \left( K_p + K_i \cos \frac{\lambda \pi}{2} + K_d \cos \frac{\mu \pi}{2} \right) - K_i \sin \lambda \frac{\pi}{2} + K_d \cos \frac{\mu \pi}{2} = 0 \tag{45}
\]

For \( \omega_{cg} = 1 \text{ rad/sec} \) in Equation (42)

\[
\left( K_p + K_i \cos \frac{\lambda \pi}{2} + K_d \cos \frac{\mu \pi}{2} \right) \left( K_d \mu \sin \mu \frac{\pi}{2} + K_i \lambda \sin \lambda \frac{\pi}{2} \right) \\
- \left( K_d \sin \mu \frac{\pi}{2} - K_i \sin \lambda \frac{\pi}{2} \right) \left( K_d \mu \cos \mu \frac{\pi}{2} - K_i \lambda \cos \lambda \frac{\pi}{2} \right) \\
- 0.55 \left( \left( K_p + K_i \cos \frac{\lambda \pi}{2} + K_d \cos \frac{\mu \pi}{2} \right)^2 + \left( K_d \sin \mu \frac{\pi}{2} - K_i \sin \lambda \frac{\pi}{2} \right)^2 \right) = 0 \tag{46}
\]

For \( \omega = 20 > \omega_t \) in Equation (42)

\[
\left( K_p + 20^{-i} \times K_i \cos \frac{\lambda \pi}{2} + 20^n \times K_d \cos \frac{\mu \pi}{2} \right)^2 + \left( 20^n \times K_d \sin \mu \frac{\pi}{2} - 20^{-i} \times K_i \sin \lambda \frac{\pi}{2} \right)^2 - 2.74 \times 10^{-3} \\
\times \left( 1.91 \left( K_p + 20^{-i} \times K_i \cos \frac{\lambda \pi}{2} + K_d \times 20^n \cos \mu \frac{\pi}{2} \right) - 8400 \right)^2 \\
+ \left( 1.91 \left( K_d \times 20^n \sin \mu \frac{\pi}{2} - 20^{-i} \times K_i \sin \lambda \frac{\pi}{2} \right) - 76000 \right)^2 \right\} \leq 0 \tag{47}
\]

For \( \omega = 0.001 \leq 0.01 \text{ rad/sec} \) in Equation (45)

\[
0.04 - 3.6481 \left( K_p + 0.001 \times K_i \cos \frac{\lambda \pi}{2} + 0.001^n \times K_d \cos \frac{\mu \pi}{2} \right)^2 \\
- \left( 1.91 \left( K_d \times 0.001^n \sin \mu \frac{\pi}{2} - 0.001^{-i} \times K_i \sin \lambda \frac{\pi}{2} \right) + 0.02 \right)^2 \leq 0 \tag{48}
\]

Equation (45) is taken as the main function to minimize and rest of the equation is taken as the constraint. Equations (38) and (39) serves as the nonlinear equality constraint and Equations (40) and (41) as nonlinear inequality constraint. Therefore the function to minimize is given as

\[
F = 20 \log \left( 0.06744 \sqrt{ \left( K_p + K_i \cos \frac{\lambda \pi}{2} + K_d \cos \frac{\mu \pi}{2} \right)^2 + \left( K_d \cos \mu \frac{\pi}{2} - K_i \sin \lambda \frac{\pi}{2} \right)^2 } \right) \tag{49}
\]

The parameters of FOPID are searched in such a way that the function \( F \) takes minimum value \( \text{fmincon()} \) result depend on the initial value, therefore it is always better to take some known good initial value. In our work we take the parameter of FOPID obtained in genetic algorithm as initial point. Parameter of \( \text{fmincon()} \) solver in active set algorithm is set as given in Table 9.

After running the algorithm in MATLAB the function converges to 25.02083 after 13 iteration. The value of FOPID parameter \( K_p, K_i, K_d, \lambda \) and \( \mu \) comes out to be 1.148, 39.419, 47.403, 0.229 and 0.813.
Now we generate step response and compare the result with the response obtained with genetic algorithm and simulated annealing method.

The curve shows a single overshoot which slowly settles down to steady states value. A steady state error is introduced of about 0.002 which is insignificant as far as our system is concerned. The response shows arise time of 0.4751 sec, peak over shoot 7.5% and settling time of 2 sec.

In the next section a comparison is made on the performance characteristics of three FOPID controller tuned by genetic algorithm simulated annealing and frequency domain design (Figure 7).

From Tables 10 and 11 it is clear that this method gives a slightly higher rise time, peak over shoot and settling time, but the response is far better than the response of integer order PID controller and simulated annealing based FOPID controller.

| Parameter of fmincon solver | Value |
|-----------------------------|-------|
| Start point                 | [132.8200 0.835 1.3] |
| Lower bound \([K_p, K_i, K_d, \mu]\) | [1, 1, 0, 0] |
| Upper bound \([K_p, K_i, K_d, \mu]\) | [1000, 500, 500, 2, 2] |
| Max iteration               | 400 |
| Max function evaluation     | \(400 \times \text{(No. of variable)} = 2,000\) |
| Parameter tolerance         | 0.000001 |

| Table 10. Response characteristic for step input |
|-------------------------------------------------|
| Response specification | FOPID tuned by GA | FOPID tuned by SA | FOPID tuned by frequency domain design |
|------------------------|-------------------|-------------------|----------------------------------------|
| Rise time              | 0.06 sec          | 0.33 sec          | 0.4751                                 |
| Peak overshoot         | 5.4%              | 15.68%            | 75%                                    |
| Settling time          | 1.1 sec           | 3.4 sec           | 2 sec                                  |
| Steady state error     | 0.00              | 0.00              | 0.002                                  |
7. Comparison of all tuning method

In this section we have compared the result of closed loop step response obtained from different tuning strategy used so far and find which tuning method is best one. The system is simulated for 60s, a step disturbance is introduced after 30 sec of step input to the system. The graph shown in Figure 8 the relative merits of PID and FOPID under different tuning strategy.

Tables 12 and 13 clearly shows that the performance characteristics obtained from FOPID is better in all respect. Among the FOPID tuned by different tuning method, genetic algorithm based approaches shows the best result in term of rise time, peak over shoot and settling time. Among FOPID SA based approaches gives highest over shoot of 15.68% and highest settling time of 3.4 sec. The results of frequency domain design and genetic algorithm is very close to each other.

| Table 11. Response characteristic for step disturbance input to the system |
|--------------------------------------------------|----------------|----------------|----------------|
| Response specification | FOPID tuned by GA | FOPID tuned by SA | FOPID tuned by frequency domain design |
| Peak undershoot | 0.6% | 0.7% | 1.75% |
| Settling time | 0.0 sec | 0.0 sec | 0.00 sec |

| Table 12. Comparison of step response of all system |
|--------------------------------------------------|----------------|----------------|----------------|
| Response specification | PID tuned by GA | FOPID tuned by GA | FOPID tuned by SA |
| Rise time | 0.815 sec | 0.06 sec | 0.33 sec |
| Peak overshoot | 16.85% | 5.4% | 15.68% |
| Settling time | 7.24 sec | 1.1 sec | 3.4 sec |
| Steady state error | 0.00 | 0.00 | 0.00 |

| Table 13. Comparison of step response of all system |
|--------------------------------------------------|----------------|----------------|----------------|
| Response specification | PID tuned by GA | FOPID tuned by GA | FOPID tuned by SA |
| Peak undershoot | 3% | 0.6% | 0.7% |
| Settling time | 3 sec | 0.00 sec | 0.00 sec |

Figure 8. Step response comparison of all tuning method.
8. Conclusion
In this paper we have used different tuning method to set the parameter of PID and FOPID to control a field controlled DC servo motor. Then performance comparison of PID and FOPID has been done on the basis of unit step response. After comparison of all the results it is found that FOPID is better in all respect whether it is rise time settling time or peak overshoot. Among the FOPID the best result is obtained by genetic algorithm based optimization approaches. Although the response obtained by frequency domain design based on constrained minimization by fmincon(.) is well in comparison with genetic algorithm approach but the complexity of obtaining the nonlinear constraint is very significant in frequency domain design. Also the output of fmincon(.) depend on the initial guess or starting point which is not the case in genetic algorithm.

The performance obtained from FOPID tuned by simulated annealing is inferior as compared to FOPID tuned by genetic algorithm and frequency domain design method but far better than integer order PID controller. Moreover simulated annealing takes large number of iteration and more time to converge and also its output depend on the initial guess or start point. Simulated annealing based optimization instead of being a global search technique, in our system seems to give local optimum point but the point is acceptable as far as the characteristic of the step response is concerned.

Funding
This work is funded by National Institute of Technology, Arunachal Pradesh under project entitled “New Simulated annealing based optimal tuning of Fractional Order Proportional-Integral-Derivative (PIλDμ) Controller” and National Research Foundation [grant number 97823]. Authors would like to thank NIT Arunachal Pradesh for funding this project.

Author details
Prashant Kumar 1
E-mail: pkab36@gmail.com
Shantanu Chatterjee 1
E-mail: instru59eng@gmail.com
ORCID ID: http://orcid.org/0000-0002-2889-0434
Devnath Shah 1
E-mail: devnath.nitap@gmail.com
U. K. Saha 2
E-mail: uksahanitap@gmail.com
SaiBal Chatterjee 3
E-mail: saibalds@ieee.org
ORCID ID: http://orcid.org/0000-0003-1680-7565

1 Department of Electrical Engineering, National Institute of Technology, Arunachal Pradesh, Papum Pare, India.
2 Department of Basic & Applied Science, National Institute of Technology, Arunachal Pradesh, Papum Pare, India.
3 Department of Electrical Engineering, NERIST, Arunachal Pradesh, India.

Citation information
Cite this article as: On comparison of tuning method of FOPID controller for controlling field controlled DC servo motor, Prashant Kumar, Shantanu Chatterjee, Devnath Shah, U. K. Saha & Saibal Chatterjee, Cogent Engineering (2017), 4: 1357875.

References
Alshehhi, K. (2014). Combined armature and field voltage control of DC motor. The British University in Dubai.
Aström, K. J., & Hägglund, T. (1988). Automatic tuning of PID controllers. Instrument Society of America (ISA).
Bandyopadhyay, S., Saha, S., Maulik, U., & Deb, K. (2008). A simulated annealing-based multiobjective optimization algorithm: AMOSA. IEEE Transactions on Evolutionary Computation, 12, 269–283.
Bouafoura, M. K., & Braiekh, N. B. (2010). PIλDμ controller design for integer and fractional plants using piecewise orthogonal functions. Communications in Nonlinear Science and Numerical Simulation, 15, 1267–1278.
Chen, Y. Q., Hu, C. H., & Moore, K. L. (2003). Relay feedback tuning of robust PID controllers with iso-damping property. 42nd IEEE Proceedings Conference on Decision and Control, 2003, 3, 2180–2185.
Chen, Y. Q., Moore, K. L., Vinagre, B. M., & Podlubny, I. (2004). Robust PID controller autotuning with a phase shaper. First IFAC workshop on fractional differentiation and its applications, 162–167.
Coleman, T. F., & Zhang, Y. (2011). Fmincon [Online]. Natick: Mathworks.
Cominos, P., & Munro, N. (2002). PID controllers: recent tuning methods and design to specification. IEEE Proceedings-Control Theory and Applications, 149, 46–53.
Davis, P. J. (1959). Leonhard Euler’s integral: A historical profile of the gamma function. In memoriam: Milton Abramowitz. The American Mathematical Monthly, 66, 849–869.
Eckert, M., Kpper, M., & Hohmann, S. (2014). Functional fractional calculus for system identification of battery cells. Automatisierungstechnik, 62, 272–281.
Edet, E., & Katebi, R. (2016). Design and tuning of fractional-order PID controllers for time-delayed processes. In Control (CONTROL), 2016 UKACC 11th International Conference (pp. 1-6). IEEE.
Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. Machine Learning, 3, 95–95.
Granda, J. J. (2002). The role of bond graph modeling and simulation in mechatronics systems: An integrated software tool: CAMP-G, MATLAB-SIMULINK. Mechatronics, 12, 1271–1295.
Hanselman, D. C., & Littlefield, B. (1995). MATLAB; Version 4: User’s Guide, Prentice Hall PTR.
Kushwah, M., & Patra, A. (2014). Tuning PID controller for speed control of DC motor using soft computing techniques–A review. Advance in Electronic and Electric Engineering, 4, 141–148.
Li, L., Yang, Y., Peng, H., & Wang, X. (2006). An optimization method inspired by “chaotic” ant behavior. International Journal of Bifurcation and Chaos, 16, 2351–2364.
Maione, G., Punzi, A., & Li, K. (2013). A comparative study on differential evolution with other heuristic methods for continuous optimization (pp. 1356–1361). 21st Mediterranean Conference on Control and Automation, Athens.
Maiti, D., Biswas, S., & Konar, A. (2008). Design of a fractional order PID controller using particle swarm optimization technique. arXiv preprint arXiv:0810.3776.

Miller, K. S., & Ross, B. (1993). An introduction to the fractional calculus and fractional differential equations. New York, NY: Wiley-Interscience.

Monje, C. A., Vinagre, B. M., Feliu, V., & Chen, Y. Q. (2008). Tuning and auto-tuning of fractional order controllers for industry applications. Control Engineering Practice, 16, 798-812.

Munkhammar, J. D. (2004). Riemann-Liouville fractional derivatives and the Taylor-Riemann series. UUDM Project Report, 7, 1-18.

Oustaloup, A., Levron, F., Mathieu, B., & Nanot, F. M. (2000). Frequency-band complex noninteger differentiator: characterization and synthesis. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 47, 25-39.

Padhee, S., Gautam, A., Singh, Y., & Kaur, G. (2011). A novel evolutionary tuning method for fractional order PID controller. International Journal of Soft Computing and Engineering, 1, 1-9.

Podlubny, I. (1999). Fractional-order systems and PI/sup/lambda/D/sup/mu/-controllers. IEEE Transactions on Automatic Control, 44, 208-214.

Singhal, R., Padhee, S., & Kaur, G. (2012). Design of Fractional order PID Controller for speed control of DC Motor. International Journal of Scientific and Research Publications, 2, 1-8.

Tepljakov, A., Petlenkov, E., & Belikov, J. (2011). FOMCON: A MATLAB toolbox for fractional-order system identification and control. International Journal of Microelectronics and Computer Science, 2, 51-62.

Vinagre, B. M., Podlubny, I., Dorcak, L., & Feliu, V. (2000). On fractional PID controllers: a frequency domain approach. In Proceedings of IFAC Workshop on Digital Control-PID00. Terrassa.

Vu, T. N., & Lee, M. (2013). Analytical design of fractional-order proportional-integral controllers for time-delay processes. ISA Transactions, 52, 583-591.

Yachen, Z., & Yueming, H. (2008). On PID controllers based on simulated annealing algorithm. 27th Chinese Control Conference, 225-228.

**Appendix 1**

**Specification of field controlled dc servo motor**

| Rated power P | 10 HP |
|---------------|-------|
| Rated armature voltage (Va) | 240 V |
| Field winding resistance (Rf) | 1.39 ohm |
| Field winding inductance (Lf) | 0.1 Henry |
| Field winding current (If) | 10 Ampere |
| Applied field voltage (Vf) | 300 V |
| Motor torque constant (K) | 0.05 N-m/A |
| Moment of inertia of motor (J) | 0.02 Kgm² |
| Viscous friction coefficient (B) | 0.02 N-m/rad/sec |
