This is a postprint version of the following published document:

Haji Haji, V. y Monje, C.A (2017) Fractional-Order PID Control of a Chopper-Fed DC Motor Drive Using a Novel Firefly Algorithm with Dynamic Control Mechanism. *Soft Computing*, 22(18), pp. 6135-6146.

DOI: <https://doi.org/10.1007/s00500-017-2677-5>

Fractional Order PID Control of a Chopper Fed DC Motor Drive Using a Novel Firefly Algorithm with Dynamic Control Mechanism

Vahab Haji Haji · Concepción A. Monje

Abstract In this paper, a dynamic control mechanism is proposed to improve the Firefly Algorithm's (FA) rate of convergence and minimization of the fitness function. The Dynamic FA (DFA) dynamically selects the best performing combinations of the step size scaling factor, the attractiveness coefficient, the absorption coefficient, and the population size along the complete evolution process of the algorithm. A fractional order PID (FOPID) controller based on DFA is proposed to improve the performance of a chopper fed Direct Current (DC) motor drive. The proposed controller is used in speed control loop to improve the response. To illustrate the efficacy of the DFA based FOPID, we compare its performance with those based on the conventional FA, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) algorithm, and Differential Evolution (DE) algorithm. The simulation results and analyses show the effectiveness of the proposed method.

Keywords Chopper fed DC motor drive · Fractional order PID · Dynamic firefly algorithm · Optimization

---

V. Haji Haji
Young Researchers and Elite Club, Borujerd Branch, Islamic Azad University, Borujerd, Iran
Tel.: +98-9365475525
Fax: +98-66-42605335
E-mail: vhbhaji@gmail.com

C. A. Monje
Systems Engineering and Automation Department, University Carlos III of Madrid, Avenida Universidad 30, 28911 Legans, Madrid, Spain
E-mail: cmonje@ing.uc3m.es

1 Introduction

There has been a continuous development of high performance motor drives for industrial applications, such as electric vehicles, steel rolling mills and robotic manipulators. A high performance motor drive should be reliable, and should have good dynamic speed command tracking and load regulating response (Rajasekhar et al. 2014). Direct Current (DC) motors drives are less complex, simpler to control and lower cost compared to Alternating Current (AC) drives systems. In this paper, we investigate the control of a DC motor speed using chopper. A chopper converts a fixed DC input voltage into a variable DC output voltage.

There has been a growing interest in the application of fractional calculus in various fields of science and engineering, such as mathematics, physics, chemistry, and mechanics (Podlubny 1999; Monje et al. 2010; Aghababa 2016). The use of fractional calculus in the area of control theory extends the opportunity to enhance the system control performance of conventional controllers. A typical FOPID controller is a generalized version of the classic PID controller and requires fine tuning of five interdependent parameters i.e., proportional (K_P), integral (K_I), derivative (K_D), derivative order (μ), and integral order (λ), in order to design a control structure and meet all design specifications. Several analytical, rule-based, and optimization approaches have been presented to offer new effective tuning techniques for FOPID controllers. The analytical methods can be used only when there are few and simple equations involved in the control system model. Rule-based methods offer great tuning rules to calculate the controller parameters easily; however, they are in general limited to the shape of the step response. Optimization methods have demonstrated to overcome

some of the previously cited problems from analytical and rule-based methods and are a good choice for tuning FOPID controllers, as shown in one of the works by Martín et al. 2015, where the optimization is carried out both in time and frequency domains.

A wide variety of fractional order controller problems and tuning methods can be found in literature following these three methodologies, such as Liu et al. (2014), Li et al. (2010), Perng et al. (2016), Monje et al. (2008), and Sharma et al. (2016). A robust controller design based on fractional calculus for a wind turbine generator is presented in Ghasemi et al. (2014). In Maâmar and Rachid (2014) a FOPID filter controller tuning method based on the Internal Model Control (IMC) method is investigated. In Pan and Das (2013) several intelligent techniques have been proposed for tuning optimal FOPID controllers based on minimizing indices including Integral Time Absolute Error (ITAE), Integral Absolute Error (IAE), Integral Time Squared Error (ITSE) and Integral Squared Error (ISE).

Evolutionary and Swarm Intelligence (SI) algorithms, such as Genetic Algorithm (GA) (Sarkar et al. 2013), Particle Swarm Optimization (PSO) (Pandey et al. 2014; Wang et al. 2017), Differential Evolution (DE) (Martín et al. 2015), and Artificial Bee Colony Algorithm (ABC) (Rajasekhar et al. 2014), have a long history in solving optimization problems of control systems. Another alternative is the Firefly Algorithm (FA), which is a nature inspired optimization algorithm that simulates the flashing behavior of fireflies and has been widely used in literature. For instance, the authors in Naidu et al. (2014) use FA with online wavelet filter in automatic generation control of a three unequal area interconnected reheat thermal power system. The classical controllers of a combined cycle gas turbine (CCGT) plant using FA is discussed in Saikia and Sahu (2013). Yang et al. (2011) uses FA to determine the feasible optimal solution of the economic dispatch problems. The performance of the FA for clustering problems is compared with ABC and PSO in Senthilnath et al. (2011). A comprehensive review of FA and its application can be found in Fister et al. (2013, 2014, 2015) and Yang and He (2013).

Even though the FA shows a significant success in solving different optimization problems, its performance is very dependent on the right combination of control parameters. A better parameter tuning affects the convergence rate and the minimization of its cost function. According to the settings of the FA parameters, there are two main ways to assign parameters properly: 1) by determining the values of parameters before the run, and keeping parameters fixed through-

out the iterations and 2) by changing the values of the parameters during the iterations (Fister et al. 2013). Yu et al. (2014) proposed a wise step strategy for FA (WSSFA) to update the step size scaling factor (α) for each firefly. In Cheung et al. (2014), a new adaptive firefly algorithm (AdaFa) is proposed. The AdaFa included five dynamic strategies to set α . In order to update the attractiveness coefficient (β_0) and the absorption coefficient (γ), a new chaos FA (CFA) is presented in Gandomi et al. (2013). Further details about FA with adaptive control parameters strategies can be found in Wang et al. (2016).

In this paper, we go a step further and propose a new mechanism to dynamically select the best performing combinations of the step size scaling factor, the attractiveness coefficient, the absorption coefficient, and the population size for the FA algorithm. Our approach not only considers a set of one or two parameters to be dynamically changed, as in the previously cited works, but it goes a step further and deals with the complete set of parameters affecting the performance of the algorithm. Then, it is expected that, when compared to standard FA, GA, PSO, ABC and DE algorithms, it brings improvements to the performance, as will be discussed later. A FOPID controller based on the DFA algorithm is implemented to improve the performance of a chopper fed DC motor drive. The well-known fitness functions ITSE, ITSE+ISCO (Integral Squared Control Output), ITAE and ITAE+ISCO are considered to show the dynamic performance of the proposed controller with optimum gains. Simulation results and analyses show the superiority of the proposed DFA algorithm over FA, GA, PSO, ABC and DE algorithms.

The structure of this paper is organized as follows: In Section 2, the mathematical modeling of a DC motor drive is presented and the basic concepts of the FOPID controller are described. The FA and DFA algorithms are discussed in Section 3. Simulation results and performance analyses are provided in Section 4. Finally, the paper is concluded in Section 5.

2 Modelling and Control Scheme

This paper focuses on the study of a separately excited DC motor. It can be regulated using chopper as a converter. The equivalent circuit of a separately excited DC motor is shown in Fig. 1. When the motor is excited by

Fig. 1 Equivalent circuit diagram of a separately excited DC motor

a field current (I_f) and as a consequence an armature current (I_a) flows in the circuit, the motor develops a Back Electromotive Force (BEMF) and a torque to balance the load torque at a particular speed (Rajasekhar et al. 2014; Gopakumar 2010). The armature voltage E_a and the equivalent torque are as follows:

$$V_a = I_a R_a + L_a \frac{dI_a}{dt} + E_b, \quad (1)$$

$$T_d = J \frac{d\omega}{dt} + B\omega + T_L, \quad (2)$$

where R_a , Armature resistance; L_a , Armature inductance; I_a , Armature current; I_f , Field current; V_a , Armature Voltage; E_b , BEMF; T_L , Load torque; ω , Angular velocity of rotor; J , Moment of Inertia; B , friction coefficient of motor; T_d , Developed torque. Assuming $B = 0$, Eqn. (2) will be reduced to:

$$T_d = J \frac{d\omega}{dt} + T_L. \quad (3)$$

The BEMF and torque developed are defined as:

$$E_b = K\Phi\omega, \quad (4)$$

$$T_d = K\Phi I_a, \quad (5)$$

where Φ and K are the field flux and the BEMF constant, respectively. By applying the Laplace transform the following equations are obtained:

$$I_a(s) = \frac{V_a - E_b}{R_a + L_a s} = \frac{V_a - K\Phi\omega}{R_a(1 + \frac{L_a}{R_a}s)}, \quad (6)$$

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

The block diagram representation of the speed control for the DC motor is shown in Fig. 2. After performing a block reduction, the relationship between $\omega(s)$ and $V_a(s)$ will be as follows:

$$\frac{\omega(s)}{V_a(s)} = \frac{\frac{1}{K_m}}{(1 + sT_m)(1 + sT_a)}, \quad (8)$$

where $T_a = L_a/R_a$, $T_m = JR_a/(K\Phi)^2$, $K_m = K\Phi$ and $T_L = 0$. T_m and T_a are the electromechanical and

Table 1 DC motor drive parameters

Power = 300kW
Base speed = 500rpm
Max rated voltage (V_a) = 460V
Max rated current (I_a) = 690A
Current limit = 1200A
Inductance (L_a) = 0.7026mH
Moment of inertia (J) = 84kgm ²
BEMF contant (K_m) = 8.5Vs/rad
Armature resistance (R_a) = 0.02342W
Controller gain (K_c) = 0.2618
Current loop filter gain (K_2) = 0.0083
Speed loop filter gain (K_1) = 0.19
Chopper constant (K_t) = 46
Armature time constant (T_a) = 30ms
Controller time constant (T_c) = 30ms
Electromechanical time constant (T_m) = 27.55ms
Tachometer constant (T_1) = 25ms
Current feedback (T_2) = 3.5ms

the electrical time constants, respectively. In Fig. 2, by assuming $T_c = T_a$, the relationship between I_a and I_a^{ref} can be reduced to the following form:

$$\frac{I_a(s)}{I_a^{ref}(s)} = \frac{1/K_2}{1 + s2T_2}, \quad (9)$$

where K_2 and T_2 are the current filter gain and the current loop filter time constant, respectively. Figure 2 can be further reduced to Fig. 3. The model parameters are provided in Table 1 (Gopakumar 2010; Rajasekhar et al. 2014).

2.1 Fractional Order PID Controller

The application and development of the fractional calculus in the area of control theory extends the opportunity to obtain further improvement in performance response of the conventional controllers. The applications of the FOPID controller in industry are presented by Monje et al. (2010). Monje et al. (2008) proposes an auto-tuning method for FOPID controllers using a relay test, where fractional order PID or $PI^\lambda D^\mu$ is a generalized version of the conventional PID controller in which the order of the derivation μ and the integration λ are not integer. The continuous transfer function of a $PI^\lambda D^\mu$ controller is defined as follows:

$$C_s = K_P + K_I s^{-\lambda} + K_D s^\mu, \quad (10)$$

where $\lambda, \mu > 0$. The $PI^\lambda D^\mu$ controller output $u(t)$ in time domain is:

$$u_t = K_P e(t) + K_I D^{-\lambda} e(t) + K_D D^\mu e(t), \quad (11)$$

where $e(t)$, K_P , K_I and K_D are the error signal and the proportional, integral, and derivative gains, respectively. In our approach, this FOPID is used for the

Fig. 2 Complete layout of the speed control in a DC motor drive.

Fig. 3 A simplified layout of the speed control in a DC motor drive.

speed control of the DC motor drive in Fig. 3. The controller will be tuned following a novel DFA technique in which the optimization parameters are dynamically selected for the sake of the convergence speed and the minimization of the fitness function.

3 Firefly Algorithm

The FA is an optimization algorithm introduced by Xin-She Yang in late 2007 and 2008 (Yang 2009; Naidu et al. 2014). The FA simulates the flashing behavior of the fireflies. The main premise of this algorithm is established upon three idealized rules: (1) All fireflies in the swarm are unisex so that one firefly is attracted to others regardless of their sex; (2) Attractiveness is directly proportional to the brightness. This means that the less bright firefly will be attracted and moved toward the brighter one. Both the attractiveness and the brightness decrease as their distance increases. If there is no firefly with higher level of brightness, the fireflies

move randomly. (3) The brightness of a firefly is dictated by the objective function (Yang and He 2013). Based on these three rules, a pseudo code of the FA is formulated in Fig. 4. The relationship between the attractiveness function β and the distance r of the fireflies can be determined by:

$$\beta(r) = \beta_0 e^{-\gamma r^2}, \quad (12)$$

where β_0 and γ are the attractiveness at $r = 0$ and the light absorption coefficient, respectively. The distance r_{ij} between any two fireflies i and j at location x_i and x_j is defined as:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}, \quad (13)$$

where $x_{i,k}$ and $x_{j,k}$ represent the k th component of the i th and j th firefly positions, and d denotes the dimensionality of the problem. The movement of firefly i which is attracted to the j th firefly is determined by:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha(\text{rand} - 0.5), \quad (14)$$

Firefly Algorithm

```

1. Generate random population of fireflies  $x_i$ 
2. Light intensity  $I_i$  at  $x_i$ 
3. while ( $t < \text{MaxGeneration}$ )
4.   for  $i = 1 : n$  all  $n$  fireflies
5.     for  $j = 1 : i$  all  $n$  fireflies
6.       if ( $I_j > I_i$ )
7.         Move firefly  $i$  towards  $j$  in all  $d$  dimensions
8.       end if
9.     Attractiveness varies with distance  $r$ 
10.    Evaluate new solutions
11.    Update light intensity
12.  end for  $j$ 
13. end for  $i$ 
14. Rank the fireflies and find the current best
15. end while

```

Fig. 4 Pseudo code of the firefly algorithm.

where α is the randomization parameter and $rand$ is a random number generator distributed within $[0, 1]$.

3.1 Dynamic Firefly Algorithm

In solving any optimization problem, the choice of the search operators and right combination of the control parameters play a pivotal role in the performance of the optimization algorithm. A dynamic parameters selection mechanism was introduced by Sarker and Elsayed (2014). This paper uses a dynamic selection mechanism to dynamically select the best performing combination of the parameters (amplification factor, crossover rate, and population size) for the DE algorithm. Authors show the superiority of the dynamic DE over other state-of-the-art algorithms. Besides, the paper shows that there is no fixed combination which is the best during the evolution process for all optimization test problems.

The FA starts with a random population and uses attraction and randomization operations to move each individual in the population. Various studies have used different values for each parameter α , β_0 , γ . Yang et al. (2011) carried out a detailed parametric study by varying α , β_0 , γ and population size PS to obtain the right parameters combination. This paper found that the best values for ranges of the parameters are: $\alpha \in [0.4, 0.9]$, $\beta_0 \in [0.5, 1.0]$, $\gamma \in [0.1, 10]$, and $PS \in [25, 50]$. Yang (2009) recommended that $\beta_0 = 1$, $\alpha \in [0, 1]$, and $\gamma \in [0.01, 100]$. Yang used the values $\alpha = 0.2$, $\gamma = 1$, and $\beta_0 = 1$ to find the global optimum of the Michalewicz function. Senthilnath et al. (2011) took $\beta_0 = 1$, $\gamma = 1$, and $PS = 20$ for clustering problem to identify homogeneous groups of objects based on the values of their attributes. Saikia and Sahu (2013) used tuned values $\beta_0 = 0.2$, $PS = 5$, $\gamma = 0.5$, and $\alpha = 0.5$

for the automatic generation control of a combined cycle gas turbine plant.

In order to overcome the right parameters selection problem, a dynamic parameter selection mechanism is developed in this work and proposed to choose the right combination of control parameters (step size scaling factor α , attractiveness coefficient β_0 , absorption coefficient γ , and population size PS) in a firefly based optimization problem. The pseudocode of the algorithm is presented in Fig. 5. The DFA starts with a random population and a random combination of α , β_0 and γ for each individual in the population. The success rate of each combination is recorded for a certain number of generations CS . Based on the success rate, the best combinations are selected and used for a number of subsequent generations. This process is named as a cycle. The fireflies are moved as follows:

$$x_i = x_i + \beta_{0,i} e^{-\gamma_i r_{ij}^2} (x_j - x_i) + \alpha_i (rand - 0.5), \quad (15)$$

where $\alpha \in \alpha_{set} = \{\alpha_1, \alpha_2, \dots, \alpha_{n\alpha}\}$, $\beta_0 \in \beta_{0,set} = \{\beta_{0,1}, \beta_{0,2}, \dots, \beta_{0,n\beta_0}\}$, $\gamma \in \gamma_{set} = \{\gamma_1, \gamma_2, \dots, \gamma_{n\gamma}\}$, and $PS \in PS_{set} = \{PS_1, PS_2, \dots, PS_{nps}\}$. Here, $n\alpha$, $n\beta_0$, $n\gamma$, and nps refer to the cardinality of the set step size scaling factor, the attractiveness coefficient, the absorption coefficient, and the population size PS , respectively.

If the new position x_i is better than its previous position, the success of a combination will be increased by one ($SR_y = SR_y + 1$), where y is a combination of parameters, $y \in y_{set}$, and y_{set} is the combination of all α , β_0 , and γ . After CS generations the population size reduces to PS_{nps-1} and the remaining $PS_{nps} - PS_{nps-1}$ fireflies are archived, where PS_{nps} is assumed to be larger than PS_{nps-1} . After that, based on Eqn. (16) the success rate of each combination $Rank_y$ is recorded and the number of combinations is reduced to the half. After every few cycles, the process restarts with all possible combinations of parameters.

$$Rank_y = \frac{SR_y}{N_y}, \quad (16)$$

where N_y is the number of times a combination y is used and the combinations with the higher $Rank_y$ are chosen for the next generations. At the end of $CS \times nps$ generations, using Eqn. (17) the best population size is selected and used for the $(\eta - nps) \times CS$ next generations.

$$Rank_{PS_i} = \frac{\sum_1^{CS} \sum_{k=1}^{TC} SR_y}{PS_k^2}, \quad (17)$$

where TC is the total number of combinations α_{set} , $\beta_{0,set}$, and γ_{set} , and for instance, if $TC = 16$, then the maximum value of η is 4 ($16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$), which means that TC can be divided by 2 for 4 times. Finally,

Dynamic Firefly Algorithm

```

1. Define the parameters,  $i_{pop} = nps, \alpha_{set}, \beta_{0,set}, \gamma_{set}$ , and  $PS_{set}$ 
2. Objective function  $f(x), x = (x_1, \dots, x_d)^T$ 
3. Generate initial random population of fireflies  $x_i (i = 1, 2, \dots, n)$ 
4. Light intensity  $I_i$  at  $x_i$  is determined by  $f(x_i)$ 
5. while ( $t < MaxGeneration$ )
6.   Assign a random combination ( $y$ ) of parameters
7.   for  $i = 1 : n$  all  $n$  fireflies
8.     for  $j = 1 : i$  all  $n$  fireflies
9.       if ( $I_j > I_i$ )
10.        Move firefly  $i$  towards  $j$  in all  $d$  dimensions
11.        if new vector is better than its previous vector, then  $SR_y = SR_y + 1$  ;end if
12.        end if
13.      end for  $j$ 
14.    end for  $i$ 
15.     $period = period + 1$ ;
16.     $PS_{prd} = PS_{prd} + 1$ ;
17.    Rank the fireflies and find the current best
18.    if  $mod(period, CS) = 0$  and  $period < (\eta * CS)$ 
19.      Select the best half combination based on the rankings using equation (16) and update  $y_{set}$ 
20.    else if  $mod(period, \eta * CS) = 0$ 
21.      Set each  $SR_y = 0$  and  $period = 0$ 
22.    end if
23.    if  $mod(PS_{prd}, CS) = 0$  and  $i_{pop} > 0$ 
24.      Calculate  $Rank_{PS_{i_{pop}}}$  using (17)
25.      set  $i_{pop} = i_{pop} - 1$ 
26.      if  $i_{pop} \sim 0$ 
27.        Archive the worst ( $PS_{i_{pop}+1} - PS_{i_{pop}}$ ) individuals
28.        Set  $PS = PS_{i_{pop}}$ 
29.      end if
30.    end if
31.    if  $i_{pop} = 0$  and  $PS_{prd} = nps * CS$ 
32.      Set  $PS$  to the population size with the best ranking
33.      Use require individuals from the archive
34.    end if
35.    if  $PS_{prd} = \eta * CS$ 
36.      Set  $PS_{prd} = 0, i_{pop} = nps$ , and  $PS = PS_{i_{pop}}$ 
37.      Use require individuals from the archive and clear the archive
38.    end if
39.  end while

```

Fig. 5 Pseudo code of the dynamic firefly algorithm.

after $\eta \times CS$ generations, the process restarts with all combinations of control parameters.

4 Simulation Results

The considered FOPID is used in speed loop of the chopper fed DC motor drive according to Fig. 3. The gains of the controller are optimized using our DFA approach. In this DFA based optimization, the maximum number of iterations is 100, $CS = 40$, $\eta = 3$, $\beta_0 \in \{0.6, 0.8, 1\}$, $\alpha \in \{0.1, 0.3, 0.5, 0.7, 1\}$, $\gamma \in \{0.1, 1, 10\}$ and $PS \in \{5, 10\}$. The parameters settings of the GA, DE, ABC, PSO, and FA algorithms are listed in Table 2. We consider the time indices ITSE, ITSE+ISCO, ITAE, ITAE+ISCO for tuning the gains of the con-

troller using the GA, PSO, DE, ABC, FA, DFA algorithms:

$$J_1 = ITSE = \int_0^\infty t.e^2(t)dt, \quad (18)$$

$$J_2 = ITSE + ISCO = \int_0^\infty [w_1.t.e^2(t) + w_2.u^2(t)]dt, \quad (19)$$

$$J_3 = ITAE = \int_0^\infty t.|e(t)|dt, \quad (20)$$

$$J_4 = ITAE + ISCO = \int_0^\infty [w_1.t.|e(t)| + w_2.u^2(t)]dt, \quad (21)$$

where e , u , w_1 , and w_2 are the error signal, the control signal, the error weight, and the control weight, respectively, and J_1 , J_2 , J_3 , and J_4 refer to the ITSE, ITSE+ISCO, ITAE, and ITAE+ISCO functions, respectively. The ISCO criterion considers the introduction of

the control signal into the objective function, allowing this way the weighting of the control law within the minimization problem in order to prevent the saturation of the system to be controlled. In our case, since there is no particular restriction on the saturation to be considered, and for the sake of simplicity, we have weighted both the error and the control signals equally, selecting $w_1 = w_2 = 1$.

In order to incorporate statistical results, all the algorithms have been run 10 times, and best, worst and average results have been included in this section and discussed in detail. This way the performance of each algorithm can be assessed in a fairer manner. The convergence characteristics of the FA, DFA, PSO, GA, DE, and ABC algorithms for time indices ITSE, ITSE+ISCO, ITAE, and ITAE+ISCO are compared in Fig. 6. which shows the best results from the algorithms. It is clear that the DFA converges faster than GA, PSO, ABC, and DE algorithms. The optimum parametric gains and time indices of the proposed FOPID controller are provided in Tables 3-6, where M_p , T_r , T_s , and Of refer to the maximum peak, rise time (sec), settling time (sec), and objective function, respectively. The simulation results obtained by the different algorithms in 10 runs are provided in Table 7 and Table 8. In this comparison, the DFA provides better average for all objective functions. As for the ITSE function, as shown in Fig. 6 and Table 5, the DFA algorithm has the ability to provide a fast convergence and a low objective function. It is clear that the GA based controller provides the worst final cost function, as also shown in Table 7. In the case of the ITSE+ISCO objective function, from Fig. 6 and Table 5, it is evident that the FA and ABC based FOPID controllers provide a fast convergence, but fail in cost function term compared to the DFA algorithm. As for the ITAE function, as shown in Fig. 6 and Table 6, the GA algorithm provides the worst final error, and FA algorithm cannot show a proper performance compared to the DE, DFA, ABC, and PSO algorithms. In the case of the ITAE+ISCO fitness function, from Fig. 6 and Table 6, it is clear that the DFA and FA algorithms show a very fast convergence rate.

The dynamic speed responses of the DC motor drive controlled with the optimally tuned FOPID controllers based on the DFA, FA, PSO, GA, DE, and ABC algorithms by minimizing the ITSE, ITSE+ISCO, ITAE, and ITAE+ISCO functions are obtained and shown in Fig. 7. As for the ITSE function, from Fig. 7 and Table 5, it is clear that the DFA based FOPID provides a low settling time and quick rise time compared to the GA and DE based controllers, while the overshoot is bigger than those with the DE and PSO based ones. The PSO based FOPID provides a very low overshoot

but fails in the minimization of the objective function. In the case of the ITSE+ISCO function, Fig. 7 and Table 5 show that the DFA provides a very smooth response with very low overshoot, settling time and objective function. Although the GA based FOPID provides a very low rise time, it fails in the minimization of the overshoot and the settling time. As for ITAE function, from Fig. 7 and Table 6, it is obvious that the DE based FOPID provides a very low overshoot and settling time. From Fig. 7 and Table 6, regarding the ITAE+ISCO function, it is clear that the DFA based FOPID, in terms of objective function minimization, outperforms the GA, PSO, ABC, DE, and FA based controllers.

Figure 8 depicts the control signals produced by all the FOPID controllers tuned for the four minimization functions. As for ITSE+ISCO and ITAE+ISCO, the DFA based FOPID provides very low intensity control signals compared to the rest of algorithms. It can be also observed that the additional term ISCO helps the ITSE+ISCO and the ITAE+ISCO objective functions to provide control signals that are lower than those from the ITSE and ITAE objective functions.

To observe the variations of α , β_0 , and γ during the course of evolution, the corresponding best parameters values during the evolution process are presented in Fig. 9. It can be seen that no fixed combination is the best during the entire evolution process.

In order to improve the statistical analysis of the results, a Mann-Whitney U test has been also conducted (Mann and Whitney 1947), which is a nonparametric test applied when the simulation data do not have normal distributions. The asymptotic significance resulting from the test is lower than 0.05, which implies that there is a significant difference between the DFA results and those from ABC, DE, GA, PSO, and FA algorithms. In other words, the results obtained in this work could not have happened by chance. This type of statistical analysis will be discussed in detail in future works.

5 Conclusions

This paper proposes a new FA algorithm improved with dynamic parameters selection. The DFA algorithm is based on the FA and dynamically selects the best performing combinations of the attractiveness coefficient, the absorption coefficient, and the population size. The analyses show that the presented dynamic parameters selection mechanism improves the convergence rate and the minimization of the cost function for the FA algorithm compared to other conventional techniques.

Table 2 Parameters settings in simulation

Algorithm	Parameters value
GA	Population size = 10, Mutation probability = 0.2, Crossover probability = 0.8
DE	Population size = 10, Scaling Factor = 0.6, Crossover probability = 0.2
PSO	Population size = 10, Velocity coefficients = 2, Inertia weight = 1
ABC	Population size = 16, Limit = 30
FA	Population size = 10, $\gamma = 1$, $\beta_0 = 1$, $\alpha = 0.9$

Fig. 6 Convergence characteristics of the ITSE, ITSE+ISCO, ITAE, and ITAE+ISCO for FA, DFA, GA, PSO, DE, and ABC algorithms.**Table 3** Optimum values of the FOPID gains obtained via minimizing J_1 and J_2 using different evolutionary algorithms

Algorithms	$K_p(J_1)$	$K_I(J_1)$	$K_D(J_1)$	$\lambda(J_1)$	$\mu(J_1)$	$K_p(J_2)$	$K_I(J_2)$	$K_D(J_2)$	$\lambda(J_2)$	$\mu(J_2)$
GA	9.9988	4.8576	0.6664	0.1938	0.9693	0.11	0.5442	0.1	0.1219	0.01
PSO	10	4.9148	0.6160	0.1582	1	0.1	0.553	0.1	0.1253	0.1514
ABC	9.9039	5.3354	0.6767	0.1936	0.9783	0.13	0.4813	0.12	0.1133	0.07826
DE	10	5.1650	0.6003	0.1622	1	0.1	0.4790	0.1	0.1066	0.011
FA	10	5.3611	0.6267	0.1782	1	0.1	0.463	0.1	0.096	0.01
DFA	10	5.4918	0.6005	0.1709	1	0.1	0.4778	0.101	0.1058	0.01

Table 4 Optimum values of the FOPID gains obtained via minimizing J_3 and J_4 using different evolutionary algorithms

Algorithms	$K_p(J_3)$	$K_I(J_3)$	$K_D(J_3)$	$\lambda(J_3)$	$\mu(J_3)$	$K_p(J_4)$	$K_I(J_4)$	$K_D(J_4)$	$\lambda(J_4)$	$\mu(J_4)$
GA	9.9784	0.27	1.7581	0.2516	0.046	0.11	1.786	0.859	0.799	0.01
PSO	9.46	2.034	0.248	0.01	0.981	0.13	0.745	0.1	0.01	0.055
ABC	9.879	1.85	0.302	0.01	0.929	0.748	0.106	1.127	0.121	0.0101
DE	10	0.106	0.184	0.011	1	0.11	1.79	0.7157	0.7148	0.012
FA	10	1.26	0.238	0.01	1	0.103	0.743	0.12	0.01	0.0604
DFA	10	1.45	0.239	0.01	0.987	0.1	0.453	0.429	0.01	0.013

Fig. 7 Step responses of the DC motor drive using ITSE, ITSE+ISCO, ITAE, and ITAE+ISCO based FOPID controllers.

Fig. 8 Control signals of the DC motor drive using ITSE, ITSE+ISCO, ITAE, and ITAE+ISCO based FOPID controllers.

Table 5 Optimum time indices parameters and objective function for J_1 and J_2 using different evolutionary algorithms

Algorithms	$M_P(J_1)$	$T_r(J_1)$	$T_s(J_1)$	$O_f(J_1)$	$M_P(J_2)$	$T_r(J_2)$	$T_s(J_2)$	$O_f(J_2)$
GA	17.1405	0.0276	0.2261	0.07171	2.1929	1.0553	3.07935	12.3916
PSO	16.3470	0.0268	0.2241	0.070539	1.7733	1.077	1.6497	12.61777
ABC	17.3137	0.0266	0.2254	0.0715	1.5222	1.1887	1.8352	12.3672
DE	16.6012	0.0272	0.2224	0.070508	1.6301	1.1785	1.8089	12.2960
FA	17.2463	0.02658	0.2231	0.070717	1.3487	1.2211	1.899	12.30451
DFA	16.9113	0.0271	0.2212	0.070515	1.61	1.1815	1.8151	12.2959

Table 6 Optimum time indices parameters and objective function for J_3 and J_4 using different evolutionary algorithms

Algorithms	$M_P(J_3)$	$T_r(J_3)$	$T_s(J_3)$	$O_f(J_3)$	$M_P(J_4)$	$T_r(J_4)$	$T_s(J_4)$	$O_f(J_4)$
GA	19.594	0.0518	0.2699	0.2102	24.168	0.371	2.089	40.35119
PSO	6.829	0.0432	0.1759	0.11416	0.018	0.919	1.665	21.1226
ABC	8.894	0.0417	0.1835	0.1182	0.01	0.399	0.738	41.535
DE	1.389	0.0496	0.1605	0.1162	24.6	0.3734	2.007	40.3742
FA	5.044	0.0432	0.175	0.148627	0.0146	0.9223	1.67	21.1145
DFA	6.3251	0.0435	0.1747	0.1116	0.002	0.8822	1.608	20.95

Table 7 Optimum objective function for J_1 and J_2 using different evolutionary algorithms in 10 runs.

Algorithms	Objective Function(J_1)			Objective Function(J_2)		
	Best	Worst	Average	Best	Worst	Average
GA	0.07171	0.07868	0.07448	12.3916	13.0162	12.6265
PSO	0.070539	0.071028	0.070755	12.6177	14.2236	13.2275
ABC	0.0715	0.07766	0.073422	12.36722	14.77907	13.17398
DE	0.070508	0.072756	0.070738	12.29603	12.83579	12.59186
FA	0.070717	0.07132	0.070976	12.30451	12.59956	12.411
DFA	0.070515	0.070686	0.070539	12.29597	12.30685	12.29968

As a case study for the validation of our approach, a fractional order PID controller has been implemented for the speed regulation of a chopper fed DC motor drive. The gains of the proposed FOPID controller are optimized successfully using the DFA algorithm. Fitness functions based on the ITSE, ITSE+ISCO, ITAE, and ITAE+ISCO time indices are used for designing the controller. The simulation studies clearly show the performance and superiority of the proposed FOPID controller based on the DFA approach over the FA, GA, PSO, ABC and DE algorithms. Besides, the FOPID controllers based on the ITSE+ISCO and ITAE+ISCO fitness functions provide lower intensity control signals than those based on ITSE and ITAE fitness functions.

In any case, it is important to remark that the optimization problem is very dependent on the particular fitness or objective function to choose and the selection of the control parameters. The fact that an optimization algorithm provides good results for a specific objective function does not guarantee the same success when applied to a different objective function. In our case, we can conclude that: 1) even if for some specific objective functions the performance of the FDA approach is better than for others, the differences are very slight; and 2) for every particular objective function consid-

ered, the FDA algorithm outperforms in general terms the rest of algorithms used for comparison.

In future works the authors will deal with the implementation of other dynamic FA variants and will compare their performances with that of the DFA approach in order to discuss on their limitations.

6 Compliance with Ethical Standards

Funding: This study was not funded.

Conflict of Interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Aghababa MP (2016) Optimal design of fractional-order PID controller for five bar linkage robot using a new particle swarm optimization algorithm. *Soft Comput* 20:4055–4067
- Chandra Saikia L, Kant Sahu S (2013) Automatic generation control of a combined cycle gas turbine plant

Table 8 Optimum objective function for J_3 and J_4 using different evolutionary algorithms in 10 runs.

Algorithms	Objective Function (J_3)			Objective Function (J_4)		
	Best	Worst	Average	Best	Worst	Average
GA	0.2102	0.2464	0.229	40.35119	42.01539	40.67022
PSO	0.114161	0.192879	0.145293	21.12269	40.41904	24.2318
ABC	0.118264	0.204557	0.163329	41.53528	47.14024	43.41272
DE	0.116207	0.193251	0.129386	40.3742	46.26457	41.9746
FA	0.148627	0.15072	0.149674	21.1145	21.36081	21.2058
DFA	0.11168	0.112188	0.111979	20.95295	21.10295	21.01903

Fig. 9 Best parameter values for ITSE, ITSE+ISCO, ITAE, and ITAE+ISCO during the evolution process.

with classical controllers using firefly algorithm. *Int J Electr Power Energy Syst* 53:27–33

Cheung NJ, Ding X-M, Shen H-B (2014) Adaptive firefly algorithm: parameter analysis and its application. PLoS One 9(11):e112634. doi:10.1371/journal.pone.0112634

Fister I, Fister IJ, Yang XS, Bret J (2013) A comprehensive review of firefly algorithms. *Swarm Evol Comput* 13:34–46

Fister IJ, Perc M, Kamal SM, Fister I (2015) A review of chaos-based firefly algorithms: Perspectives and research challenges. *Appl Math Comput* 252:155–165

Fister I, Yang XS, Fister D, Fister IJ (2014) Firefly Algorithm: a brief review of the expanding literature. In: Yang XS (ed) *Cuckoo search and firefly algorithm*, Springer, New York, USA, pp 347–360

Gandomi AH, Yang X-S, Talatahari S, Alavi AH (2013) Firefly algorithm with chaos. Commun

Nonlinear Sci Numer Simul 18(1):89–98

Ghasemi S, Tabesh A, Askari-Marnani J (2014) Application of fractional calculus theory to robust controller design for wind turbine generators. *IEEE Trans on Energy Conversion* 29:780–787

Gopakumar K (2010) Power electronics and electrical drives video lectures 1–25. Center for electronics and technology, IISC, Bangalore.

Li H, Luo Y, Chen Y (2010) A fractional order proportional and derivative (FOPD) motion controller: Tuning rule and experiments. *IEEE Trans Control Syst Technol* 18:516–520

Liu L, Pan F, Xue D (2014) Fractional-order optimal fuzzy control for network delay. *Optik - International Journal for Light and Electron Optics* 125:7020–7024

Maamar B, Rachid M (2014) IMC-PID-fractional-order-filter controllers design for integer order systems. *ISA Trans* 53:1620–1628

- Mann HB, Whitney DR (1947) On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics* 18(1):50–60
- Martín F, Monje CA, Moreno L, Balaguer C (2015) DE-based tuning of $PI^\lambda D^\mu$ controllers. *ISA Trans* 59:398–407
- Monje CA, Chen YQ, Vinagre BM, Xue D, Feliu V (2010) Fractional-order systems and controls. Fundamentals and applications, Springer
- Monje CA, Vinagre BM, Feliu V, Chen YQ (2008) Tuning and auto-tuning of fractional order controllers for industry applications. *Control Eng Pract* 16:798–812
- Naidu K, Mokhlis H, Bakar AHA, Terzija V, Illias HA (2014) Application of firefly algorithm with online wavelet filter in automatic generation control of an interconnected reheat thermal power system. *Int J Electr Power Energy Syst* 63:401–413
- Pan I, Das S (2013) Intelligent fractional order systems and control. Studies in computational intelligence, Springer
- Pandey SK, Mohanty SR, Kishor N, Catalão JPS (2014) Frequency regulation in hybrid power systems using particle swarm optimization and linear matrix inequalities based robust controller design. *Int J Elec Power* 63:887–900
- Perng JW, Chen GY, Hsu YW (2016) FOPID controller optimization based on SIWPSO-RBFNN algorithm for fractional-order time delay systems. *Soft Comput* 1–14. doi:10.1007/s00500-016-2050-0
- Podlubny I (1999) Fractional Differential Equations. Academic Press, San Diego, CA
- Rajasekhar A, Jatoth RK, Abraham A (2014) Design of intelligent $PID/PI^\lambda D^\mu$ speed controller for chopper fed DC motor drive using opposition based artificial bee colony algorithm. *Eng Appl Artif Intel* 29:13–32
- Sarker RA, Elsayed SM, Ray T (2014) Differential evolution with dynamic parameters selection for optimization problems. *IEEE Trans Evol Comput* 18:689–707
- Sarkar BK, Mandal P, Saha R, Mookherjee S, Sanyal D (2013) GA-optimized feedforward-PID tracking control for a rugged electrohydraulic system design. *ISA Trans* 52:583–861
- Senthilnath J, Omkar S, Mani V (2011) Clustering using firefly algorithm: performance study. *Swarm Evol Comput* 1:164–171
- Sharma R, Gaur P, Mittal AP (2016) Design of two-layered fractional order fuzzy logic controllers applied to robotic manipulator with variable payload. *Appl Soft Comput* 47:565–576
- Wang D, Tan D, Liu L (2017) Particle swarm optimization algorithm: an overview. *Soft Comput* 1–22. doi:10.1007/s00500-016-2474-6
- Wang H, Zhou X, Sun H, Yu X, Zhao J, Zhang H, Cui Laizhong (2016) Firefly algorithm with adaptive control parameters. *Soft Comput* 1–12. doi:10.1007/s00500-016-2104-3
- Yang XS (2009) Firefly algorithms for multimodal optimization. In: Proceeding of the conference on stochastic algorithms, foundations and applications, Springer, pp 169–178
- Yang XS, He X (2013) Firefly Algorithm: recent advances and applications. *Int J Swarm Intelligence* 1:36–60
- Yang XS, Hosseini SSS, Gandomi AH (2011) Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Appl Soft Comput* 12:1180–1186
- Yu S, Su S, Lu Q, Huang L (2014) A novel wise step strategy for firefly algorithm. *Int J Comput Math* 91(12):2507–2513