sensors

MDPI

Article

A Fast Firefly Algorithm for Function Optimization: Application to the Control of BLDC Motor

Smail Bazi [1,2], Redha Benzid [3], Yakoub Bazi [4,*] and Mohamd Mahmoud Al Rahhal [5]

[1] Department of Science and Technology, University Center of Abdelhafid Bousouf Mila, Mila 43000, Algeria; s.bazi@centre-univ-mila.dz
[2] Research Laboratory of Electromagnetic Induction and Propulsion Systems, Department of Electrical Engineering, University of Batna 2, Batna 05000, Algeria
[3] Laboratoire d'Automatique Avancée et d'Analyse des Systèmes: LAAAS, Department of Electronics, University of Batna 2, Batna 05000, Algeria; r.benzid@univ-batna2.dz
[4] Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[5] Applied Computer Science Department, College of Applied Computer Science, King Saud University, Riyadh 11543, Saudi Arabia; mmalrahhal@ksu.edu.sa
[*] Correspondence: ybazi@ksu.edu.sa; Tel.: +966-101469629

Abstract: Firefly Algorithm (FA) is a recent swarm intelligence first introduced by X.S. Yang in 2008. It has been widely used to solve several optimization problems. Since then, many research works were elaborated presenting modified versions intending to improve performances of the standard one. Consequently, this article aims to present an accelerated variant compared to the original Algorithm. Through the resolving of some benchmark functions to reach optimal solution, obtained results demonstrate the superiority of the suggested alternative, so-called Fast Firefly Algorithm (FFA), when faced with those of the standard FA in term of convergence fastness to the global solution according to an almost similar precision. Additionally, a successful application for the control of a brushless direct current electric motor (BLDC) motor by optimization of the Proportional Integral (PI) regulator parameters is given. These parameters are optimized by the FFA, FA, GA, PSO and ABC algorithms using the IAE, ISE, ITAE and ISTE performance criteria.

Keywords: Fast Firefly Algorithm; optimization; benchmark functions; BLDC motor; PI controller; nature inspired algorithm

check for updates

Citation: Bazi, S.; Benzid, R.; Bazi, Y.; Rahhal, M.M.A. A Fast Firefly Algorithm for Function Optimization: Application to the Control of BLDC Motor. *Sensors* **2021**, *21*, 5267. https://doi.org/10.3390/s21165267

Academic Editor: Andrey V. Savkin

Received: 3 June 2021
Accepted: 1 August 2021
Published: 4 August 2021

1. Introduction

Optimization is one of the methods that seek to solve complex problems in engineering or other fields. The objective of optimization is to locate the optimal value of a cost function in a well-defined research space under different constraints [1]. Among the techniques used, in optimization, are those of swarm intelligence algorithms which are nature-inspired algorithms, these optimization techniques have spread over the past two decades [2]. Thus, the significant performance of swarm intelligence algorithms compared to other conventional optimization methods motivates researchers and are still to be attractive to exploit them in several complex optimization problems at different fields [3]. These algorithms operate on two different search properties: exploitation and exploration, where exploration scans the entire search space and prevents the algorithm from falling into the local optima, while exploitation ensures the efficiency of the search and the convergence of the algorithm towards the optimal solution [4]. Since the appearance of Genetic Algorithm [5], many optimization algorithms have been proposed such as Ant Colony Optimization [6,7], Artificial Bee Colony (ABC) [8,9], Particle Swarm Optimization (PSO) [10,11] Modified Particle Swarm Optimization [12] Cuckoo Search (CS) [13,14], Bat Algorithm (BA) [15,16], Gray Wolf Optimizer (GWO) [17,18], Firefly Algorithm [19,20] and so on.

Recently, Firefly Algorithm is one of the famous swarm intelligence algorithms for optimization problems that have been introduced in 2008. Due to its ease of design, implementation and flexibility in nature, it has become popular in the field of optimization and has been widely applied to diverse engineering optimization problems such as in [21,22]. Despite all these advantages, it has drawbacks such as the problem of local minima and it was unable to guarantee a balance between exploration and exploitation [2,23]. Therefore, several improvement algorithms have been proposed to overcome such drawbacks which make them more widely applied successfully in engineering like optimizing Proportional Integral Derivative (PID) parameters in machine control [24–28].

The PID controller and its variants are mainly used in control process to have a better dynamic performance of the controlled systems. Therefore, the optimal value of the corrector parameters is needed. In this context, the choice of controller gains has become an optimization problem [29]. FA and rival algorithms were successfully applied in the optimization of the parameters of PID mainly in electrical engineering and other fields [30]. One of the prominent applications in electrical engineering is the control of BLDC motor driven by a tuned and optimized PID. However, a BLDC motor is developed on the basis of Brushed DC motor and it is one of the special electrical synchronous motor. It is driven by DC voltage, but current commutation is obtained by solid-state switches. The commutation time is fixed by the rotor position which is detected by hall sensor position [31].

It is noticeable that BLDC motor has the advantages that are: high efficiency, long operating time, low noise, small size and well speed–torque features. In general, it has a great improvement in automotive, aerospace and industry of engineering and so on. Therefore, its use has been exposed to many types of load disturbances. Conventional control methods cannot resist these alterations and lose their precision. Thus, it was necessary to implement advanced control techniques to solve this problem, especially those based on the artificial intelligence, such as: fuzzy control [32,33], neural control [34,35], Genetic Algorithm (GA) control [36,37], PSO control [38], BAT control [31] and recently, FA control and Improved Firefly Algorithm (IFA) or Modified Firefly Algorithm (MFA) [24–28]. These methods are based essentially on the optimization of the PID corrector parameters and its derivatives to obtain optimal performance.

In this paper, we propose an improved version of the FA for function optimization by reducing the search space. We apply this method to several benchmark problems and also to the design of a controller for BLDC motor. The paper contains two experimental parts, the first concerns the search for the global optimum of several benchmark functions according to the FA and FFA algorithms and then a comparative study is carried out. In order to consolidate its efficiency, a second application of PI parameters' optimization for the BLDC motor control is achieved through simulation in the MATLAB platform. This application used the FFA, FA, GA, PSO and ABC algorithms according to the IAE, ISE, ITAE and ISTE performance criteria, to test the competitiveness of the FFA algorithm. Finally, by comparison of the obtained results, it is found that the performances of the FFA are better than those of the other algorithms and it can be concluded that this new algorithm can be a valid concurrent meta-heuristic optimization method.

The paper is organized as follows. Section 2 introduces the mathematical background of the standard FA and the suggested FFA. In Section 3, the two algorithms are compared through optimum finding of several standard test functions. The mathematical model of BLDC motor and the PI controller with description of the experimental results are presented in Section 4. Finally, drawn conclusion summarizing the achieved work is given in Section 5.

2. Firefly Algorithm and Proposed Fast Firefly Algorithm

2.1. Standard Firefly Algorithm

Firefly Algorithm is inspired by the natural behavior of fireflies by using their self-luminosity to get closer to each other in the dark. Three assumptions have been suggested by Yang to clarify the behavior of fireflies [19,20]. Firstly, all fireflies are unisex. Thus, each firefly can be attracted to other fireflies regardless of gender. Secondly, the attractiveness is linked to the intensity which is a function of the distance between the firefly concerned and the other fireflies. The attractiveness decreases as the distance increases. Finally, the luminosity or the luminous intensity of a firefly is given by the value of the cost function of the problem posed. Mathematically, the FA algorithm can be given by the following equations [19].

The light intensity of a firefly is given by Equation (1).

$$I(r) = I_0 \exp(-\gamma . r_{ij}) \tag{1}$$

where: γ is the absorption coefficient and (I_0) is the initial value at (r = 0).

The attractiveness is expressed by Equation (2) where β_0 is the initial value at (r = 0):

$$\beta = \beta_0 \exp(-\gamma . r_{ij}^{m}) \, , m \geq 1 \tag{2}$$

Equation (3) evaluates the distance between two fireflies i and j, at positions x_i and x_{j}, respectively, and can be defined as Cartesian distance. Where x_{ik} is the kth element of the spatial coordinate x_j of the ith firefly and D denotes the dimensionality of the problem [19].

$$r_{ij} = \left| r_i - r_j \right| = \sqrt{\sum_{k=1}^{D} (x_{ik} - x_{jk})^2} \tag{3}$$

The motion equation of the i^{th} firefly to *the* j^{th} one is determined by Equation (4).

$$x_i(t+1) = x_i(t) + \beta(x_j(t) - x_i(t)) + \alpha(rand - 0.5) \tag{4}$$

where $x_i(t$ + 1) is the position of firefly i at iteration t +1 displacement. As it can be seen, the first part of the right side of Equation (4) is the position of firefly i at iteration t, the second term is relative to the attractiveness and the last one is randomization (blind flying if there is no light) where α is the random walk parameter $\alpha \in [0,1)$, [19].

The FA Algorithm 1 is given as follows [19]:

Algorithm 1. Firefly Algorithm

```
Initialization of the parameters of FA (Population size, $\alpha$, $\beta_o$, $\gamma$ and the number of iterations).
The Light intensity is defined by the cost function f($x_i$) where $x_i$(i = 1, . . . ,n).
While (iter < Max Generation).
        for i = 1:n (all n fireflies)
          for j = 1:n (all n fireflies)
            if (f($x_i$) < f($x_j$)), move firefly i towards j,
            end if.
            Update attractiveness $\beta$ with distance r.
            Evaluate new solution and update f($x_i$) in the same way as (4).
          end for j
        end for i
rank the solutions and find the best global optimal.
end while.
Show the results.
```

2.2. Fast Firefly Algorithm

It is worth noting that the original algorithm of Xin-She Yang performs (Max generation $n.n$) tests. However, in the proposed version, ($K.n$) tests only are performed, where K is an integer. It means that the conventional one is hugely time consuming when compared to the suggested one. The proposed Algorithm 2 is summarized as follows:

Algorithm 2. Fast Firefly Algorithm

```
While (iter < Max Generation)
    for k = 1:K.n (all n fireflies) // Here it is the first modification
        i = rand(n) // Here it is the second modification
        j = rand(n) // Here it is the third modification
        if (f(xi) < f(xj)), move firefly i towards j,
        end if.
        Update attractiveness $\beta$ with distance r.
        Evaluate new solution and update f(xi) in the same way as Equation (4).
        Modify the new position obtained by Equation (4) according to Equation (5).
        end for k
rank the solutions and find the best global optimal.
end while.
Show the results.
```

As above mentioned, the new position obtained by Equation (4) is modified according to Equation (5):

$$x_i(t+1) = \alpha.x_i(t) \tag{5}$$

It should be noted that the values of α and γ are given empirically in the original version according to each test function, β_0 is equal to unity. However, on the other hand, the α in FFA is taken equal to:

$$\alpha = \exp(-10.iter/(iter+100)) \tag{6}$$

where the convergence is reached easily and γ still chosen equal to 1. The randomization parameter α is reduced exponentially from a maximum value to a minimum value according to successive iterations instead of keeping it constant; with this injected artifice, we can maintain the research balance between the exploitation and the exploration of the proposed algorithm and it can give better results than its rival FA [4].

In the original version of the FA, the technique of updating the motion of fireflies can be improved to be more faster. Thus, it is beneficial for each firefly in the swarm to find a promising region by reorienting its motion in order to easily reach the overall optimum. Consequently, the updated term is redirected to have a better exploration and exploitation of the algorithm and the speed of its convergence is, thus, guaranteed [1,39].

The essence of the proposed method is the reduction of the search space (exploration) while keeping the search efficiency satisfactory to reach the optimal solution. It means that ($K.n$) evaluated tests were found clearly sufficient to obtain the optimal solution for the large number of benchmark functions and other applications [40].

3. Simulation Results and Analysis

3.1. Benchmark Functions

Standards' functions are essential to prove and compare the characteristics of optimization algorithms. The most terms of evaluation are: The convergence speed and the precision. Hence, 12 different test functions are used to compare the performance of the original algorithm FA and the proposed one FFA according to the previously mentioned evaluation terms. The used test functions are listed in Table 1, highlighting the variables, ranges and values of the global optimum to reach [41,42].

Table 1. Benchmark functions.

Function	Name	Expression	Range	f(x*)
F1	Schaffer N.1	$F1(x,y) = \frac{\sin^2\left(x^2+y^2\right)^2-0.5}{\left(1+0.001\left(x^2+y^2\right)\right)^2}$	[−100,100]	0 for x* = (0,0)
F2	Matyas	$F2(x,y) = 0.26\left(x^2+y^2\right) - 0.48xy$	[−10,10]	0 for x* = (0,0)
F3	BohachevskyN1	$F3(x,y) = x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7$	[−100,100]	0 for x* = (0,0)
F4	Xin-SheYang N.2	$F4(x) = \sum_{i=1}^{D} \lvert x_i \rvert \exp\left(\sum_{i}^{D} \sin\left(x_i^2\right)\right)$	[−2π,2π]	0 for x* = (0, ... ,0)
F5	Zakharov	$F5(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^4$	[−5,10]	0 for x* = (0, ... ,0)
F6	Ackley	$F6(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2})-$ $\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i) + 20 + \exp(1)$	[−32,32]	0 for x* = (0, ... ,0)
F7	Powell	$F7(x) = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	[−1,1]	0 for x* = (0, ... ,0)
F8	Rastrigin	$F8(x) = 10D + \sum_{i=1}^{D}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	[−5.12,5.12]	0 for x* = (0, ... ,0)
F9	Schewel223	$F9(x) = \sum_{i=1}^{D} x_i^{10}$	[−10,10]	0 for x* = (0,0)
F10	Alpinen1	$F10(x) = \sum_{i=1}^{D} \lvert x_i\sin(x_i) + 0.1x_i \rvert$	[0,10]	0 for x* = (0,0)
F11	Grienwak	$F11(x) = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	[−600,600]	0 for x* = (0, ... ,0)
F12	Brown	$F12(x) = \sum_{i=1}^{D-1} \left(x_i^2\right)^{\left(x_{i+1}^2+1\right)} + \left(x_{i+1}^2\right)^{\left(x_i^2+1\right)}$	[−1,4]	0 for x* = (0, ... ,0)
F13	Sphere	$F13(x) = \sum_{i=1}^{D} x_i^2$	[−5.12,5.12]	0 for x* = (0, ... ,0)
F14	Salomon	$F14(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2}$	[−100,100]	0 for x* = (0, ... ,0)
F15	Three Hump Camel	$F15(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	[−5,5]	0 for x* = (0, ... ,0)

3.2. Parameter Settings

The parameter settings of FA and FFA are showed in Table 2.

Table 2. Parameter settings of FA and FFA.

Symbol	Quantity	Value
N	Population size	30
Iter	Number of iterations	1000
α	Randomization parameter	[0,1]
β_0	Attractiveness	1
γ	Absorption coefficient	[0,1]

3.3. Functions' Experimental Results

The two algorithms are applied to minimize a set of test functions of dimensions 2D, 10D, 20D and 30D, respectively. The experimental environment is the MATLAB R2017a software, the CPU is an E5700@3.00 GHZ, the RAM is of size 6 GB. To compare their performance, minimum, mean, standard deviation and the computational time are taken over 10 runs. For each function, the two algorithms operate independently. The results of the optimization are summarized in Table 3.

Table 3. Comparative simulation results of FA and FFA for the 12 benchmark test functions.

Function	Algorithm	Dim. D	Theoretical Optimal Value	Minimum Value	Computational Time (s)	Average Speed up Ratio of 10 Runs	Std	Mean
F1	FA	2	0	1.6542×10^{-13}	86.483887	12.0295:1	2.9426×10^{-12}	3.4053×10^{-12}
	FFA			2.2204×10^{-16}	7.189327		1.1466×10^{-16}	3.5527×10^{-16}
F2	FA	2	0	5.8618×10^{-15}	110.617901	11.7869:1	2.2829×10^{-15}	4.2810×10^{-15}
	FFA			1.1794×10^{-22}	9.384792		5.2167×10^{-22}	3.4229×10^{-22}
F3	FA	2	0	8.0766×10^{-10}	114.399397	11.9121:1	2.2452×10^{-9}	2.2950×10^{-9}
	FFA			1.1102×10^{-16}	9.603642		1.0320×10^{-15}	1.4211×10^{-15}
F4	FA	10	0	5.6623×10^{-4}	165.318332	13.0661:1	1.3542×10^{-8}	5.6625×10^{-8}
	FFA			3.4134×10^{-11}	12.652510		2.4298×10^{-11}	3.6237×10^{-11}
F5	FA	10	0	1.9635×10^{-7}	102.111224	12.0521:1	1.1191×10^{-7}	3.1165×10^{-7}
	FFA			5.0686×10^{-22}	8.472478		9.9126×10^{-38}	5.0686×10^{-22}
F6	FA	10	0	0.0252	174.645121	12.6941:1	0.0061	0.0357
	FFA			7.5286×10^{-11}	13.758015		5.5179×10^{-11}	8.6060×10^{-11}
F7	FA	20	0	5.4225×10^{-8}	254.090816	11.9330:1	4.4293×10^{-8}	6.0622×10^{-8}
	FFA			6.0701×10^{-27}	21.293091		1.6065×10^{-25}	2.1326×10^{-25}
F8	FA	20	0	1.7397×10^{-9}	86.135286	11.9768:1	2.2366×10^{-9}	2.5533×10^{-9}
	FFA			7.1054×10^{-15}	7.191834		1.1235×10^{-14}	1.0658×10^{-14}
F9	FA	20	0	5.4774×10^{-26}	199.923385	12.2129:1	6.8916×10^{-26}	7.4821×10^{-26}
	FFA			2.9153×10^{-84}	16.369803		4.8193×10^{-100}	2.9153×10^{-84}
F10	FA	30	0	1.7988×10^{-4}	139.507606	12.2676:1	1.5871×10^{-5}	2.1766×10^{-4}
	FFA			5.6687×10^{-9}	11.372018		3.6465×10^{-10}	6.0921×10^{-9}
F11	FA	30	0	8.0295×10^{-6}	130.429223	17.9004:1	2.0834×10^{-7}	1.2770×10^{-6}
	FFA			3.3304×10^{-16}	7.286405		2.0572×10^{-16}	6.5503×10^{-16}
F12	FA	30	0	2.0832×10^{-4}	312.021582	16.2503:1	2.1256×10^{-5}	1.7312×10^{-4}
	FFA			3.5141×10^{-16}	19.200945		2.1964×10^{-17}	3.5897×10^{-16}

In terms of precision of convergence towards the global optimum, by visualizing the results in Table 3, it can be seen that the mean and the standard deviation of the reached optimum, after 10 runs for each test function, of FFA in all dimensions are better than of FA.

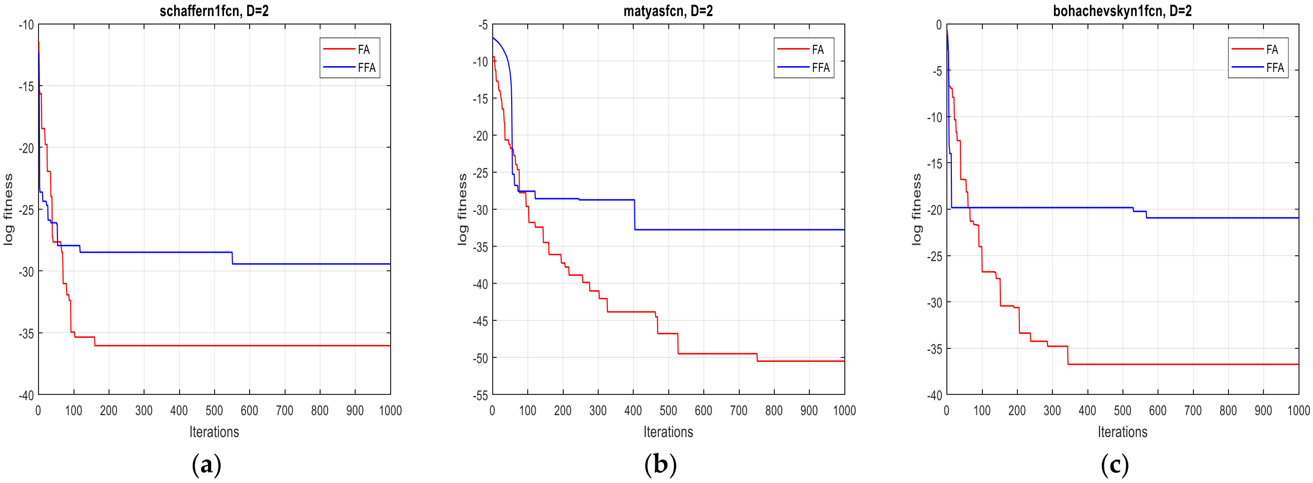
Concerning the convergence fastness to the global optimum, it can be clearly remarked, from extensive simulation tests, that the proposed method outperforms the original one and it is significantly faster (see Table 3). Accordingly, the average speed up ratio, when applying the two algorithms on the 12 test functions, is 12:1, which confirms the effectiveness of the suggested technique.

It is worthy to note that the speed up ratio is defined by:

$$SR = \frac{t_{FA}}{t_{FFA}} \tag{7}$$

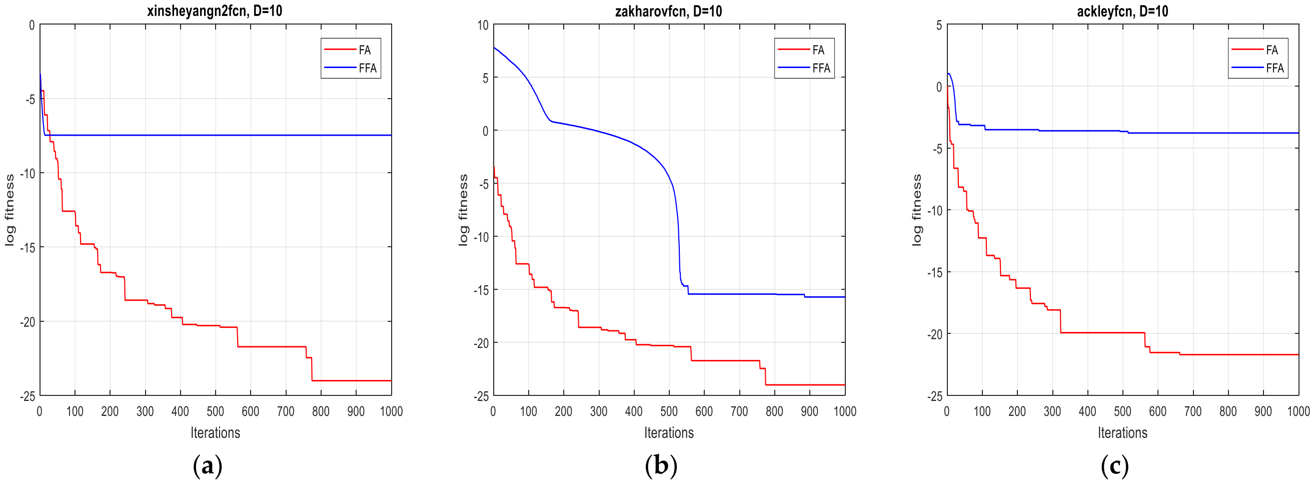
where t_{FA} is the execution time of the original algorithm FA, and the t_{FFA} is the execution time of the proposed one FFA.

As is shown in Figures 1–4 below, the proposed algorithm reaches all solutions of all test functions with high precision outperforming, accordingly, those obtained from the standard one.



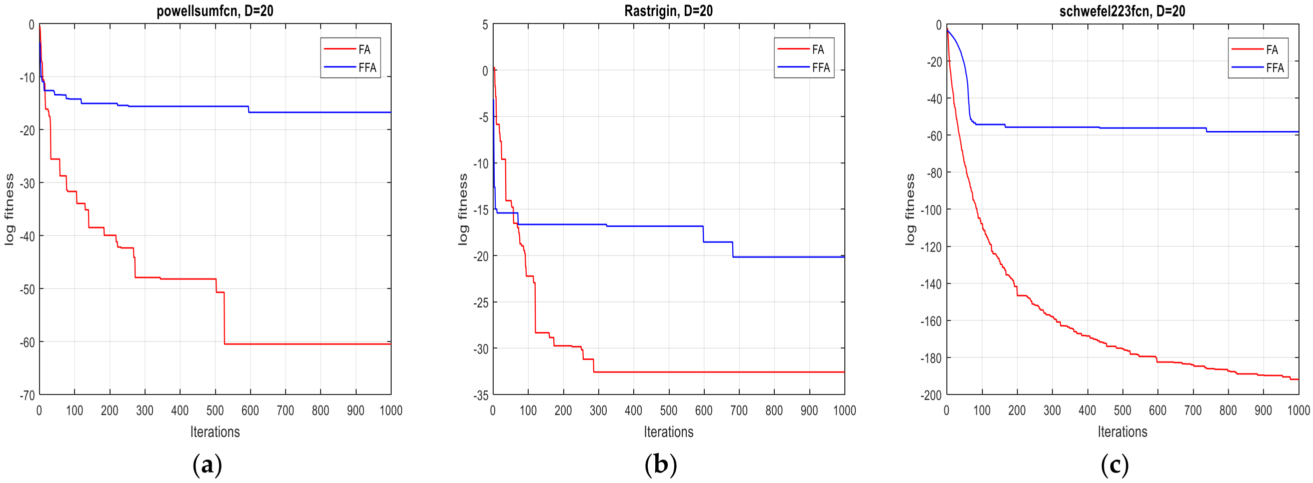
(**a**) (**b**) (**c**)

Figure 1. Convergence curves of FFA and FA for the functions: (**a**) F1; (**b**) F2 and (**c**) F3 (2D).



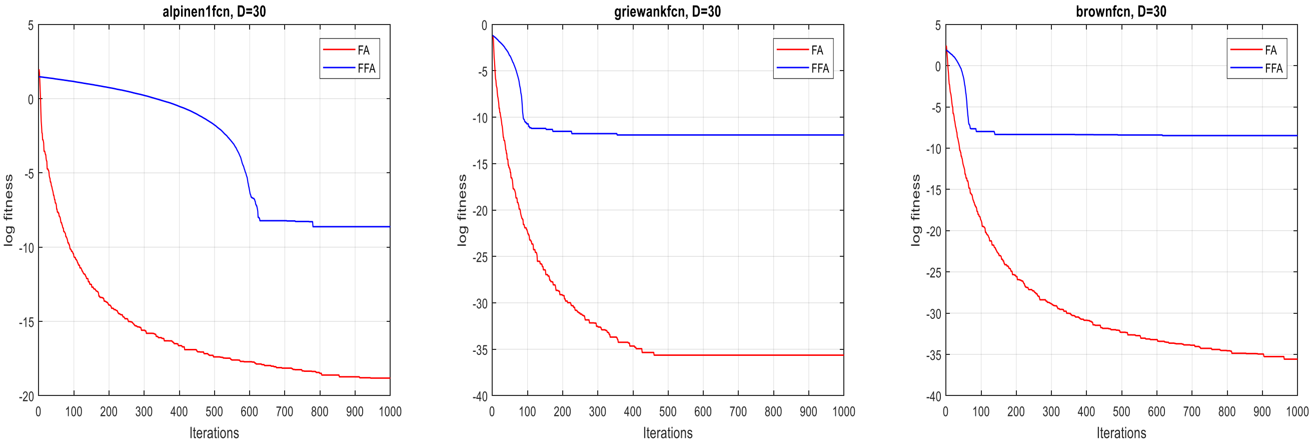
(**a**) (**b**) (**c**)

Figure 2. Convergence curves of FFA and FA for the functions: (**a**) F4; (**b**) F5 and (**c**) F6 (10D).



(**a**) (**b**) (**c**)

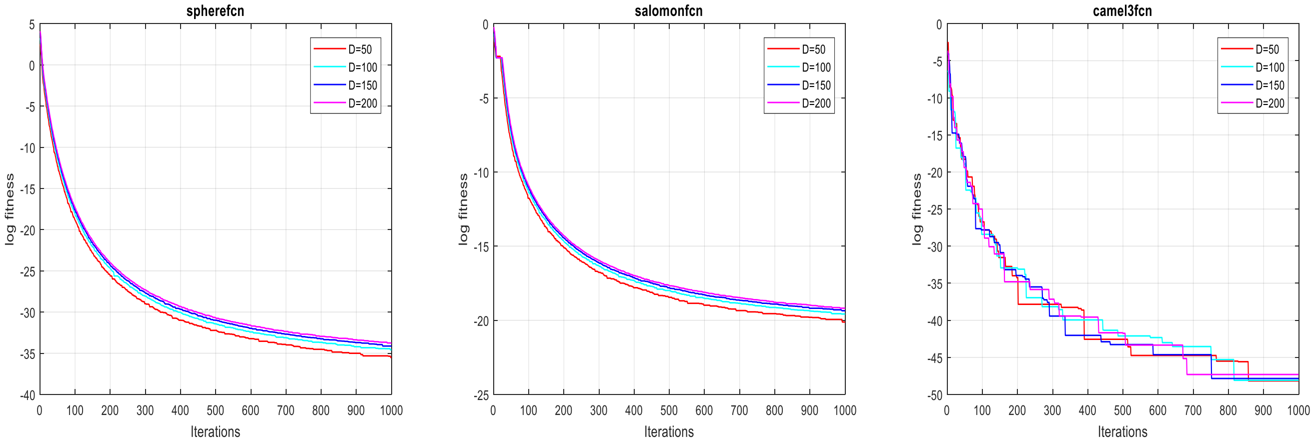
Figure 3. Convergence curves of FFA and FA for the functions: (**a**) F7; (**b**) F8 and (**c**) F9 (20D).



(**a**) (**b**) (**c**)

Figure 4. Convergence curves of FFA and FA for the functions: (**a**) F10; (**b**) F11 and (**c**) F12 (30D).

As can be seen from Table 3, the proposed algorithm is more unbiased (the statistical expected value of obtained cost function of FFA is more tending to the theoretical value than FA) and more consistent (the standard deviation of obtained cost function when applying FFA is more tending to 0 than the FA). The reported remarks hold for the twelve test functions as previously shown in Table 3 for dimensions 2D, 10D, 20D and 30D, respectively. For more convincing, robustness and stability of FFA in higher dimensions are evaluated by using the test functions (F13, F14 and F15) for dimensions 50D, 100D, 150D and 200D, respectively. Table 4 gives the results of these tests with a 10 times run for each test function. Finally, it can be concluded that the stability of FFA is not affected by increasing significantly dimensions (high precision remains obtained). The graphs of Figure 5 reflect these results.



(**a**) (**b**) (**c**)

Figure 5. Convergence curves of FFA for the functions: (**a**) F13; (**b**) F14 and (**c**) F15 on 50D, 100D, 150D and 200D.

Table 4. Stability of FFA in higher dimensions.

Function	Algorithm	Dim. D	Theoretical Optimal Value	Minimum Value	Std	Mean	Computational Time of 10 Runs (Seconds)	Iterations
F13	FFA	50	0	3.7950×10^{-16}	2.9222×10^{-17}	3.9330×10^{-16}	11.743543	1000
		100		8.9766×10^{-16}	5.3287×10^{-17}	9.2937×10^{-16}	12.630883	
		150		1.5083×10^{-15}	4.4389×10^{-17}	1.5223×10^{-15}	13.399011	
		200		2.1964×10^{-15}	8.7426×10^{-20}	2.1966×10^{-15}	14.696007	

Table 4. *Cont.*

Function	Algorithm	Dim. D	Theoretical Optimal Value	Minimum Value	Std	Mean	Computational Time of 10 Runs (Seconds)	Iterations
F14	FFA	50	0	1.9615×10^{-9}	5.8730×10^{-11}	2.0092×10^{-9}	12.619130	1000
		100		3.0618×10^{-9}	6.2137×10^{-11}	3.1208×10^{-9}	13.483702	
		150		3.9315×10^{-9}	1.7846×10^{-11}	3.9107×10^{-9}	14.451983	
		200		4.6921×10^{-9}	1.1344×10^{-11}	4.6959×10^{-9}	15.718014	
F15	FFA	50	0	1.1834×10^{-21}	7.7571×10^{-22}	2.0847×10^{-21}	8.676127	1000
		100		1.3134×10^{-21}	9.5957×10^{-22}	1.7686×10^{-21}	9.238064	
		150		1.6926×10^{-21}	4.3289×10^{-21}	3.9281×10^{-21}	10.033184	
		200		2.8146×10^{-21}	3.7922×10^{-21}	5.1335×10^{-21}	11.103394	

4. Application for the Control of Brushless DC Motor

4.1. Description

BLDC motor is a permanent magnet synchronous motor that has trapezoidal Back-EMF and an almost rectangular current. It uses position detectors and an inverter to control the armature currents. It becomes popular for industrial applications because of its high efficiency, reliability, noiseless operation, low maintenance and an optimized volume. BLDC motors are available in several different configurations, but three-phase is the most common type due to its high speed and low torque ripple [43].

The drive model of a BLDC motor is shown in Figure 6. It is divided into two blocks. The first one is the inverter and the second is the BLDC motor. Accordingly, the BLDC motor is powered by a six-switch inverter where, for each control step, two phases operate simultaneously while the third is eliminated. Note that the signals of the Hall Effect position sensor (Ha, Hb, Hc) shifted by 120°, electrically govern these switches by generation of the pulses (S1, . . . ,S2) at every 60° electrical angle [43–45].

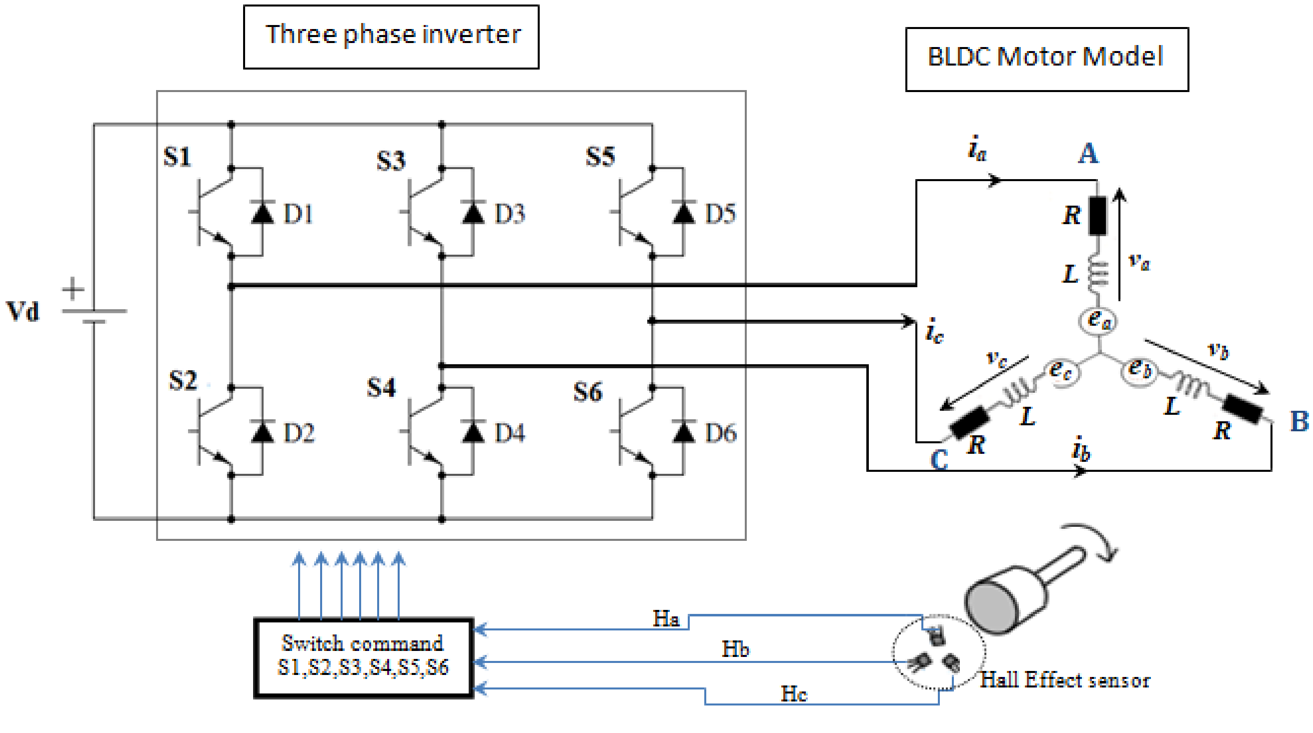


Figure 6. Drive model of a BLDC motor [43–47].

4.2. Mathematical Modeling of a BLDC Motor

By consideration of the symmetry of the phases, it is assumed that the three phases' resistances are identical as well as the inductances. Consequently, the equations describing the model of the equivalent circuit of the motor are [43–45]:

$$v_a = Ri_a + L\frac{d}{dt}i_a + e_a \tag{8}$$

$$v_b = Ri_b + L\frac{d}{dt}i_b + e_b \tag{9}$$

$$v_c = Ri_c + L\frac{d}{dt}i_c + e_c \tag{10}$$

Then, the line voltage equation can be obtained by subtraction of the phase voltage equation as:

$$v_{ab} = R(i_a - i_b) + L\frac{d}{dt}(i_a - i_b) + e_a - e_b \tag{11}$$

$$v_{bc} = R(i_b - i_c) + L\frac{d}{dt}(i_b - i_c) + e_b - e_c \tag{12}$$

$$v_{ca} = R(i_c - i_a) + L\frac{d}{dt}(i_c - i_a) + e_c - e_a \tag{13}$$

where:

R: resistance of a stator phase [Ω].
L: inductance of a stator phase [H].
v_a, v_b and v_c are the stator phase voltages [V].
v_{ab}, v_{bc} and v_{ca} are the stator phase to phase voltages [V].
i_a, i_b and i_c are stator phase currents [A]
e_a, e_b *and* e_c are motor Back-EMFs [V].

The relationship between phase currents is given by the equation:

$$i_a + i_b + i_c = 0 \tag{14}$$

Since each voltage is a linear combination of the other two voltages, two equations are sufficient. Using relation 14, Equations (11) and (12) become [44]:

$$v_{ab} = R(i_a - i_b) + L\frac{d}{dt}(i_a - i_b) + e_a - e_b \tag{15}$$

$$v_{bc} = R(i_a + 2i_b) + L\frac{d}{dt}(i_a + 2i_b) + e_b - e_c \tag{16}$$

The equation of mechanical part represents as follows:

$$T_e = k_f\omega_m + J\frac{d\omega_m}{dt} + T_L \tag{17}$$

$$\omega_m = \frac{d\theta_m}{dt} \tag{18}$$

where:

T_e and T_L are the electromagnetic torque and the load torque [Nm].
J is the rotor inertia, k_f is a friction constant and ωm is the rotor speed [rad/s].

The Back-EMF and electromagnetic torque can be expressed as:

$$e_a = k_e\omega_m F(\theta_e) \tag{19}$$

$$e_b = k_e\omega_m F\left((\theta_e - \frac{2\pi}{3})\right) \tag{20}$$

$$e_c = k_e \omega_m F\left((\theta_e - \frac{4\pi}{3}) \right) \tag{21}$$

where:

k_e is the Back-EMF's constant.

θ_e is equal to the rotor angle (θ_e= p. $\theta_m/2$), θ_m the mechanic angle and p the number of pole pairs. $F(\theta_e)$ is trapezoidal waveform of Back-EMFs.

Thus, the torque equation can be defined as:

$$T_e = \frac{(e_a i_a + e_b i_b + e_c i_c)}{\omega_m} = \frac{k_t}{2}\left[F(\theta_e) i_a + F\left(\theta_e - \frac{2\pi}{3}\right) i_b + F\left(\theta_e - \frac{4\pi}{3}\right) i_c \right] \tag{22}$$

kt: the torque constant.

Therefore, the function $F(\theta_e)$ is a function of rotor position, which gives the trapezoidal waveform of Back-EMF. One period of function can be written as:

$$F(\theta_e) = \begin{cases} 1 & 0 \leq \theta_e < \frac{2\pi}{3} \\ 1 - \frac{6}{\pi}\left(\theta_e - \frac{2\pi}{3}\right) & \frac{2\pi}{3} \leq \theta_e < \pi \\ -1 & \pi \leq \theta_e < \frac{5\pi}{3} \\ -1 + \frac{6}{\pi}\left(\theta_e - \frac{5\pi}{3}\right) & \frac{5\pi}{3} \leq \theta_e < 2\pi \end{cases} \tag{23}$$

For illustration, Figure 7 shows Back-EMF, Hall Effect sensor signal and the current in the three phases. In the trapezoidal motor Back-EMF induced in the stator has a trapezoidal shape and its phases must be supplied with quasi square currents for ripple free torque operation [44,46].

Finally, Equations (15)–(18) can be converted to a state space form. The resulting complete model is given as:

$$\begin{bmatrix} \frac{di_a}{dt} \\ \frac{di_b}{dt} \\ \frac{d\omega_m}{dt} \\ \frac{d\theta_m}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & 0 & 0 & 0 \\ 0 & -\frac{R}{L} & 0 & 0 \\ 0 & 0 & -\frac{k_f}{J} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ \omega_m \\ \theta_m \end{bmatrix} + \begin{bmatrix} \frac{2}{3L} & \frac{1}{3L} & 0 \\ -\frac{1}{3L} & \frac{1}{3L} & 0 \\ 0 & 0 & \frac{1}{J} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{ab} - e_{ab} \\ v_{bc} - e_{bc} \\ T_e - T_L \end{bmatrix} \tag{24}$$

$$i_c = -(i_a + i_b) \tag{25}$$

where: $e_{ab} = e_a - e_b$ and $e_{bc} = e_b - e_c$

4.3. Hall Effect Sensor and Transistor Switching Sequence

According to the angular position of the rotor evolution between 0° and 360°, the position produced by Hall Effect sensors is given which is described in Table 5 below.

Table 5. Switching sequence by using Hall Effect sensor signals.

Electrical Angle (°)	Sequence Number	Hall Sensors			Phase Current			Switch Closed	
		Ha	Hb	Hc	ia	ib	ic		
0–60	1	1	0	1	+	−	off	S1	S4
60–120	2	1	0	0	+	off	−	S1	S6
120–180	3	1	1	0	off	+	−	S3	S6
180–240	4	0	1	0	−	+	off	S3	S2
240–300	5	0	1	1	−	off	+	S5	S2
300–360	6	0	0	1	off	−	+	S5	S4

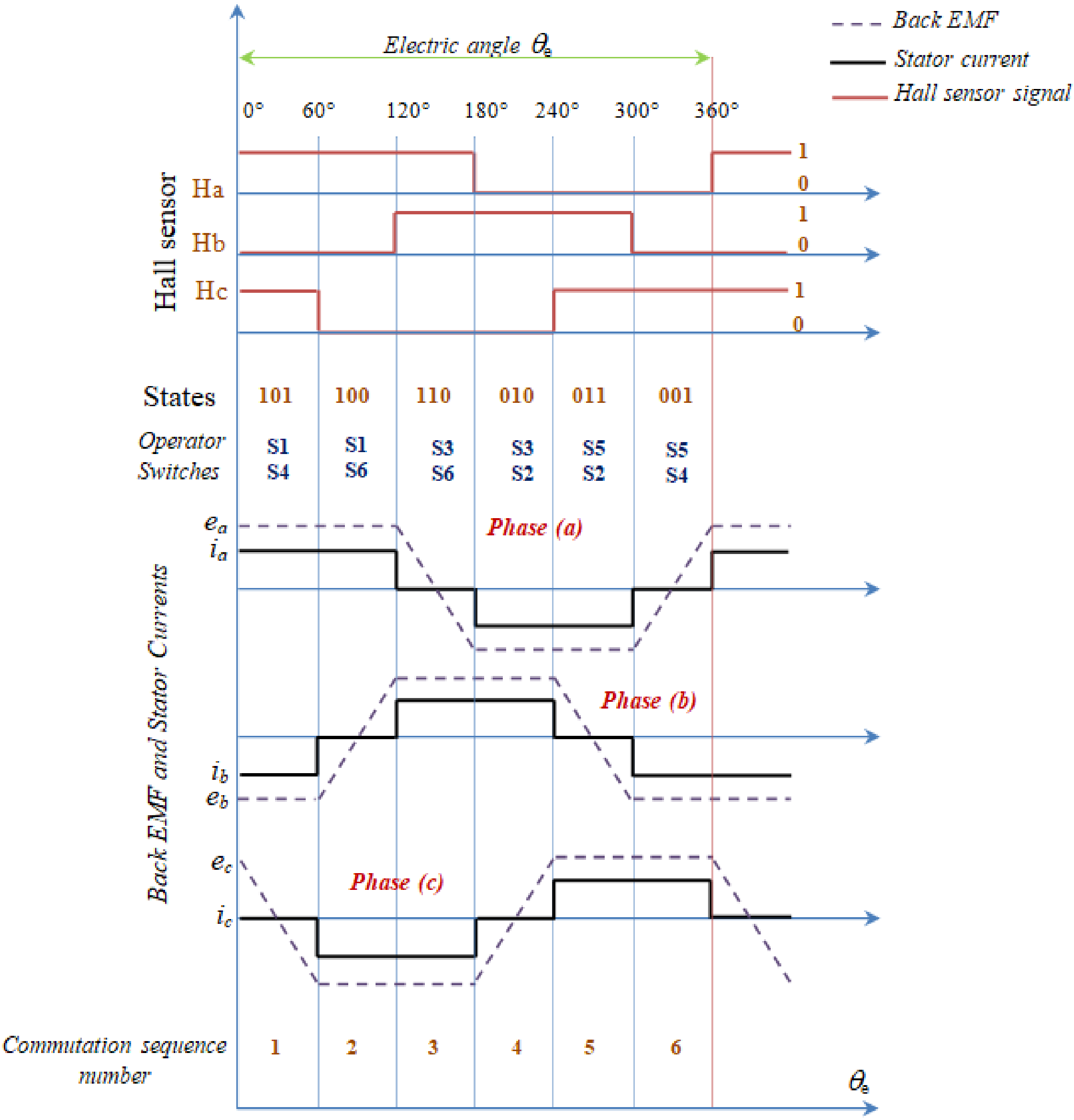


Figure 7. State of Hall sensor signals, Back-EMF and currents [47].

Each Hall Effect sensor operates during the passage of the poles based on the rising and falling edges. Thus, the rising front for the north pole and falling for the south pole. Accordingly, the sensor indicates 1 or 0, respectively. Following this switching logic of Hall Effect sensors, the switching sequence of the inverter is expressed in Table 5, where the switching sequence for shaft rotation is clockwise [45,47].

According to the circuit in Figure 6, the three-phase voltages are calculated with the following formulas [45]:

$$v_a = \frac{v_d}{2}(S1 - S2) \quad (26)$$

$$v_b = \frac{v_d}{2}(S3 - S4) \quad (27)$$

$$v_c = \frac{v_d}{2}(S5 - S6) \quad (28)$$

where v_d is the DC supply voltage.

4.4. Speed Control of Brushless DC Motor

The principle diagram for speed control of the three-phase BLDC motor is shown in Figure 8. At the regulator input, the reference speed is compared to the actual speed of the motor to generate a control voltage at its output.

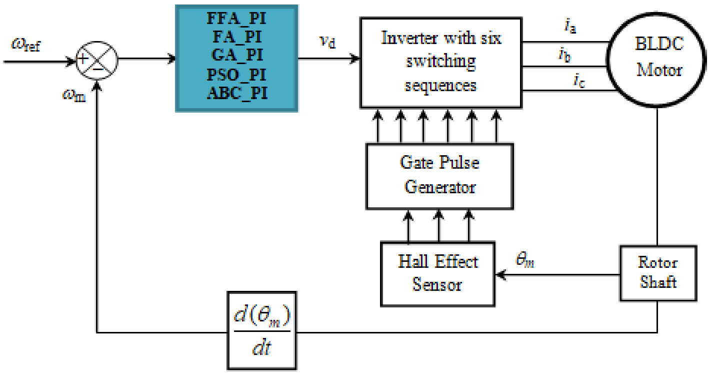


Figure 8. Speed control principle of BLDC motor.

The signals of the switching sequences are obtained from the position of the motor shaft. The motor stator is excited by the three-phase currents [45].

4.5. PI Controller

PI controller is a derivative of PID controller. It has been extensively used in industrial applications due to its simplicity, robustness, reliability and easy tuning gains in simple control [21].

The equation of the PI controller is specified by:

$$y(t) = k_p \varepsilon(t) + k_i \int_0^t \varepsilon(\tau) d\tau \tag{29}$$

The Laplace transfer function is:

$$C(S) = k_p + \frac{k_i}{s} \tag{30}$$

where:

k_p: proportional gain,
k_i: integral gain,
s: Laplace operator.

4.6. Simulation Results and Discussion

To ensure efficient performance of the system to be monitored, the performance criteria defined by Equations (31)–(34) are used. The objective functions are chosen for minimizing the time response characteristics due to the dependency of error on time [27]:

$$J_1 = IAE = \int_0^T \varepsilon(t) dt = \int_0^T (\omega_{ref} - \omega_m) dt \tag{31}$$

$$J_2 = ISE = \int_0^T \varepsilon^2(t) dtdt = \int_0^T (\omega_{ref} - \omega_m)^2 dt \tag{32}$$

$$J_3 = ITAE = \int_0^T t.\varepsilon(t)dt = \int_0^T t.(\omega_{ref} - \omega_m)dt \tag{33}$$

$$J_4 = ITSE = \int_0^T t.\varepsilon^2(t)dt = \int_0^T t.(\omega_{ref} - \omega_m)^2dt \tag{34}$$

The problem can be represented as:
Minimize J subjected to:

$$k_{pmin} \leq kp \leq k_{pmax}$$

$$k_{imin} \leq ki \leq k_{imax}$$

where ω_{ref} is the reference speed and ω_{m} is the actual one. Figure 9 shows PI controller block of the control. In this problem, the values of overshoot, rise time and stabilization time are controlled indirectly. These parameters are directly linked to the objective function so they are optimized implicitly [27].

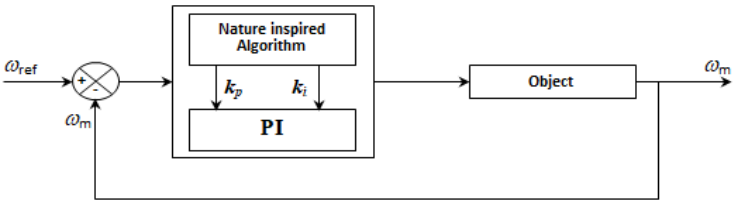


Figure 9. Scheme of PI controller parameters' optimization based on a nature-inspired algorithm.

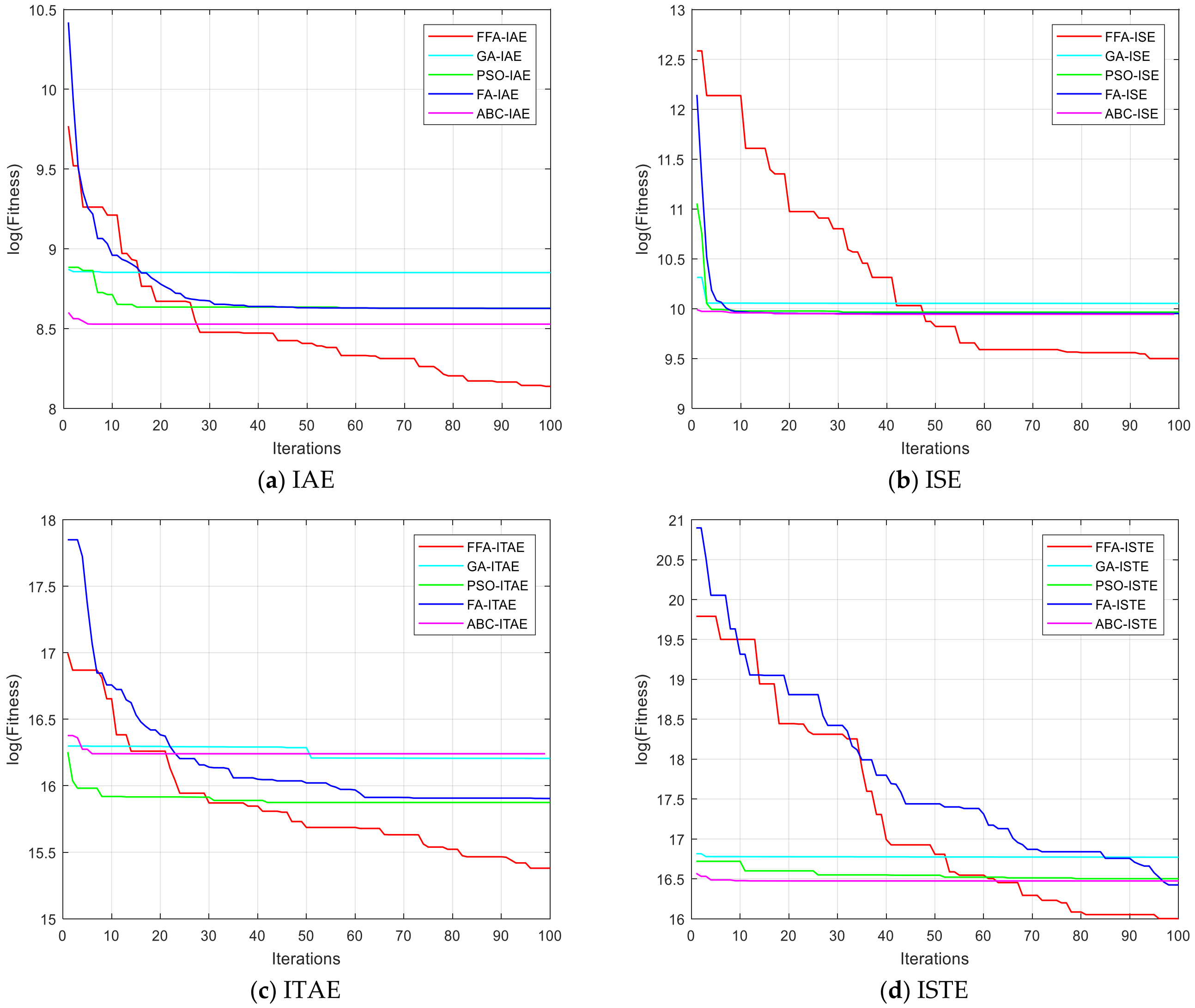
The model of BLDC motor drive is simulated in MATLAB. The parameters of the BLDC motor are reported in Table 6.

Table 6. Parameters of BLDC motor.

Parameters	Values
Number of pole	4
Nominal voltage vd	114 V
Stator resistance R	1.2 Ω
Stator inductance L	1.2 mH
Torque coefficient k_t	0.3262 Nm/A
Back-EMF coefficient k_e	0.3262 Vs/rad
Rotor inertia J	0.00085 kgm^2
Rated speed Nr	3000 rpm
Friction coefficient k_f	0.0001 Nms/rad

To control the BLDC motor, a conventional PI controller is used. However, it is not easy to adjust its parameters in order to have an efficient control. Therefore, the FFA_PI controller is used and it is compared to other algorithms to evaluate its competitiveness. The simulation is performed by considering the well-known algorithms GA, PSO, ABC and the standard FA. The simulation is run with 100 iterations and a population size of 10.

Figure 10 shows the evolution of the different performance criteria with the different algorithms. The results of FFA, with the different criteria, are all the better than those presented by the other algorithms. Figure 11, also, presents the cost functions IAE, ISE, ITAE and ISTE obtained by FFA algorithm.



(**a**) IAE (**b**) ISE

(**c**) ITAE (**d**) ISTE

Figure 10. Convergence curves for different algorithms with several criteria.

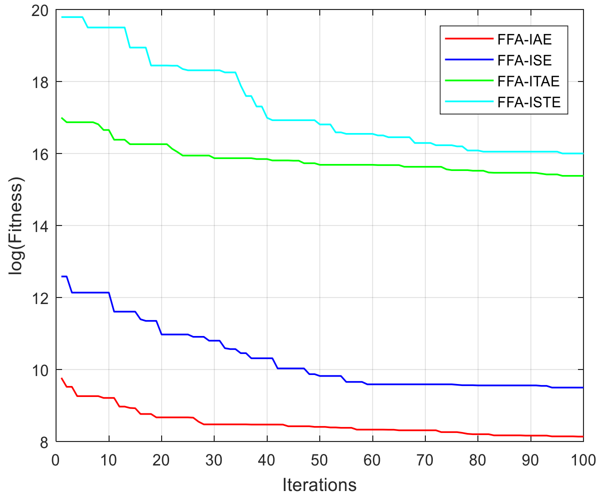


Figure 11. Convergence curves for different criteria with FFA algorithms.

The values of the PI controller, obtained by different simulations, are shown in Table 7. The values are obtained by the five algorithms used, with different criteria.

Table 7. kp, ki parameters obtained by various objective functions and various algorithms.

Algorithm	Parameters/Criterion	*IAE*	*ISE*	*ITAE*	*ITSE*
FFA	kp_FFA	18.19	24.5	24.56	24.06
	ki_FFA	4468.8	4435.2	4132.2	4002.32
FA	kp_FA	26.54	20.21	24.08	24.08
	ki_FA	2207.2	3615.8	3451.2	2996.2
GA	kp_GA	19.45	23.14	19.11	24.76
	ki_GA	1685.1	2474.5	3220.2	2896.1
PSO	kp_PSO	25.8	17.68	22.06	21.72
	ki_PSO	2081.5	3440.8	3451.2	3601.66
ABC	kp_ABC	24.51	19.53	20.16	24.17
	ki_ABC	3140.9	3796.6	3650.8	2901.12

In the chosen cost functions, the values of the overshoot, the rise time and the settling time can be controlled indirectly. Based on their optimization, the cost functions force the values of the other parameters to be optimum [27]. Table 8 shows the values of the different correctors used in this simulation. The values of the rise time, settling time, peak time, peak and overshoot are reported in Table 8. Accordingly, the results concerning the time are better for the FFA algorithm as well as for the peaks and the overshoots which are alternated with the other algorithms.

Table 8. Performance of the different controllers.

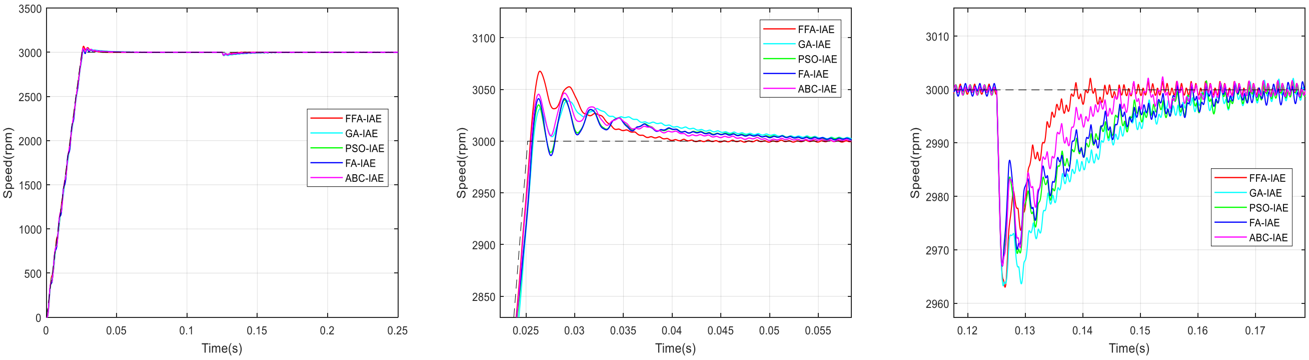
Controller	Criterion	Rise Time(s)	Settling Time(s)	Peak	Peak Time(s)	% Overshoot
FFA_PI	IAE	**0.0217**	**0.413**	3067.3	**0.0264**	2.2497
FA_PI		0.0219	0.0722	3041.1	0.0289	1.3708
GA_PI		0.0219	0.0731	**3038.8**	0.0293	**1.2922**
PSO_PI		0.0219	0.0730	3039.9	0.0289	1.3299
ABC_PI		0.0219	0.0579	3046.5	0.0289	1.5486
FFA_PI	ISE	**0.0217**	**0.0478**	**3058.2**	**0.0263**	1.7516
FA_PI		0.0218	0.0488	3060.7	0.0264	2.0226
GA_PI		0.0219	0.0646	3044.3	0.0290	**1.4769**
PSO_PI		**0.0217**	0.0488	3062.4	0.0265	2.0804
ABC_PI		**0.0217**	0.0488	3062.1	0.0264	2.0713
FFA_PI	ITAE	**0.0218**	**0.0488**	3051.5	**0.0263**	1.7163
FA_PI		0.0219	0.0546	**3049.7**	**0.0263**	**1.6554**
GA_PI		**0.0218**	**0.0488**	3058.9	0.0264	1.9620
PSO_PI		0.0219	0.0612	3055.2	**0.0263**	1.8398
ABC_PI		**0.0218**	**0.0488**	3061.2	0.0264	2.0393
FFA_PI	ISTE	**0.0218**	**0.0513**	3051.0	**0.0263**	1.7013
FA_PI		0.0219	0.0613	3046.4	0.0290	1.5480
GA_PI		0.0219	0.0613	3046.4	0.0290	**1.5473**
PSO_PI		**0.0218**	**0.0513**	3058.9	**0.0263**	1.9625
ABC_PI		0.0219	0.0596	**3045.8**	0.0290	1.5269

Moreover, the execution simulation time comparison is given between the different correctors and shown in Table 9. It can be reported that the calculation time using the FFA_PI is faster than those obtained with the FA_PI, GA_PI, PSO_PI and ABC_PI when using 50 or 100 iterations.

Table 9. Simulation time of the five algorithms.

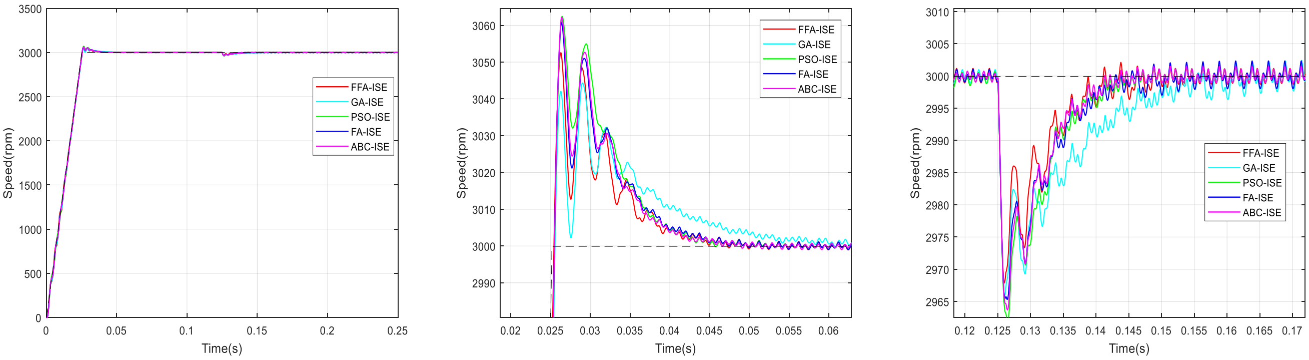
	Simulation Time (s)				
Iteration	**FFA**	**FA**	**GA**	**PSO**	**ABC**
50	108.53	284.15	119.55	242.23	135.95
100	216.57	570.06	239.77	486.82	268.20

According to the used criterions, Figures 12–15 represent the BLDC motor speeds obtained with the different corrector optimized. Consequently, the figures are given for comparison and they justify the values in Table 8.



(**a**) (**b**) (**c**)

Figure 12. Comparison of speed responses with different algorithms using IAE criterion: (**a**) original; (**b**) and (**c**) zoomed version.



(**a**) (**b**) (**c**)

Figure 13. Comparison of speed responses with different algorithms using ISE criterion: (**a**) original; (**b**) and (**c**) zoomed version.

The graphs are zoomed in the area of the overshoot and the rejection of the disturbance for better visualization of signals.

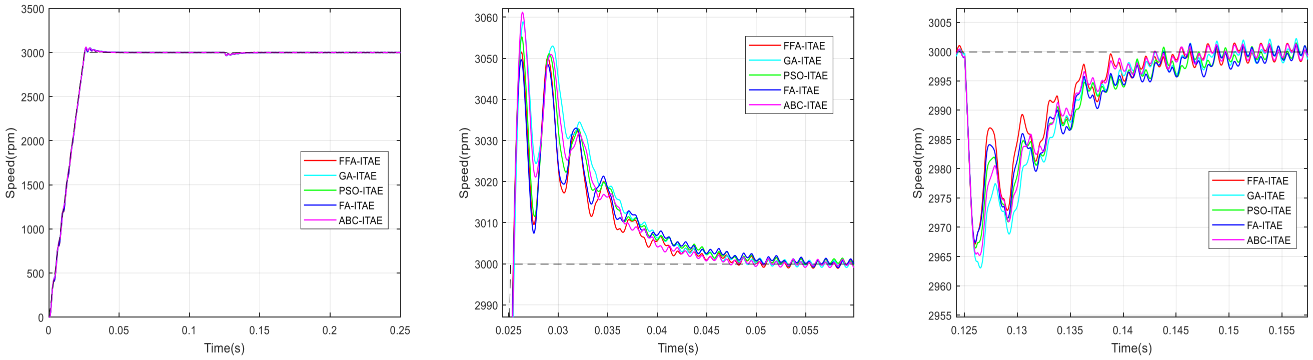


Figure 14. Comparison of speed responses with different algorithms using ITAE criterion: (**a**) original; (**b**) and (**c**) zoomed version.

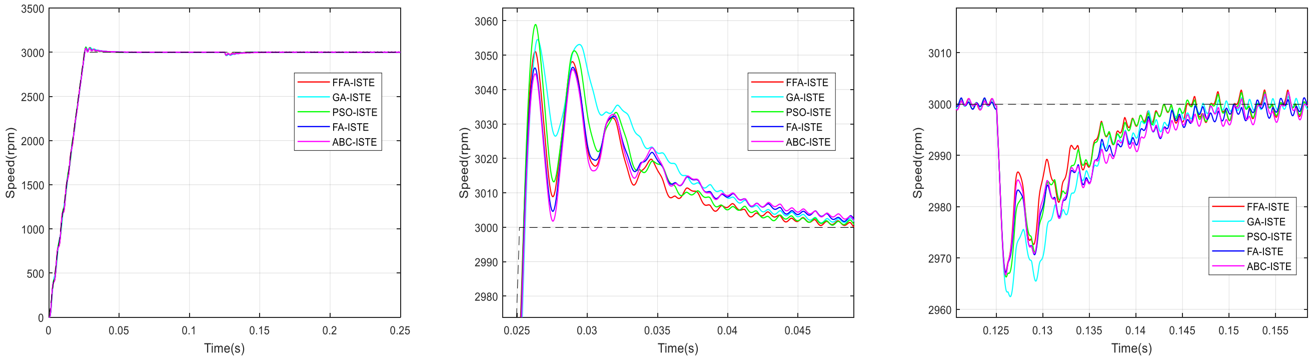
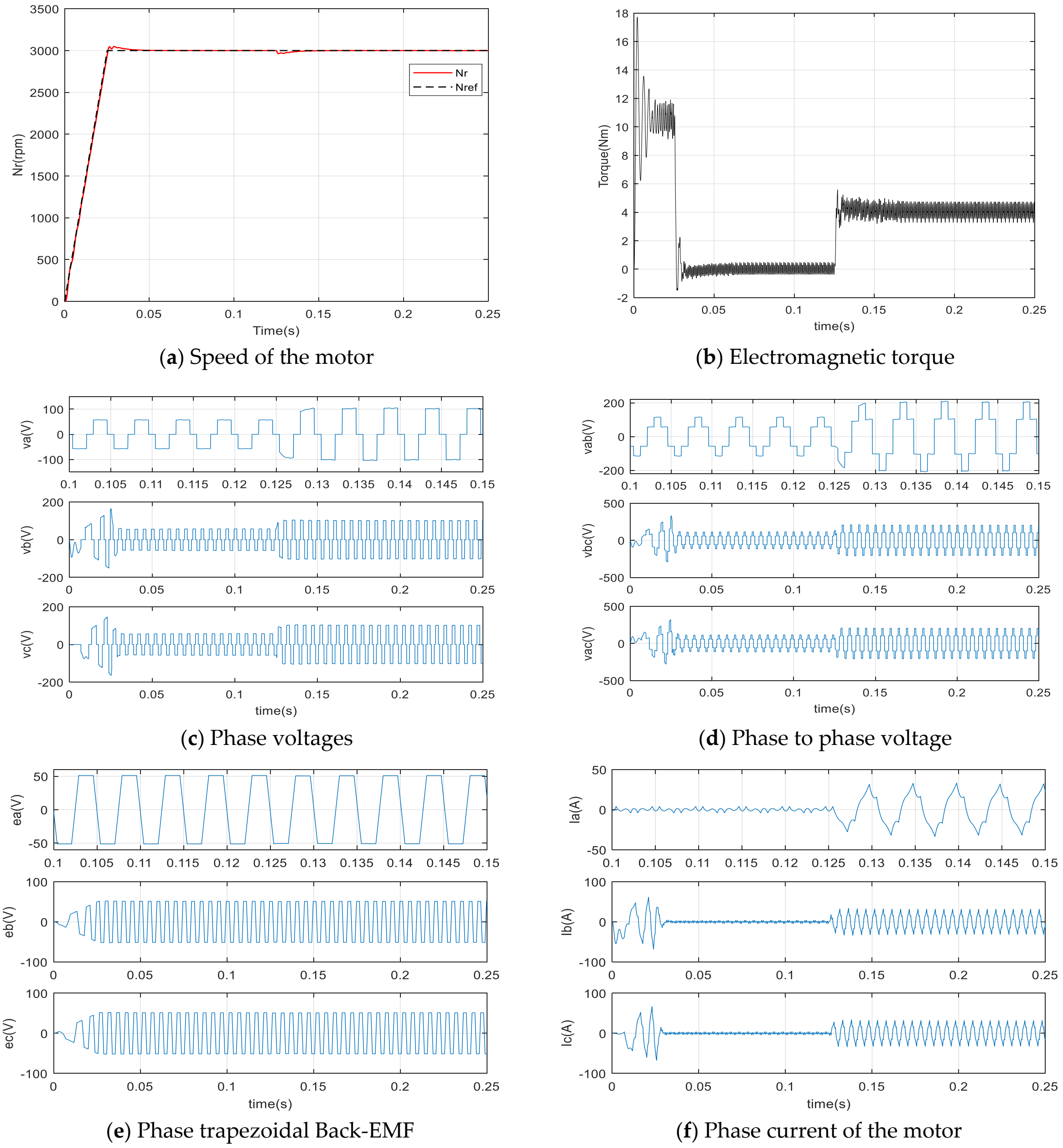


Figure 15. Comparison of speed responses with different algorithms using ISTE criterion: (**a**) original; (**b**) and (**c**) zoomed version.

From the previous numerical results and the figures' responses, it can be concluded that the optimized PI controller-based FFA showed a better capacity to compete with its FA counterpart, and its rivals GA, PSO and ABC. Thus, it provided the fastest rise and response times in addition to the minimum peak time.

Figure 16 show the simulation results of the various variables of the BLDC motor using the FFA_PI using (ki = 2468, kp = 18.19). Accordingly, Figure 16a presents the speed of the BLDC motor where the reference speed *ω*ref is chosen as a ramp in order to dampen the current at start-up and to avoid peaks as well as for the electromagnetic torque on the Figure 16b. At 0.125 s, a torque load TL = 4 Nm is applied and a good rejection by the control is observed. The effect of the load is very apparent on the figure of the speed, the torque, the voltages and the current.

On each figure presented, there are three phases, where the first phase is zoomed-in to clearly visualize the behavior of the signals. Thus, Figure 16c,d show the phase voltages and the phase to phase voltage simultaneously. The trapezoidal Back-EMF shape is well illustrated on the Figure 16e. Finally, the shape of the currents of the three phases of the stator is given by the Figure 16f. As can be seen, there is a distortion in the torque signals which is due to the trapezoidal shape of the Back-EMF and the nature of the currents containing harmonics. Finally, Figure 17 gives the evolution, until the convergence, of the parameters of the FFA_PI and FA_PI on the control technique.



(**a**) Speed of the motor

(**b**) Electromagnetic torque

(**c**) Phase voltages

(**d**) Phase to phase voltage

(**e**) Phase trapezoidal Back-EMF

(**f**) Phase current of the motor

Figure 16. Results of simulation by using FFA_PI controller.

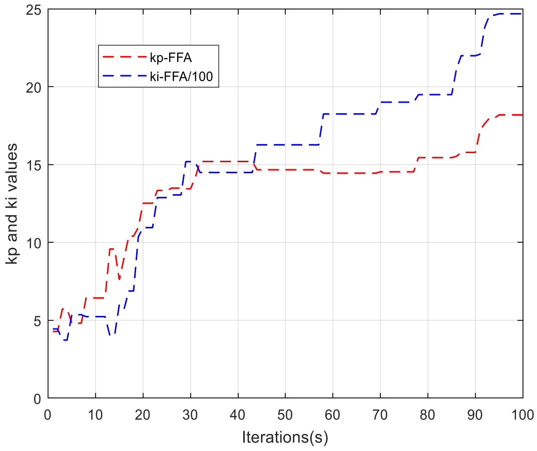


Figure 17. Evolution of parameters of FFA_PI until convergence.

5. Conclusions

A fast FA algorithm so-called FFA is presented and compared with the standard FA through searching the global optimum by using different standard benchmark functions in a first application. The simulation results were compared, taking in consideration the precision and the speed of convergence criteria for the two algorithms. The reached results prove that those obtained by FFA are better than those of FA. A second application concerning the optimization of the gains of a PI controlling a BLDC motor is carried out through the ITSE performance criterion. The results obtained show the robustness of the two algorithms with superiority for FFA. The acceleration of the proposed algorithm is due to the search space reduction by a random election of a significantly small set of moving fireflies while the whole search space stills covered. It should be noted that the acceleration, in the optimization function, is in the average 12:1, with respect to FA. Additionally, for the complex problem (BLDC motor control), the acceleration is clearly remarked for the modified algorithm FFA than FA, GA, PSO and ABC algorithms. Globally, the suggested FFA algorithm can be considered as most state of the art metaheuristic algorithms such as FA, GA, PSO and ABC, and presents superior fastness against all reported optimizers.

Furthermore, a modification on the α parameter is given and this guarantees the robustness and precision through the enhancement of search directions toward the global optimal solution.

Author Contributions: Conceptualization, methodology, software, writing—original draft preparation, validation S.B. and R.B.; Supervision, R.B. formal analysis, writing—review and editing, funding acquisition, Y.B. and M.M.A.R. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group No. RG-1441-502.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group No. RG-1441-502.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

Symbols used in this paper	
ABC	Artificial Bee Colony
PSO	Particle Swarm Optimization
CS	Cuckoo Search
BA	Bat Algorithm
GWO	Gray Wolf Optimizer
FA	Firefly Algorithm
FFA	Fast Firefly Algorithm
GA	Genetic Algorithms
PI	Proportional Integral
PID	Proportional Integral & Derivative
kp, ki	Proportional and Integral gains of the PI controller
IFA	Improved Firefly Algorithm
MFA	Modified Firefly Algorithm
t_{FA}	Execution time of the original algorithm FA
t_{FFA}	Execution time of the proposed algorithm FFA
R	Resistance of a stator phase, [Ω]
L	Inductance of a stator phase, [H]

v_a, v_c, v_c	Stator phase voltages, [V]
v_{ab}, v_{bc}, v_{ca}	Stator phase to phase voltages, [V]
i_a, i_b, i_c	Stator phase currents, [A]
e_a, e_b, e_c	Motor Back-EMFs, [V]
T_e, T_L	Electromagnetic and load torques, [Nm]
J	Rotor inertia, [kgm^2]
k_f	Friction constant, [Nms/rad]
kt	Torque coefficient, [Nm/A]
ω_m	Rotor speed, [rad/s]
Nr	Rated speed, [rpm]
θ_e	Electric angle of rotor, [rad]
θ_m	Mechanic angle of rotor, [rad]
$F(\theta_e)$	Back-EMF reference function
$\varepsilon(t)$	Error input signal
$y(t)$	Output signal
Ha, Hb, Hc	Hall Effect Sensors for the three phases
IAE	Integral Absolute Error
ISE	Integral Square Error
ITAE	Integral Time Absolute Error
ISTE	Integral Square Time Error

References

1. Sarangi, S.K.; Panda, R.; Priyadarshini, S.; Sarangi, A. A New Modified Firefly Algorithm for Function Optimization. In Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016; pp. 2944–2949.
2. Yelghi, A.; Köse, C. A Modified Firefly Algorithm for Global Minimum Optimization. *Appl. Soft Comput.* **2018**, *62*, 29–44. [CrossRef]
3. Ariyaratne, M.K.A.; Fernando, T.G.I.; Weerakoon, S. A Modified Firefly Algorithm to Solve Univariate Nonlinear Equations with Complex Roots. In Proceedings of the 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 24–26 August 2015; pp. 160–167.
4. Ariyaratne, M.K.A.; Fernando, T.G.I.; Weerakoon, S. Solving systems of nonlinear equations using a modified firefly algorithm (MODFA). *Swarm Evol. Comput.* **2019**, *48*, 72–92. [CrossRef]
5. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed.; Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1989; ISBN 978-0-201-15767-3.
6. Dorigo, M.; Birattari, M.; Stutzle, T. Ant Colony Optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [CrossRef]
7. Zeng, Q.; Tan, G. Optimal Design of PID Controller Using Modified Ant Colony System Algorithm. In Proceedings of the Third International Conference on Natural Computation (ICNC 2007), Haikou, China, 24–27 August 2007; Volume 5, pp. 436–440.
8. Karaboga, D.; Akay, B.; Ozturk, C. Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks. In Proceedings of the Modeling Decisions for Artificial Intelligence, Haikou, China, 24–27 August 2007; Torra, V., Narukawa, Y., Yoshida, Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 318–329.
9. Singh, K.; Sundar, S. Artifical Bee Colony Algorithm Using Problem-Specific Neighborhood Strategies for the Tree t-Spanner Problem. *Appl. Soft Comput.* **2018**, *62*, 110–118. [CrossRef]
10. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
11. Nema, S.; Padhy, P.K. PI-PD Controller for Stable and Unstable Processes. *Int. J. Syst. Control Commun.* **2013**, *5*, 156–165. [CrossRef]
12. Nangru, D.; Bairwa, D.K.; Singh, K.; Nema, S.; Padhy, P.K. Modified PSO Based PID Controller for Stable Processes. In Proceedings of the 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE), Jabalpur, India, 16–18 December 2013; pp. 1–5.
13. Yang, X.-S.; Deb, S. Cuckoo Search via Lévy Flights. In Proceedings of the 2009 World Congress on Nature Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214.
14. Yang, X.-S.; Deb, S. Engineering Optimisation by Cuckoo Search. *Int. J. Math. Model. Numer. Optim.* **2010**, *1*, 330–343. [CrossRef]
15. Yang, X.; He, X. Bat Algorithm: Literature Review and Applications. *Int. J. Bio-Inspired Comput.* **2013**, *5*, 141–149. [CrossRef]
16. Yang, X.-S. A New Metaheuristic Bat-Inspired Algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N., Eds.; Studies in Computational Intelligence; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74. ISBN 978-3-642-12538-6.
17. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
18. Pallav, D.; Santanu, K.N. Grey Wolf Optimizer Based PID Controller for Speed Control of BLDC Motor. *J. Electr. Eng. Technol.* **2021**, *16*, 955–961. [CrossRef]

19. Yang, X.-S. Firefly Algorithms for Multimodal Optimization. In Proceedings of the Stochastic Algorithms: Foundations and Applications; 5th International Symposium, SAGA 2009, Sapporo, Japan, 26–28 October, 2009; pp. 169–178.
20. Yang, X.-S. *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: Beckington, UK, 2010; ISBN-13: 978-1-905986-28-6.
21. Bazi, S.; Benzid, R.; Said, M.S.N. Optimum PI Controller Design in PMSM Using Firefly Algorithm and Genetic Algorithm. In Proceedings of the 2017 6th International Conference on Systems and Control (ICSC), Batna, Algeria, 7–9 May 2017; pp. 85–89.
22. Pradhan, P.C.; Sahu, R.K.; Panda, S. Firefly Algorithm Optimized Fuzzy PID Controller for AGC of Multi-Area Multi-Source Power Systems with UPFC and SMES. *Eng. Sci. Technol. Int. J.* **2016**, *19*, 338–354. [CrossRef]
23. Jinran, W.; You-Gan, W.; Kevin, B.; Yu-Chu, T.; Brodie, L.; Zhe, D. An improved firefly algorithm for global continuous optimization problems. *Expert Syst. Appl.* **2020**, *149*, 113340. [CrossRef]
24. Boris, J.; Paolo, L.; Guido, M. Control of double-loop permanent magnet synchronous motor drives by optimized fractional and distributed-order PID controllers. *Eur. J. Control* **2021**, *58*, 232–244. [CrossRef]
25. Mat Hussin, A.T.; Intan Zaurah, M.D.; Pakharuddin, M.S.; Hanim, M.Y.; Mohd Ibthisham, A.; Nik Mohd, R.S.; Muhamad, S.H. Vibration control of semi-active suspension system using PID controller with advanced firefly algorithm and particle swarm optimization. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 1119–1137.
26. He, S.; Chen, Z.; Gao, X. Parameter Solving of DC Servo Motor PID Controller Based on Improved Firefly Algorithm. In Proceedings of the 2018 3rd International Conference on Robotics and Automation Engineering (ICRAE), Guangzhou, China, 17–19 November 2018; pp. 136–140.
27. Rosy, P.; Santosh, K.M.; Jatin, K.P.; Bibhuti, B.P. Optimal fractional order PID controller design using Ant Lion Optimizer. *Ain Shams Eng. J.* **2020**, *11*, 281–291. [CrossRef]
28. Devarapalli, R.; Naga Lakshmi, N.J.; Prasad, U. Application of a Novel Political Optimization in Optimal Parameter Design of PI Controller for the BLDC Motor Speed Control. In Proceedings of the 2020 International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET), Patna, India, 10–11 July 2020; pp. 1–6.
29. BARAN, H. Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm. *IEEE Access* **2019**, *7*, 38100–38114.
30. Jagatheesan, K.; Anand, B.; Samanta, S.; Dey, N.; Ashour, A.S.; Balas, V.E. Design of a Proportional-Integral-Derivative Controller for an Automatic Generation Control of Multi-Area Power Thermal Systems Using Firefly Algorithm. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 503–515. [CrossRef]
31. Chaib, L.; Choucha, A.; Arif, S. Optimal Design and Tuning of Novel Fractional Order PID Power System Stabilizer Using a New Metaheuristic Bat Algorithm. *Ain Shams Eng. J.* **2017**, *8*, 113–125. [CrossRef]
32. Xiaoyan, P.; Mingfei, J.; Lei, H.; Xiang, Y.; Yibin, L. Fuzzy sliding mode control based on longitudinal force estimation for electro-mechanical braking systems using BLDC motor. *CES Trans. Electr. Mach. Syst.* **2018**, *2*, 142–151.
33. Walekar, V.R.; Murkute, S.V. Speed Control of BLDC Motor Using PI Fuzzy Approach: A Comparative Study. In Proceedings of the 2018 International Conference on Information, Communication, Engineering and Technology (ICICET), Pune, India, 29–31 August 2018; pp. 1–4.
34. Ahmed, R.; Paul, Y. Hardware/Software Implementation of Fuzzy-Neural-Network Self-Learning Control Methods for Brushless DC Motor Drives. *IEEE Trans. Ind. Appl.* **2016**, *52*, 414–424.
35. Xiong, S.; Junguo, G.; Jian, C.; Biao, J. Research on Speed Control System of Brushless DC Motor Based on Neural Network. In Proceedings of the 2015 8th International Conference on Intelligent Computation Technology and Automation (ICICTA), Nanchang, China, 14–15 June 2015; pp. 761–764.
36. Putra, E.H.; Zulfatman, Z.; Effendy, M. Robust Adaptive Sliding Mode Control Design with Genetic Algorithm for Brushless DC Motor. In Proceedings of the International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Malang, Indonesia, 16–18 October 2018; Volume 5, pp. 330–335, ISBN 978-1-5386-8402-3.
37. Li, J.; Li, W. On-Line PID Parameters Optimization Control for Wind Power Generation System Based on Genetic Algorithm. *IEEE Access* **2020**, *8*, 137094–137100. [CrossRef]
38. Merugumalla, M.K.; Navuri, P.K. PSO and Firefly Algorithms Based Control of BLDC Motor Drive. In Proceedings of the 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2018; pp. 994–999.
39. Ho, S.L.; Yang, S.; Ni, G.; Lo, E.W.C.; Wong, H.C. A Particle Swarm Optimization-Based Method for Multiobjective Design Optimizations. *IEEE Trans. Magn.* **2005**, *41*, 1756–1759. [CrossRef]
40. Fateen, S.-E.K.; Bonilla-Petriciolet, A. Intelligent firefly algorithm for global optimization. In *Cuckoo Search and Firefly algorithm*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 315–330.
41. Ardeh, M.A. Benchmark Functions. Available online: Benchmarkfcns.xyz/fcns.html (accessed on 26 May 2021).
42. Jamil, M.; Yang, X.-S. A Literature Survey of Benchmark Functions for Global Optimisation Problems. *Int. J. Math. Model. Numer. Optim.* **2013**, *4*, 150–194. [CrossRef]
43. Permanent Magnet Brushless DC Motor Drives and Controls | Wiley. Available online: https://www.wiley.com/en-ar/Permanent+Magnet+Brushless+DC+Motor+Drives+and+Controls-p-9781118188330 (accessed on 27 May 2021).
44. Mondal, S.; Mitra, A.; Chattopadhyay, M. Mathematical Modeling and Simulation of Brushless DC Motor with Ideal Back EMF for a Precision Speed Control. In Proceedings of the 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 5–7 March 2015; pp. 1–5.

45. Prasad, G.; Ramya, N.S.; Prasad, P.V.N.; Das, G.T.R. Modeling and Simulation Analysis of the Brushless DC Motor by Using MATLAB. *Int. J. Innov. Technol. Explor. Eng. (IJITEE)* **2012**, *1*, 27–31.
46. Kumar, D.; Gupta, R.A.; Gupta, N. Modeling and Simulation of Four Switch Three-Phase BLDC Motor Using Anti-Windup PI Controller. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017; pp. 1–6.
47. Jaromír, J.; Miloslav, Č. Position Measurement with Hall Effect Sensors. *Am. J. Mech. Eng.* **2013**, *1*, 231–235. [CrossRef]