electronics

MDPI

Article

Minimization of Torque Ripple in the Brushless DC Motor Using Constrained Cuckoo Search Algorithm

Łukasz Knypiński [1,*], Sebastian Kuroczycki [1] and Fausto Pedro García Márquez [2]

[1] Institute of Electrical Engineering and Electronics, Poznan University of Technology, Piotrowo 3A, 60-965 Poznan, Poland; Kuroczycki@protonmail.com

[2] Ingenium Research Group, University of Castilla-La Mancha, 13071 Ciudad Real, Spain; FaustoPedro.Garcia@uclm.es

* Correspondence: lukasz.knypinski@put.poznan.pl

Abstract: This paper presents the application of the cuckoo search (CS) algorithm in attempts to the minimization of the commutation torque ripple in the brushless DC motor (BLDC). The optimization algorithm was created based on the cuckoo's reproductive behavior. The lumped-parameters mathematical model of the BLDC motor was developed. The values of self-inductances, mutual inductances, and back-electromotive force waveforms applied in the mathematical model were calculated by the use of the finite element method. The optimization algorithm was developed in Python 3.8. The CS algorithm was coupled with the static penalty function. During the optimization process, the shape of the voltage supplying the stator windings was determined to minimize the commutation torque ripple. Selected results of computer simulation are presented and discussed.

Keywords: heuristic optimization algorithm; cuckoo search algorithm; brushless DC motor; torque ripple; constrained optimization; static penalty function

check for updates

Citation: Knypiński, Ł.; Kuroczycki, S.; Márquez, F.P.G. Minimization of Torque Ripple in the Brushless DC Motor Using Constrained Cuckoo Search Algorithm. *Electronics* **2021**, *10*, 2299. https://doi.org/10.3390/electronics10182299

Academic Editor: Pablo García Triviño

Received: 31 July 2021
Accepted: 15 September 2021
Published: 18 September 2021

1. Introduction

Currently, computational simulations are commonly used when designing the processes of technical devices. These computational simulations enable the modeling of the various phenomena and processes which occur in a device being designed, with a variety of complexity levels. During the designing process of electromagnetic devices, complex field models of phenomena in the device can be applied [1–3]. These field models include equations of: (a) the electromagnetic field; (b) the external supply circuit; (c) the mechanical equilibrium; and (d) the thermal processes in the designed devices. Such models of these phenomena are created using the finite element method (FEM). The time it takes to determine a single electromagnetic field distribution in the analyzed devices is very long [4,5]. Therefore, the optimization process using heuristic algorithms requires a high computational cost. On the other hand, lumped-parameter models, or analytical models, are computationally less complex [6–8]. Such types of models are commonly used to evaluate the correctness and efficiency of new algorithms [9]. They can also be used in the initial stage of the development of electrical machines.

Decisions made by a designer are carried out automatically. The automation of the designing processes consists of applying an optimization software (script), where the main task is to 'show' or 'point out' the optimal solution. The software usually consists of two independent parts (the optimization algorithm and the mathematical model of a designed device) [1]. An optimal solution is a solution fulfilling all the criteria defined by a designer.

In the case of designing electromagnetic devices, the nondeterministic (heuristic, metaheuristic) algorithms are used most often [9]. The advantage of these methods is that they allow efficient searching through the space under investigation to find the global extreme point. Heuristic algorithms are able to search for optimal solutions within a short time. Some algorithms can be categorized as simple or complex algorithms, which are based on interactions within a group of individuals (i.e., a population). Individuals grouped into a population can compete (e.g., genetic algorithms) or cooperate, sharing information about the localization of a leader (e.g., particle swarm method or gray wolf method) [2,10–13].

Nondeterministic algorithms are effective, and are often used to solve difficult problems arising in the synthesis of the electromagnetic transducers, where a solution is looked for within multidimensional sets of design variables [14]. Among these nondeterministic methods, the nature-inspired algorithms are popular and based on observations of phenomena in the natural environment. The most popular and most used algorithms are the genetic algorithms (GA) and the particle swarm algorithm (PSO). However, the gray wolf method (GWO), the cuckoo search (CS) algorithm, the ant colony algorithm (ACO), or the method based on the echolocational behavior of bats (BA) are rarely used methods [15]. Researchers are still looking for new, more effective optimization algorithms, which can solve the optimization task faster. As a result of this research work, new algorithms are developed and known algorithms are also modified.

The cuckoo search (CS) algorithm is very often used for optimization without constraints [16,17]. A much smaller number of manuscripts have addressed the application of the CS algorithm when solving constrained optimization problems [18,19]. The external penalty function can be applied for most of the heuristic algorithms [1,9,11,14,15]. In the case of using the CS algorithm to solve constraint optimization problems, as in the case of other heuristic algorithms, scientists should select the method they use to take into account constraints very carefully. The correct selection of the approach and the individual penalty coefficient may significantly affect the quality and efficiency of the obtained results [9]. In order to use a static penalty, it is enormously important to choose a constant number. This constant number must be carefully selected and is dependent on the type of problem that is being solved. The value of the constant number determines the value of the penalty, and this is chosen in the preliminary trials of the algorithm [20].

The major research purpose of this paper is to elaborate upon the computer script for the minimization of the pulsation factor in a BLDC motor. The computer script was developed in Python 3.8. The novelty of this work can be summarized as follows:

- The CS algorithm was tied with the static penalty function approach;
- The results of our FEM calculations were used to determine the parameters of the simplified model, which would allow us to obtain more accurate results from the optimization process;
- The modified objective function is composed of: (a) the main objective function (electromagnetic torque); and (b) the penalty term representing the required value of the permissible pulsation factor.

The rest of the paper is organized as follows: Section 2 describes the reproductive behavior of the cuckoo and the mathematical model of the cuckoo search algorithm; Section 3 outlines the mathematical model of a brushless DC (BLDC) motor; the optimization task is formulated in Section 4; Section 5 presents the results of the optimization calculations; finally, discussions and conclusions are given in Section 6.

2. The Cuckoo Search Optimization Algorithm

2.1. The Reproductive Behavior of the Cuckoo in the Natural Environment

The common cuckoo inhabits territories in Europe and Asia. Adult birds do not pair. It has been observed that in their natural environment there are more males than females. Cuckoos do not build their own nests and they do not brood their own eggs. Rather, they are reproductive predators. In order to propagate their species, female cuckoos slip their eggs into the nests of different species of birds [21].

A male cuckoo observes other birds in their nests while they incubate their eggs, and, when the host leaves the nest, the female cuckoo visits, throwing away one of the eggs that is already there and laying her own egg. These female cuckoos usually choose a species of birds whose eggs are similar to those of the cuckoo, including most commonly the red-ramped parrot, wagtails, the great reed warbler, and the tooth-billed bowerbird.

Sometimes, when the hosts spot a foreign egg, they throw it away. There are also known cases when, after recognition of a foreign egg in the nest, the hosts decide to leave the nest for good and built a new nest in another location.

Young cuckoos hatch after around 11 days, which is typically faster than the nestling period of the hosts. As soon as they are able to move, young cuckoos use their backs to push the other eggs out of the nest. The cuckoo's nestlings are typically bigger than the ones of the host, and so they need more space in the nest, as well as more food. If they did not push out the other eggs, none of the nestlings would survive because the hosts would not be able to provide enough food for them all [22]. Female cuckoos typically lay more eggs than the female birds of the species that they choose for hosts. In the natural environment, around 35% of cuckoos do not reach adulthood.

2.2. Cuckoo Search Algorithm

The mathematical model of the cuckoo algorithm (CS) was developed by Xin-She Yang and S. Deb [23]. The cuckoo method belongs to a group of algorithms called 'nature-inspired algorithms'. The method was developed based on the of observation of the cuckoo's reproductive process. In the mathematical model, the following assumptions are included [24]:

(1) Each cuckoo lays one egg in a randomly chosen nest;
(2) Nests with the best eggs (i.e., solutions to the problem) are carried on to the next iteration of the algorithm;
(3) The number of hosts' nests is constant. A host can discover a foreign egg with a given probability $p_a \in (0, 1)$ [25]. In such a case, a host can remove the egg (the solution is eliminated) or abandon the nest and build a new one in a new localization.

In this algorithm, the last rule is carried out by replacing some part of the existing solutions with new, randomly generated solutions. Similar to all nondeterministic algorithms, a starting population is set at the starting point with an equal number of cuckoos. In the CS algorithm, the nest is a single solution to the problem being analyzed. In each iteration of the algorithm, 25% of nests change their location, i.e., the p_a probability is 0.25 [26,27]. The cuckoo starts from the i-th nest and lays an egg in the new nest, while N denotes the total number of cuckoos, and n is the nests numbers in the optimization algorithm.

Taking into account the three theoretical assumptions outlined above, the potential new position of each nest in the current iteration is determined according to Equation (1) [28]:

$$x_t^i = x_{t-1}^i + \alpha\kappa(\lambda)\left(x_b - x_{t-1}^i\right) \tag{1}$$

where t is the number of the iteration, $\alpha > 0$ is the step size scaling factor of the number, $\kappa(\lambda)$ is the Lévy flight coefficient, and x_b is the position of the best nest.

The random walk coefficient $\kappa(\lambda)$ is drawn from a Lévy distribution [26,29] and calculated by Equation (2) [30]:

$$\kappa(\lambda) = \sqrt{\frac{\Psi}{2\pi}\frac{1}{t^{1.5}}},\ (1 < \lambda \leq 3) \tag{2}$$

where Ψ is the constant number.

After determining the new nest locations for a total number of cuckoos, the new positions of nests with a given probability p_a are drawn. The worst nest positions are removed in each iteration.

The block diagram (in the case of maximization) of the proposed algorithm is presented in Figure 1. The maximum number of assigned iterations ($t_{\max}$) was adopted as the stop criterion.

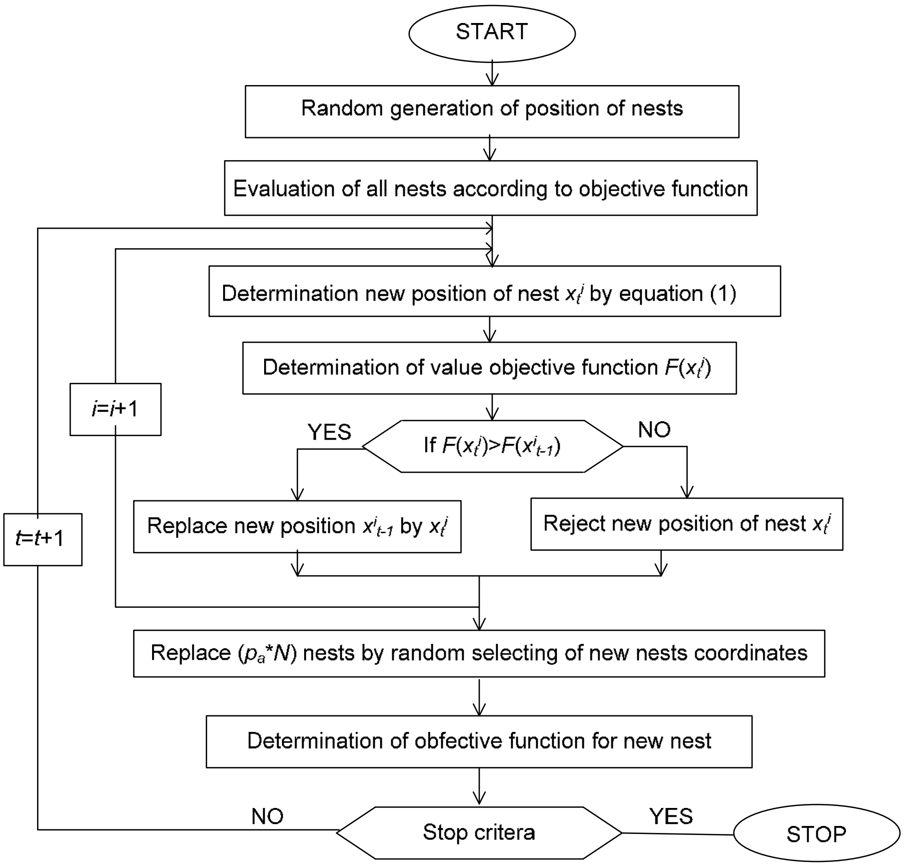

START
Random generation of position of nests
Evaluation of all nests according to objective function
Determination new position of nest x_t^i by equation (1)
Determination of value objective function $F(x_t^i)$
i=i+1
YES
If $F(x_t^i)$>$F(x^i_{t-1})$
NO
t=t+1
Replace new position x^i_{t-1} by x_t^i
Reject new position of nest x_t^i
Replace (p_a*N) nests by random selecting of new nests coordinates
Determination of obfective function for new nest
NO
Stop critera
YES
STOP

Figure 1. Block diagram of the optimization procedure.

2.3. Test of the Optimization Procedure Using a Test Function

The optimization procedure was developed in Python 3.8. The correctness test was performed using the multimodal Himmelblau's analytical function. The test function is defined by Equation (3):

$$f(x,y) = \left(x^2 + y - 11\right)^2 + \left(x + y^2 - 7\right)^2 \tag{3}$$

wherein $-5 \le x \le 5$ and $-5 \le y \le 5$.

The Himmelblau's function has one global maximum at the point (−0.270845, −0.923039), where $f(x,y)$ = 181.617, and four local minima at the following points: (3,2), (−2.805118, 3.131312), (−3.779310, −3.283186), and (3.58428, −1.848126). All local minima are identical and equal to zero ($f(x,y)$ = 0). The 3D visualization of Himmelblau's function is presented in Figure 2.

The optimization calculations were performed for the following parameters: there is a number of cuckoos equal to N = 100, a number of nests equal to n = 125, and a probability p_a = 0.25, while the maximum number of iterations is $t_{\max}$ = 40. The optimization procedure was run 10 times, and the best results are presented in Table 1. In each successive column is listed: the number of the iteration (t), the x and y coordinates, the absolute error between the optimal and actual value of the objective function (Δf), and the number of calls to the objective function (N_{of}).

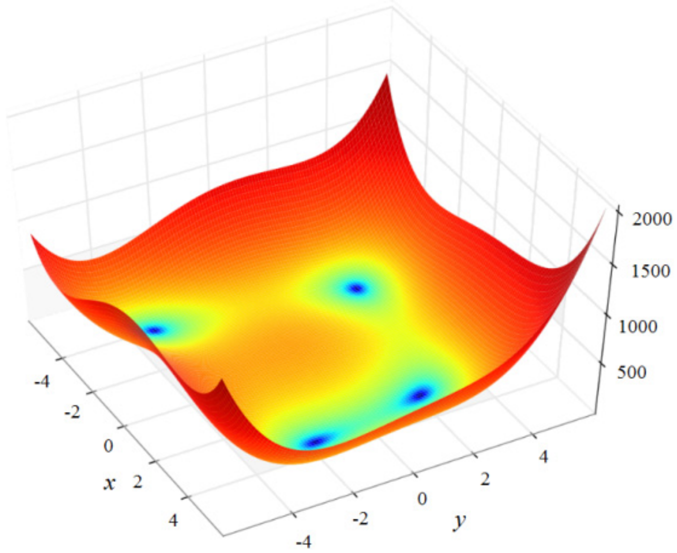


Figure 2. The visualization of Himmelblau's functions in a 3D coordinate system.

Table 1. Results from selected iterations of the cuckoo search algorithm.

t	x	y	$\Delta f(x,y)$	N_{of}
1	2.8881171	2.4550472	3.710164	100
2	3.7677678	−1.9077475	1.825653	225
4	3.7453086	−1.8046252	1.495028	475
6	2.9771054	2.04594154	0.034826	725
8	3.5913946	−1.8473621	0.003278	975
10	3.5913946	−1.8473621	0.002378	1225
20	3.5913946	−1.8473621	0.003278	2325
40	3.5909618	−1.8480482	0.002914	4525

On the basis of the results obtained from the CS algorithm, we can conclude that in the first iterations, the algorithm-determined solution approximates to the global extreme. The value of the objective function improves in successive iterations. After about eight iterations, a result is obtained which is close to the optimal one.

Figure 3 shows the distribution of the cuckoos in selected iterations of the algorithm. Based on the obtained results, it can be observed that after 25 iterations, the cuckoos are near three of the minimum points and about the same value as the objective function. After 40 iterations, all individuals are clustered near two minima points.

Figure 4 shows the distribution of the swarm particles in selected iterations.

Based on the distributions of the cuckoos (Figure 3) and swarm particles (Figure 4) in selected iterations, it can be observed that the process of searching for the global extreme is different. The PSO algorithm focuses on and goes towards the selected minimum, while the CS algorithm looks for a better extreme point throughout the whole optimization process.

If the optimization processes for the PSO and the CS algorithms are compared, it can be noted that for the PSO algorithm, all individuals in the swarm are moving towards a leader. On the other hand, for the CS algorithm, the process of focusing around the leader is much slower in relation to the PSO algorithm. Therefore, for the PSO algorithm, the external penalty function should be applied. The external penalty must systematically increase in subsequent iterations [14]. In the case of the CS algorithm, a gentler way of considering constraints can be employed.

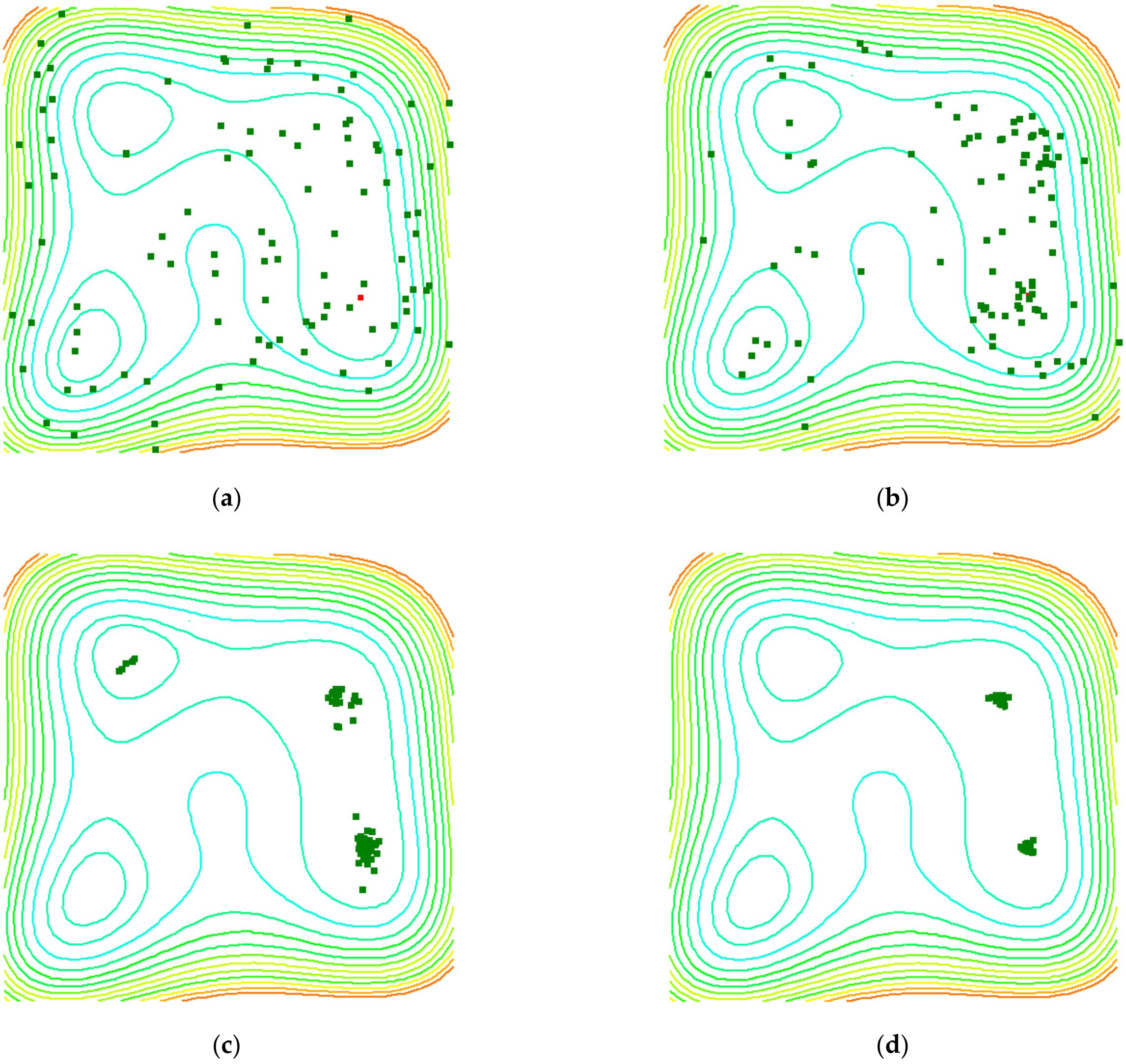


Figure 3. The distribution of cuckoos in selected iterations: (**a**) $t = 1$; (**b**) $t = 6$; (**c**) $t = 25$; (**d**) $t = 40$.

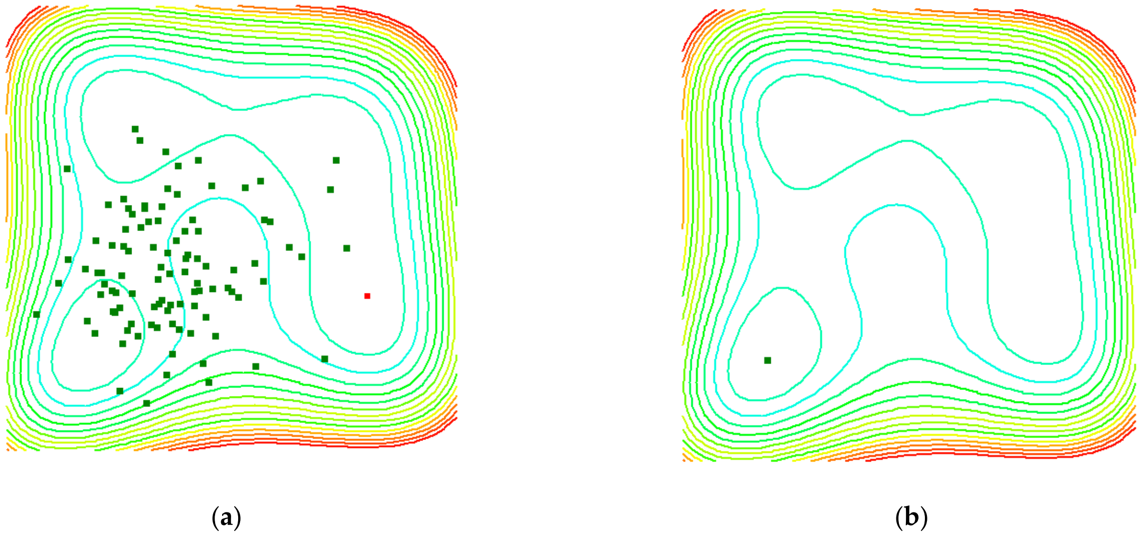


Figure 4. The distribution of swarm particles in selected iterations: (**a**) $t = 5$; (**b**) $t = 25$.

Here, the results obtained using the CS algorithm are compared with those obtained using the PSO algorithm [31–33]. The calculations were repeated over ten runs for the same numbers of particles and a maximum number of iterations, as was the case for the CS algorithm. The following parameters of the PSO algorithm were applied: $w = 0.2$, $c_1 = 0.35$, and $c_2 = 0.45$ [14]. Additionally, the best, worst, and mean values of the objective functions and standard deviations for both algorithms were compared. The results are presented in Table 2.

Table 2. Comparison of results for Himmelblau's function.

Algorithm	Best	Worst	Mean	Standard Deviation
CS	0.002914	0.041602	0.058439	0.010225
PSO	0.004196	0.097256	0.081736	0.092668

Table 2 shows that the best result for the CS algorithm is 0.002914, and for the PSO algorithm is 0.004196, which indicates that the best is lower for the CS compared to the PSO. Moreover, better standard deviation and mean values were obtained for the CS method. The CS method allows us to obtain a solution near to the optimal value for all runs, which demonstrates that the efficiency of the CS algorithm is good. Thus, the probability of finding the global extreme is greater for the CS algorithm.

3. Mathematical Model of the BLDC Motor

In the mathematical model of the BLDC model, it was assumed that there are three concentrated phase windings in the stator of the motor. The magnetic flux in the rotor is excited by the permanent magnets mounted on its surface. The structure of the motor and the system of the electronic commutator is presented in Figure 5. The system of the electronic commutator is built from: (a) a microcontroller; (b) three Hall sensors; (c) six transistors ($T1$–$T6$); and (d) six diodes ($D1$–$D6$). The microprocessor supplies the transistor system with voltage U_s. The transistors distribute the voltage U_s on each BLDC phase winding.

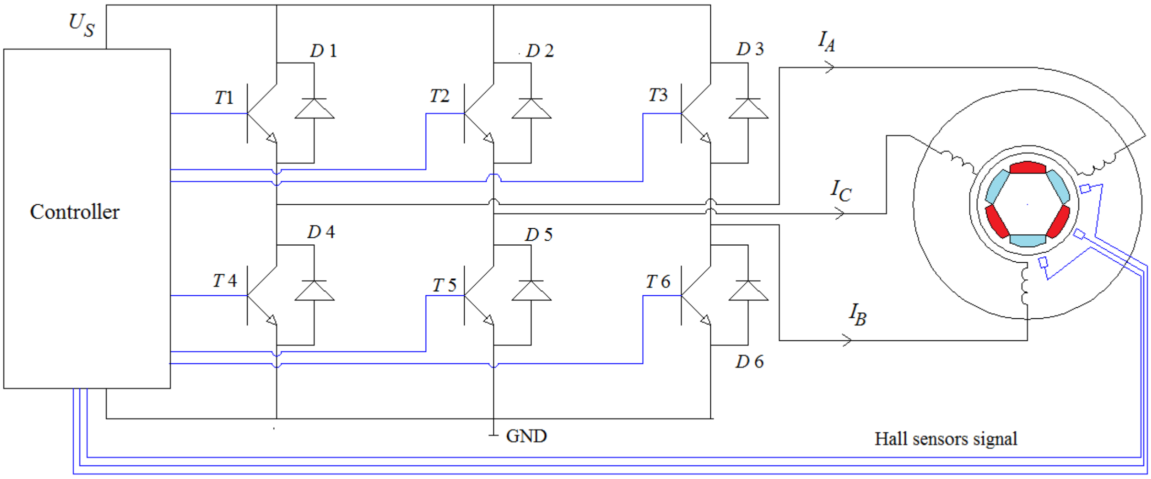


Figure 5. System of an electronic commutator with a BLDC motor.

The set of voltage equations for a three-phase motor are given in Equation (4):

$$\begin{bmatrix} u_A \\ u_b \\ u_C \end{bmatrix} = \begin{bmatrix} R_A & 0 & 0 \\ 0 & R_B & 0 \\ 0 & 0 & R_C \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \Psi_A \\ \Psi_B \\ \Psi_C \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} \Psi_A \\ \Psi_B \\ \Psi_C \end{bmatrix} = \begin{bmatrix} L_{AA} & L_{AB} & L_{AC} \\ L_{BA} & L_{BB} & L_{BC} \\ L_{CA} & L_{CB} & L_{CC} \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \begin{bmatrix} \Psi_{pA} \\ \Psi_{pB} \\ \Psi_{pC} \end{bmatrix} \tag{5}$$

where u_A, u_B, and u_C are the feeding phase voltage of the phases, i_A, i_B, and i_C are the phase currents, R_A, R_B, and R_C are the phase resistances, Ψ_A, Ψ_B and Ψ_C are the flux linkages coupled with the phase winding L_{AA}, L_{BB}, and L_{CC} are the self-inductances of the phase windings, L_{AB}, L_{AC}, L_{BA}, L_{BC}, L_{CA}, and L_{CB} are the mutual inductances between phases, and Ψ_{pA}, Ψ_{pB}, Ψ_{pC} are the fluxes generated by the permanent magnets linked to the phase windings.

The distribution of the magnetic field excited by the permanent magnets depends on the position of a rotor [33]. The flux linked with particularly concentrated windings made by the permanent magnets is recorded with the use of a dimensionless period function $\lambda(\alpha)$ defined in [32], and depends on the number and volume of magnets used in a rotor, following Equation (6):

$$\left[\Psi_{pA}\ \Psi_{pB}\ \Psi_{pC}\right]^T = \left[\Psi_m\lambda(\alpha)\ \Psi_m\lambda(\alpha-120)\ \Psi_m\lambda(\alpha+120)\right] \tag{6}$$

where Ψ_m is the amplitude of the 'linkage' flux generated by the permanent magnets, and $\lambda(\alpha)$ is the dimensionless periodic function.

If the core of the BLDC motor stator is symmetrical, then Equation (7) is obtained according to reference [34]:

$$\begin{bmatrix} \Psi_A \\ \Psi_B \\ \Psi_C \end{bmatrix} = \begin{bmatrix} L_{AA} & M & M \\ M & L_{BB} & M \\ M & M & L_{CC} \end{bmatrix} \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} + \begin{bmatrix} \Psi_{pA} \\ \Psi_{pB} \\ \Psi_{pC} \end{bmatrix} \tag{7}$$

where M is the mutual inductance between the phases.

In the case where a winding is connected without the zero wire, the voltage equations are even simpler. The flux equation of phase A is given by Equation (8):

$$\Psi_A = (L_{AA} - M)i_A + \Psi_{pA} \tag{8}$$

and it is given in this matrix form by Equation (9):

$$[\Psi] = [L_p][i] + [\Psi_p] \tag{9}$$

where $[L_p] = [L_{AA} - M\, L_{BB} - M\, L_{CC} - M]$ is the vector phase inductance, $[\Psi] = [\Psi_A\ \Psi_B\ \Psi_C]^T$ is the vector flux linkage, $[i] = [i_A\ i_B\ i_C]^T$ is the vector of phase currents, and $[\Psi_p] = [\Psi_{pA}\ \Psi_{pB}\ \Psi_{pC}]^T$ is the vector of fluxes generated by the permanent magnets that are linked to specific windings.

The full set of voltage equations of the BLDC motor is, therefore, obtained by Equation (10):

$$[u] = [R][i] + [L_p]\frac{\mathrm{d}}{\mathrm{d}t}[i] + \Psi_m\omega\frac{\mathrm{d}}{\mathrm{d}\alpha}[\Psi_p] \tag{10}$$

where ω is the angular velocity of the rotor, and $[u] = [u_A\ u_B\ u_C]^T$ is the vector of the phase supply voltage.

Assuming, that the electrical power generated in every winding is changed into mechanical power [35,36], the electromagnetic torque of the motor can be determined by Equation (11):

$$T = \frac{1}{\omega}(e_A i_A + e_B i_B + e_C i_C) \tag{11}$$

where e_A, e_B, and e_C are the values of back electromotive forces (EMF) inducted in particular phases [31].

During the analysis of the dynamic states (starting or a load torque change), the rotational velocity was changed significantly. In this case, it is crucial to include the mechanical equilibrium equation in the mathematical model of the motor according to Equation (12):

$$J\frac{\mathrm{d}\omega}{\mathrm{d}t} + B\omega = T - T_o \tag{12}$$

where J is the rotor's moment of inertia, B is the friction constant, T_o is the load torque of the motor, and T is the electromagnetic torque generated on the motor shaft.

One of the most important parameters of the permanent magnet motor in the steady-state analysis is the torque pulsation factor [37–39]. In the proposed mathematical model, the pulsation factor is calculated by Equation (13) according to [40]:

$$\varepsilon = \frac{T_{\max} - T_{\min}}{T_{av}} 100\% \tag{13}$$

where $T_{\max}$ and $T_{\min}$ are the maximum and minimum torque values, and T_{av} is the average torque.

4. Software to BLDC Analysis

The computer software for the analysis of the operating state of the BLDC motor was developed on the basis of the mathematical model described in Section 3. The script has been developed in Python 3.8 and Windows 10.

Before starting calculations, the user has to provide the rated data of the simulated BLDC motor. All the simulation calculations were performed for the BLDC motor with catalogue number 57BLR50. The rated data of the motor are listed in Table 3.

Table 3. The rated parameters of the BLDC motor.

Parameter	Unit	Value
Number of poles	-	4
Rated voltage	V	24
Rated current	A	5 ± 10%
Rated power	W	84
Rated speed	rpm	3500 ± 10%
Rated torque	mNm	230

In order to avoid imperfections in the production process of real devices [41], the values of the phase stator winding resistance, the self-phase inductance, and the mutual inductance between phases were determined by the use of the magnetostatic FEA model [42]. The FEM mesh with 34960 triangular elements has been applied. The demagnetization curve of permanent magnets has been approximated by a linear function. The backelectromotive forces (EMF) for the stator winding were also determined to estimate the maximum value and obtain more accurate results in the simplified model of a BLDC motor. The waveforms of back EMF are presented in Figure 6.

As a result of the FEM calculations, the following values were obtained: $R_A = R_B = R_C = 0.624\ \Omega$, $L_{AA} = L_{BB} = L_{CC} = 1.123$ mH and $L_{AB} = L_{AC} = L_{BA} = L_{BC} = L_{CA} = L_{CB} = 4.26$ µH.

Only some input data of the optimization procedure were determined by the use of the FEM model. The time of calculation of FEM parameters is less than 1% of the total calculation time of a single optimization procedure. Therefore, the numerical algorithm for determining the electromagnetic field that has been elaborated by the authors was applied. The algorithm was based on the Newton–Raphson process [43]. There are more effective methods for calculating the electromagnetic field distribution [44,45]. Due to the relatively short calculation time, we did not use these methods.

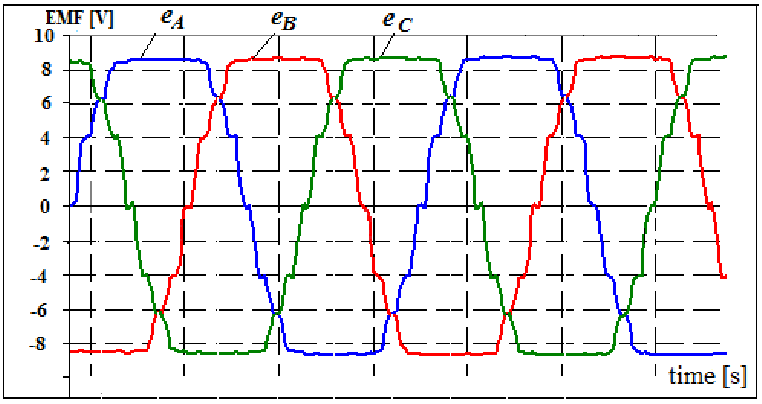
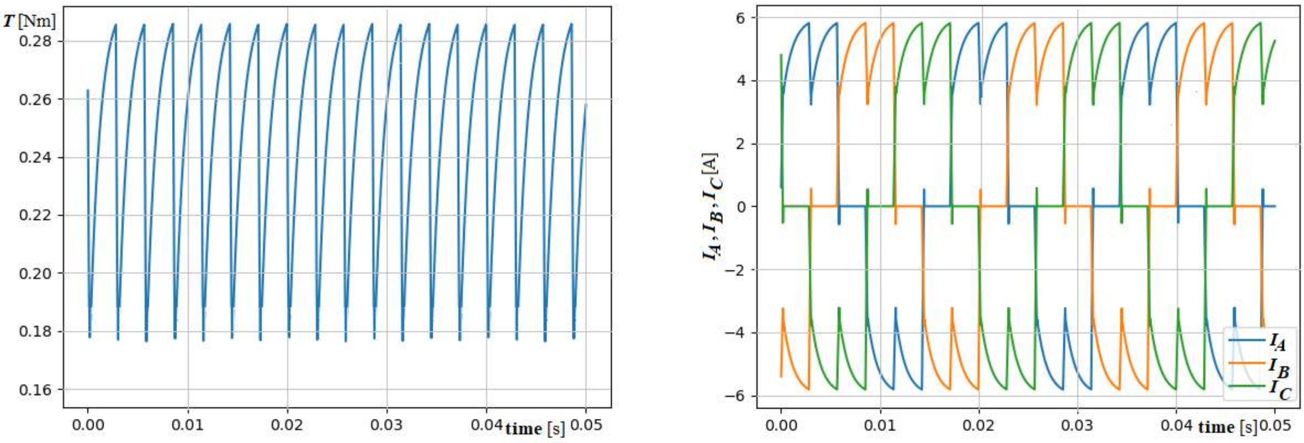


Figure 6. The backelectromotive forces induced in stator winding.

By using this software, the phase current waveforms and electromagnetic torque waveforms for the rated supply parameters and load torque equal to T_N were determined. The values of the rotor's moment of inertia is J = 128 g·cm^2, the friction constant is B = 0.085 mN·m·s/rad [46], and the time step length is Δt = 0.02 ms. These values were adopted. A waveform of the electromagnetic torque is shown in Figure 7a. The waveforms of the phase currents for the rated torque are presented in Figure 7b.



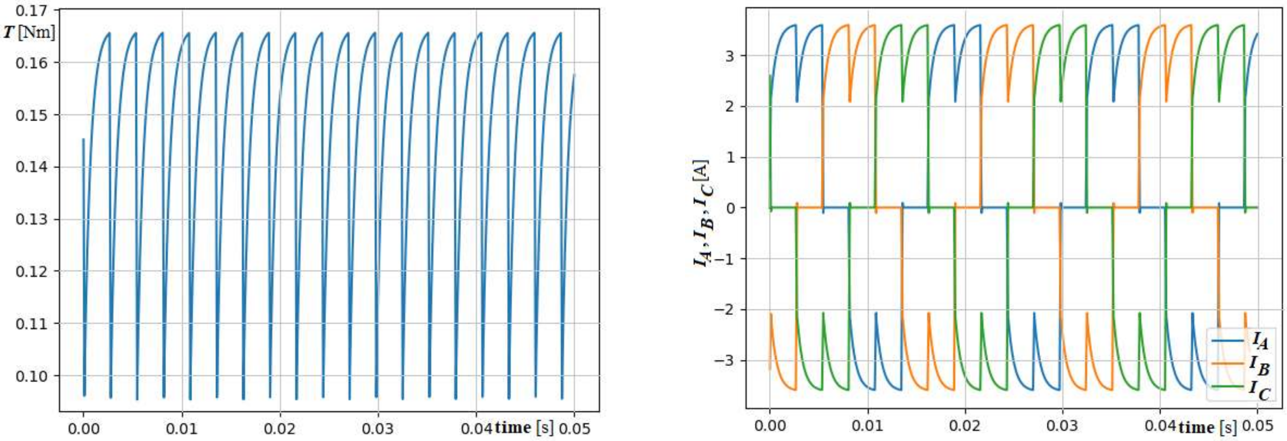
(**a**) (**b**)

Figure 7. Mechanical and electrical quantities waveforms: (**a**) the electromagnetic torque waveform for T_N; (**b**) the phase current waveforms for T_N.

The value of the pulsation factor was determined for the simulation time to be ranging between 0 s and 0.032 s. The value of this factor is ε = 44.32%.

Then calculations were made with the value of the load torque at 0.6 T_N, with ε = 53.19%. The resulting phase currents and electromagnetic torque are presented in Figure 8.

The simulation calculations for different values of load torque and rotational velocity were performed. The obtained results of simulation calculations are presented in Table 4.



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

Figure 8. Mechanical and electrical quantities waveforms for load torque equal to 0.6 T_N: (**a**) the electromagnetic torque waveform; (**b**) the phase current waveforms.

Table 4. Pulsation factors for various load torque.

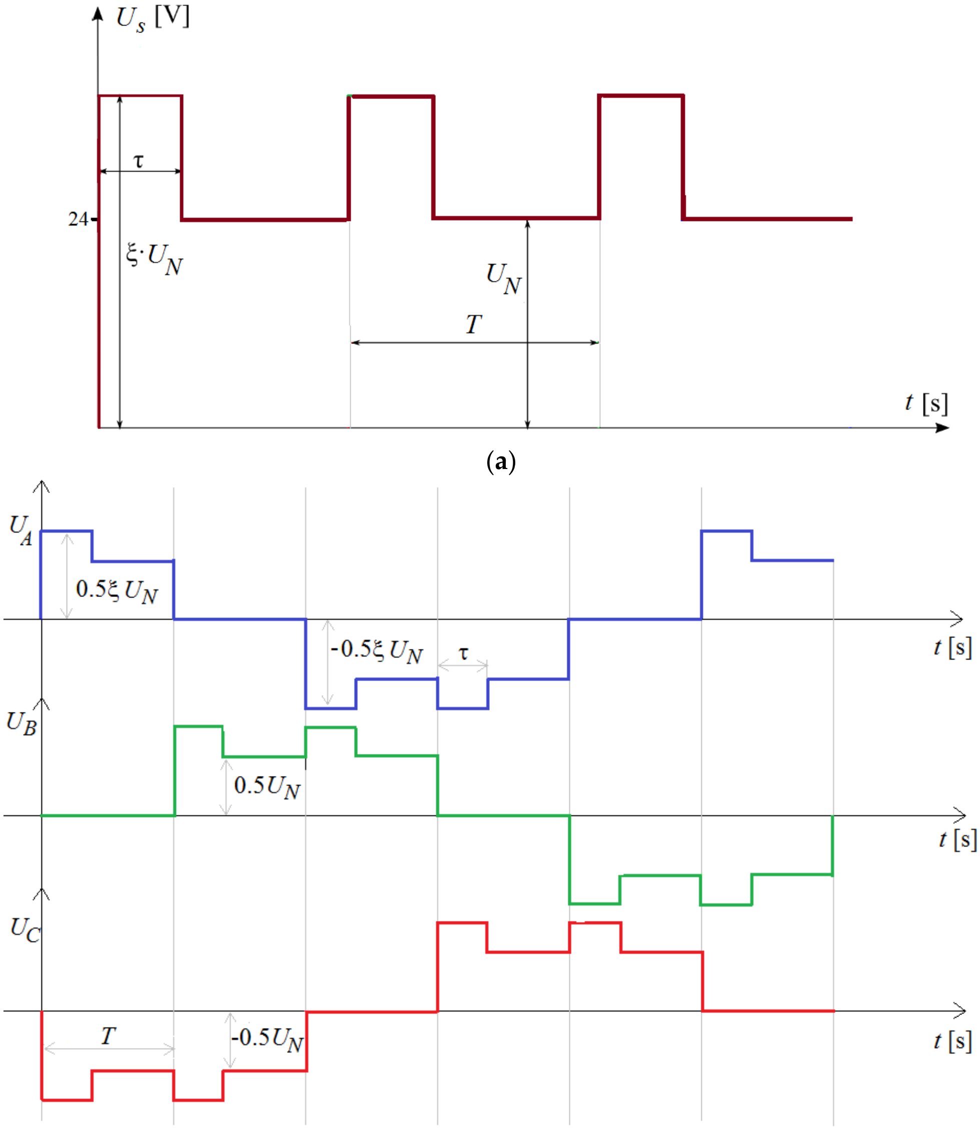
T_l (mNm)	ε (%)	n (rpm)
0.0	60.59	4000
50	56.48	3895
100	55.28	3790
120	54.26	3748
150	51.76	3685
170	48.58	3660
200	46.26	3580
230	44.23	3500

Based on the simulation results presented here, it can be observed that a higher value of pulsation factors was obtained for the lower value of the load torque. A higher mechanical load increases the mechanical power on the shaft. At constant voltage, the test machine draws a higher value of current from the stator windings for a bigger load torque. The value of winding currents depends on the mechanical loading. In order to obtain a constant value of the pulsation coefficient in the BLDC motor, it is necessary to modify the value of the waveform's supply voltages.

5. Formulation of an Optimization Problem

Torque ripples in the BLDC motors are caused by two different phenomena. The first source of torque ripple pulsation is related to interactions between the stator with slots and the permanent magnet rotor, referred to as the cogging torque [47]. The second source of torque ripple pulsation is the commutation process during the exchange of current between the two phases [48,49]. The commutation torque ripple can even reach about 50% of the average torque generated by the BLDC motor [49].

The optimization task consists of the selection of the two parameters describing the shape of a supply voltage waveform in order to obtain the maximum value of the average torque at the required value of the pulsation factor. Two design variables were taken into account: $s_1 = \xi$, and $s_2 = \tau$ (see Figure 9a,b). Design parameter ξ determines the value of the phase supply voltage in relation to the rated phase voltage, while variable τ defines the time in which the winding is supplied by increased voltage.



(**b**)

Figure 9. The waveforms of voltages in the system of an electronic commutator: (**a**) the supply voltage U_s waveform; (**b**) the waveforms of phases voltages.

The supply voltage U_s waveform is generated by the microcontroller (see Figure 5). U_s voltage is given to the electronic commutator. The controller controls the operation of an electronic commutator, i.e., a system of six transistors. The power supply to two phases of the stator winding is performed by switching transistors $T1$–$T6$ (see Figure 5). The two phases of the stator winding are supplied by U_s voltage at any time. Then, the voltage on a single phase is equal to half the voltage U_s. The waveforms of the voltages (U_A, U_B, and U_C) on individual phases are shown in Figure 9b. The phase voltages are input data in the mathematical model of the BLDC motor. They are used in Equation (4). The phase voltage values are calculated on the basis of the U_s waveform (see Figure 9a). The shape of the U_s waveform is defined by two design variables, s_1 and s_2. The optimization procedure through the variables s_1 and s_2 influences the functional parameters of the motor, i.e., T_{av} and the pulsation factor ε.

The objective function for the *i*-th cuckoo (the *i*-th variant of the supply voltage waveform shape) is given by Equation (14):

$$f_i(s_1,\, s_2) = \frac{T_i(s_1,\, s_2)}{T_{av0}} \tag{14}$$

where $T_i(s_1, s_2)$ is the value of the average electromagnetic torque for the *i*-th cuckoo, T_{av0} is the value of the average electromagnetic torque after the generation of the initial positions of the cuckoos/nests.

The optimization problem consists of the maximization of the objective function (14) taking into account the permissible value of the pulsation factor $\varepsilon \leq \varepsilon_p$. In the case of the constrained optimization problem, the primary objective function $f(s_1, s_2)$ is modified by penalizing for overstepping the required constraints [50–52]. Thus, the penalty term $p(s_1, s_2)$ is added to the primary objective function.

The constant static penalty approach has been taken into account in the developed algorithm [53], and $p(s_1, s_2)$ is obtained by Equation (15):

$$p_i(s_1,\, s_2) = \gamma\left(\frac{\varepsilon_i(s_1,\, s_2) - \varepsilon_p}{\varepsilon_p}\right) \tag{15}$$

where $\varepsilon_i(s_1, s_2)$ is the value of the pulsation factor for *i*-th the cuckoo, ε_p is the required value of the pulsation factor, and γ is the constant number dependent on solving the task [54].

Finally, the modified objective function [55,56] $h_i(s_1, s_2)$ consists of two components [8] and is calculated for the *i*-th cuckoo by the Equation (16):

$$h_i(s_1,\, s_2) = \begin{cases} f_i(s_1,\, s_2) & \text{for } \varepsilon_i(s_1,\, s_2) \leq \varepsilon_p \\ f_i(s_1,\, s_2) - p_i(s_1,\, s_2) & \text{for } \varepsilon_i(s_1,\, s_2) \geq \varepsilon_p \end{cases} \tag{16}$$

6. Results of Constrained Optimization

The constrained optimization calculations were performed using computer software developed in Python. The CS algorithm was selected to solve this optimization task. This is because in the last two years an increased interest in this optimization algorithm has been observed [57–59]. The software was designed by the authors and consists of two independent and cooperating parts. The first part contains a mathematical model of the BLDC. The master module forms a optimization algorithm that manages the work of the mathematical model of the motor. A block diagram of the software is presented in Figure 10.

Before starting on the optimization calculation, the user must declare the input data, which consists of: (a) the rated parameters of the studied BLDC motor; (b) self- and mutual inductance between phases; (c) the rotor's moment of inertia; (d) the friction constant; and (e) the permissible value of the pulsation factors. Next, the optimization procedure is run. The procedure creates an initial population of cuckoos and executes the calculations for the declared maximum number of iterations. From each individual, the optimization procedure imitates the calculations of the mathematical model for the given design variables (s_1 and s_2). The module with a mathematical model calculated the values of the average electromagnetic torque and the pulsation factor. After the mathematical model finishes the calculation for *i*-th individual, the values of the primary objective function (f_i), the penalty term (p_i) and the secondary objective function (h_i) are determined. The modified values of the above parameters (f, p, and h) are sent to the optimization procedure to evaluate of the *i*-th cuckoo.

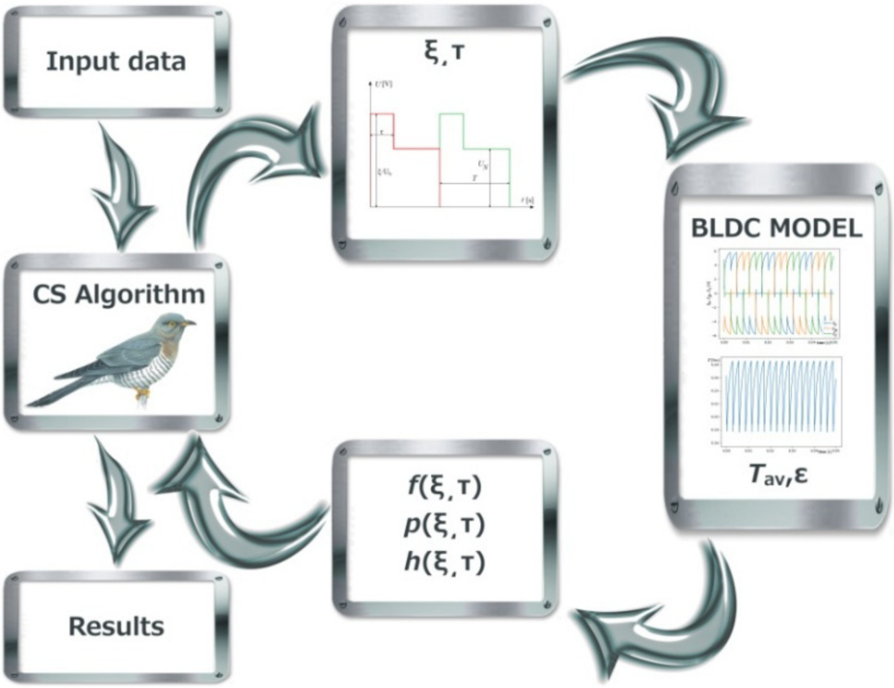


Figure 10. The block chart of the software.

In order to evaluate the performance of the computer software, a large number of simulations were conducted. The optimization calculations were performed for the following CS algorithm settings: the number of cuckoos is N = 54 [60–62], the number of nests is n = 72, the maximum number of iterations is $t_{\max}$ = 30, and the probability is p_a = 0.25. The value of the constant coefficient that determines the size of the penalty is γ = 0.55. For each of the analyzed permissible pulsation factors, five independent runs of the optimization software were carried out. The ranges of the design variables assumed during the optimization process are presented in Table 5.

Table 5. The assumed range of design variables.

Design Variable	Nature	Assumed Range
s_1	continuous	1.0–2.0
s_2	continuous	0.0–0.7

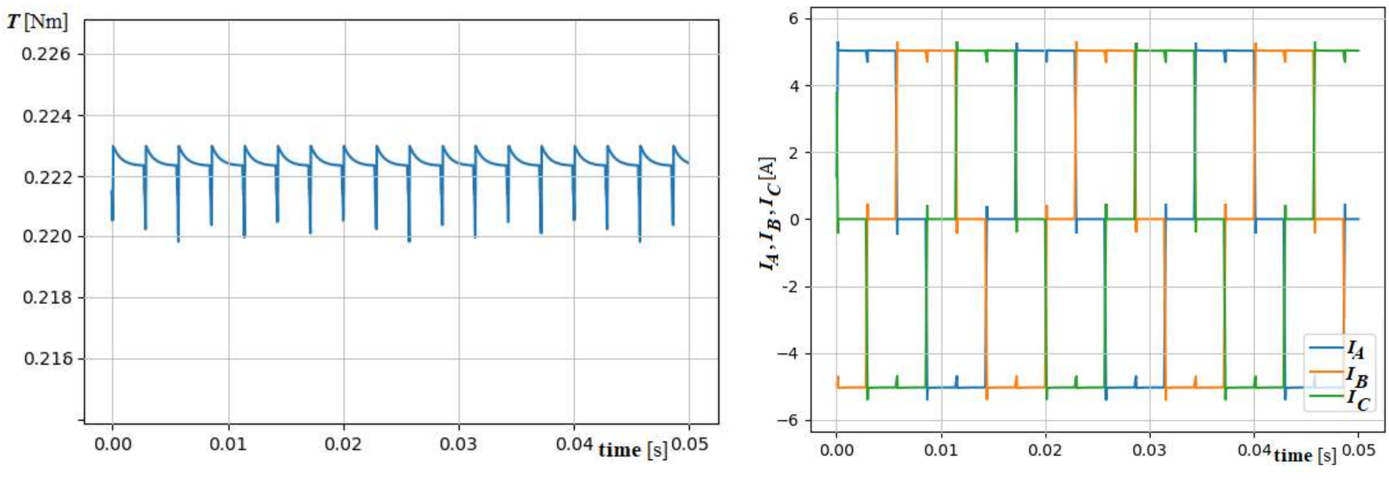
At the beginning, calculations were made for the required value of the pulsation factor ε_p = 5%. The optimization process was repeated five times for the different positions of nests. The best obtained result is presented in Table 6. The course of the optimization process for the subsequent iterations of the CS algorithm is presented in Table 6.

Table 6. The course of optimization process for selected iterations.

t	s_1	s_2	ε (%)	T_{av} (Nm)	p (s_1, s_2)	H (s_1, s_2)
1	1.8284	0.0541	13.962	0.2271	0.98582	0.010250
2	1.6508	0.1046	6.217	0.2263	0.13387	0.833364
3	1.6607	0.1012	5.977	0.2254	0.10747	0.855934
4	1.6618	0.0805	5.518	0.2239	0.05698	0.904297
5	1.6775	0.0785	5.108	0.2232	0.01188	0.946414
6	1.6843	0.0771	4.996	0.2222	0	0.946418
7	1.6843	0.0771	4.996	0.2222	0	0.946418
10	1.6843	0.0771	4.996	0.2222	0	0.946418
30	1.6843	0.0771	4.996	0.2222	0	0.946418

Based on the results obtained in Table 6, it can be noted that the optimal solution was achieved after six iterations of the cuckoo search optimization algorithm. The subsequent iterations of the algorithm did not lead to any better-quality solution. It is necessary to point out that the best improvements concerning the pulsation factor were obtained between the first and second iteration of the optimization algorithm.

The waveform of the electromagnetic torque for the required ε_p is shown in Figure 11a. The waveforms of the phase currents for the required ε_p are presented in Figure 11b.



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

Figure 11. Mechanical and electrical quantities waveforms for the required ε_p = 5%: (**a**) the electromagnetic torque waveform; (**b**) the phase current waveforms.

Next, optimization calculations were made for the less restrictive value of the pulsation factor, which is equal to ε_p = 22%. The optimization process was performed for the rated load torque at first. The optimization parameters for the selected iterations of the optimization algorithm are presented in Table 7.

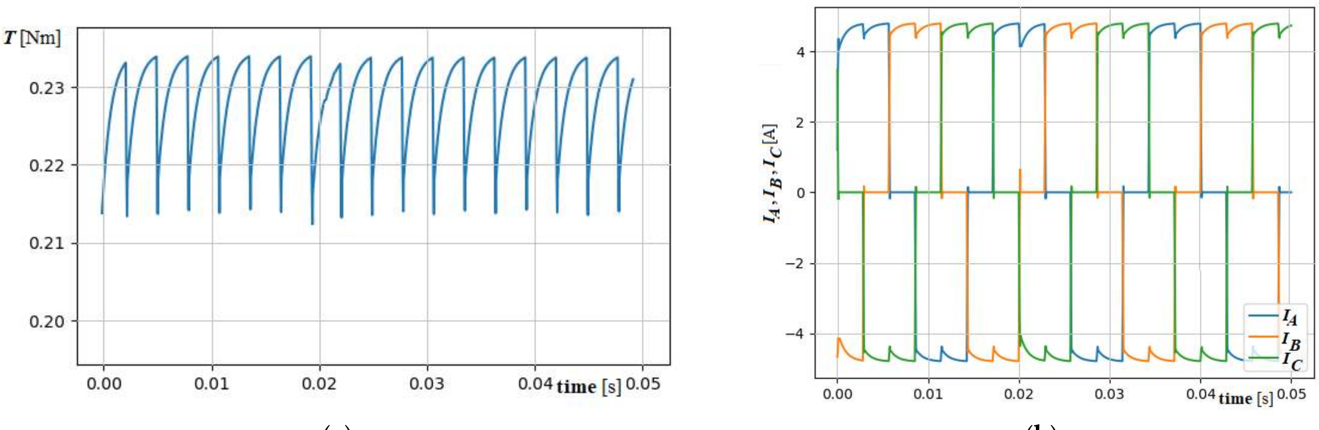
Table 7. The course of optimization process for selected iterations.

t	s_1	s_2	ε (%)	T_{av} (Nm)	p (s_1, s_2)	H (s_1, s_2)
1	1.2613	0.08971	27.018	0.2284	0.1255	0.8850
2	1.3325	0.12561	22.981	0.2274	0.0245	0.9729
3	1.3412	0.13029	22.164	0.2274	0.0041	0.9933
4	1.3421	0.13061	22.007	0.2270	0.0002	0.9955
5	1.3421	0.13061	22.007	0.2270	0.0002	0.9955
10	1.3421	0.13061	22.007	0.2270	0.0002	0.9955
30	1.3421	0.13061	22.007	0.2270	0.0002	0.9955

The waveforms for the optimal solution are presented in Figure 12a,b.

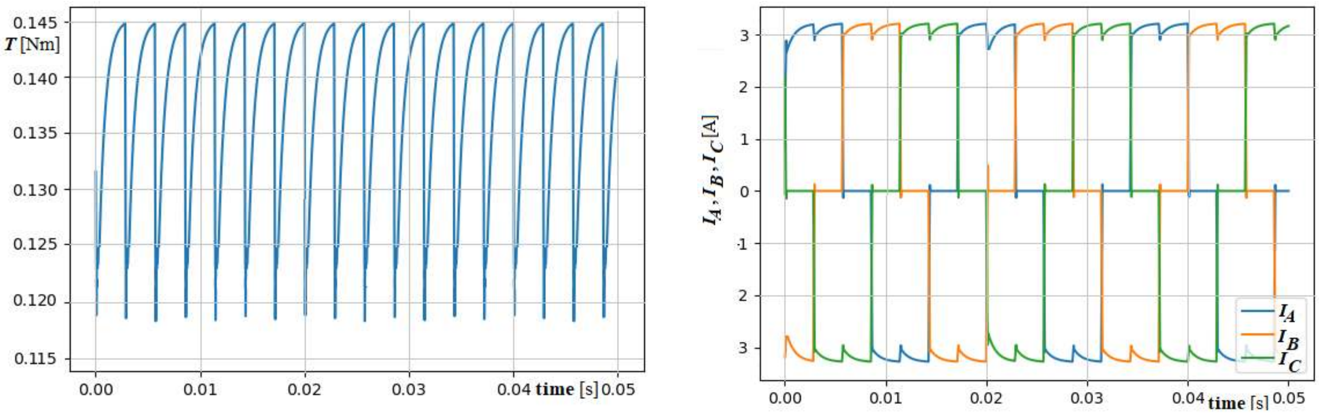
Then, optimization calculations were made for the lower value of the load torque of 0.6 T_N. The waveforms of electromagnetic torque and phase currents for permissible pulsation factor ε_p = 22% are shown in Figure 13a,b, respectively.

Phase currents for the rated voltage (see Figure 8b) and phase currents for the modified shape depend on the value of the load torque (see Figure 13b). The average currents for loading torque equal to 0.6 T_N are similar and do not depend on the pulsation factor. If we load machines by the same load torque, we will obtain a similar value of phase current. A similar phenomenon is observed for a brush DC motor, in which the current value is proportional to the load torque value.



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

Figure 12. Mechanical and electrical quantities waveforms for the required ε_p = 22%: (**a**) the electromagnetic torque waveform; (**b**) the phase current waveforms.



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

Figure 13. Mechanical and electrical quantities waveforms for the required ε_p = 22% and 0.6 T_N: (**a**) the electromagnetic torque waveform; (**b**) the phase current waveforms.

When comparing the courses of the optimization processes for the two different constraints, it can be concluded that, in the case of the less restrictive constraint (ε_p = 22%), the optimal solution was determined faster, i.e., after four iterations of the CS algorithm.

Figure 14 presents a comparison of the convergence curves for the two values of permissible pulsation factors equal to ε_p = 5% and ε_p = 22%. The figure shows the relative convergence of the modified objective function related to the optimal value. In the case of a gentler permissible pulsation factor, the result close to the optimal one will be obtained faster (see Figure 14).

Finally, the optimal results of the constrained optimization for the different values of load torque were calculated. The permissible value of the pulsation factor was equal to ε_p = 22%. Five independent optimization processes for each load torque were made. The initial populations were selected randomly. The comparison of the obtained results is shown in Table 8.

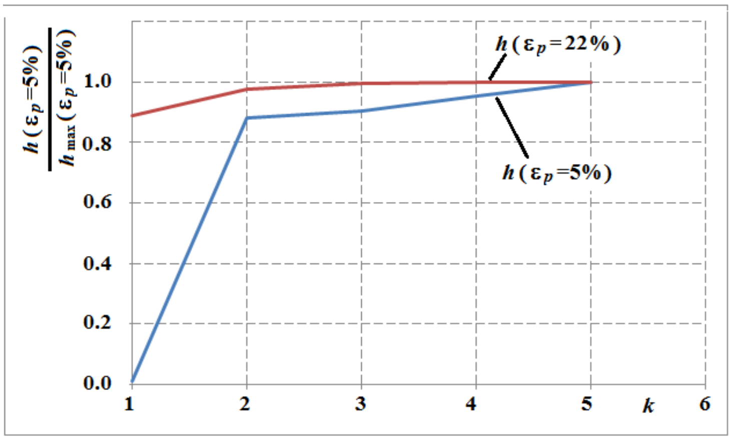


Figure 14. The convergence curves for different value of permissible values of pulsation factors.

Table 8. The value of pulsation factors for various load torque.

T_l (mNm)	s_1	s_2	ε (%)
20	1.4799	0.1032	21.995
70	1.4512	0.1101	21.999
100	1.4099	0.1193	21.997
150	1.3798	0.1256	21.998
170	1.3354	0.1143	21.993
200	1.3398	0.1287	22.001
230	1.3421	0.1306	22.007

To achieve the permissible value of the pulsation factor of ε = 22% for different values of load torque, the various parameters (ξ, and τ) describing the supply waveform should be applied. The analysis of obtained results shows that a high value of the pulsation factors requires the application of a higher modified supply voltage waveform.

7. Conclusions

The paper presents the application of the CS algorithm for the minimization of the commutation torque ripple in a brushless DC motor. The code was elaborated upon in Python for Windows. The structure of the computer software with all components has been presented in detail. The optimization task was the maximization of the average electromagnetic torque coupled with the constraint on the pulsation factor.

The obtained results of the computer simulation for the elaborated CS procedure are encouraging. Based on the presented results, in order to test the analytical function of a BLDC, the proposed CS procedure is much better in terms of effectiveness and quality when compared to the classical PSO approach. Moreover, according to the authors, the CS algorithm should guarantee a higher probability of finding the global extreme. After analyzing the optimization process for the CS algorithm, the static penalty function was applied to take into account the constraints.

The course of the optimization process was investigated for two different constraints: in the first case, a more restrictive permissible value of pulsation factor ε_p = 5% was imposed. The average value of the electromagnetic torque was 0.222, which is 5% lower in relation to the rated torque. The optimal solution was obtained after six iterations. In the second case, the optimization process was performed for the softer value of permissible pulsation factor ε_p = 22%. The optimal solution was obtained after four iterations of optimization algorithms, and the value of the average electromagnetic torque was 0.228. The obtained results confirm that for the less restrictive constraints the optimal solution can be found faster.

Based on the obtained results of the computer simulation, it can be concluded that the proposed approach allows obtaining the results of calculations with satisfactory accuracy with an acceptable calculation time. During the commutation phenomena, especially in supply systems with shaped values, there can be highly saturated subareas in the magnetic circuit of the motor. The difficulty is correctly estimating phase self-inductance and mutual inductance between two phases. Therefore, the values of the ξ coefficient must be carefully selected.

In future research, the authors want to focus on the design and execution of a laboratory stand to perform an experimental verification of research. Next, the performance of the CS algorithm will be compared with other heuristic optimization algorithms such as the genetic algorithm, particle swarm algorithm, or gray wolf algorithm. Moreover, additional constraint functions could be taken into consideration in future research to achieve a much more complicated constrained optimization task. Finally, the mathematical model of the BLDC motor will be expanded upon with thermal equations.

Author Contributions: Conceptualization, Ł.K. and F.P.G.M.; methodology, Ł.K.; optimization procedure, Ł.K.; BLDC model, Ł.K.; software, S.K. and Ł.K.; validation, Ł.K. and S.K.; formal analysis, Ł.K.; investigation, Ł.K.; formal analysis, Ł.K.; data curation, Ł.K.; writing—original draft preparation, Ł.K.; review and editing, F.P.G.M. and Ł.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Polish Government, grant number [0212/SBAD/0543], and the APC was funded by [0212/SBAD/0543].

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

References

1. Knypiński, Ł.; Pawełoszek, K.; Le Manech, Y. Optimization of low-power line-start PM motor using gray wolf metaheuristic algorithm. *Energies* **2020**, *13*, 1186. [CrossRef]
2. Nekoubin, A.; Soltani, J.; Dowlatshah, M. Multi-objective design optimization of five-phase fractional-slot concentrated-winding surface-mounted permanent-magnet machine. *Arch. Electr. Eng.* **2020**, *69*, 873–889.
3. Faramarzi Palangar, M.; Mahmoudi, A.; Solmaz Kahourzade, S.; Soong, W. Optimum Design of Line-Start Permanent-Magnet Synchronous Motor Using Mathematical Method. In Proceedings of the IEEE Energy Conversion Congress and Exposition (ECCE), Detroit, MI, USA, 11–15 October 2020. [CrossRef]
4. Belahcen, A.; Floran, M.; El-Hadi-Zaim, Z.; Kolondzovski, Z. Combined FE and particle swarm algorithm for optimization high speed PM synchronous machine. *COMPEL* **2015**, *34*, 475–484. [CrossRef]
5. Barański, M. FE analysis of coupled electromagnetic-thermal phenomena in the squirrel cage motor working at hight ambient temperature. *COMPEL* **2019**, *38*, 1120–1132. [CrossRef]
6. Knebl, L.; Bianchi, N.; Bacco, G.; Ondrusek, C. Synchronous Reluctance Motor analytical model cross-saturation and magnetization analysis. *Przegląd Elektrotechniczny* **2020**, *96*, 108–112. [CrossRef]
7. Singh Patel, K.; Pandey, A.K. Modeling and simulation of brushless DC motor using PWM control techniqe. *Int. J. Eng. Res. Appl.* **2013**, *3*, 612–620.
8. Nageswara Rao, P.; Devarapalli, R.; Márquez, F.P.G.; Malik, H. Global Sliding-Mode Suspension Control of Bearingless Switched Reluctance Motor under Eccentric Faults to Increase Reliability of Motor. *Energies* **2020**, *13*, 5485.
9. Knypiński, Ł. Adaptation of the penalty function method to genetic algorithm in electromagnetic devices designing. *COMPEL* **2019**, *38*, 1285–1294. [CrossRef]
10. Bora, T.C.; Coelho, L.; Lebensztajn, L. Bat-inspired optimization approach for brushless DC motor problem. *IEEE Trans. Magn.* **2012**, *48*, 947–950. [CrossRef]
11. Brisset, S.; Brochet, P. Analytical model for the optimal design of a brushless DC wheel motor. *COMPEL* **2005**, *20*, 829–848. [CrossRef]
12. Kiranyaz, S.; Ince, T.; Yildirim, A.; Gabbouj, M. Fractional particle swarm optimization in multidimensional search space. *IEEE Trans. Syst. Man Cybern.* **2010**, *40*, 298–319. [CrossRef] [PubMed]
13. Dey, B.; García Márquez, F.P.; Basak, S. Smart energy management of residential microgrid system by a novel hybrid MGWOSCACSA algorithm. *Energies* **2020**, *13*, 3500. [CrossRef]
14. Knypiński, Ł. Constrained optimization of line-start PM motor based on the gray wolf optimizer. *Eksploatacja i Niezawodnosc—Maint. Reliab.* **2021**, *23*, 1–10. [CrossRef]
15. Nagaraj, B.; Vijayakumar, P. A Comparative Study of PID Controller Tuning Using GA, EP, PSO and ACO. *J. Autom. Mob. Robot. Intell. Syst.* **2011**, *5*, 42–48.

16. Peng, K.; Chen, Z.; Huang, L.; Wu, X. Application of cuckoo search algorithm for texture recognition based on water areas. In Proceedings of the Tenth International Conference on Digital Image Processing, Shanghai, China, 11–14 May 2018. [CrossRef]
17. Aziz, M.; Marwanza, I.; Hartrnti, N.; Ghifari, M.; Angrahadi, A. Application of Cuckoo Search method in 3D slope stability analysis for limestone quarry mine. *Indones. Min. J.* **2020**, *23*, 7–65.
18. Stojanovic, V.; Nedic, N.; Prsic, D.; Dubonjic, L.; Djordjevic, V. Application of cuckoo search algorithm to constrained control problem of a parallel robot platform. *Int. J. Adv. Manuf. Technol.* **2016**, *87*, 2497–2507. [CrossRef]
19. Jalal, M.; Goharzay, M. Cuckoo search algorithm for applied structural and design optimization: Float system for experimental setups. *J. Comput. Des. Eng.* **2019**, *6*, 159–172. [CrossRef]
20. Kulkarni, O.; Kulkarni, N.; Kulkarni, A.; Kakandikar, G. Constrained Cohort Intelligence using Static and Dynamic Penalty Function Approach for Mechanical Components Design. *Int. J. Parallel Emergent Distrib. Syst.* **2018**, *33*, 27–41. [CrossRef]
21. Cvetkovski, G.; Petkovska, L.; Lefley, P. Cuckoo search as a tool for optimal design of PM brushless DC motor. *COMPEL* **2018**, *37*, 1732–1743. [CrossRef]
22. Morelli, F.; Møller, A.P.; Nelson, E.; Benedetti, Y.; Liang, W.; Šímová, P.; Moretti, M.; Tryjanowski, P. The common cuckoo is an effective indicator of high bird species richness in Asia and Europe. *Sci. Rep.* **2017**, *7*, 4376. [CrossRef]
23. Yang, X.S.; Deb, S. Cuckoo search via Levy flights. In Proceedings of the World Congress on Nature and Biologically Inspired Computing–NaBIC'2009, Coimbatore, India, 9–11 December 2009. [CrossRef]
24. Li, L.; Xiao, D.; Lie, H.; Zhang, T.; Tian, T. Using Cuckoo Search algorithm with Q-L learning genetic operation to solve the problem of logistic distribution center location. *Mathematics* **2020**, *8*, 149. [CrossRef]
25. Li, J.; Li, Y.; Tian, S.; Xia, J. An improved cuckoo search algorithm with self-adaptive knowledge learning. *Neural Comput. Appl.* **2020**, *32*, 11967–11997. [CrossRef]
26. Marelia, M.; Twalab, B. An adaptive Cuckoo search algorithm for optimization. *Appl. Comput. Inform.* **2018**, *14*, 107–115. [CrossRef]
27. Khluabwannarat, P.; Puangdownreong, D. Parallel Flower Pollination Algorithm and Its Application to Fractional-Order PID Controller Design Optimization for BLDC Motor Speed Control System. *Przegląd Elektrotechniczny* **2020**, *96*, 78–83. [CrossRef]
28. Mesa, A.; Castromayor, K.; Garillos-Manliguez, C.; Calag, V. Cuckoo search via Levy flights applied to uncapacitated facility location problem. *J. Ind. Eng. Int.* **2018**, *14*, 585–592. [CrossRef]
29. Wang, L.; Guo, H.; Marignetti, F.; Shaver, C.; Bianchi, N. Cuckoo Search Algorithm for Multi-Objective Optimization of Transient Starting Characteristics of a Self-Starting HVPMSM. *IEEE Trans. Energy Convers.* **2021**. early access.
30. Yang, X.S.; Deb, S. Multiobjective cuckoo search for design optimization. *Comput. Oper. Res.* **2013**, *40*, 1616–1624. [CrossRef]
31. Devarapalli, R.; Kumar Sinh, N.; Bathinavenkateswara, R.; Knypiński, Ł.; Jaya Naga, N.; Garcia Marquez, F.P. Allocation of real power generation based on computing over all generation cost: An approach of Salp Swarm Algorithm. *Arch. Electr. Eng.* **2021**, *78*, 337–349.
32. Zhu, X.; Li, N.; Pan, Y. Optimization Performance Comparison of Three Different Group Intelligence Algorithms on a SVM for Hyperspectral Imagery Classification. *Remote Sens.* **2019**, *11*, 734. [CrossRef]
33. Zawirski, K.; Deskur, J.; Kaczmarek, T. *Electric Drive Automation*; Poznan University of Technology Press: Poznan, Poland, 2012.
34. Mümtaz, M. Analysis and Design Optimization of Tubular Linear Voice Coil Motor for High Thrust Force and Low Copper Loss. *IEEE Can. J. Electr. Eng. Comput. Eng.* **2021**, *44*, 165–170.
35. Zhu, Z.Q.; Howe, D. Electrical machines and drives for electric, hybrid and fuel cell vehicles. *Proc. IEEE* **2007**, *85*, 746–765. [CrossRef]
36. Coballes-Pantoja, J.; Gómez-Fuentes, R.; Noriega, J.R.; García-Delgado, L.A. Parallel loop control for torque and angular velocity of BLDC motors with DTC commutation. *Electronics* **2020**, *9*, 279. [CrossRef]
37. Kommula, B.N.; Kota, V.R. Direct instantaneous torque control of Brushless DC motor using firefly Algorithm based fractional order PID controller. *J. King Saud Univ. Eng. Sci.* **2020**, *32*, 133–140. [CrossRef]
38. Młot, A.; Kowol, M.; Kołodziej, J.; Lechowicz, A.; Skrobotowicz, P. Analysis of IPM motor parameters in an 80-kW traction motor. *Arch. Electr. Eng.* **2020**, *69*, 467–481.
39. Jędryczka, C.; Danielczyk, D.; Szeląg, W. Torque Ripple Minimization of the Permanent Magnet Synchronous Machine by Modulation of the Phase Currents. *Sensors* **2020**, *20*, 2406. [CrossRef]
40. Stachowiak, D. Edge element analysis of brushless motors with inhomogeneously. *COMPEL* **2004**, *23*, 1119–1128. [CrossRef]
41. Bucolo, M.; Buscarino, A.; Famoso, C.; Fortuna, L.; Frasca, M. Control of imperfect dynamical systems. *Nonlinear Dyn.* **2019**, *98*, 2989–2999. [CrossRef]
42. Knypiński, Ł.; Nowak, L. Fidel-circuit simulation of the dynamics of the outer rotor permanent magnet brushless DC motor. *COMPEL* **2011**, *30*, 440–929. [CrossRef]
43. Idziak, P.; Kowalski, K.; Nowak, L.; Knypiński, Ł. FE transient analysis of magneostrictive actuator. *Int. J. Appl. Electomagnet. Mech.* **2016**, *51*, S81–S87. [CrossRef]
44. Cherif, R.; Tang, Z.; Guyomarch, F.; Chevallier, L.; Le Menach, Y. An Improved Newton Method Based on Choosing Initial Guess Applied to Scalar Formulation in Nonlinear Magnetostatics. *IEEE Trans. Magn.* **2019**, *55*, 7202204. [CrossRef]
45. Demenko, A.; Sykulski, J. On the equivalence of finite difference and edge element formulations in magnetic field analysis using vector potential. *COMPEL* **2014**, *33*, 47–55. [CrossRef]

46. Hoyos Velasco, F.; Candelo-Becerra, J.; Rincón Santamaría, A. Dynamic Analysis of a Permanent Magnet DC Motor Using a Buck Converter Controlled by ZAD-FPIC. *Energies* **2018**, *11*, 3388. [CrossRef]
47. Friedrch, L.; Curti, M.; Gysen, B.; Lomonova, L. High-order methods applied to nonlinear magnetostatic problems. *Math. Comput. Appl.* **2019**, *24*, 19. [CrossRef]
48. Dini, P.; Saponara, S. Cogging torque reduction in Brushless Motors by a nonlinear control technique. *Energies* **2019**, *12*, 87. [CrossRef]
49. Baszynski, M. Torque ripple reduction in BLDC motor based on a PWM technique for open-end winding. *Arch. Electr. Eng.* **2021**, *70*, 5–23.
50. Li, X.; Jiang, G.; Chen, W.; Shi, T.; Zhang, G.; Geng, Q. Commutation Torque Ripple Suppression Strategy of Brushless DC Motor Considering Back Electromotive Force Variation. *Energies* **2019**, *12*, 1932. [CrossRef]
51. Liu, Y.; Zhu, Z.Q.; Howe, D. Commutation-torque-ripple minimization in direct-torque-controlled PM Brushless DC drives. *IEEE Trans. Ind. Appl.* **2007**, *43*, 1012–1021. [CrossRef]
52. Yeniay, O. Penalty function methods for constrained optimization with genetic algorithms. *Math. Comput. Appl.* **2005**, *10*, 45–56. [CrossRef]
53. Mutluer, M.; Sahman, A.; Cunkas, M. Heuristic optimization based on penalty approach for surface permanent magnet synchronous machines. *Arab. J. Sci. Eng.* **2020**, *45*, 6751–6767. [CrossRef]
54. Puzzi, S.; Carpinteri, A. A double-multiplicative dynamic penalty approach for constrained evolutionary optimization. *Struct. Multidiscip. Optim.* **2008**, *35*, 431–445. [CrossRef]
55. Jayswala, A.; Antczak, T.; Jha, S. Second order modified objective function method for twice differentiable vector optimization problems over cone constraints. *Numer. Algebra Control Optim.* **2019**, *9*, 133–145. [CrossRef]
56. Nowak, L.; Knypiński, Ł.; Jędryczka, C.; Kowalski, K. Decomposition of the compromise objective function in the permanent magnet synchronous motor optimization. *COMPEL* **2015**, *34*, 496–504. [CrossRef]
57. Tsipianitis, A.; Tsompanakis, Y. Improved Cuckoo Search algorithmic variants for constrained nonlinear optimization. *Adv. Eng. Softw.* **2020**, *149*, 102865. [CrossRef]
58. Prakash Agrawal, A.; Kaur, A. Performance comparison of Bat search and Cuckoo search using software artifact infrastructure repository and regression testing. *Int. J. Adv. Intell. Paradig.* **2021**, *18*, 99–118. [CrossRef]
59. Zheyuan, C.; ThaeerHammid, A.; Noori Kareem, A.; Jiang, M.; Mohammed, M.; Manoj Kumar, N.A. Rigid Cuckoo Search Algorithm for Solving Short-Term Hydrothermal Scheduling Problem. *Sustainability* **2021**, *13*, 4277. [CrossRef]
60. Barbosa, C.; Vasconcelos, G. Analysis of Cuckoo Search Efficiency. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019. [CrossRef]
61. Nguyen, K.; Fujita, G.; Dieu, V. Cuckoo search algorithm for optimal placement and sizing of static var compensator in large-escale power systems. *JAISCR* **2016**, *6*, 59–68.
62. Zhang, L.; Lu, Y.; Luo, Y.; Zhang, S. Improved cuckoo search algorithm and its application to permutation flow shop scheduling problem. *J. Algorithms Comput. Technol.* **2020**, *14*, 1748302620962403. [CrossRef]