processes

MDPI

Article

Modified Whale Algorithm-Based Optimization for Fractional Order Concurrent Diminution of Torque Ripple in Switch Reluctance Motor for EV Applications

Nutan Saha [1] and Prakash Chandra Mishra [2,*]

[1] Department of Electrical Engineering, Veer Surendra Sai University of Technology, Burla 768018, India
[2] Department of Mechanical Engineering, Veer Surendra Sai University of Technology, Burla 768018, India
* Correspondence: pcmishra_me@vssut.ac.in

Abstract: This work proposes a new multi-objective optimization technique for concurrent diminution of torque ripple with the regulation of the speed of a 75 Kilowatt, 8/6, 4-phase SRM based on a double close loop, modified whale algorithm optimized fractional order proportional integral (MWAO FO-PI) control with a commutation angle controller. The system is analyzed and designed in MATLAB/SIMULINK environment. First, the performance of MWAO is tested on 30-dimensional standard benchmark functions. It is found that MWAO performance is better when examined on 30-dimensional standard benchmark functions, compared with WOA, and another six recently proposed state-of-art functions. Then, a double loop control based on the MWAO FO-PI controller is designed and implemented for concurrent diminution of torque ripple with the regulation of the speed of a 75 Kilowatt, 8/6, 4-phase SRM with a commutation angle controller. It was found that the percentage improvement achieved in the combined objective optimization function with the MWAO FO-PI controller was 10.044% in comparison with the MWAO PI controller, and 9.0597% compared with the WOA PI controller. It is also proved that MWAO FO-PI-based double close loop control of SRM provides less torque ripple, better tracking of speed with a reference value of speed and a better current profile in comparison with the MWAO PI controller and WOA PI controller. From all the above analysis, the conclusion is reached that the MWAO FO-PI controller provides very good overall system operational performance compared with MWAO PI and WOA PI controllers. The conclusion is reached based on simulation analysis and experimental validation is lacking.

Keywords: switched reluctance motor (SRM); torque ripple; speed tracking; cosine adapted modified whale algorithm optimized fractional order proportional integral (MWAO FO-PI) control

Citation: Saha, N.; Mishra, P.C. Modified Whale Algorithm-Based Optimization for Fractional Order Concurrent Diminution of Torque Ripple in Switch Reluctance Motor for EV Applications. *Processes* **2023**, *11*, 1226. https://doi.org/10.3390/pr11041226

Academic Editor: Enrique Rosales-Asensio

Received: 10 March 2023
Revised: 4 April 2023
Accepted: 6 April 2023
Published: 16 April 2023

1. Introduction

Cost arbitrariness and non-availableness of rare earth elements led researchers to look for motor drives for electric vehicle(EV) applications, that are free from magnet configurations. The magnet-free drive configurations are maintenance-free, high-power density, and rugged [1–3]. But when used in EV drive applications, SRM is assisted with high torque ripple and acoustic noise. As SRM is governed by the non-linear relationship of the torque current profile with the rotor position [4]. This causes SRM to be assisted with acoustic noise and high torque pulsation [1–3] in EV drive applications. To overcome this gloomy area of SRM many improvements are suggested by researchers [5]. The saliency in the construction of stator and rotor and nonlinear characteristics cause it's difficult to control SRM [6]. Progressive development in power electronic devices, leads to the designing of a proper controller. With the advent of such controller, the minimization of torque pulsation and speed control of SRM is possible.

Thus, a control system could be designed keeping in mind its nonlinear magnetic characteristics, and intelligent selection of switching angle, current, and voltage values [5,7–9]. It is reported that controlling the current profile can also decrease the torque ripple. Many

methods have been used for controlling SRM's stator current profile and hence reducing torque ripple. Many methods [10,11] such as fuzzy logic and artificial neural networks can be used for current profile control so that it lessens the torque ripple. Whereas, implementing such intelligent techniques requires expert depth of knowledge. In spite of the unavailability of a formal model concept/theory and again having high mathematical complications, such systems are suitable for implementing such approaches. Optimization methods bring in the use of huge amounts of system information for addressing system problems with deep-seated models. Thus, they have strong workability. Contemporary heuristic optimization methods are very suitable for the designing and formulation of controllers [12]. Metaheuristic optimization techniques [12] are reported to be of high computational efficiency. These techniques are extensively classified [13] into evolutionary methods, swarm methods, and trajectory methods.

The whale optimization algorithm (WOA) [14] imitates the hunting style of humpback whales with the bubble net hunting technique. The differential search algorithm (DSA) [15] is encouraged by a Brownian-type random walk motion adopted by organisms for migration. The lightning search algorithm (LSA) [16] is developed on the natural occurrence of lightning and with the system of step leader action. To address the projectile transition, three types of projectiles were developed, step leader population, space projectile, and lead projectile. The harmonic search algorithm (HSA) [17] is inspired by musicians and used for producing perfect harmony. In the backtracking search algorithm (BSA) [18], adaptive control parameters depending on global and local knowledge of the swarm in the present iteration is adopted to adapt the individual search length. This brings a balance between exploitation and exploration ability. Particle swarm optimization (PSO) [19] uses the intelligence and propagation skill of a swarm. This technique solves a problem by using the social interaction skill of a swarm. It uses a number of searching agents moving around in searching for the best solution. The firefly algorithm (FFA) [20] implements global communication skills between various swarming agents.

Optimization algorithms [17,21] are efficient methods to find solutions to many nonlinear real-world problems. As NFL (No Free Lunch) theory explains, one single optimization algorithm cannot provide satisfactory results to all types of problems. Thus, there is always an opportunity for the discovery and advancement of new optimization techniques to suit that concerned problem. This causes motivation for the present research work.

Optimization techniques are nature inspired and broadly categorized into three types. These are the evolutionary method, trajectory method, and swarm method [15,22]. Researchers all over the world suggest that [23] these three methods can lead to improvement in optimization techniques. These are (i) putting forward new optimization techniques, (ii) improving present techniques, and (iii) hybridization of optimization techniques. The third one, hybridization of optimization techniques [23], is one of the popular and efficient methods.

Problem-solving with a multi-objective optimization process has many advantages in comparison with solving problems with single-objective problem formulation [24]. It is reported in [24] that multi-objective optimization can deal with many objectives at a time. Multi-objective optimization-solving scheme outputs better results when the objectives are correlated to each other as compared with single-objective optimization problem-solving methods. Again, single objective optimization problem-solving method takes more time and is also a cumbersome process for finding the parameters. Whereas a multi-objective optimization problem takes less time but is difficult to frame and requires the depth of optimization knowledge.

Mirjalili et al. [14] introduced a whale optimization algorithm (WOA). It emulates the hunting proficiency of humpback whales. WOA has a propensity to get trapped in local optima [25,26]. As there is an expansion in the search space dimension, it tends to have low convergence. This increases the chance of making changes in the technique for performance enhancement. One of the well-known controllers is the PI controller, favorable to industrial personnel as it is simple and easy to handle. However, the FO-PI [27,28] controller has some more variables to tune as compared with the PI controller. This gives some additional

degree of freedom to enable advancement in execution. Podlubny et al. [27] explain the workings of the FO-PI controller. FO-PI can be simply explained by, where γ represents the integrator order. Ghoudelbourk [29] discusses the application of FO-PI control in the application of SRM as an electric vehicle using a fuzzy logic controller.

In the present work, performance assessment and comparison of the cosine adapted modified whale algorithm optimized fractional order proportional integral (MWAO FO-PI) controller and MWAO PI controller is executed for concurrent diminution of torque ripple with regulation of speed of a 75 KW, 8/6, 4-phase SRM based on double close loop control. The controllers are designed depending on a modified whale optimization algorithm (MWAO) and WOA optimization scheme. In the modified whale algorithm of the optimization method, WOA is remodeled by incorporating the following changes to bring MWAO into existence. The first one is implementing a cosine function for drooping the control parameter and the second one is imbibing amendment factors for upgrading the position of search agents.

In the present work, the efficacy of MWAO is analyzed for 30-dimensional benchmark functions. Adding to it, a multiobjective optimization real world engineering problem for SRM's control based on MWAO is also devised using a fractional order proportional integral (FO-PI) controller. Mathematical modeling of SRM is provided in Section 2. The control of SRM is focused in Section 3. Section 4 discusses the MWAO technique. Section 5 justifies the correction factors selection. Section 6 analyses the outcomes derived by executing the MWAO method. Section 7 discusses the control of SRM by implementing MWAO-based fractional order controller and formulating a problem for multiobjective optimization.

2. Analytical Modeling and Analysis of an SRM Drive System

A switched reluctance motor is characterized by double saliency. Windings are only present in the stator and no windings are present in the rotor. The rotor is constructed of laminated silicon steel. With the supply of electric power to the stator, the rotor tries to attend a minimum reluctance position, and hence it experiences a reluctance torque to move. When an electric supply is provided in the proper sequence to stator windings, the rotor experiences a continuous rotating torque. Analytical modeling of four phases, 8/6, 75 KW, and SRM is detailed below. The input voltage is applied across each phase of winding. For simplicity of analysis, the effect of mutual inductances between phases is ignored. As the stator and rotor have saliency, the stator flux exhibits a nonlinear correlation with the rotor position (θ) and stator current (i).

The input phase voltage is applied to the stator winding of SRM. For simplicity of analysis, the coupled mutual inductance among the different phases of the stator is not taken into account.

The saliency of the rotor and stator structure is derived for nonlinear correlation [11,12] of stator flux linkage with the rotor position (θ) and stator current (i). The flux linkage $\Psi_s(t)$ is expressed as:

$$\Psi_s(t) = \Psi_s(i, \theta) \tag{1}$$

The stator's phase voltage is obtained by the below equation:

$$\Psi_s(t) = \int_0^t (V_S - R_S I_S) dt \tag{2}$$

In the above equation, R_S symbolizes resistances of the stator winding. The flux linkage is $\Psi_s(t)$. The stator phase voltage is V_S. The rotor position is θ. The stator current is I_S, obtained out of magnetic characteristics. The flux linkage ($\Psi_s(t)$) exhibits a non-linear relationship to I_S and θ. The mathematical equation for stator current $i(\Psi, \theta)$ is derived out of magnetization characteristics. $\Psi_s(t)$, V_S, and I_S are vector quantities. The magnetization characteristics of the concerned machine is shown in Figure 1.

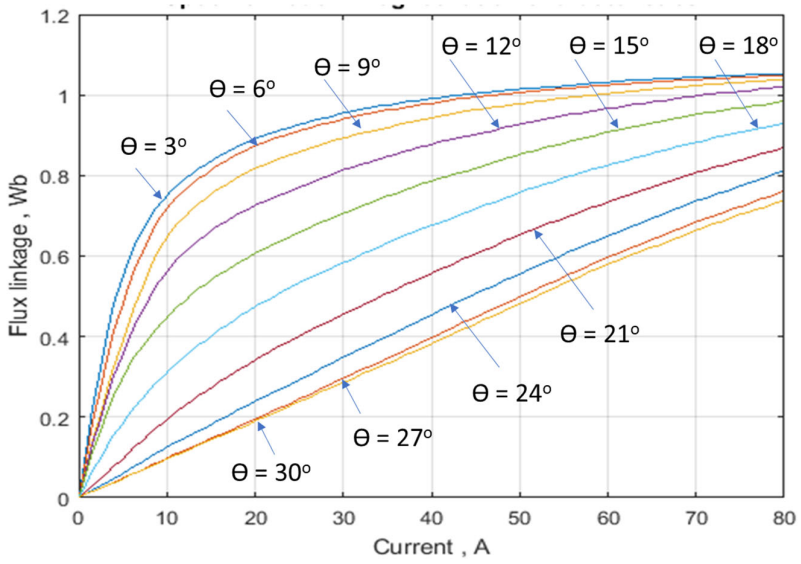


Figure 1. Magnetization characteristics of 75 KW, 8/6, 4 phase SRM at different rotor positions.

In Equation (3), $T_e(i,\theta)$ is the electromagnetic torque and is extracted from the machine's coenergy:

$$T_e(i,\theta) = \frac{\partial w'(i,\theta)}{\partial \theta} \tag{3}$$

The term $w'(i,\theta)$ is coenergy and is obtained below:

$$w'(i,\theta) = \int_0^i \Psi(i,\theta) \tag{4}$$

The sum total of all phase torque is the total electromagnetic torque (T_e) produced by the machine.

$$T_e = J\frac{\partial \omega_m}{\partial t} + B\omega_m + T_l \tag{5}$$

Here, ω_m is the angular velocity of the motor, J represents moment of inertia, load torque is T_l, and the coefficient of friction is B.

3. Switched Reluctance Motor Control

Figure 2 demonstrates a multi-objective optimization scheme of concurrent diminution of torque ripple with tracking of speed and reduction of current error using a double close loop MWAO optimized FO-PI controller. This control shown in Figure 2 accomplishes the following goals.

1. Torque ripple diminution,
2. Tracking of reference speed,
3. Reducing current error.

The task is accomplished by formulating a multi-objective optimization problem for a double close loop cosine adapted modified whale optimized fractional order proportional integral (MWAO FO-PI) speed controller, MWAO FO-PI current controller, and commutation angle controller. Rotor position feedback information is provided to the commutation angle controller. The commutation angle controller takes the decision of providing a positive current pulse during the stator's rising inductance profile. Thus, command for the turn-on angle (θ_{ON}) and turn-off angle (θ_{OFF}) for electrical switches is provided by the commutation angle controller.

The current hysteresis controller regulates the stator current.

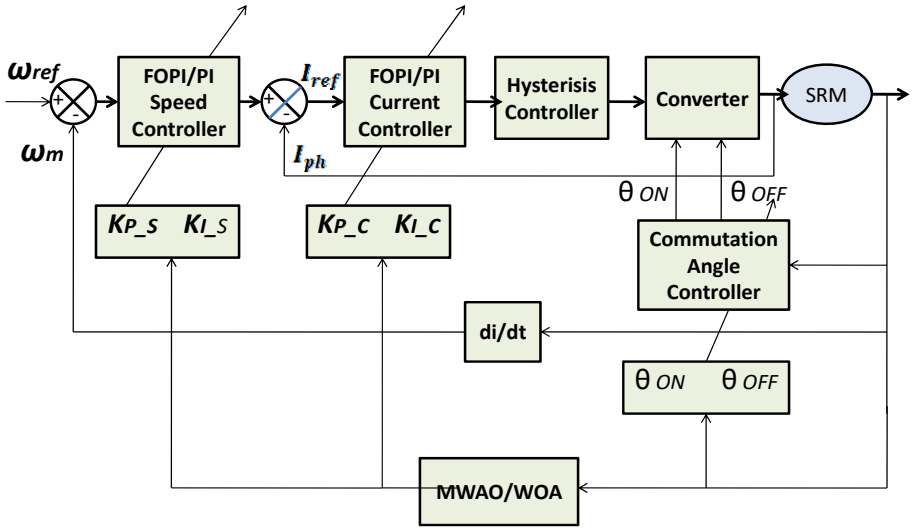


Figure 2. Multiobjective optimized double close loop MWAO FO-PI controller for control of SRM.

3.1. FO-PI Controller

In the present work, the MWAO FO-PI controller for SRM control is implemented. The usage of the FO-PI controller for governing SRM is the most efficient method of control in comparison to a simple PI controller. The classical PI controller is a particular case of fractional PI^{λ} controller (FO-PI controller), where λ represents the integrator order. As shown, the fractional PI^{λ} has one more variable to tune as compared with the PI controller. This gives some additional degree of freedom for performance advancement of the system [6]. Podlubny et al. [27] explain the working of the FO-PI controller.

3.2. FO-PI Speed Controller

An MWAO FO-PI controller is proposed for implementation of the speed controller. The error speed between the actual speed and reference speed act as input to the controller whereas the output signal obtained from the speed controller becomes the input to the current controller. The transfer function of the MWAO FO-PI speed controller in the Laplace domain is explained below.

$$T_S(S) = K_{P\_S} + \frac{K_{I\_S}}{S^{\lambda}} \tag{6}$$

Here, K_{P_S} and K_{I_S} are the proportional and integral gain constant, having an integrator of order λ.

3.3. MWAO FO-PI Current Controller

The MWAO FO-PI current controller transfer function is explained below:

$$T_C(S) = K_{P\_C} + \frac{K_{I\_C}}{S^{\mu}} \tag{7}$$

The K_{P_C} and K_{I_C} are proportional and integral gain constant of FO-PI current controller, having an integrator order of μ.

3.4. Commutation Angle Controller

Rotor positions are sensed and accordingly current pulses are supplied to each phase of SRM. This function is performed by the commutation angle controller. SRM's operation during the saturation condition is avoided [7,8]. A positive current pulse [7] is applied when the stator sees a rising inductance profile. The stator inductance profile with reference to the rotor position is shown in Figure 3. For this, it is required that the flux should decay earlier than the rotor enters in the region of negative torque. This causes selection of the turn-off angle with the turn-on angle, a crucial process. The turn-off time is chosen in the

vicinity of maximum inductance location. The parameters of the FO-PI speed controller, FO-PI current controller and the turn-on with turn-off angle controller are tuned by MWAO and WOA techniques.

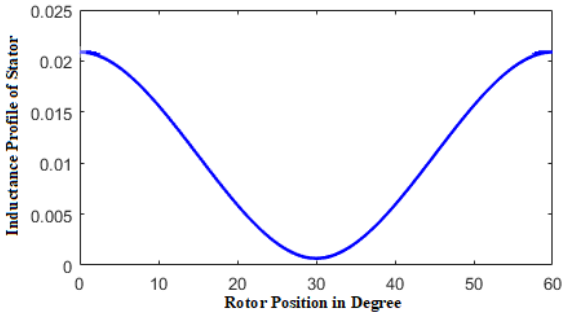


Figure 3. Stator inductance in accordance to rotor position.

4. Modified Whale Algorithm Optimization (MWAO)

In the modified whale optimization technique, improvement is incorporated by two steps. In the first step, correction factors are introduced to decrease the search step size during stance upgradation of the search particle. This causes a fine forage.

In the second step of amendment, the cosine trigonometrical function is implemented for decaying of the control parameter (d) of the whale optimization algorithm in the course of iteration. Incorporation of the cosine function in WOA leads to a balance between the exploitation and exploration property during the search process, which again helps in arriving at the exact estimated global optima.

The cyclic motif leads to the solution to reposition around over other solutions. This leads to good exploitation of the forage area among the two solutions. During the exploration forage, searching is conducted over the search region and also between respective goals.

The present best fit solution is taken as prey krill. All remaining search agents update their situation in conformity to the set target as given by (8).

$$\gamma = \frac{\left| C.\vec{n^*}(t) - \vec{n}(t) \right|}{\zeta_1} \tag{8}$$

$$\vec{n}(t+1) = \frac{\vec{n^*}(t) - \vec{\mathcal{A}}.\gamma}{\zeta_2} \tag{9}$$

$\vec{n^*}(t)$ represents the ongoing best solution, t represents the ongoing iteration and $\vec{m}(t)$ is the position vector. Vector coefficients $\vec{C}$ & $\vec{\mathcal{A}}$ are deduced from Equations (10) and (11) as shown below.

$$\vec{C} = 2.\vec{\mathfrak{r}} \tag{10}$$

$$\vec{\mathcal{A}} = 2.\vec{d}.\vec{\mathfrak{r}} - \vec{d} \tag{11}$$

Here $\vec{\mathfrak{r}}$ is a random lying in the range of 0 and 1. $\vec{\mathcal{A}}$ and $\vec{C}$ are the adjustment vectors, which are implemented to attain varying place surrounding the best particle.

In WOA, the parameter $\vec{d}$ decreases linearly starting from 2 to 0, for ensuring the shrinking nature of the encircling prey. MWAO uses a Cosine trigonometrical function to the decay control parameter 'd' in the iteration process, explained in (12).

$$d = 1 + 0.5 * Cosine\left(\pi \frac{ITER}{ITER_{MAX}}\right) \tag{12}$$

Here in $ITER_{MAX}$ the utmost iterations happened.

4.1. Attacking by Bubble Net Tactic (i.e., Exploitation Phase)

The bubble net attacking tactic brings two strategies into action. These are mentioned below. In MWAO, the cosine trigonometrical function is incorporated as explained in Equations (11) and (12). It causes $\vec{\mathcal{A}}$ to become any random number between $[-d, d]$. Appointing $\vec{\mathcal{A}}$ in the range of $[-1, 1]$, the search agent's place is ensured between the original and present best place of the search particle.

4.2. Spiral Updating Position

Separation between the whale position (m, n) and prey position (m^*, n^*) is assessed. The humpback whales moved in a helix-shaped path, defined by the equation below.

$$\vec{m}(t+1) = \frac{\left(\vec{\gamma'}.e^{bl}.cos(2\pi l) + \vec{m^*}(t)\right)}{\zeta_2} \tag{13}$$

$$\vec{\gamma'} = \frac{\left|\vec{m^*}(t) - \vec{m}(t)\right|}{\zeta_1} \tag{14}$$

In Equation (13), γ' symbolizes the separation between the i^{th} and best solution reached at present. The symbol 'b' is a constant and it determines the shape of the spiral path. 'l' is a random value selected between $[-1, 1]$.

The swimming habit of humpback whales is configured by assuming a probability of 50 percent, each for the shrinking encircling and spiral path for updating whales' position in the process of optimization, which is explained below:

$$\vec{m}(t+1) = \begin{cases} \frac{\vec{m^*}(t) - \vec{\mathcal{A}}.\vec{\gamma'}}{\zeta_1} \text{ if } p < 0.5 \\ \vec{m}(t+1) = \frac{\left(\vec{\gamma'}.e^{bl}.cos(2\pi l) + \vec{m^*}(t)\right)}{\zeta_2} \text{ if } p \geq 0.5 \end{cases} \tag{15}$$

where p is any random value selected between 0 and 1.

4.3. Foraging for Locating Prey (Exploration Phase)

Here, search agents are fetched up randomly. Here, $\vec{\mathcal{A}}$ is any random value that is more than 1 and less than -1. The parameter $\vec{\mathcal{A}}$ causes search particles to run away from the reference search particle. The searching agent's place is updated in accordance with the randomly chosen forage agent in lieu of the best search entity found. Vector $\vec{\mathcal{A}}$ acts like a global optimizer. The exploitation process is encouraged in the case of $|\mathcal{A}| < 1$ and exploration is encouraged when $|\vec{\mathcal{A}}| > 1$. The particle search position is updated with the random chosen search particle. This leads to the random movement of whales. In the mentioned MWAO technique, correction factors ζ_1 & ζ_2 are incorporated for the position upgrading of the whale, understood by the equation below.

$$\gamma = \frac{\left|\vec{C}.\vec{m_{rand}} - \vec{m}\right|}{\zeta_1} \tag{16}$$

$$\vec{m}(t+1) = \frac{\vec{m_{rand}} - \vec{\mathcal{A}}.\vec{\gamma}}{\zeta_2} \tag{17}$$

Here, $\vec{m}_{rand}$ is a random search agent picked up from the present population.

In this process, the rest of the solutions are updated corresponding to the best solutions achieved. Whereas $\vec{A}$ decreases the size of change adapted in the solution. This process guarantees convergence. The convergence takes place in proportion to the computed iteration.

5. Correction Factors Selection for MWAO

Accurate selection of correction factors (ζ_1 and ζ_2) is a cumbersome process. Different values were assigned to ζ_1 and ζ_2 for testing on 23 benchmark functions. Fifty search agents were selected during the implementation of MWAO. The algorithm was run 50 times with 500 iterations. The standard deviation and average value of the objective function obtained during 50 runs were agglomerated. During the first group of testing, ζ_1 is kept fixed at 1.0 and ζ_2 is changed from 0.5 to 3.0 incrementing by 0.5. During the Group 1 testing, ζ_2 is fixed at 2.5 and ζ_1 is allowed to change from 1.0 to 3.5 incrementing by 0.5. These results are provided by Appendix A (Tables A1 and A2). It is deduced from Table A1 that with 13 out of 23 functions, i.e., $p_2(y)$, $p_4(y)$, $p_5(y)$, $p_7(y)$, $p_9(y)$, $p_{10}(y)$, $p_{11}(y)$, $p_{15}(y)$, $p_{16}(y)$, $p_{17}(y)$, $p_{19}(y)$, $p_{20}(y)$, $p_{21}(y)$, $p_{23}(y)$, the best results were reached with correction factors ζ_1 and ζ_2 assigned the value of 1.0 and 2.5, respectively, in comparison to other values in the test. The second best values are provided for functions $p_1(y)$, $p_3(y)$, $p_8(y)$, $p_{11}(y)$, $p_{12}(y)$ and ranked at the third position for functions $p_6(y)$, $p_{13}(y)$, $p_{14}(y)$ with correction factors ζ_1 and ζ_2 assigned the value 1.0 and 2.5, respectively.

From Table A2, it is deduced that for 17 out of 23 functions, i.e., $p_5(y)$, $p_6(y)$, $p_8(y)$, $p_9(y)$, $p_{11}(y)$, $p_{12}(y)$, $p_{13}(y)$, $p_{14}(y)$, $p_{15}(y)$, $p_{16}(y)$, $p_{17}(y)$, $p_{18}(y)$, $p_{19}(y)$, $p_{20}(y)$, $p_{21}(y)$, $p_{22}(y)$, $p_{23}(y)$, the best outcomes are deduced with correction factors ζ_1 and ζ_2 assigned the values 1.0 and 2.5, respectively, in comparison with other values during the test. It is concluded that for maximum cases, the best or 2nd best outcomes were found with ζ_1 = 1.0 and ζ_2 = 2.5. For the above case in this work, these values for correction factors i.e., ζ_1 = 1.0 and ζ_2 = 2.5, are employed for execution of the MWAO algorithm.

6. Performance Assessment of MWAO Technique

Metaheuristic algorithms are stochastic by nature. The starting population is constructed by random selection. The evolution of the algorithm depends on the selection of the starting population. Thus, various runs were suggested for the metaheuristic algorithm. The efficacy of MWAO was analyzed by testing it on 23 benchmark functions [19,20,30,31]. These benchmark functions included fixed dimensional functions, unimodal functions to test exploitation capability, and multimodal functions and fixed dimensional multimodal functions for checking exploration capability.

Unimodal functions have no local optima whereas unimodal functions have unique global optima. Multimodal and fixed dimensional multimodal functions have many local and unique global optima. To prove the efficacy of CamAO, it is again compared with recent metaheuristics algorithms including the WOA [14], differential search algorithm (DSA) [15], lightning search algorithm (LSA) [16], harmonic search algorithm (HSA) [17], backtracking search algorithm (BSA) [18], particle swarm optimization (PSO) [19], and firefly algorithm (FFA) [20], as depicted in [16]. The performance of any optimization algorithm depends on control parameters, which are presented in Appendix A, Table A4, to provide a fair comparison among all the mentioned optimization maximum algorithms.

The maximum count of the generation and population size was fixed as the same as the common parameters for all mentioned techniques. The total number of iterations was kept fixed at 500 and total number of search agents was 50. Each of the mentioned techniques was run 50 times for every benchmark function. In the subsequent subsection, the analysis results are presented. It is also tested on a challenging engineering issue to prove its practical applicability. All these analyses were carried out on a Windows 7 Professional on Intel(R) Core(TM) i7 M640 2.8 GHz processor with 8GB RAM used. The system designing and analysis were done on a MATLAB/SIMULINK platform.

6.1. Analysis of Exploitation Quality ($p_1(y) - p_7(y)$)

Table 1 enlists 30-dimensional unimodal functions ($p_1 - p_7$). The statistical outcome of MWAO and WOA obtained after 50 runs on the mentioned unimodal functions and outcomes of other state-of-the-art functions mentioned in [16] are tabulated in Table 2. From the analysis of Table 2, it is inferred that MWAO is capable of surpassing all other optimization techniques for 4 unimodal functions, and these are $p_3(y)$, $p_4(y)$, $p_5(y)$, $p_7(y)$ out of seven unimodal functions. For two functions $p_1(y)$ and $p_2(y)$, MWAO provides competitive results by producing second-best results. For function $p_6(y)$, MWAO occupies the second-best result.

Table 1. Unimodal benchmark function.

Functions	Dimension	Range	f_{min}
$p_1(y) = \sum_{i=1}^{n} y_i^2$	30	$[-100, 100]^{30}$	0
$p_2(y) = \sum_{i=1}^{n} \lvert y_i \rvert + \prod_{i=1}^{n} y_i$	30	$[-10, 10]^{30}$	0
$p_3(y) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} y_j\right)^2$	30	$[-100, 100]^{30}$	0
$p_4(y) = max_i\{\lvert y_i \rvert, 1 \leq i \leq n\}$	30	$[-100, 100]^{30}$	0
$p_5(y) = \sum_{i=1}^{n-1} \left[100\left(y_{i+1} - y_i^2\right)^2 + (y_i - 1)^2\right]$	30	$[-30, 30]^{30}$	0
$p_6(y) = \sum_{i=1}^{n} (\lvert y_i + 0.5 \rvert)^2$	30	$[-100, 100]^{30}$	0
$p_7(y) = \sum_{i=1}^{n} iy_i^4 + random[0, 1)$	30	$[-1.28, 1.28]^{30}$	0

Table 2. Statistical outcomes of algorithms obtained from 30-dimensional unimodal functions after 50 runs.

Function	MWAO		WOA		LSA		DSA	
	Av.	St.Dev	Av.	St.Dev	Av.	St.Dev	Av.	St.Dev
$p_1(y)$	1.1593×10^{-59}	4.877×10^{-59}	3.0063×10^{-72}	1.6466×10^{-71}	4.81067×10^{-8}	3.4013×10^{-7}	11.58475	6.93844
$p_2(y)$	2.5745×10^{-33}	2.8745×10^{-33}	1.1189×10^{-51}	2.8691×10^{-51}	0.0368065	0.1562330	1.00603663	0.35791079
$p_3(y)$	1.6209×10^{-56}	6.0977×10^{-56}	42289.253	14705.725	43.240804	29.921944	20,888.9331	6907.30897
$p_4(y)$	6.2449×10^{-32}	2.8174×10^{-31}	49.2251	29.2213	1.4932757	1.3028270	27.8103282	7.07708310
$p_5(y)$	26.3645	0.353131	28.1028	0.489595	64.281603	43.755761	1108.18071	572.420941
$p_6(y)$	0.1047	0.046469	0.44119	0.28478	3.3400000	2.0860078	15.74000000	11.29531240
$p_7(y)$	0.0001146	0.0001284	0.0037228	0.0048686	0.0240797	0.0057262	0.123075150	0.065346271
Function	**BSA**		**FFA**		**PSO**		**HSA**	
	Av.	**St.Dev**	**Av.**	**St.Dev**	**Av.**	**St.Dev**	**Av.**	**St.Dev**
$p_1(y)$	9.9673617	9.8122468	0.0116106	0.0042959	2.76284×10^{-5}	4.3213×10^{-5}	24.7111807	6.671103
$p_2(y)$	1.1971508	0.5285104	0.3733268	0.1014310	0.0049233	0.0033347	1.457490046	0.268102707
$p_3(y)$	2720.3105	1182.1965	1808.8064	659.65397	27.863965	9.5794981	6878.658480	1943.088569
$p_4(y)$	9.8347514	2.2732874	0.0766947	0.0146061	0.6102370	0.1460184	9.385431891	1.226512199
$p_5(y)$	471.54854	231.14198	128.28961	278.63448	68.722926	57.810769	830.0325560	474.1996651
$p_6(y)$	13.940000	17.300949	0.0000000	0.0000000	0.1000000	0.3030458	25.1000000	7.532920944
$p_7(y)$	0.0544985	0.0161183	0.0352289	0.0239832	138.83431	22.077449	0.463241118	0.112722402

6.2. Analysis of Exploration Quality ($p_8(y) - p_{23}(y)$)

6.2.1. Analysis of Exploration Ability for Multimodal Functions ($p_8(y) - p_{13}(y)$)

Optimization algorithms are tested for exploration ability by testing on multimodal benchmark functions as described with many local optima. Table 3 tabulates 30-dimensional multimodal benchmark functions. Table 4 tabulates the statistical results of MWAO, WOA for 50 runs on the mentioned benchmark functions, and other six algorithms mentioned in [16]. It is observed from Table 4 that for $p_8(y)$, $p_9(y)$, $p_{10}(y)$, $p_{11}(y)$ the proposed MWAO is able to supersede all other algorithms. For $p_{12}(y)$, the proposed MWAO gives competitive results.

Table 3. Multimodal benchmark function (30-dimensional).

Function	Dimension	Range	f_{min}
$p_8(y) = \sum_{i=1}^{n} -y_i \sin\left(\sqrt{\lvert y_i \rvert}\right)$	30	$[-500, 500]^{30}$	-12569.5
$p_9(y) = \sum_{i=1}^{n} \left[y_i^2 - 10\cos(2\pi y_i) + 10\right]$	30	$[-5.12, 5.12]^{30}$	0
$p_{10}(y) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} y_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi y_i)\right) + 20 + e$	30	$[-32, 32]^{30}$	0
$p_{11}(y) = \frac{1}{4000}\sum_{i=1}^{n} y_i^2 - \prod_{j=1}^{n} \cos\left(\frac{y_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^{30}$	0
$p_{12}(y) = \frac{\pi}{n}\left\{10\sin(\pi x_i) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi x_{i+1})\right]\right\}$ $+(x_n - 1)^2 + \sum_{i=1}^{n} u(y_i, 10, 100, 4)$ $x_i = 1 + \frac{y_i+1}{4}$ $u(y_i, a, k.m) = \begin{cases} k(y_i - a)^m y_i > a \\ 0 - a < y_i < a \\ k(-y_i - a)^m y_i < a \end{cases}$	30	$[-50, 50]^{30}$	0
$p_{13}(y) =$ $0.1\left\{\sin^2(3\pi y_i) + \sum_{i=1}^{n}(y_i - 1)^2\left[1 + \sin^2(3\pi y_i + 1)\right] + (y_n - 1)^2\left[1 + \sin^2(2\pi y_n)\right]\right\} +$ $\sum_{i=1}^{n} u(y_i, 5, 100.4)$	30	$[-50, 50]^{30}$	0

Table 4. Statistical result of different algorithms derived on multimodal 30-dimensional benchmark functions for 50 runs.

Function	MWAO		WOA		LSA		DSA	
	Av.	St.Dev	Av.	St.Dev	Av.	St.Dev	Av.	St.Dev
$p_8(y)$	$-12{,}502.007$	163.77647	$-10{,}175.947$	1956.1498	-8058.7179	669.15931	$-10{,}005.1120$	278.960960
$p_9(y)$	0	0	1.8948×10^{-15}	1.0378×10^{-14}	59.697425	14.915302	46.6551322	7.19733732
$p_{10}(y)$	1.0066×10^{-15}	6.4863×10^{-16}	0.023891	0.01689	2.5372704	0.9108028	3.646161760	1.984370448
$p_{11}(y)$	0	0	0.56157	0.2582	0.0073960	0.0067533	1.091219304	0.093916433
$p_{12}(y)$	0.006556	0.0023295	1.8197	1.8725	0.1036690	0.7439600	0.170607599	0.33670129
$p_{13}(y)$	0.15038	0.058981	0.0006597	0.0003747	0.0109874	0.0472792	1.015913093	0.784576410
Function	**BSA**		**FFA**		**PSO**		**HSA**	
	Av.	**St.Dev**	**Av.**	**St.Dev**	**Av.**	**St.Dev**	**Av.**	**St.Dev**
$p_8(y)$	-9611.699	253.00510	-5867.317	655.51928	-3979.339	869.14843	-7388.96021	350.4224603
$p_9(y)$	66.364367	8.0299007	24.615080	9.1492405	42.619764	10.628658	85.55923249	10.78576220
$p_{10}(y)$	2.9164642	1.4912506	0.0507003	0.0137381	0.0028502	0.0110875	8.796876785	0.612494464
$p_{11}(y)$	1.0668755	0.0868654	0.0056496	0.0014346	0.0098795	0.0242059	1.243666737	0.064079805
$p_{12}(y)$	0.0759286	0.1533844	0.0002308	0.0001001	1.9034×10^{-7}	0.0541695	0.159434981	0.097411015
$p_{13}(y)$	0.5269402	0.7494317	0.0019041	0.0010403	1.1485×10^{-5}	0.0054067	1.700684317	0.549887450

6.2.2. Analysis of Exploration Capability for Fixed Dimension Multimodal Functions ($p_{14}(y) - p_{23}(y)$)

Fixed dimension multimodal benchmark functions are tabulated in Table 5. Statistical results are given in Table 6. From analyzing the outcome from Table 6, it is concluded that MWAO provides global optima/near optima values for 5 functions, i.e., $p_{15}(y)$, $p_{16}(y)$, $p_{17}(y)$, $p_{18}(y)$, $p_{19}(y)$ from 10 test functions. For functions $p_{15}(y)$, $p_{16}(y)$, $p_{17}(y)$, $p_{18}(y)$, MWAO is able to provide the best result. For function $p_{20}(y)$, the second-best result is obtained by MWAO. For function $p_{21}(y)$, the third-best result was obtained by MWAO. For function $p_{19}(y)$, the outcome produced by MWAO is positioned fourth. For functions $p_{14}(y)$, $p_{22}(y)$ and $p_{23}(y)$, MWAO provides competitive results. Thus, it is seen that for unimodal functions and also multimodal functions, MWAO is much more competent than fixed-dimensional multimodal functions.

Table 5. Fixed dimension multimodal benchmark functions.

Function	Range	f_{min}
$p_{14}(y) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}\left(y_i - a_{ij}\right)^2}\right)^{-1}$	$[-65, 65]^2$	1
$p_{15}(y) = \sum_{i=1}^{11}\left[a_i - \frac{y_1\left(b_i^2+b_i y_2\right)}{b_i^2+b_i y_3+y_4}\right]^2$	$[-5, 5]^4$	0.00030
$p_{16} = 4y_1^2 - 2.1y_1^4 = \frac{1}{3}y_1^6 + y_1y_2 - 4y_2^2 + 4y_2^4$	$[-5, 5]^2$	-1.0316
$p_{17}(y) = \left(y_2 - \frac{5.1}{4y^2}y_1^2 + \frac{5}{\pi}y_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(y_1) + 10$	$[-5, 5]^2$	0.398
$p_{18}(y) = \left[1 + (y_1 + y_2 + 1)^2\left(19 - 14y_1 + 3y_1^2 - 14y_2 + 6y_1y_2 + 3y_2^2\right)\right] \times \left[30 + (2y_1 - 3y_2)^2 \times \left(18 - 32y_1 + 12y_1^2 + 48y_2 - 36y_1y_2 + 27y_2^2\right)\right]$	$[-2, 2]^2$	3
$p_{19}(y) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{4} a_{ij}\left(y_j - p_{ij}\right)^2\right)$	$[1, 3]^3$	-3.86
$p_{20}(y) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(y_j - p_{ij}\right)^2\right)$	$[0, 1]^6$	-3.32
$p_{21}(y) = -\sum_{i=1}^{5}\left[(Y - a_i)(Y - a_i)^T + c_i\right]^{-1}$	$[0, 10]^4$	-10.1532
$p_{22}(y) = -\sum_{i=1}^{7}\left[(Y - a_i)(Y - a_i)^T + c_i\right]^{-1}$	$[0, 10]^4$	-10.4028
$p_{23}(y) = -\sum_{i=1}^{10}\left[(Y - a_i)(Y - a_i)^T + c_i\right]^{-1}$	$[0, 10]^4$	-10.5363

Table 6. Statistical result of different algorithms derived for 50 runs on fixed-dimensional multimodal benchmark functions.

Function	MWAO		WOA		LSA		DSA	
	Av.	St.Dev	Av.	St.Dev	Av.	St.Dev	Av.	St.Dev
$p_{14}(y)$	2.17811	2.4943	4.678×10^{-15}	2.628×10^{-15}	0.9980038	0.3379795	0.99800384	3.36448×10^{-16}
$p_{15}(y)$	0.0003848	7.982×10^{-5}	0.0006597	0.0003747	0.0241485	0.0472792	1.17000617	0.784576410
$p_{16}(y)$	-1.0316	4.8164×10^{-6}	0.39789	9.23×10^{-7}	-1.031628	0.0000000	-1.0316285	0.000000000
$p_{17}(y)$	0.39826	0.0009911	0.0054253	0.029715	0.3978874	1.682×10^{-16}	0.39788736	7.79967×10^{-12}
$p_{18}(y)$	3.0001	0.0003218	3.0000	0.0001023	3.000000	3.345×10^{-15}	3.00000000	5.38203×10^{-8}
$p_{19}(y)$	-3.8588	0.0060555	-3.8599	0.0060714	-3.862782	0.0000000	-3.8627821	0.000000000
$p_{20}(y)$	-3.2749	0.082639	-3.1969	0.11385	-3.272060	0.0592765	-3.3219952	2.32293×10^{-8}
$p_{21}(y)$	-9.6997	1.8733	-8.4587	1.8387	-7.02732	3.1561521	-10.152834	0.001254219
$p_{22}(y)$	-9.4978	1.7935	-7.3764	3.4035	-7.136702	3.5149774	-10.393586	0.044169900
$p_{23}(y)$	-10.0826	2.1227	-8.4166	3.4307	-10.53641	3.5960426	-10.536396	0.012504559
Function	**BSA**		**FFA**		**PSO**		**HSA**	
	Av.	**St.Dev**	**Av.**	**St.Dev**	**Av.**	**St.Dev**	**Av.**	**St.Dev**
$p_{14}(y)$	0.9980038	3.3645×10^{-16}	1.9920878	0.6688734	1.7903980	1.2598958	1.605937	1.371648196
$p_{15}(y)$	0.7564660	0.7494317	0.0022405	0.0010403	0.0044428	0.0054067	1.83258210	0.5498874505
$p_{16}(y)$	-1.031628	0.0000000	-1.031624	2.827×10^{-9}	-1.035284	0.0000000	-0.9965145	0.0681095561
$p_{17}(y)$	0.3978874	1.4332×10^{-12}	0.3978874	3.131×10^{-9}	0.3978873	1.684×10^{-16}	0.407772879	0.021637622
$p_{18}(y)$	3.0000000	3.50940×10^{-15}	3.0000001	2.561×10^{-8}	3.000000	4.113×10^{-15}	3.000052375	0.0000538472
$p_{19}(y)$	-3.8627821	0.000000	-3.862782	9.161×10^{-10}	-3.508608	0.3077822	-3.86021051	0.0032566497
$p_{20}(y)$	-3.3219945	2.6286×10^{-60}	-3.267674	0.0619827	-1.852705	0.6552864	-3.12163665	0.133471100
$p_{21}(y)$	-10.153199	0.00058358	-8.427091	3.1202941	-8.653837	2.8082488	-2.78267147	1.8136031039
$p_{22}(y)$	-10.402947	0.00010680	-10.27848	0.8800407	-10.08649	1.2652499	-3.04577891	1.645499759
$p_{23}(y)$	-10.536409	6.41563×10^{-50}	-10.53640	1.115×10^{-6}	-10.32051	1.0609041	-4.20434150	3.0085418461

6.3. Examination of Convergence Nature

The convergence curves of MWAO, WOA are shown in Figures 1–3, which are tabulated in Tables 1, 3 and 5. Convergence curves give an idea about performance of the algorithm. From the analysis of Figures 4–6, it is clear that for all the cases, the convergence characteristic obtained from MWAO is better than that of WOA with the exception of functions $p_{21}(y)$ and $p_{22}(y)$. For functions $p_{21}(y)$ and $p_{22}(y)$, WOA converges faster to reach local optima with the MWOA. Here, the average best score so far means average of the best solution reached so far in every iteration for 50 runs. By analyzing figures (i.e., from Figures 4–6), it is deduced that MWAO accelerates faster with the iteration for all mentioned functions in Tables 1–3. This is due to the application of the Cosine function for decay of the control parameter '*d*' during the iteration process of MWAO as depicted by Equation (11). Whereas WOA uses a linear function as given by Equation (5). The use of the Cosine function provides good balancing between a combination of exploration and exploitation over complete iteration and thus gives good convergence characteristics for arriving at the best optimal result. Further position vector correction factors are employed to decrease step change size. This helps MWAO to search challenging areas during the earlier steps of iteration and then quickly converges close to optima after performing few earlier iterations. The high exploration nature of MWAO is because of the improvised position upgrading mechanism using correction factors.

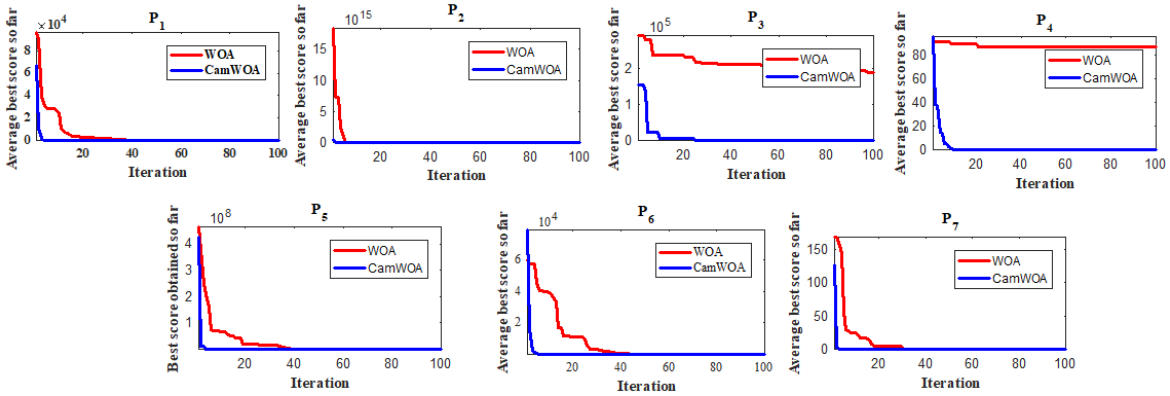


Figure 4. Convergence plot analysis for 30-dimensional unimodal functions, p_1, p_2, p_3, p_4, p_5, p_6, p_7.

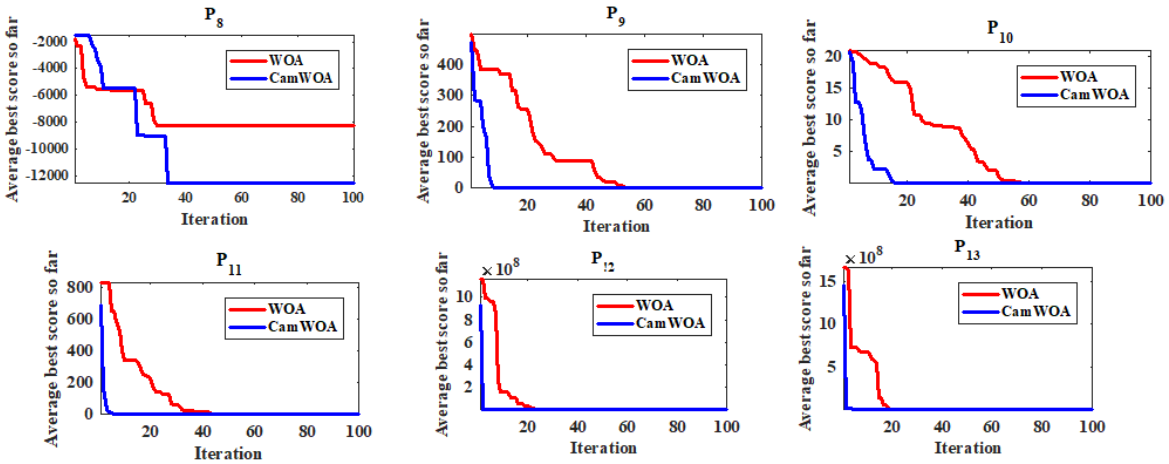


Figure 5. Convergence plot analysis for 30-dimensional functions p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}.

6.4. Comparison of Execution Time

The time of execution of WOA and MWAO is computed by running the algorithms 50 times having 50 search particles and for 500 iterations on 23 benchmark functions. All these analyses were carried out. For the above intention, a Windows 7 Professional on

Intel(R) Core(TM) i7 M640 2.8 GHz processor with 8GB RAM is used. The outcomes are shown in Table A3. It is concluded from Table A3 that for unimodal as well as multimodal benchmark functions, MWAO consumes little additional time for execution in comparison to WOA. This is due to the use of a linear equation for algorithm parameter '*d*' in WOA, whereas a Cosine trigonometrical term is used in MWAO. It is also seen that for some benchmark test functions ($p_3(y)$, $p_{16}(y)$, $p_{17}(y)$, $p_{20}(y)$, $p_{21}(y)$), MWAO needs a shorter execution time as compared with WOA. Altogether it is inferred that, MWAO is capable of providing good results as compared with WOA with a little more execution time.

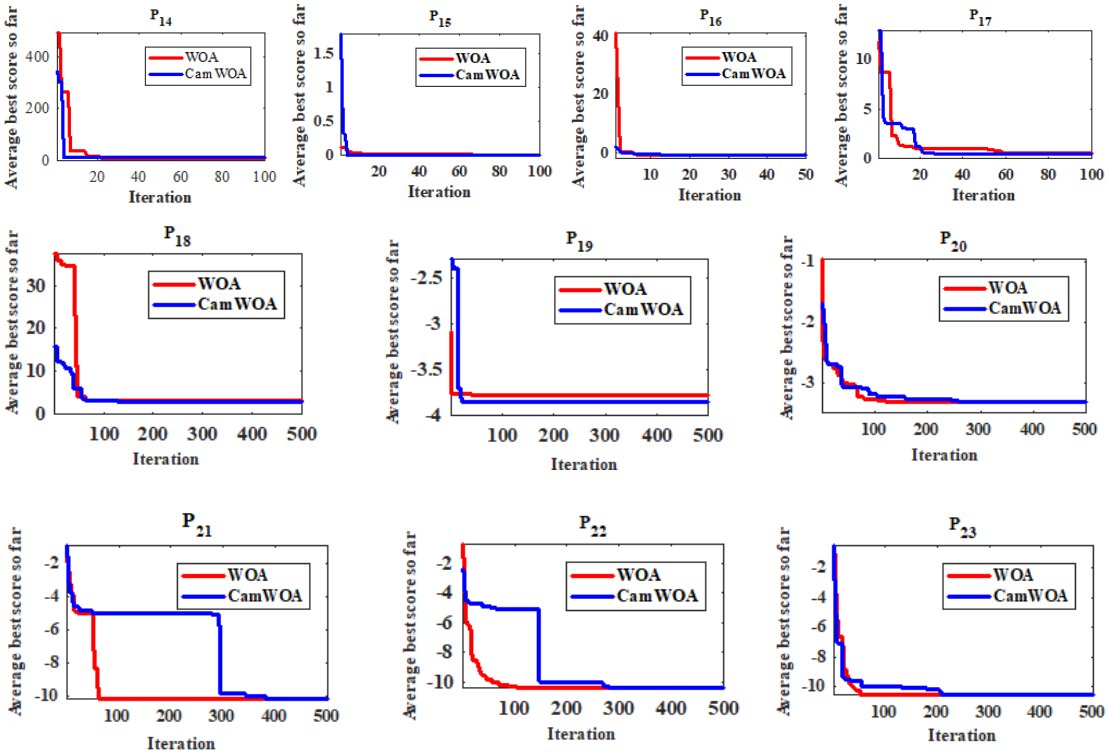


Figure 6. Convergence plot for functions p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}, p_{20}, p_{21}, p_{22}, p_{23}.

7. Proposed Approach: Concurrent Diminution of Torque Ripple with Speed Regulation of SRM Using MWAO Optimized Fractional Order Controller

Because of the nonlinear function of stator flux linkage with rotor current, the SRM drive is subjected to high torque ripple and acoustic noise. The controlling profile of the stator current, and with intelligent selection of the turn-on (θ_{ON})/turn-off angle (θ_{OFF}) [6], the dimunition of torque ripple is achieved. By the appropriate and exact computation of the switching angle and by controlling the stator current profile, the control of SRM can be achieved [20,32–35].

As a multiobjective optimization technique using the proposed MWAO FO-PI controller, for simultaneous reduction of torque ripple, the tracking of speed with reference speed and reducing current error is implemented. The objective is achieved by employing the MWAO FO-PI speed controller, MWAO FO-PI current controller, and a commutation angle controller.

Optimal parameter combinations of MWAO FO-PI speed controller, MWAO FO-PI current controller and commutation angle controller are utilized for performance advancement of SRM.

Equation (17) explains the measurement of the integral square error (ISE) of speed. Equation (18) explains measurement the ISE of current.

$$ISE(speed) = \int \left(\omega_{ref} - \omega_m\right)^2 dt \tag{18}$$

$$ISE(current) = \int \left(I_{ref} - I_{phase}\right)^2 dt \tag{19}$$

Here, the $ISE(speed)$ is the Integral Squared Error of speed, $ISE(current)$ is the Integral Squared Error of current.

Torque ripple coefficient (T_{ripple}) is [4] as explained below.

$$T_{ripple} = \frac{T_{max} - T_{min}}{T_{mean}} \tag{19a}$$

where T_{min} and T_{max} represents the minimum value and maximum value of torque. T_{mean} is the mean value of torque.

7.1. Multiobjective Problem Formulation

A multiobjective problem for optimization is formulated by combining $ISE(speed)$, $ISE(current)$, and T_{ripple}.

Diminution of ISE of speed is formulated below:

$$y_1 = min(ISE(speed)) \tag{20}$$

Diminution of T_{ripple} is explained by (21):

$$y_2 = min\left(T_{ripple}\right) \tag{21}$$

Diminution of ISE of current as:

$$y_3 = min(ISE(current)) \tag{22}$$

The multiobjective optimization problem having the final objective function Y is formulated below:

$$\min(Y) = \beta_1 y_1 + \beta_2 y_2 + \beta_3 y_3 \tag{22a}$$

β_1, β_2, β_3 are weighing factors. The weights are selected such that to make each term competitive during the optimization process or normalizing the objectives y_1, y_2, and y_3 in a uniform scale.

7.2. Execution of MWAO & WOA Technique

Table 7 shows the range of gain used for the FO-PI controller. Table 8 shows the bounds of the PI controller.

Table 7. Range of gains adopted for CamAO FO-PI controller, commutation angle controller.

Gains	Lower Limit	Upper Limit
K_{P_S}	0	200
K_{I_S}	0	200
Speed controller integrator order, λ	0.1	1
K_{P_C}	0	2000
K_{I_C}	0	100
Current controller integrator order, μ	0.1	1
θ_{ON}	32	36
θ_{OFF}	54	58

Table 8. Range of gains for PI controller and commutation angle controller.

Gains	Lower Limit	Upper Limit
K_{P_S}	0	200
K_{I_S}	0	200
K_{P_C}	0	2000
K_{I_C}	0	100
θ_{ON}	32	36
θ_{OFF}	54	58

8. Results and Discussion

As MATLAB software is user-friendly, easy for programming applications, and excellent at graphics, we were inspired to develop the proposed control technique in the MATLAB/SIMULINK environment.

The parameters considered for the designing of four phase SRM, considered in this paper, are provided in Appendix B. To have a comparison of operational efficiency of each algorithm, 50 independent trials were conducted for WOA, as well as for MWAO. The performance appraisal of SRM implementing different controllers was conducted by analyzing different statistical outcomes like mean, best, standard deviation of the ISE of current, ISE of speed, and torque ripple coefficient, listed in Table 9.

Table 9. Statistical execution of (PI/FO-PI) controller, commutation angle controller.

Method	Parameters	Best Value	Worst Value	Mean Value	Std Deviation
MWAO (FO-PI controller)	T_{ripple}	29.2972	29.5669	29.4097	0.1403
	ISE(speed)	1.3353×10^4	1.3507×10^4	1.3426×10^4	77.3649
	ISE(current)	34.3528	59.9811	47.6706	11.4450
	Y	2.5981×10^6	2.71802×10^6	2.7080×10^6	9.9606×10^3
MWAO (PI controller)	T_{ripple}	29.5523	29.5746	29.5667	0.0099
	ISE(speed)	1.4348×10^4	1.4562×10^4	1.4468×10^4	89.0206
	ISE(current)	37.1105	34.9143	34.5690	0.4137
	Y	2.626535×10^6	2.734226×10^6	2.7290×10^6	3.5509×10^3
WOA	T_{ripple}	31.9311	33.0552	32.15857	0.0628
	ISE(speed)	1.4997×10^4	1.8238×10^4	1.6513×10^4	1.8712×10^3
	ISE(current)	212.7761	310.1984	254.9331	50.0165
	Y	2.8882×10^6	2.9379×10^6	2.9036×10^6	3.1041×10^4

The final stage selection of the controller's parameters related to the minimum objective function obtained by the WOA and MWAO method are listed in Table 10.

Table 10. Optimal gain of different parameter and commutation angle of FO-PI and PI controller.

Technique/Parameter	K_{I_speed}	K_{P_speed}	λ	$K_{I\_current}$	$K_{P\_current}$	μ	θ_{ON}	θ_{OFF}
MWAO (FO-PI Controller)	1.0001	1.00012	0.5833	48.0830	435.4619	0.5051	36	54
MWAO (PI Controller)	1.001	1.0024	NA	100	541.041	NA	36	54
WOA (PI Controller)	3.0355	1.0036	NA	9.5044	77.8519	NA	36	58

As shown in Table 9, the ISE (current), coefficient for torque ripple is minimized by MWAO as compared with the WOA algorithm. Table 9 shows that the torque ripples produced by the MWAO FO-PI controller, MWAO PI controller and WOA with PI controller, and MWAO-based commutation angle controller are 29.2972, 29.5523, and 31.9311, respectively. The betterment in torque ripple obtained by MWAO with FO-PI-based controllers is 8.24% compared with the WOA with PI controller. The improvement in the torque ripple obtained by MWAO with PI dependent controllers was 7.449% compared with the WOA with PI controller. The integral square error of the current provided by the MWAO FO-PI controller and MWAO PI controller was 34.3528 and 37.1105, respectively. Whereas the ISE of the current deduced by the WOA with PI controller was 212.7761. The improvement in the ISE of the current deduced by the MWAO FO-PI-dependent controllers was 83.96% compared with the WOA PI controller. The improvement in the ISE of the current derived by implementing MWAO PI controllers was 82.44% compared with the WOA PI controller.

The integral square error of the speed provided by the MWAO FO-PI controller and MWAO PI controller was 1.3353×10^4 and 1.4348×10^4, respectively. The ISE of the speed obtained by the WOA with the PI controller was 1.4997×10^4. The improvement in ISE of the speed obtained by MWAO FO-PI dependent controllers was 10.96% compared with the WOA PI controller. The improvement in the ISE of speed deduced by the MWAO PI-based controllers was 4.3% in comparison with the WOA PI controller.

The combined objective function deduced by the MWAO FO-PI controller, MWAO PI controller, and WOA PI controller was 2.5981×10^6, 2.626535×10^6, and 2.8882×10^6, respectively. The percentage advancement in the combined objective function obtained by the MWAO FO-PI controller was 10.044% compared with the WOA PI controller. The percentage advancement in the combined objective function by the MWAO PI controller was 9.0597% compared with the WOA PI controller. Thus, the MWAO with FO-PI controller and MWAO with PI controller-dependent speed controller provides altogether better system's performance as compared with the WOA with PI controller.

The stator current profile of SRM during its control by using an optimal combination of parameters for attaining a minimum objective function presented in Table 10, shown in Figures 7–9 for MWAO with FO-PI controller, MWAO PI controller, and WOA PI controller, respectively. Figure 10 shows comparison of the torque profile. Figures 11 and 12 show speed tracking with reference speed.

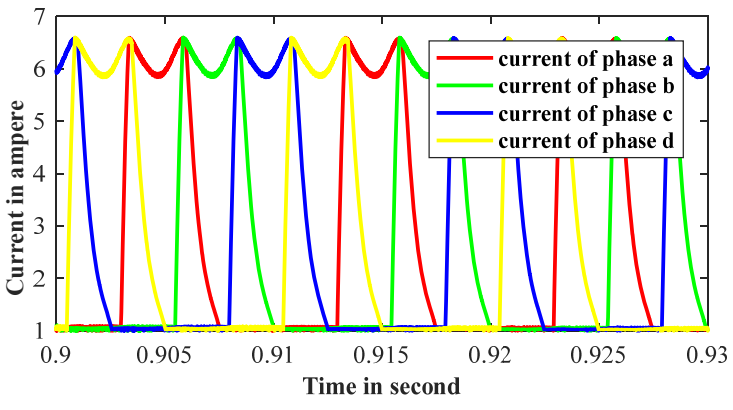


Figure 7. Stator current vs. time of 4-phase SRM drive using FO-PI controller and MWAO optimization algorithm.

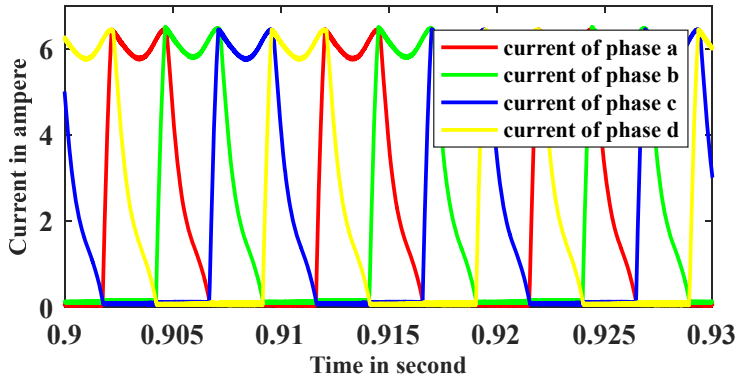


Figure 8. Stator current profile of 4-phase SRM drive using PI controller and MWAO algorithm.

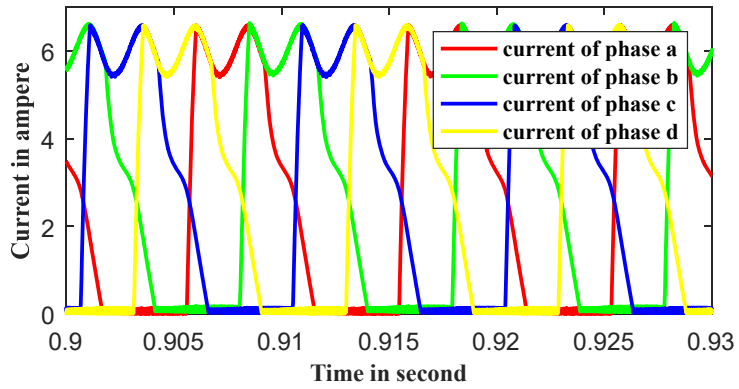


Figure 9. Stator 4-phase current with time in second of SRM drive using PI controller and WOA algorithm.

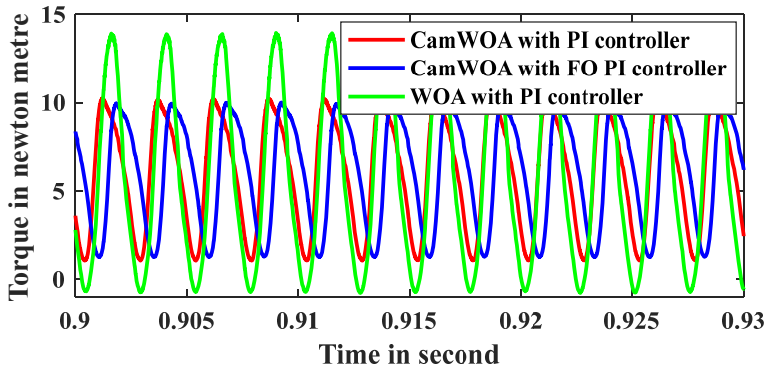


Figure 10. Comparison of torque profile.

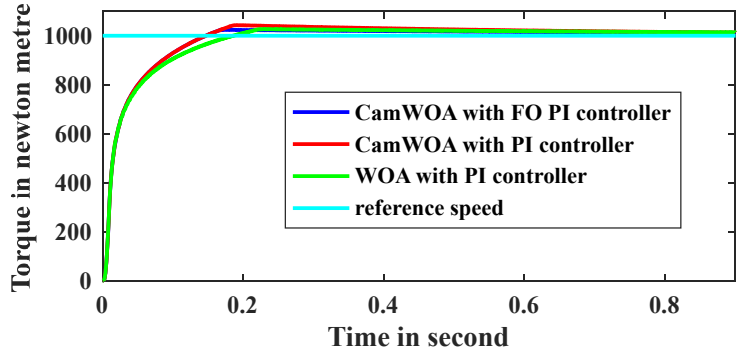


Figure 11. Comparison of tracking of speed with reference speed by MWAO and WOA controller.

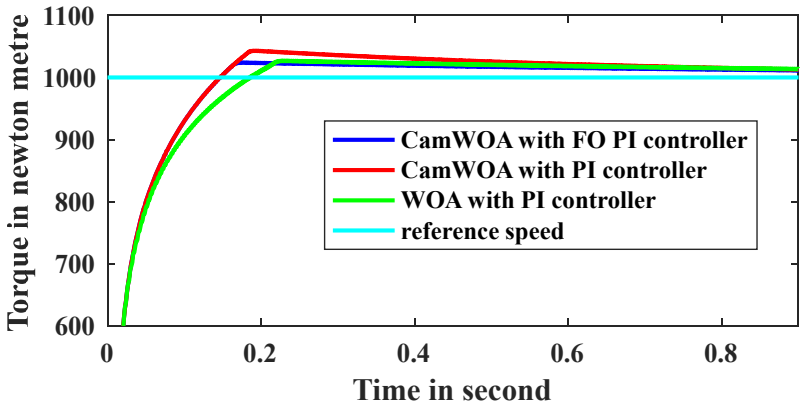


Figure 12. Zoom in of tracking of speed with reference speed by MWAO and WOA controller.

9. Conclusions and Future Scope

MWAO implements a Cosine trigonometrical function for control parameter '*d*' in an iteration process. For reducing the step size in the course of the search process, correction factors are implemented in MWAO. MWAO's performance is examined on 23 benchmark multimodal, fixed dimension multimodal and unimodal test functions. It was shown that MWAO provides very good operational/working performance results compared with other state-of-art WOA, PSO, DSA, LSA, FFA, BSA, and HAS optimization methods, in 4 from 7 unimodal benchmark functions, 4 from 6 multimodal benchmark test functions, and 4 from 10 fixed dimensions benchmark test functions.

MWAO performance supersedes other optimization techniques for four unimodal functions, i.e., $p_3(y)$, $p_4(y)$, $p_5(y)$, $p_7(y)$ from 7 unimodal functions. For functions $p_{14}(y)$, $p_{22}(y)$ and $p_{23}(y)$, MWAO provides competitive results. Thus, the conclusion reached is that MWAO is very competent in the case of multimodal and unimodal functions in comparison with fixed dimensional multimodal functions. To test practical efficacy, an MWAO based FO-PI controller was designed and implemented for advancement of the performance of the SRM drive, with the objective of speed control, diminution of torque ripple, and improvement in the current profile. The betterment in torque ripple obtained by MWAO with FO-PI based controllers was 8.24% compared with WOA with the PI controller. The improvement in torque ripple obtained by MWAO with PI-dependent controllers was 7.449% compared with WOA with the PI controller. The improvement in the ISE of the current deduced by MWAO FO-PI-dependent controllers was 83.96% compared with the WOA PI controller. The improvement in the ISE of current derived by implementing MWAO PI controllers was 82.44% compared with the WOA PI controller. The improvement in the ISE of speed obtained by MWAO FO-PI-dependent controllers was 10.96% compared with the WOA PI controller. The improvement in the ISE of speed deduced by MWAO PI-based controllers was 4.3% in comparison with the WOA PI controller. The percentage advancement in the combined objective function obtained by the MWAO FO-PI controller was 10.044% compared with the WOA PI controller. The percentage advancement in the combined objective function by the MWAO PI controller was 9.0597% compared with the WOA PI controller.

From the above analysis, the conclusion reached is that the MWAO FO-PI controller provides very good overall system operational performance compared with the MWAO PI and WOA PI controller.

The future scope of the present work is to test the performance of the controller with other improved optimization techniques. Advanced controllers such as the fuzzy PI controller and PID controller need to be explored. The limitation of the present work is a lack of experimental verification. The system validation in the hardware setup includes the future scope of the present research work.

Author Contributions: Conceptualization, N.S. and P.C.M.; methodology, N.S.; software, N.S.; validation, N.S. and P.C.M.; formal analysis, N.S.; resources, N.S.; data curation, N.S. and P.C.M.; writing—original draft preparation, N.S. and P.C.M.; writing—review and editing, N.S. and P.C.M.; visualization N.S.; supervision, N.S.; project administration, N.S.; funding acquisition, N.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is included in the manuscript.

Conflicts of Interest: No conflict of interest.

Abbreviations

$\mathcal{A}$	Vector coefficients
B	Coefficient of friction
β_1, β_2, β_3	Weighing factors
BSA	Backtracking Search Algorithm
$\vec{C}$	Vector Coefficients
MWAO	Modified Whale Algorithm Optimization
d	Control Parameter of MWAO
DSA	Differential Search Algorithm
EV	Electric Vehicle
FFA	Firefly Algorithm
$\Psi_s(t)$	Flux linkage
FO-PI	Fractional Order Proportional Integral
HSA	Harmonic Search Algorithm
i	Current
$ISE(current)$	Integral Square Error of current
$ISE(speed)$	Integral Square Error of Speed
I_{ref}	Reference current
I_{phase}	Actual current
I_S	Stator current
$ITER_{MAX}$	Utmost iterations happened
J	Moment of Inertia
l	Random value selected between [−1,1]
$\vec{m_{rand}}$	Random search agent picked up from present population
$\vec{m_{rand}}$	Random search agent picked up from present population
λ	Speed controller integrator order
μ	Current controller integrator order
ζ_1 and ζ_2	Correction factors
K_{I_S}	Integral Gain Constant of Speed Controller
K_{P_S}	Proportional Gain Constant of Speed Controller
WOA	Whale Optimization Algorithm
K_{P_C}	Proportional Gain Constant of Current Controller
K_{I_C}	Integral Gain Constant of Current Controller
KW	Kilowatt
γ'	Separation between the i^{th} and best solution reached
LSA	Lightninig Search Algorithm
$m(t)$	Position Vector
$n^*(t)$	Ongoing Best Solution
PI	Proportional Integral
PSO	Particle Swarm Optimization
R_S	Stator winding resistance
$\vec{\mathfrak{r}}$	Random number
SRM	Switched Reluctance Motor

t	Ongoing iteration
T_l	Load torque
λ	Order of Integrator
$w'(i,\theta)$	Machine's coenergy
ω_m	Angular velocity of motor
$T_S(S)$	Transfer function of Speed Controller
$T_C(S)$	Transfer function of Current Controller
θ	Rotor position in degrees
T_{ripple}	Torque ripple coefficient
$T_e(i,\theta)$	Electromagnetic Torque
T-i-θ	Torque
T_{max}	Maximum value of torque
T_{min}	Minimum value of torque
T_{mean}	Mean value of torque
V_S	Stator phase voltage
ω_m	Actual speed
ω_{ref}	Reference speed

Appendix A. Sensitivity Analysis (ζ_1) and (ζ_2)

Table A1. ζ_1 is kept fixed at 1.0 and ζ_2 varies from 0.5 to 3.0.

Function		ζ_1 = 1.0, ζ_2 = 0.5	ζ_1 = 1.0, ζ_2 = 1.0	ζ_1 = 1.0, ζ_2 = 1.5	ζ_1 = 1.0 ζ_2 = 2.0	ζ_1 = 1.0, ζ_2 = 2.5	ζ_1 = 1.0, ζ_2 = 3.0
$p_1(y)$	Average	7.6556×10^{-41}	3.423×10^{-40}	4.0393×10^{-55}	4.363×10^{-52}	1.1593×10^{-59}	5.8225×10^{-63}
	Std. deviation	2.782×10^{-40}	1.5208×10^{-39}	1.7657×10^{-54}	2.377×10^{-51}	4.8737×10^{-59}	1.7546×10^{-62}
$p_2(y)$	Average	1.5828×10^{-23}	8.333×10^{-22}	1.2976×10^{-30}	1.1353×10^{-32}	2.5745×10^{-33}	1.2019×10^{-30}
	Std. deviation	6.6982×10^{-23}	3.9056×10^{-21}	5.4201×10^{-30}	3.5642×10^{-32}	2.8745×10^{-33}	5.44×10^{-30}
$p_3(y)$	Average	110363.2492	56408.4512	2.5512×10^{-49}	1.4876×10^{-51}	1.6209×10^{-56}	1.6397×10^{-60}
	Std. deviation	33450.53221	17537.996	1.2992×10^{-48}	8.1215×10^{-51}	6.0977×10^{-56}	7.7957×10^{-60}
$p_4(y)$	Average	54.3092	43.9519	2.4977×10^{-29}	7.735×10^{-32}	6.2449×10^{-32}	1.0886×10^{-28}
	Std. deviation	29.0378	29.2169	8.6221×10^{-29}	3.0035×10^{-31}	2.8174×10^{-31}	5.9426×10^{-28}
$p_5(y)$	Average	27.7379	27.1548	27.2754	27.2844	26.3645	26.5783
	Std. deviation	0.443013	0.654897	0.303547	0.323847	0.353131	4.22645
$p_6(y)$	Average	0.64878	0.14746	0.09098	0.098967	0.1047 (3rd)	0.12331
	Std. deviation	0.24186	0.11647	0.045069	0.055668	0.046469	0.055738
$p_7(y)$	Average	0.003287	0.003679	0.00018764	0.000115	0.00011465	0.00017722
	Std. deviation	0.0039455	0.0048482	0.00017255	0.00010374	0.00012841	0.00024685
$p_8(y)$	Average	−9283.7537	−11,113.2137	−12,519.5639	−0.000115	−12,502.0073	−12,447.0115
	Std. deviation	1335.5325	1964.45009	126.81916	0.00010374	163.776476	238.758513
$p_9(y)$	Average	0	0	0	0	0	0
	Std. deviation	0	0	0	0	0	0
$p_{10}(y)$	Average	3.1382×10^{-15}	3.0198×10^{-15}	3.3751×10^{-15}	1.5987×10^{-15}	1.0066×10^{-15}	1.0066×10^{-15}
	Std. deviation	2.7174×10^{-15}	2.001×10^{-15}	1.6559×10^{-15}	1.4454×10^{-15}	6.4863×10^{-15}	6.4863×10^{-15}
$p_{11}(y)$	Average	0	0.0032514	0	0	0	0
	Std. deviation	0	0.017809	0	0	0	0
$p_{12}(y)$	Average	0.029789	0.0065026	0.0048319	0.0060906	0.006556	0.0070208
	Std. deviation	0.016391	0.0051216	0.002176	0.0027322	0.0023295	0.003844
$p_{13}(y)$	Average	0.63642	0.29502	0.11033	0.12464	0.15038	0.14982
	Std. deviation	0.3227	0.2354	0.066306	0.05788	0.05898	0.086802
$p_{14}(y)$	Average	3.3582	1.8857	2.0458	2.4375	2.17811	2.7936
	Std. deviation	3.0923	1.8955	2.501	2.9447	2.4943	3.4751
$p_{15}(y)$	Average	0.00081561	0.00056235	0.00046141	0.00040091	0.00038476	0.00038418
	Std. deviation	0.00051826	0.00029764	0.00024184	0.00011491	7.9821×10^{-5}	8.9299×10^{-5}
$p_{16}(y)$	Average	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	Std. deviation	8.7441×10^{-6}	2.3488×10^{-8}	1.3637×10^{-6}	4.4862×10^{-6}	4.8164×10^{-6}	1.9395×10^{-6}
$p_{17}(y)$	Average	0.39955	0.3979	0.39885	0.39894	0.39826	0.40058
	Std. deviation	0.0040552	3.2115×10^{-5}	0.0018938	0.0018712	0.0009911	0.0043102
$p_{18}(y)$	Average	3	3	3.0002	3.0003	3.0001	3.0003
	Std. deviation	6.0224×10^{-5}	6.6015×10^{-5}	0.00040859	0.00058213	0.00032177	0.00066644

Table A1. *Cont.*

Function		$\zeta_1 = 1.0$, $\zeta_2 = 0.5$	$\zeta_1 = 1.0$, $\zeta_2 = 1.0$	$\zeta_1 = 1.0$, $\zeta_2 = 1.5$	$\zeta_1 = 1.0$ $\zeta_2 = 2.0$	$\zeta_1 = 1.0$, $\zeta_2 = 2.5$	$\zeta_1 = 1.0$, $\zeta_2 = 3.0$
$p_{19}(y)$	Average	−3.8437	−3.8593	−3.8545	−3.8568	−3.8588	−3.8576
	Std. deviation	0.035037	0.0037188	0.010259	0.0055391	0.0060555	0.00571
$p_{20}(y)$	Average	−3.1236	−3.2228	−3.2427	−3.2412	−3.266	−3.2382
	Std. deviation	0.24137	0.094241	0.094653	0.08854	0.082639	0.091723
$p_{21}(y)$	Average	−10.0684	−9.3038	−8.9106	−9.1935	−10.1108	−9.2981
	Std. deviation	0.0859104	2.2249	2.1651	1.8867	1.8733	1.7044
$p_{22}(y)$	Average	−10.3287	−8.5547	−9.3393	−8.8644	−9.4978	−8.719
	Std. deviation	0.0854108	2.6061	1.5836	2.3254	1.7935	2.2637
$p_{23}(y)$	Average	−10.1785	−9.282	−9.1738	−9.5958	−10.4457	−9.7806
	Std. deviation	1.46931	2.5818	2.2747	1.8494	2.1227	1.504

Table A2. ζ_1 is kept fixed at 2.5 and ζ_2 is varied from 1.0 to 3.5.

Function		$\zeta_1 = 1.0$, $\zeta_2 = 2.5$	$\zeta_1 = 1.5$, $\zeta_2 = 2.5$	$\zeta_1 = 2.0$, $\zeta_2 = 2.5$	$\zeta_1 = 2.5$ $\zeta_2 = 2.5$	$\zeta_1 = 3.0$, $\zeta_2 = 2.5$	$\zeta_1 = 3.5$, $\zeta_2 = 2.5$
$p_1(y)$	Av.	1.1593×10^{-59}	8.4873×10^{-224}	9.8813×10^{-324}	0	0	0
	Std.	4.8737×10^{-59}	0	0	0	0	0
$p_2(y)$	Av.	2.5745×10^{-33}	1.7253×10^{-113}	1.1593×10^{-162}	1.4723×10^{-192}	2.3936×10^{-222}	9.9903×10^{-243}
	Std.	2.8745×10^{-33}	9.4167×10^{-113}	4.4455×10^{-162}	0	0	0
$p_3(y)$	Av.	1.6209×10^{-56}	2.3263×10^{-217}	7.6088×10^{-311}	0	0	0
	Std.	6.0977×10^{-56}	0	0	0	0	0
$p_4(y)$	Av.	6.2449×10^{-32}	7.185×10^{-114}	5.3626×10^{-163}	2.0152×10^{-189}	1.8936×10^{-219}	1.503×10^{-242}
	Std.	2.8174×10^{-31}	3.4089×10^{-113}	2.2228×10^{-162}	0	0	0
$p_5(y)$	Av.	26.3645	27.4518	27.7293	27.9926	28.0469	28.0833
	Std.	0.353131	0.400047	0.412319	0.363797	0.3473	0.343142
$p_6(y)$	Av.	0.1047	0.22679	0.51077	0.53919	0.58355	0.61437
	Std.	0.046469	0.069648	0.17753	0.15796	0.24981	0.32066
$p_7(y)$	Av.	0.00011465	8.9378×10^{-5}	0.00010015	8.5411×10^{-5}	7.8268×10^{-5}	0.00010142
	Std.	0.00012841	7.4012×10^{-5}	9.3399×10^{-5}	8.0125×10^{-5}	6.886×10^{-5}	9.2815×10^{-5}
$p_8(y)$	Av.	−12,502.0073	−11,439.6176	−11,380.9642	−11,356.6927	−11,516.3283	−11,129.1044
	Std.	163.776476	1245.59677	1755.65974	1443.79102	1815.34216	1764.70062
$p_9(y)$	Av.	0	0	0	0	0	0
	Std.	0	0	0	0	0	0
$p_{10}(y)$	Av.	1.0066×10^{-15}	1.0066×10^{-15}	8.8818×10^{-16}	8.8818×10^{-16}	8.8818×10^{-16}	8.8818×10^{-16}
	Std.	6.4863×10^{-16}	6.4863×10^{-16}	0	0	0	0
$p_{11}(y)$	Av.	0	0	0	0	0	0
	Std.	0	0	0	0	0	0
$p_{12}(y)$	Av.	0.006556	0.012229	0.024117	0.027424	0.027679	0.042597
	Std.	0.0023295	0.0061196	0.0097346	0.01098	0.013716	0.031859
$p_{13}(y)$	Av.	0.15038	0.25599	0.37991	0.47182	0.51315	0.51355
	Std.	0.058981	0.090468	0.14704	0.20594	0.20568	0.23737
$p_{14}(y)$	Av.	2.17811	5.5874	5.8163	8.2108	9.1477	7.7524
	Std.	2.4943	4.9629	4.911	5.0777	4.8873	5.3965
$p_{15}(y)$	Av.	0.00038476	0.00038663	0.00039224	0.00042184	0.00046921	0.00043481
	Std.	7.9821×10^{-5}	8.5211×10^{-5}	0.00011758	0.00012183	0.00013442	0.0001309
$p_{16}(y)$	Av.	−1.0316	−1.0117	−1.0055	−1.0048	−1.0041	−1.0057
	Std.	4.8164×10^{-6}	0.012736	0.0094093	0.0079115	0.0083002	0.010042
$p_{17}(y)$	Av.	0.39826	0.39887	0.39864	0.40292	0.39938	0.40198
	Std.	0.0009911	0.00016374	0.0011056	0.024402	0.0042679	0.019078
$p_{18}(y)$	Av.	3.0001	3.0014	3.0036	3.9119	3.9112	8.42953
	Std.	0.00032177	0.0021175	0.005737	4.93	4.9296	10.9732
$p_{19}(y)$	Av.	−3.8588	−3.8532	−3.8355	−3.8313	−3.8385	−3.8155
	Std.	0.0060555	0.014227	0.031627	0.039658	0.033109	0.054432
$p_{20}(y)$	Av.	−3.266	−3.2332	−3.1838	−3.1241	−3.1664	−3.0841
	Std.	0.082639	0.10405	0.1321	0.19023	0.13984	0.22011
$p_{21}(y)$	Av.	−10.1108	−6.6502	−6.1211	−6.3317	−6.5192	−6.7025
	Std.	1.8733	2.3266	1.8503	1.9532	2.0645	2.0424
$p_{22}(y)$	Av.	−9.4978	−7.3499	−6.3975	−6.8133	−6.8929	−6.6846
	Std.	1.7935	2.4942	2.1603	2.4119	2.0272	2.1886
$p_{23}(y)$	Av.	−10.4457	−6.4657	−7.1001	−6.5468	−7.1618	−7.0914
	Std.	2.1227	2.2941	2.3042	2.2093	2.335	2.2302

Table A3. Comparison of implementation time (in seconds) of WOA and MWAO.

Functions	$p_1(y)$	$p_2(y)$	$p_3(y)$	$p_4(y)$	$p_5(y)$	$p_6(y)$	$p_7(y)$
WOA	5.359231	6.21212	31.8832	5.48073	6.664643	5.393869	9.180304
MWAO	5.464056	6.2987	31.6330	5.517608	6.820316	5.396708	9.314385
Functions	$p_8(y)$	$p_9(y)$	$p_{10}(y)$	$p_{11}(y)$	$p_{12}(y)$	$p_{13}(y)$	
WOA	7.165297	5.652841	6.528609	7.405994	18.840883	18.814588	
MWAO	7.178957	5.711984	6.665277	7.448466	18.857660	19.073341	
Functions	$p_{14}(y)$	$p_{15}(y)$	$p_{16}(y)$	$p_{17}(y)$	$p_{18}(y)$		
WOA	46.340362	4.496783	3.87674	2.967898	2.960074		
MWAO	47.109380	4.514427	3.336196	2.910601	3.017865		
Functions	$p_{19}(y)$	$p_{20}(y)$	$p_{21}(y)$	$p_{22}(y)$	$p_{23}(y)$		
WOA	6.961318	7.226696	11.19802	14.01065	18.852677		
MWAO	7.038459	7.217759	11.136932	14.11498	19.03258		

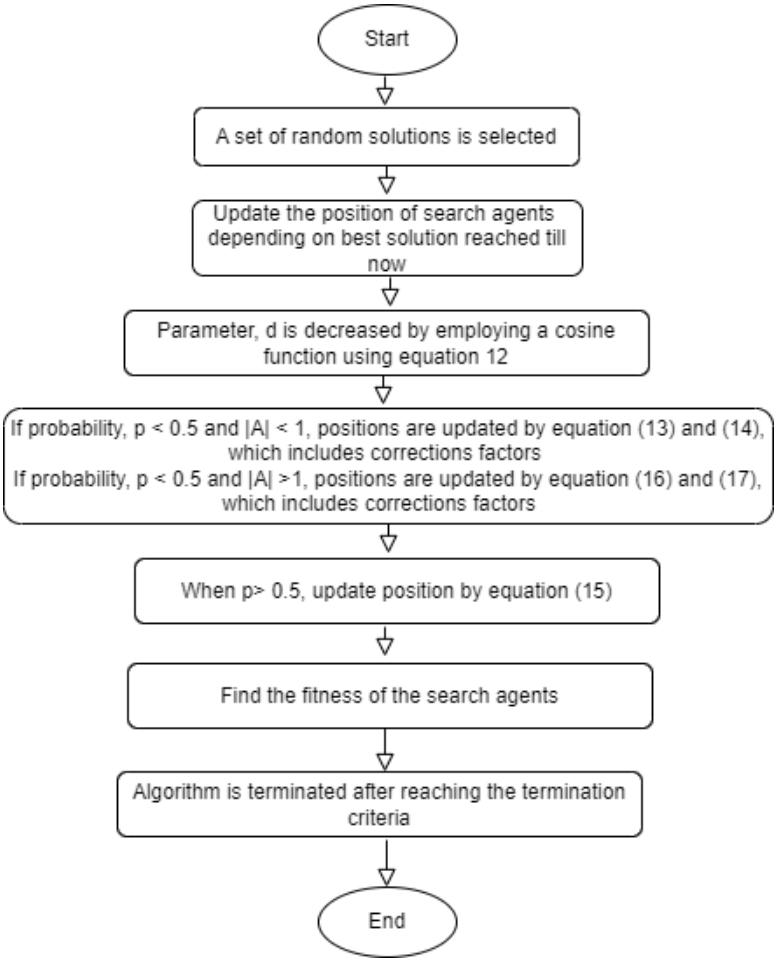
Table A4. Parameters setup of various metaheuristic algorithms.

Algorithm	**Parameter**	**Value**
PSO	c_1	2
	c_2	2
FFA	γ	1
	β	1
	α	0.2
BSA	F	3.rand
DSA	p1and p2	0.3.rand
LSA	Channel time	10
WOA	Parameter *d* of coefficient vector	Decreases linearly from 2 to 0.
	Parameter α of coefficient vector	Decreases from 2 to 0.
MWAO	Parameter *d* of coefficient vector	Varies between 2 to 0.
	Parameter α of coefficient vector	Varies between 2 to 0.
	Correction factor, CF_1	2.5
	Correction factor, CF_2	1.5

Appendix B. Dimension of SRM Adopted for Design

Machine Parameter	**Value**	**Machine Parameter**	**Value**
Power (output)	75 KW	Load torque	4 nm
Rotor speed	1000 RPM	Aligned inductance	23.62 mH
Resistance of stator	0.05 ohm	Unaligned inductance	0.67 mH
Friction	0.02 Nms	DC link voltage (Input)	220 V
Inertia	0.025 Kg mm	Maximum current	450 A
Number of stator pole	8	Maximum flux linkage	0.486 mH
Number of rotor pole	6	Saturated inductance	0.15 mH

Appendix B.1. Flow Chart for Computation



References

1. Harris, M.R. Comparison of Design and Performance Parameters in Switched Reluctance and Induction Motors. In Proceedings of the 1989 Fourth International Conference on Electrical Machines and Drives, Oxford, UK, 13–15 September 1989; pp. 303–307.
2. Hu, Y.; Song, X.; Cao, W.; Ji, B. New SR drive with integrated charging capacity for plug-in hybrid electric vehicles (PHEVs). *IEEE Trans. Ind. Electron.* **2014**, *61*, 5722–5731. [CrossRef]
3. Valdivia, V.; Todd, R.; Bryan, F.J.; Barrado, A.; Lázaro, A.; Forsyth, A.J. Behavioral modeling of a switched reluctance generator for aircraft power systems. *IEEE Trans. Ind. Electron.* **2014**, *61*, 2690–2699. [CrossRef]
4. Miler, T.J.E. *Switched Reluctance Motor and Their Control*; Magna Physics Publishing: Lebanon, OH, USA, 1993.
5. Anwar, M.N.; Hussain, I.; Radun, A.V. Comprehensive design methodology for switched reluctance machines. *IEEE Trans. Ind. Appl.* **2001**, *37*, 1684–1692. [CrossRef]
6. Hussain, I. Minimization of torque ripple in SRM drives. *IEEE Trans. Ind. Electron.* **2002**, *49*, 28–39. [CrossRef]
7. Torrey, D.A.; Niu, X.M.; Unkauf, E.J. Analytical modeling of variable reluctance machine magnetization characteristics. *IEEE Proc. Elctr. Power Appl.* **1995**, *142*, 14–22. [CrossRef]
8. Hoang, L.H.; Brunelle, P. A Versatile Nonlinear Switched Reluctance Motor in Simulink Using Realistic and Analytical Magnetic Characteristics. In Proceedings of the 31st Annual Conference of the IEEE Industrial Electronics Society, Raleigh, NC, USA, 6–10 November 2005.
9. Li, D.; Yu, H.; Tee, K.P.; Wu, Y.; Ge, S.S.; Lee, T.H. On time-synchronized stability and control. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 2450–2463. [CrossRef]
10. Brandsletter, P.; Hrbac, R. Application of Fuzzy Logic in Control of Switched Reluctance Motor. *Adv. Electr. Electron. Eng.* **2006**, *5*, 68–71.
11. Lin, Z.; Reay, S.; Williams, W. Torque ripple reduction in switched reluctance motor drives using B-Spline neural networks. *IEEE Trans. Ind. Appl.* **2006**, *2*, 1445–1453. [CrossRef]
12. Nesmachnow, S. An overview of metaheuristics: Accurate and efficient methods for optimisation. *Int. J. Metaheuristics* **2014**, *3*, 320–347. [CrossRef]

13. Boussa, D.I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. *Inf. Sci.* **2013**, *237*, 82–117. [CrossRef]
14. Sayedali; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67.
15. Civicioglu, P. Transforming geocentric Cartesian coordinates to geodetic coordinates by using differential search algorithm. *Comput. Geosci.* **2012**, *46*, 229–247. [CrossRef]
16. Shareef, H.; Ibrahim, A.A.; Mutlag, A.H. Lightninig Search Algorithm. *Appl. Soft Comput.* **2015**, *36*, 315–333. [CrossRef]
17. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–70. [CrossRef]
18. Civicioglu, P. Backtracking search optimization algorithm for numerical optimization problems. *Appl. Math. Comput.* **2013**, *219*, 8121–8814. [CrossRef]
19. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
20. Yang, X.S. *Firefly Algorithm in Engineering Optimization*; John Wiley & Sons Inc.: New York, NY, USA, 2010.
21. Saha, N.; Panda, S. Speed control with torque ripple reduction of switched reluctance motor by Hybrid Many Optimizing Liaison Gravitational Search technique. *Eng. Sci. Technol. Int. J.* **2017**, *20*, 909–921. [CrossRef]
22. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.* **2003**, *35*, 263–308. [CrossRef]
23. Wang, G.; Tan, Y. Improving Metaheuristics Algorithms with Information Feedback Model. *IEEE Trans. Cybernatics* **2017**, *49*, 542–555. [CrossRef]
24. Mahrach, M.; Miranda, G.; León, C.; Segredo, E. Comparison between Single and Multi-Objective Evolutionary Algorithms to Solve the Knapsack Problem and the Travelling Salesman Problem. *Mathematics* **2020**, *8*, 2018. [CrossRef]
25. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper optimization algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [CrossRef]
26. Civicioglu, P.; Besdok, E. Bernstain-search differential evolution algorithm for numerical function optimization. *Expert Syst. Appl.* **2019**, *138*, 112831. [CrossRef]
27. Podlubny, I. *Fractional-Order Systems and Fractional-Order Controllers*; Institute of Experimental Physics, Slovak Academy of Sciences: Kosice, Slovakia, 1994; Volume 12, pp. 1–18.
28. Tytiuk, V.; Ilchenko, O.; Chornyi, O.; Zachepa, I.; Serhiienko, S.; Berdai, A. SRM Identification with Fractional Order Transfer Functions. In Proceedings of the 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), Lviv, Ukraine, 2–6 July 2019; pp. 271–274. [CrossRef]
29. Ghoudelbourk, S.; Azar, A.T.; Dib, D.; Rechach, A. Fractional order control of switched reluctance motor. *Int. J. Adv. Intell. Paradig.* **2022**, *21*, 247–266. [CrossRef]
30. Digalakis, J.; Margaritis, K. On benchmarking functions for genetic algorithms. *Int. J. Comput. Math.* **2001**, *77*, 481–506. [CrossRef]
31. Molga, M.; Smutnicki, C. Test functions for optimization needs. *Robert Marks.org* **2005**, *101*, 48.
32. Evangeline, S.J.; Venmathi, K.; Ajayan, S. Speed Control of Switched Reluctance Motor Using Fractional Order Control. In Proceedings of the International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT), Coimbatore, India, 3–4 February 2017; pp. 367–372.
33. Hamouda, M.; Menaem, A.A.; Rezk, H.; Ibrahim, M.N.; Számel, L. Numerical Estimation of Switched Reluctance Motor Excitation Parameters Based on a Simplified Structure Average Torque Control Strategy for Electric Vehicles. *Mathematics* **2020**, *8*, 1213. [CrossRef]
34. Boukhnifer, M.; Chaibet, A.; Ouddah, N.; Monmasson, E. Speed robust design of switched reluctance motor for electric vehicle system. *Adv. Mech. Eng.* **2017**, *9*. [CrossRef]
35. Rechach, A.; Ghoudelbourk, S.; Aoulmi, Z.; Djalel, D. Smart Controls for Switched Reluctance Motor 8/6 Used for Electric Vehicles Underground Mining Security. *Eur. J. Electr. Eng.* **2021**, *23*, 423–432. [CrossRef]