

Article

Metaheuristic Parameter Identification of Motors Using Dynamic Response Relations

Omar Rodríguez-Abreo ^{1,*}, Juvenal Rodríguez-Reséndiz ² and José Manuel Álvarez-Alvarado ²
and Alfonso García-Cerezo ^{1,*}

¹ Space Robotics Laboratory, Department of Systems Engineering and Automation, Universidad de Malaga, C/Ortiz Ramos s/n, 29071 Malaga, Spain

² Faculty of Engineering, Universidad Autónoma de Querétaro, Santiago de Querétaro 76010, Mexico; juvenal@uaq.edu.mx (J.R.-R.); jmalvarez@uaq.mx (J.M.Á.-A.)

* Correspondence: omar.rodriguez@upq.edu.mx (O.R.-A.); ajgarcia@uma.es (A.G.-C.)

Abstract: This article presents the use of the equations of the dynamic response to a step input in metaheuristic algorithm for the parametric estimation of a motor model. The model equations are analyzed, and the relations in steady-state and transient-state are used as delimiters in the search. These relations reduce the number of random parameters in algorithm search and reduce the iterations to find an acceptable result. The tests were implemented in two motors of known parameters to estimate the performance of the modifications in the algorithms. Tests were carried out with three algorithms (Gray Wolf Optimizer, Jaya Algorithm, and Cuckoo Search Algorithm) to prove that the benefits can be extended to various metaheuristics. The search parameters were also varied, and tests were developed with different iterations and populations. The results show an improvement for all the algorithms used, achieving the same error as the original method but with 10 to 50% fewer iterations.

Keywords: Cuckoo Search; metaheuristic; parameter estimation; DC motor; Jaya; Grey Wolf Optimizer



Citation: Rodríguez-Abreo, O.;

Rodríguez-Reséndiz, J.;

Álvarez-Alvarado, J.M.;

García-Cerezo, A. Metaheuristic

Parameter Identification of Motors

Using Dynamic Response Relations.

Sensors **2022**, *22*, 4050. [https://](https://doi.org/10.3390/s22114050)

doi.org/10.3390/s22114050

Academic Editor: Juan M. Corchado

Received: 5 May 2022

Accepted: 22 May 2022

Published: 27 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Parameter identification is a highly studied problem due to its multiple fields of application. For example, Hammerstein controlled auto-regressive moving average systems and autonomous robot navigation in the state of charge estimation of lithium-ion batteries [1–3]. One of the studied systems for their parametric identification is the direct current motors due to their increasing use in high-demand applications [4].

The parametric identification techniques are diverse. Some authors opt for using intelligent methods, as in [5], where a backpropagation network is used for parametric identification. The disadvantage is that a large amount of data is required to train accurately the network, which is a weakness of all ANNs of this type. Another intelligent technique was used in [6], where the authors estimate the parameters with fuzzy techniques. In [7], a Sparse estimation is developed by combining SPARSEVA and the Steiglitz–McBride method. Methods such as least squares and Steiglitz–McBride are heuristic methods that generally provide adequate results. However, metaheuristic algorithms represent a more straightforward option in complex problems [8]. This optimization algorithm is biologically inspired, and its use is widespread in multiple areas [9].

Metaheuristic algorithms have been widely developed and implemented in multiple areas. These algorithms are iterative which complicates their implementation in real-time [10]. However, its complexity is reduced with respect to other heuristic methods [8]. Some examples of applications are found in works such as [11,12], where authors use metaheuristics algorithms to tune controllers. In [13], a GA with a neural network convolutional is proposed to successfully indicate the likelihood of microscopy images belonging to different classes. In the estimation of parameters, the metaheuristic algorithms have an application such as that shown in [14].

In [15], the authors use a metaheuristic algorithm improved for parameter estimation in permanent magnet synchronous motor. This methodology is similar; however, the dynamic response of the system is not considered. The work contribution is that using existing relationships both in the transient and the steady-state of the motor response. These relations make it possible to reduce an estimation problem of five random variables (model parameters) to use two and calculate the remaining ones. Reducing random variables allows this type of algorithm to improve since the method must comply with these relationships, taken as restrictions. The use of restrictions in the search improves the performance of this type of algorithm [15]. However, no previous work has used the relations of the transient-state and the steady-state as constraints to improve the parameter search process.

The relevance of this work lies in the fact that the engine is a system of second-order differential equations. Therefore, the method can be extended to systems that are modeled in the same way. Second-order systems represent a high percentage of physical systems in engineering systems [16]. Few systems are of one-order, and higher-order systems are usually reduced to second-order for their analysis [16]. Consequently, the method can be adapted by changing the fitness/cost function and the relations between the equations.

The second contribution is improving the three metaheuristic algorithms (the Cuckoo Search, the Gray Wolf Optimizer, and the Jaya algorithm) with dynamic response relations. The tests in the three algorithms are intended to reveal that these relations can be extended to other algorithms of this type and are not only applied in a particular case. Each of the algorithms were tested in their original version and the version proposed in this research.

The standard procedure to develop the parametric estimation was used, where the motor step response is taken as input data. The tests were carried out with two motors of known nominal parameters to compare the error of the improved algorithms. Tests were executed first with simulated signals and later with experimental signals to observe the performance of the algorithms against signals with the typical noise associated with the acquisition of signals.

Finally, the processing time, the accuracy, and the number of necessary iterations of each algorithm are compared. The results indicate a substantial improvement in each algorithm, ranging between 3% and 5% reduction in magnitude error compared to the original algorithm until 50% faster. Additionally, the result indicates that the CS_p proposed algorithm is the one that converges in the fewest iterations.

The rest of the work is divided as follows. In Section 2, a literature review is presented to provide an introspective of the proposed work. In Section 3, the motor model and the dynamic response relationships, both in their transient and steady-state parts. Section 3 described the proposed improvement in each algorithm to include the relations described in part 2. In Section 5, the results at the simulation level and using experimental signals are exhibited. Finally, in Section 6, the conclusions remarking on parameter identification are made.

2. Related Work

Parameter estimation has already been studied from a different point of view. For example, the parameters of the motors when controlling in the open and closed-loop have been proposed not only for AC electrical machines [17], but also for DC motors [18,19]. Knowing the parameters of the dynamic model helps to perform multiple control and automation tasks. For example, to improve the machining performance of the toolset when speaking in a mechatronic process [20].

Some authors focus their investigations with heuristic methods, such as the works sent in [21,22]. Another option highly used in parametric estimation is metaheuristic algorithms. They are widely used for their versatility and adaptability to multiple problems [8]. Works such as [23–25] have explored the analysis of parametric studies of motors with metaheuristic algorithms.

There are a large number of population-based metaheuristic algorithms that can be adapted to parametric estimation. The most extensive metaheuristic algorithm is the genetic algorithms that can be observed as parametric estimators of motors in works such as [26,27]. However, several algorithms were adapted to this aim; for example, the research [28–30] uses the dynamic encoding algorithm for searches (DEAS) for parametric estimation of a motor. In the research [31], a search for harmony (HS) is carried out to obtain the inductance model of an electrical machine. Another example is investigation [32], where a PSO approach is applied to have the parameters of an induction motor. In works [33,34], algorithms such as the whale and the bat are used. Some authors explore the advantages of combining two different metaheuristics algorithms [30]. Finally, some authors such as [29] analyze the online implementation of these algorithms as estimators through parallel processing.

Despite the wide variety of similar works for parameter estimation in motors with metaheuristic algorithms, no work has analyzed the use of dynamic motor relationships as search constraints in both the steady-state and the transient-state. This work aims to analyze the effect of its use in the search with different algorithms, which must be fulfilled. In the Table 1 is shown a comparison between several similar works and the present investigation.

Table 1. Comparison of parameter estimation of DC motor with metaheuristics algorithms in similar works.

Work	Metaheuristic algorithm	Improvement
Our work	Cuckoo Search Algorithm, Gray Wolf Optimizer, Jaya	Dynamic relations used as search limiter
[15]	Cuckoo Search	Pa adjustment based on fuzzy logic
[23]	Cuckoo Search	Use of steady-state
[26]	Genetic Algorithms	Geared DC motor optimal estimation
[24]	Smart Collaborative Performance	Quantify the contribution of the solutions
[25]	Gray Wolf Optimizer	Torque estimation
[26]	Genetic Algorithms	Optimal estimation
[28]	Dynamic Encoding Algorithm for Searches	Control and fault detection
[33]	Whale Optimization Algorithm	Use of WOA as parameter estimator

The table above shows that the research presented here differs from the related works that use dynamic relationships to search for the parameters. Therefore, this approach can be implemented as an additional improvement in the works mentioned and can be extended to any stable open-loop system.

3. Background

This section briefly explains the D.C. motor equations and their dynamic relations, and describes the three metaheuristic algorithms used in this work. The variables involved in the mathematical model of the DC motor are described in Table 2.

Table 2. Variables involved in the mathematical model of the DC motor.

Parameter	Description
$v(t)$	Voltage applied to the motor (V)
$\tau(t)$	Torque generated by the motor (Nm)
$i(t)$	Current consumed by the motor (A)
$\omega(t)$	Rotor angular speed (rad/s)
T_L	Load torque (N.m)
R	Armature resistance (Ω)
L	Armature self-inductance (H)
K	The equivalent value of both the electrical and the mechanical constant
B	Friction coefficient ($\frac{Kgm^2}{s^2}$)
J	Moment of inertia (Nm)

3.1. DC Motor and Its Dynamic Response

The electric motor is a hybrid system composed of an electrical part and a mechanical part. Its dynamic behavior is observed through the electrical and mechanical equations in Equation (1).

$$\left. \begin{aligned} v(t) &= Ri(t) + L \frac{di(t)}{dt} + E(t) \\ \tau(t) &= J \frac{d\omega(t)}{dt} + B\omega(t) + T_L \end{aligned} \right\} \quad (1)$$

The set of equations above is related to the two connection equations depicted in Equation (2).

$$\left. \begin{aligned} E(t) &= K_e \omega(t) \\ \tau(t) &= K_m i(t) \end{aligned} \right\} \quad (2)$$

Substituting Equation (2) in Equation (1) and considering a motor without load ($T_L = 0$), Equation (3) is obtained, and its representation is observed in Figure 1.

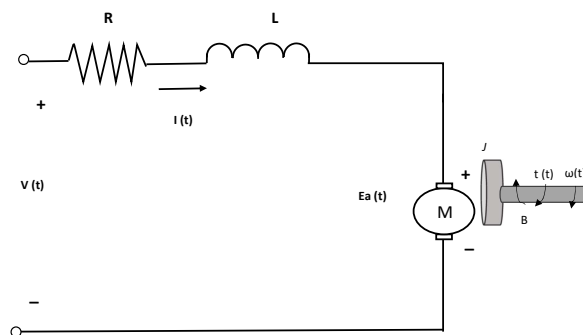


Figure 1. Schematic diagram for a DC motor.

$$\left. \begin{aligned} v(t) &= Ri(t) + L \frac{di(t)}{dt} + K_e \omega(t) \\ k_m i(t) &= J \frac{d\omega(t)}{dt} + B\omega(t) \end{aligned} \right\} \quad (3)$$

The previous differential equation system is the model of a DC motor with six unknown parameters. In addition, if null initial conditions are considered, the system can be rewritten in a second-order transfer function. The second-order transfer functions have a well-known typical dynamic step response, as displayed in Figure 2. The dynamic response is divided into two phases, the transient and the steady-state stage.

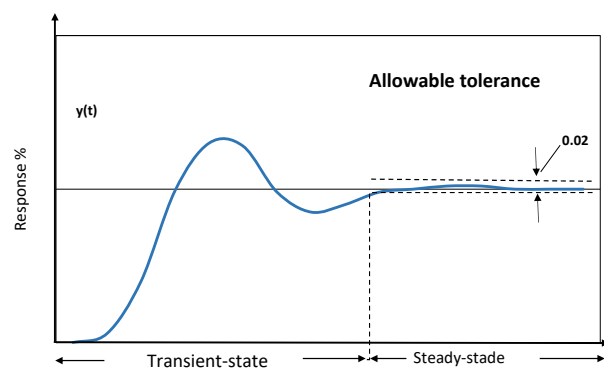


Figure 2. States in dynamic step response.

3.1.1. Steady-State Equations of the Motor

The steady-state is when the output of the system has stabilized, and its response varies less than 2% with respect to time. In this sense, the current and the speed remain constant. Therefore, the derivative of current and the derivative of velocity becomes zero. A fact well-known is that the magnitude of both electrical and mechanical constants is similar.

Therefore the same K value is used for both. Using these considerations, Equation (3) can be rewritten as:

$$\left. \begin{aligned} v_{ss} &= Ri_{ss} + K\omega_{ss} \\ Ki_{ss} &= B\omega_{ss} \end{aligned} \right\} \quad (4)$$

Equation (4) represents the equations of the motor in steady-state of the six initial parameters, the only parameter that remains in both equations is K . Therefore, the system is rewritten, leaving the value of the other parameters as a function of K :

$$\left. \begin{aligned} R &= \frac{v_{ss} - K\omega_{ss}}{i_{ss}} \\ B &= \frac{Ki_{ss}}{\omega_{ss}} \end{aligned} \right\} \quad (5)$$

3.1.2. Transient-State Equations of the Motor

The transient part is when the output of the system has variations with respect to time. However, the relationships in this phase are not obvious if it is known that the current has a directly proportional influence on the derivative of the speed. This relation causes the maximum value of the current to cause the maximum value in the derivative of the current, that is:

$$Kmax(i) = Jmax\left(\frac{d\omega}{dt}\right) + B\omega(t_{max(i)}) \quad (6)$$

From Equation (6), it can be seen that the parameter J can be a function of B and K , since B is already defined as a function of K , if J is cleared, it is only defined as a function of K :

$$J = \frac{B\omega(t_{max(i)}) - Kmax(i)}{max\left(\frac{d\omega}{dt}\right)} \quad (7)$$

Equations (5) and (7) are the relationships of the dynamic response.

3.2. Metaheuristics Algorithms

The use of metaheuristic algorithms has spread in the past few years. This is mainly due to the ease of implementation and adaptation to various problems [8]. This work decided to verify the effect of using dynamic relationships in population-based metaheuristic algorithms. Although multiple tasks can be solved with heuristic methods, the similarity and adaptability to numerous problems and restrictions make metaheuristic algorithms ideal for this study. This work compares three algorithms to determine the effect of dynamic response relations on them.

There are a large number of population-based metaheuristic algorithms in the literature. However, the Cuckoo Search algorithm, the Gray Wolf Optimizer, and the Jaya algorithm were chosen because the first contains an explicit, specific parameter, the GWO has an implicit specific parameter. The Jaya algorithm does not have any specific parameters. Algorithms with two or more specific parameters were avoided because it is required to adjust them. Nevertheless, it should be considered that recent adjustment techniques can solve the problem, such as the hyperparameters adjustment technique, as shown in [35]. The tuning of hyperparameters is another way of improvement for metaheuristics. However, to see only the improvement of the algorithms with the use of dynamic relationships, they were not considered in this work.

3.2.1. Grey Wolf Optimizer

The GWO_O is inspired by how the wolf hunts its prey. The herd is organized by leading wolves (α) that coordinate the hunt in the space and subordinate wolves (β, δ and ω_{wolf}). The GWO_O is fully described in [36], and its process is displayed in Figure 3a. On the other hand, Figure 3b has displayed the GWO_P with the dynamic response relations.

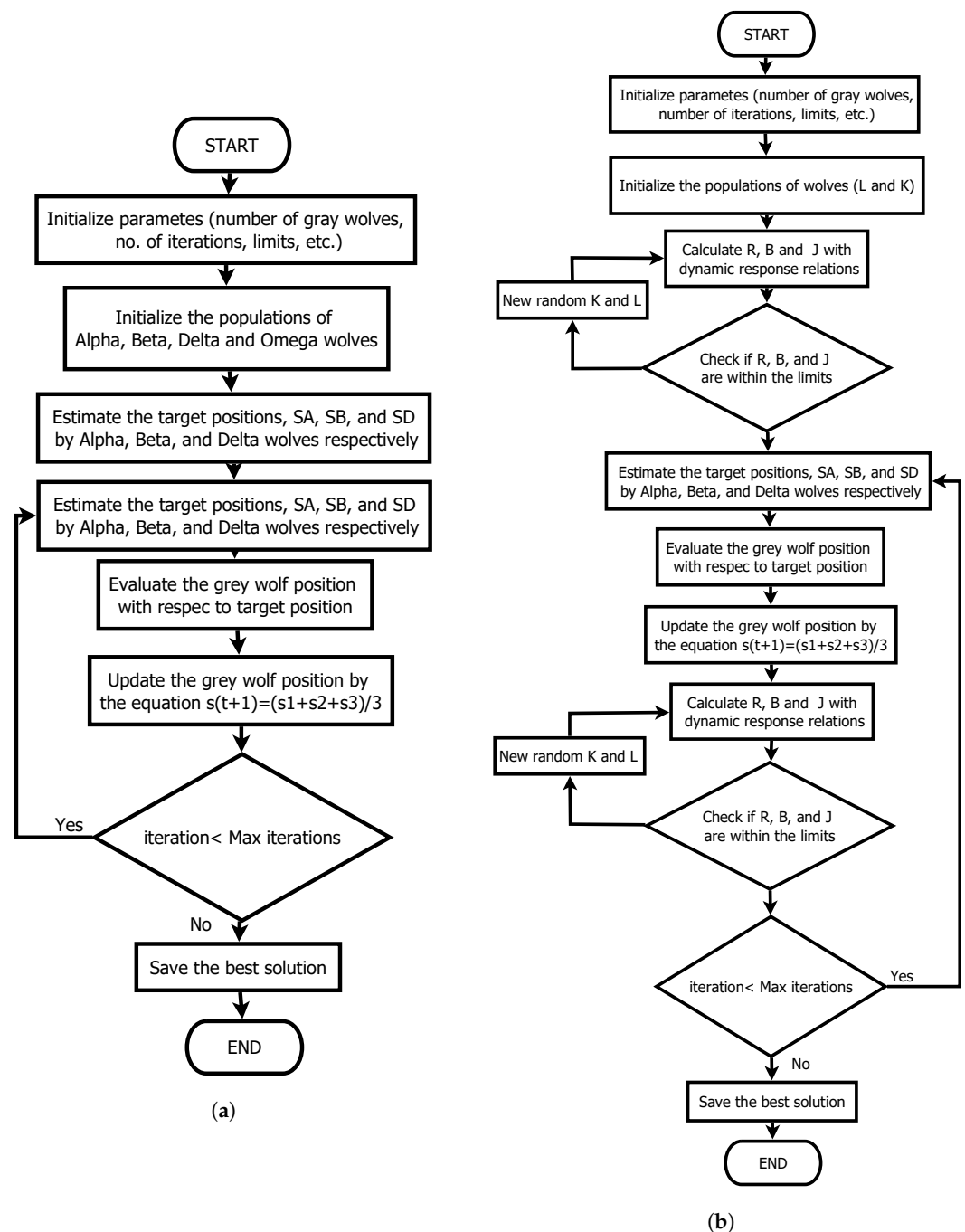


Figure 3. Grey Wolf Optimizer. (a) Original Algorithm; (b) Modified algorithm with dynamics relations.

The GWO_O was chosen among the metaheuristic algorithms since it has a similar performance to the most well-known algorithms in the area, such as Genetic algorithms. It has a simple structure, which allows for fewer computational requirements. It also reduces its search space increasing its convergence speed compared to other algorithms [37].

3.2.2. Algorithm of Jaya

Jaya is a relatively new algorithm. However, like GWO_O , it has similar results to the more popular search methods like GA. Its main feature is simplicity since it has only general parameters and no specific parameters. It is only based on the best and worst solution, looking for values close to the best solution and moving away from the worst [38].

The complete method can be studied in detail in [38]. Figure 4a displays the $Jaya_O$ while in Figure 4b, the $Jaya_P$ with the dynamic relationships of the motor is depicted.

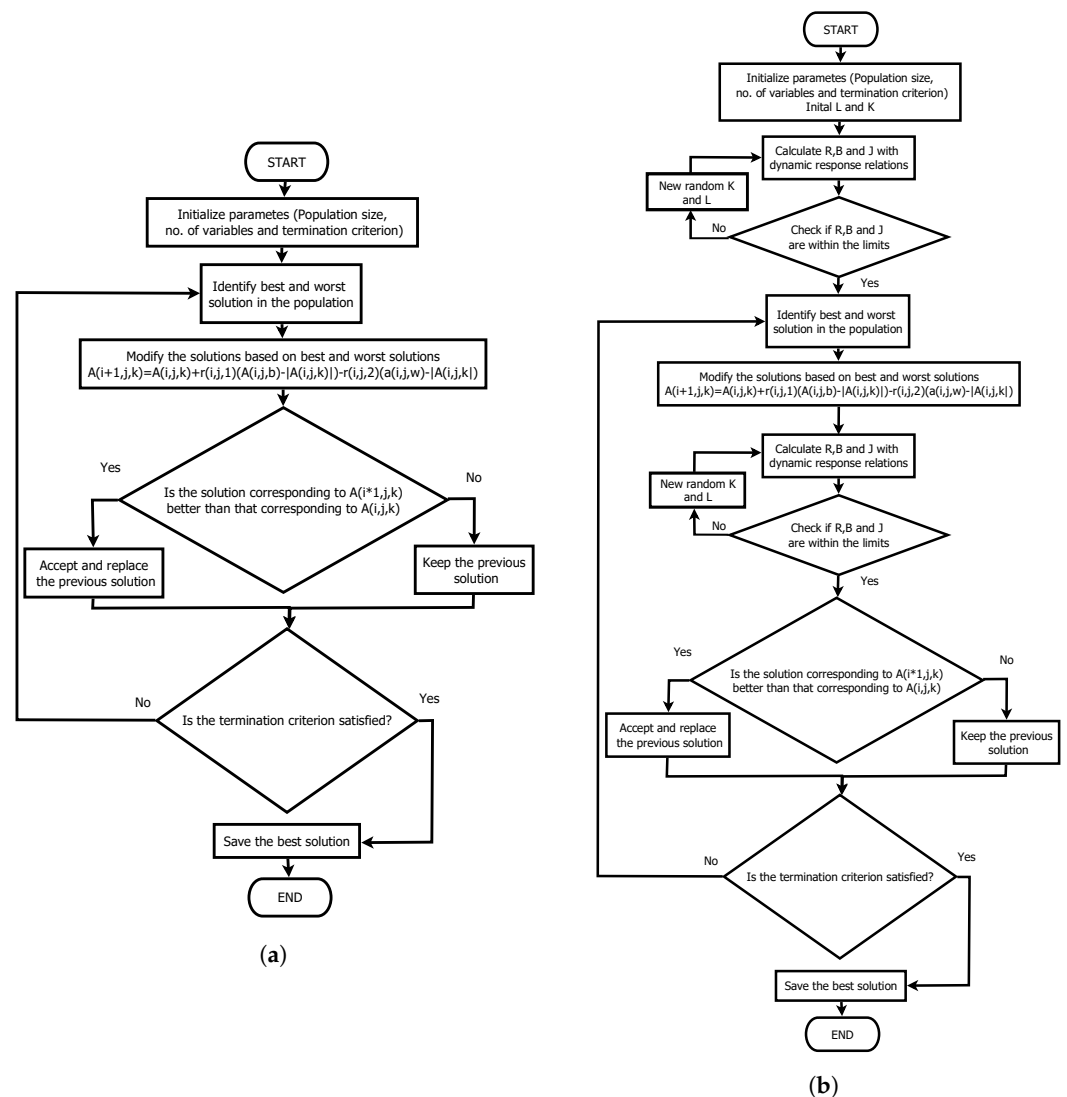


Figure 4. Jaya algorithm. (a) Original Algorithm; (b) Modified algorithm with dynamics relations.

The Jaya algorithm was chosen for reasons similar to the GWO_O . The algorithm exhibits a fast convergence rate and is among the simplest metaheuristic algorithms to implement.

3.3. The Cuckoo Search Algorithm

The CS_O is not found among the simplest algorithms unlike the other two algorithms. Unlike the other two algorithms. However, like the previous ones, it has shown similar results to the GA, but the convergence is observed in a smaller number of iterations. Like GWO_O , it is a bioinspired algorithm based on the peculiar way Cuckoo birds reproduce by pinning their eggs in other nests for different birds to raise. Unlike the other two algorithms, the CS_O has a specific parameter. Determining the value of any particular parameter is a complex subject of study. However, in the literature, a value of 25% is recommended [39]. The complete method is described in [39] where the author of the algorithm tests the algorithm in problems of a different nature. The differences between the CS_O and the CS_P can be observed in the flow diagram of both procedures in Figure 5.

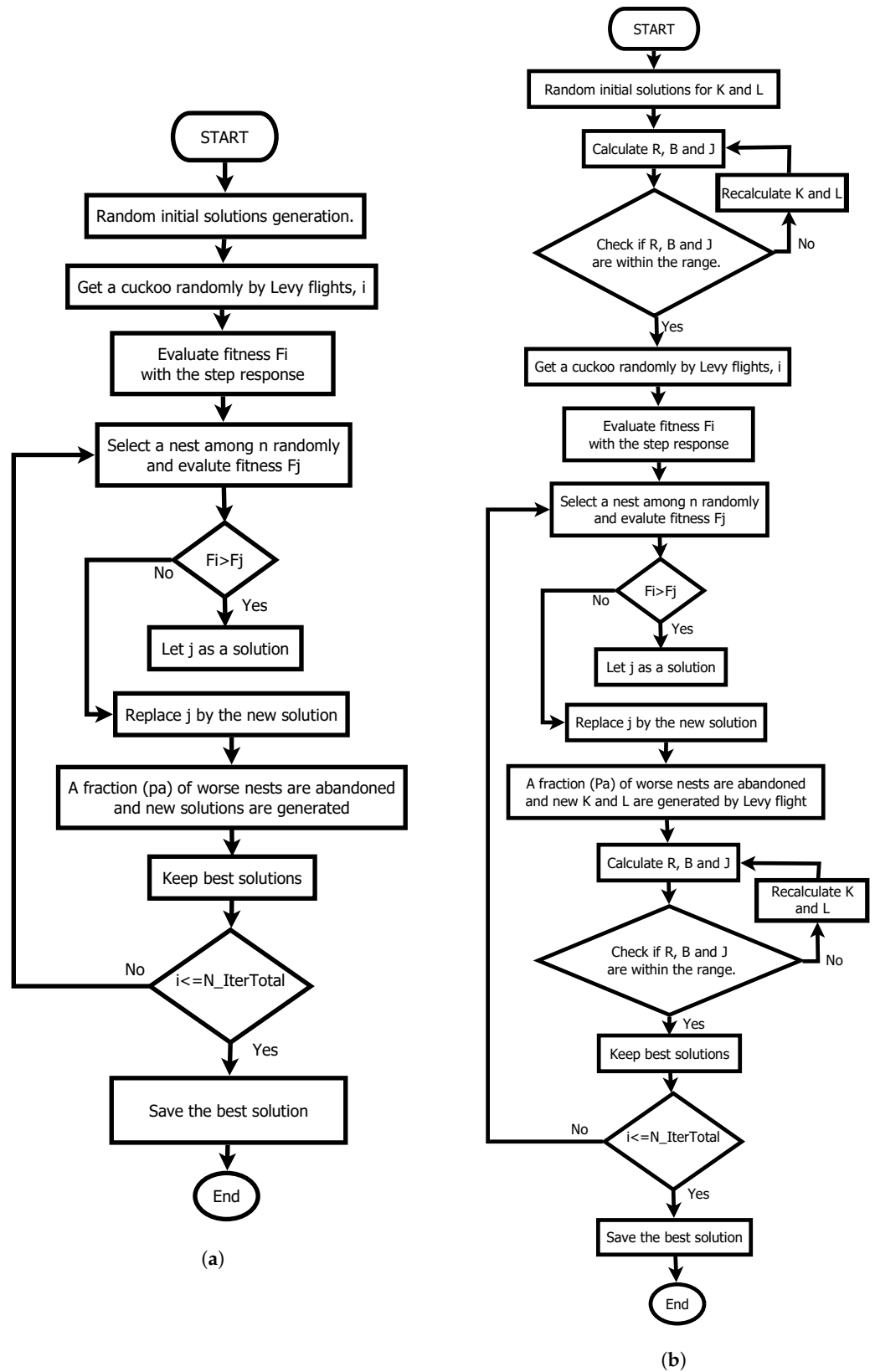


Figure 5. Cuckoo Search algorithm. (a) Original Cuckoo search; (b) Modified algorithm with dynamics relations.

The methods used were the Cuckoo Search, the Gray Wolf Optimizer, and the Jaya algorithm was chosen to contrast with the other two algorithms as it has a specific parameter and uses the levy flight hence implementation complexity is greater. However, its performance and application have been tested in multiple areas. Therefore, it is presented as a logical option since, in contrast, the most used metaheuristic (GA) has multiple specific parameters, making a comparison complicated since they must be justified in selecting each parameter.

4. Methodology

The methods used were the Cuckoo Search, the Gray Wolf Optimizer, and the Jaya algorithm. All techniques started from original random populations and generated optimal solutions for a fitness/cost function.

The proposal of the present investigation lies in the use of Equation (5) and (7) to reduce the search for random parameters. Instead of searching five parameters simultaneously, only K and L are randomly generated. For this, the dynamic relations are implemented as search restrictions. In this way, the parameters J , B , and R can be determined by the dynamic equations and as a function of the value proposed by K and L . Thanks to this, the algorithm focuses only on finding these last two parameters.

In this work, the fitness/cost function is the Euclidean distance between the sum of error in current and the sum of error in velocity, that is:

$$f = \sqrt{\sum (i_r - i_s)^2 + \sum (w_r - w_s)^2} \quad (8)$$

The goal of metaheuristic algorithms is to reduce the value of the Equation (8). The three methods are population-based and have several candidate solutions that report data about the search, causing abrupt jumps in the direction of the hopeful solution. In addition, having an initial population with a high number of individuals generates a greater diversity which helps to avoid a local optimum [40].

All metaheuristic algorithms contain parameters that modify their performance. Some parameters are similar or even equivalent. On the other hand, some parameters are specific for each type of method. Table 3 shows the parameters used for each algorithm with the original names given by the respective author and the similar or equivalent parameters.

Table 3. Name parameters and equivalence in each algorithm.

Parameter	GWO	Jaya	CS	Description
Population	Search Agents_no	pop	nest	Number of initial random solution
Iterations	Max_iteration	maxGen	N_IterTotal	
Limits	lb,ub	mini, maxi	lb,ub	upper and lower boundaries of the search
Variables to search	dim	var	nd	Number of variables to search
Fitness/cost function	fobj	objective	fobj	Fitness function used (same for three algorithms)
Pa	-	-	Pa	Probability of the alien egg being discovered

The three algorithms have the distinction of having a reduced number of parameters, and most of these have their equivalence in other algorithms (see Table 3). This characteristic is one of the reasons for which they were chosen.

The tests of the original and modified algorithms with the dynamic relationships are developed (a total of six algorithms). In addition, two different DC motors will be used to analyze the performance. As a performance measure of each algorithm, the value of the cost function that it reaches was taken, with a cost of 0 being the maximum possible performance. Another factor measured in the tests is the time it takes for the algorithm to complete the search task. The population parameters and maximum iterations were varied by performing the following tests:

- Test 1: Population of 50 individuals and 100 iterations.
- Test 2: Population of 30 individuals and 100 iterations.
- Test 3: Population of 50 individuals and 50 iterations.
- Test 4: Population of 30 individuals and 50 iterations.

A large number of individuals provides greater diversity, and a large number of iterations provides an exhaustive search. However, it should be considered that a greater number of individuals and iterations increasing the computational cost exponentially. The chosen values show allow observing the change in results between runs with twice the number of individuals or iterations. The other common parameters were remain fixed for all the tests. As shown in Table 3, the cost function was in common for the six algorithms, which is determined by Equation (8). The number of variables to be proposed randomly is two (K and L), and the number of variables to be calculated is three (R, B, and J). The initial search limits will also be the same for all algorithms. The upper limit is determined by $U_b = [5 \ 0.1 \ 0.5 \ 0.001 \ 0.001]$ and the lower limit for $L_b = [0.1 \ 0.01 \ 0.005 \ 0.0000001 \ 0.00001]$ corresponding to the vector [R, K, L, J, B].

The first motor used is the Robokits RMCS2004 motor with the following nominal parameters = $[0.921042 \ \Omega, 0.073472, 0.007759 \text{ H}, 0.000136 \text{ Nm}, 0.000678 \frac{\text{Kgm}^3}{\text{s}^2}]$, referred to in this work as M1. Motor 2 is the Mavilor CLM-050 (M2) with the following nominal parameters = $[3.1363 \ \Omega, 0.048774, 0.01307 \text{ H}, 0.000009 \text{ Nm}, 0.000169 \frac{\text{Kgm}^3}{\text{s}^2}]$, this motor is referred in this work as M2.

The tests are performed with step response signals obtained with simulation and measured by sensors and a data acquisition system. The signals obtained by simulation are called simulated signals and those obtained by measurements are called experimental signals. For the simulation tests, the motor model described by Equation (3) is simulated by Simulink, while the metaheuristic algorithms are developed using Matlab. Thus Matlab obtains a random combination of each method and sends the data to Simulink, which returns a current and velocity signal to Matlab. The software calculates the dynamic relationships and estimates a cost according to each algorithm with these signals. In the case of the experimental signals, only measured data were used.

5. Results

In this section, the results obtained for both tests carried out with simulated signals, and those with experimental signals. The following results were found:

- the use of dynamic relationships and the estimation of parameters through metaheuristic algorithms improve the search speed of the parameters;
- this limits the number of results to valid combinations for all parameters.

5.1. Results with Simulation Signals

The results of the tests performed with original and proposed algorithms are seen in Table 4.

Table 4. Cost/Fitness result for each test in M1 and M2.

Test	Motor Cost	GWO_O Cost	GWO_P Cost	$Jaya_O$ Cost	$Jaya_P$ Cost	CS_O Cost	CS_P Cost
Test 1	M1	0.19	0.08	14.66	4.98×10^{-4}	0.003	3.091×10^{-5}
Test 2	M1	0.08	0.17	8.25	4.97×10^{-4}	0.004	1.232×10^{-4}
Test 3	M1	0.12	0.03	99.66	5.63×10^{-4}	0.455	4.837×10^{-4}
Test 4	M1	24.33	0.07	94.34	7.67×10^{-4}	0.170	1.911×10^{-3}
Test 1	M2	16.96	0.05	1.59	2.65×10^{-3}	0.001	4.973×10^{-6}
Test 2	M2	0.88	0.55	7.34	2.65×10^{-3}	0.001	1.692×10^{-5}
Test 3	M2	28.28	0.13	36.37	2.70×10^{-3}	0.194	4.554×10^{-4}
Test 4	M2	0.63	0.10	24.52	2.71×10^{-3}	0.671	1.434×10^{-3}

The performance of each algorithm in the different tests was represented in the following graphs. Figure 6 displays the performance of GWO_O against the GWO_P for the M1.

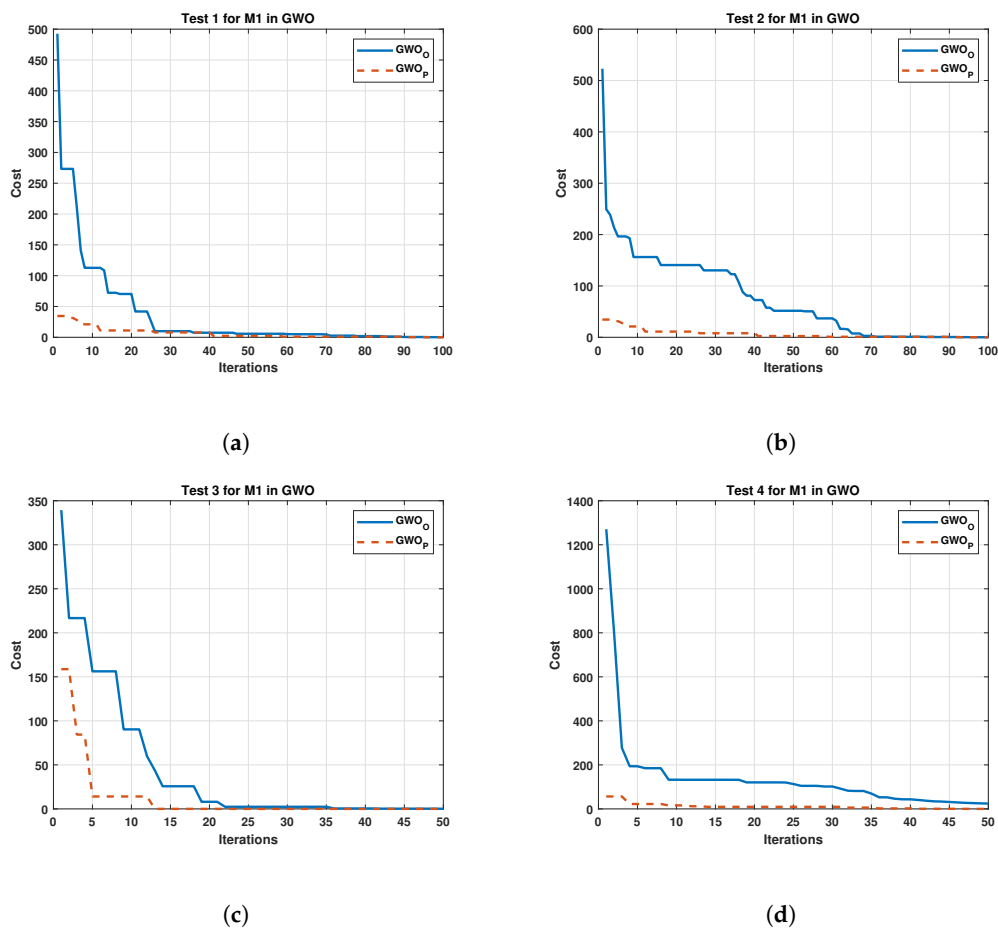


Figure 6. Comparison between the Original and the Gray Wolf Optimizer Proposed in four test for Motor 1. (a) cost and iterations of the GWO in test 1 for Motor 1; (b) cost and iterations of the GWO in test 2 for Motor 1; (c) cost and iterations of the GWO in test 3 for Motor 1; (d) cost and iterations of the GWO in test 4 for Motor 1.

The test results for Motor 2 are exhibit in Figure 7.

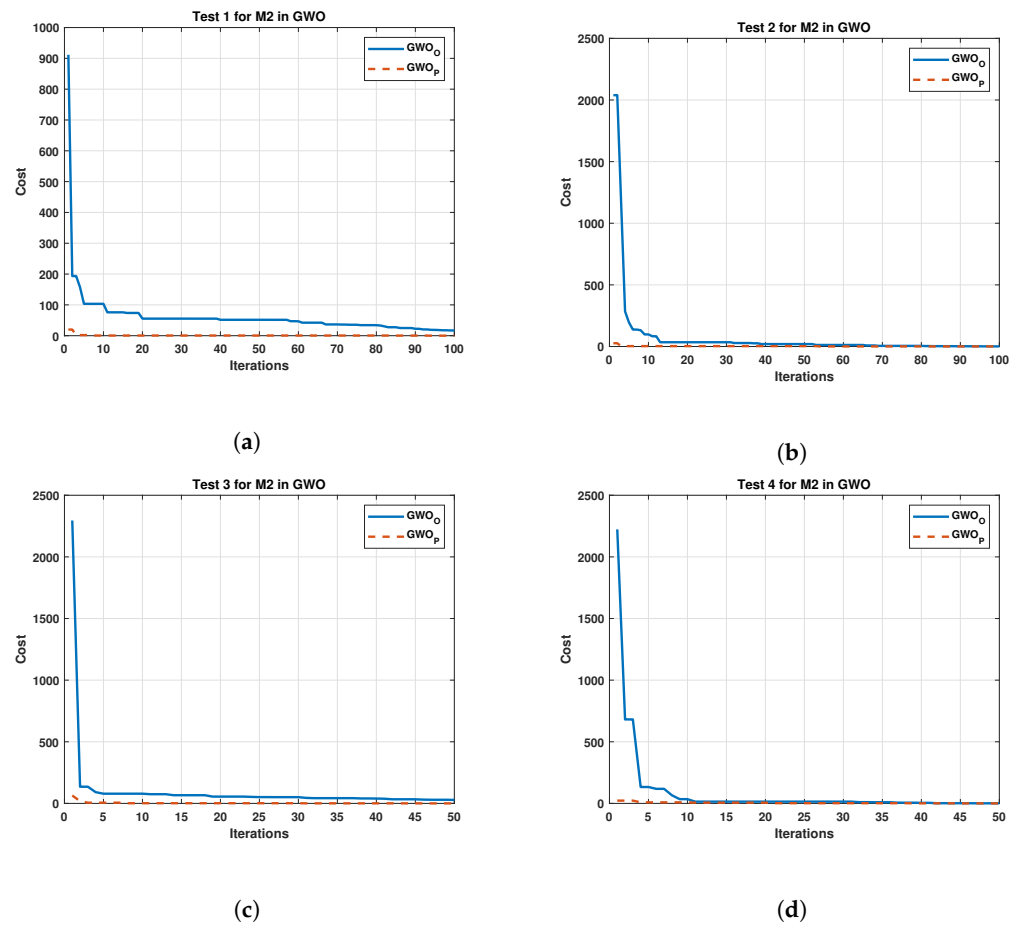


Figure 7. Comparison between the Original and Proposed Gray Wolf Optimizer in four test for Motor 2. (a) cost and iterations of the GWO in test 1 for Motor 2; (b) cost and iterations of the GWO in test 2 for Motor 2; (c) cost and iterations of the GWO in test 3 for Motor 2; (d) cost and iterations of the GWO in test 4 for Motor 2.

Both Figures 6 and 7 present an improvement for the GWO_O , achieving a noticeable initial reduction in cost. The GWO_P shows a cost reduction below the original in the following iterations. On the other hand, Figure 8 exhibits the results of the algorithm of Jaya in tests carried out in M1.

The Jaya algorithm results have a similar effect as the GWO, which significantly improves the initial iterations. For tests 1 and 2, the cost reduces similarly in both cases as the iterations increase. However, in cases 3 and 4, the original algorithm does not have proper convergence, unlike the proposed one. Similarly, Figure 9 displays the results obtained by Jaya for M2.

Considering the results of Jaya in motor 2, a similar behavior to motor 1 is observed, suggesting that the motor used does not affect the algorithm. Finally, the results of the Cuckoo search algorithm are shown in Figure 10.

In the same way, the results of the tests on the M2 are depicted in Figure 11.

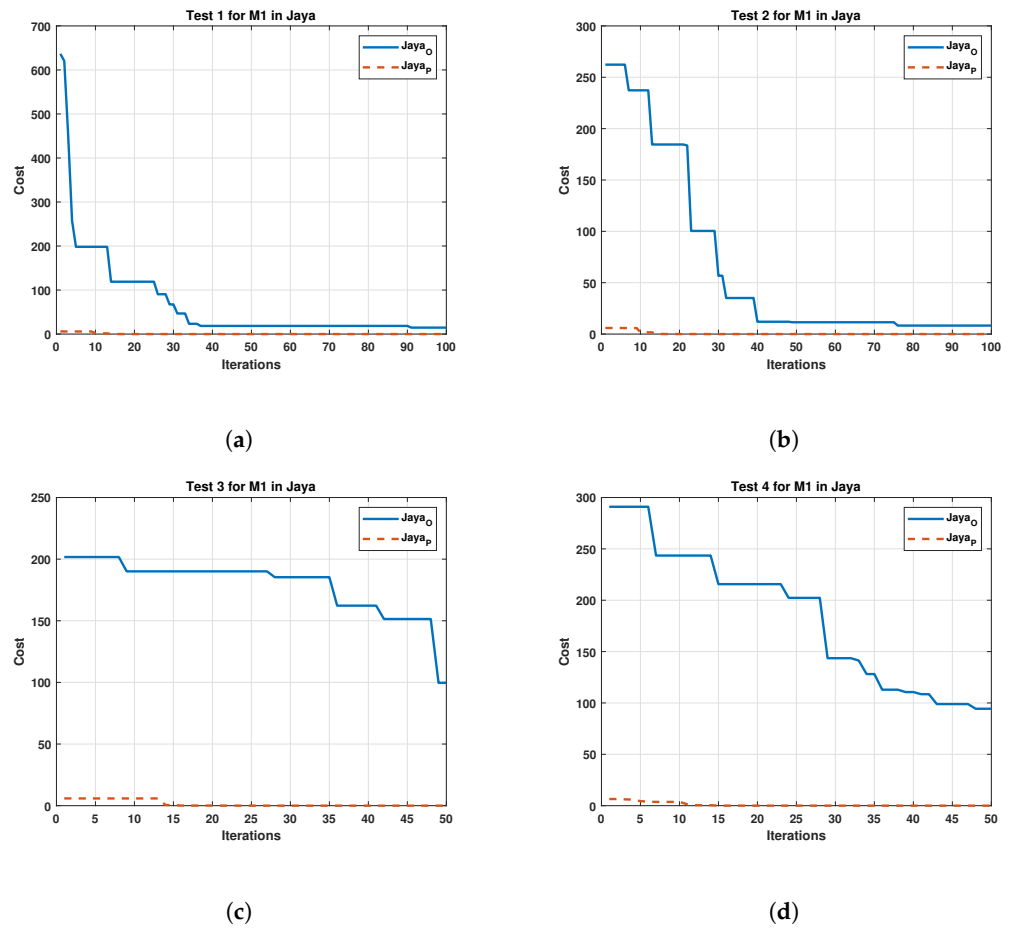


Figure 8. Comparison between the Original and the Proposed Jaya in four test for Motor 1. (a) cost and iterations of the Jaya in test 1 for Motor 1; (b) cost and iterations of the Jaya in test 2 for Motor 1; (c) cost and iterations of the Jaya in test 3 for Motor 1; (d) cost and iterations of the Jaya in test 4 for Motor 1.

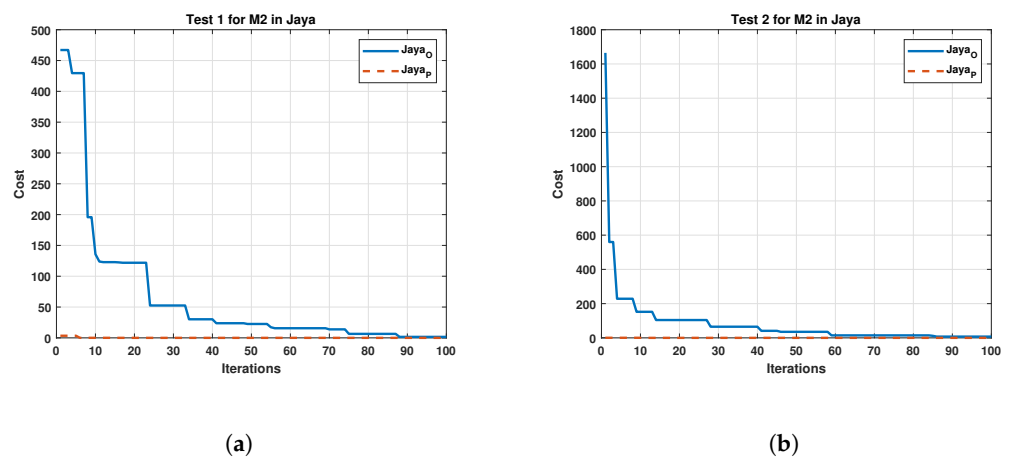


Figure 9. Cont.

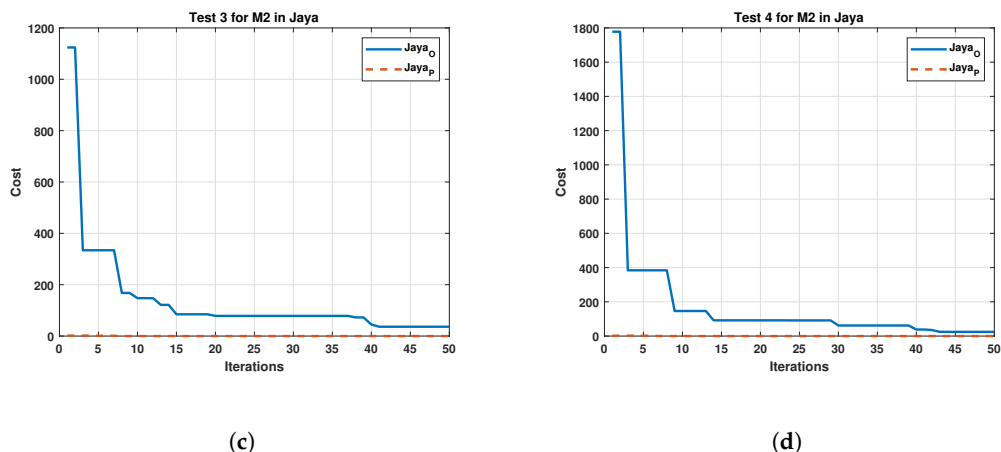


Figure 9. Comparison between the original and the proposed Jaya in four test for Motor 2. (a) cost and iterations of the Jaya in test 1 for Motor 2; (b) cost and iterations of the Jaya in test 2 for Motor 2; (c) cost and iterations of the Jaya in test 3 for Motor 2; (d) cost and iterations of the Jaya in test 4 for Motor 2.

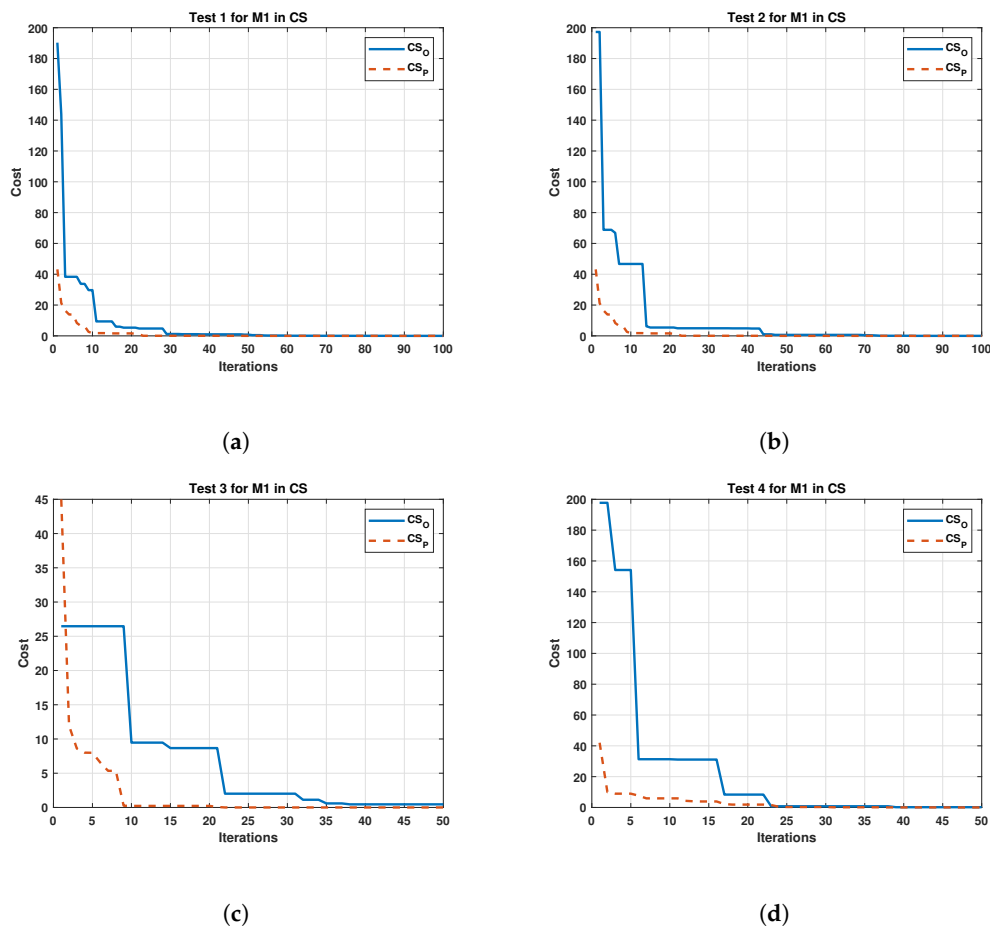


Figure 10. Comparison between the original and the proposed Cuckoo Search in four tests for Motor 1. (a) cost and iterations of the Cuckoo Search in test 1 for Motor 1; (b) cost and iterations of the Cuckoo Search in test 2 for Motor 1; (c) cost and iterations of the Cuckoo Search in test 3 for Motor 1; (d) cost and iterations of the Cuckoo Search in test 4 for Motor 1.

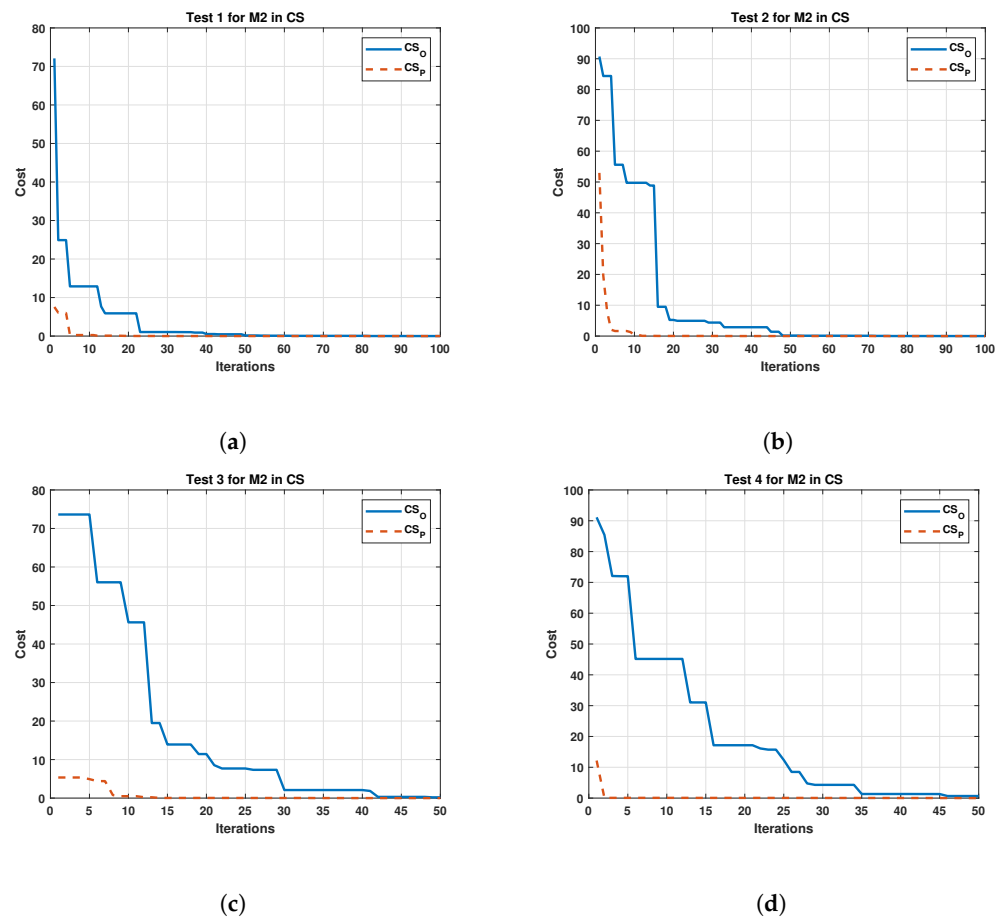


Figure 11. Comparison between the original and the proposed Cuckoo Search in four test for Motor 2. (a) cost and iterations of the Cuckoo Search in test 1 for Motor 2; (b) cost and iterations of the Cuckoo Search in test 2 for Motor 2; (c) cost and iterations of the Cuckoo Search in test 3 for Motor 2; (d) cost and iterations of the Cuckoo Search in test 4 for Motor 2.

The absolute percentage error was calculated in each test with respect to the nominal parameters to observe the performance of the algorithm in terms of precision. The results were summarized in Table 5 and Table 6 for M1 and M2, respectively.

Both numerical and tabular improvements are observed in the modified algorithms compared to the originals. Therefore, subsequent results will focus on the modified algorithms. According to the observed results, the modified algorithm convergence is faster, adjusting the cost curves in the first iterations. Considering the adjustment speed was programmed, a condition for when the cost function obtains a value less than 0.1, the search stopped. The next test was executed in this way for the algorithm convergence speed comparison and was developed with 30 individuals for all algorithms. The results are displayed in Figure 12. Figure 12a shows the convergence speed in iterations for Motor 1, and Figure 12b depicted the convergence speed in iterations for Motor 2.

It can be observed that convergence in the least number of iterations occurs in the Cuckoo search algorithm.

Table 5. The absolute percentage error for test in Motor 1.

Test	Algorithm	R %	K %	L %	J %	B %
Test1	<i>GWO_O</i>	1.557	0.355	6.721	2.317	0.182
	<i>GWO_P</i>	1.473	0.173	0.591	0.397	0.173
	<i>Jaya_O</i>	15.081	3.801	55.180	12.917	0.340
	<i>Jaya_P</i>	0.012	0.001	0.210	0.225	0.001
	<i>CS_O</i>	0.252	0.029	0.068	0.290	0.029
	<i>CS_P</i>	0.008	0.001	0.041	0.027	0.001
Test2	<i>GWO_O</i>	0.262	0.161	4.030	0.198	0.077
	<i>GWO_P</i>	0.653	0.076	4.476	0.300	0.076
	<i>Jaya_O</i>	5.623	0.396	55.180	20.540	0.899
	<i>Jaya_P</i>	0.013	0.001	0.194	0.225	0.001
	<i>CS_O</i>	0.198	0.023	1.084	0.226	0.023
	<i>CS_P</i>	0.022	0.002	0.069	0.098	0.002
Test3	<i>GWO_O</i>	1.445	0.020	3.555	0.508	0.391
	<i>GWO_P</i>	1.355	0.159	1.391	0.383	0.159
	<i>Jaya_O</i>	20.539	0.572	55.180	168.091	12.048
	<i>Jaya_P</i>	0.014	0.002	0.300	0.225	0.002
	<i>CS_O</i>	3.137	0.376	9.907	3.560	0.376
	<i>CS_P</i>	0.004	0.001	0.212	0.215	0.001
Test4	<i>GWO_O</i>	13.269	2.785	53.598	66.302	0.375
	<i>GWO_P</i>	1.233	0.141	0.437	0.082	0.141
	<i>Jaya_O</i>	63.585	21.087	57.780	55.245	35.513
	<i>Jaya_P</i>	0.151	0.018	0.191	0.241	0.018
	<i>CS_O</i>	0.369	0.043	5.891	3.551	0.043
	<i>CS_P</i>	0.241	0.028	0.269	0.251	0.028

Table 6. The absolute percentage error for test in Motor 2.

Test	Algorithm	R %	K %	L %	J %	B %
Test1	<i>GWO_O</i>	29.525	4.733	61.008	70.750	5.774
	<i>GWO_P</i>	0.244	0.054	5.024	0.755	0.054
	<i>Jaya_O</i>	9.779	2.480	3.253	4.529	5.228
	<i>Jaya_P</i>	0.087	0.020	0.667	0.829	0.020
	<i>CS_O</i>	0.203	0.045	0.086	0.264	0.045
	<i>CS_P</i>	0.025	0.005	0.024	0.023	0.005
Test2	<i>GWO_O</i>	5.792	1.448	19.533	6.690	1.554
	<i>GWO_P</i>	2.773	0.640	18.245	1.454	0.640
	<i>Jaya_O</i>	9.550	1.637	28.856	17.807	0.466
	<i>Jaya_P</i>	0.060	0.013	0.703	0.823	0.013
	<i>CS_O</i>	0.307	0.069	0.632	0.459	0.069
	<i>CS_P</i>	0.053	0.012	0.025	0.013	0.012
Test3	<i>GWO_O</i>	11.298	1.530	67.539	192.484	2.660
	<i>GWO_P</i>	0.936	0.206	8.692	0.601	0.206
	<i>Jaya_O</i>	6.186	0.080	65.769	162.739	6.260
	<i>Jaya_P</i>	0.121	0.027	0.637	0.836	0.027
	<i>CS_O</i>	0.291	0.065	10.560	5.329	0.065
	<i>CS_P</i>	0.028	0.006	0.551	0.102	0.006
Test4	<i>GWO_O</i>	4.496	0.966	12.593	12.692	0.737
	<i>GWO_P</i>	2.004	0.436	4.146	0.370	0.436
	<i>Jaya_O</i>	35.942	14.237	161.400	28.310	6.117
	<i>Jaya_P</i>	0.000	0.000	0.665	0.809	0.000
	<i>CS_O</i>	4.296	0.909	13.716	11.149	0.909
	<i>CS_P</i>	0.113	0.025	0.035	0.556	0.025

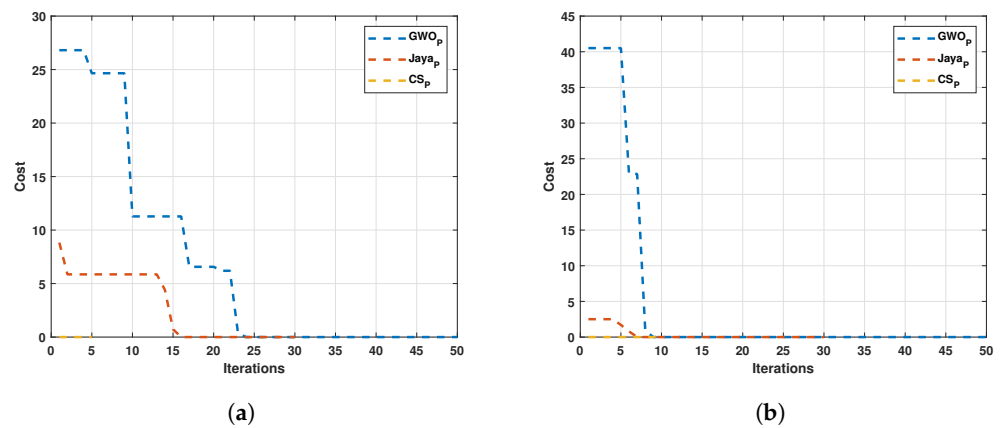


Figure 12. Comparison between the three modified algorithms. (a) Tests for Motor 1; (b) Tests for Motor 1.

5.2. Results with Experimental Signals

In this section we tested the proposed algorithms with experimental signals using an acquisition system of our design based in a simple PIC18F4550 with a sampling period of 0.001 s and implemented in M2. The algorithms work with experimental signals to verify the performance of the three modified algorithms under real conditions, unlike the previous section, the signals acquired from the real world usually have noise and a higher uncertainty which is a function of the measurement system and the conditions in which the measurement was taken. For this case, all algorithms were used with a population of 30 individuals and a maximum of 50 iterations. The sampling process was as follows: at 0.5 s, a voltage of 10.5 constant was applied, and the measured variables were voltage, current, and motor speed, for which a 12-bit ADC, the Hall effect sensor Acs712 and the encoder integrated to the motor are used. Figure 13 displays the setup of the acquisition system.

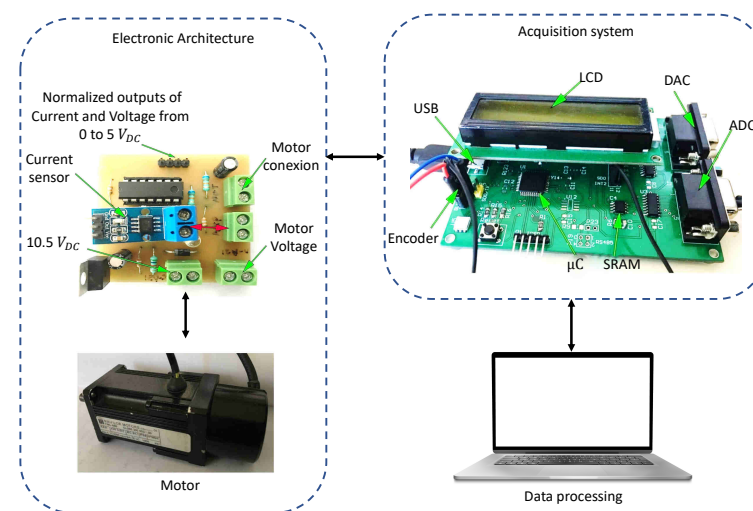


Figure 13. Data acquisition system used for the experimental setup.

Although signals can be acquired with such basic equipment, a faster acquisition card and better performing current sensor would help to obtain better quality signals. The results of the acquisition for the step signal used can be observed in Figure 14.

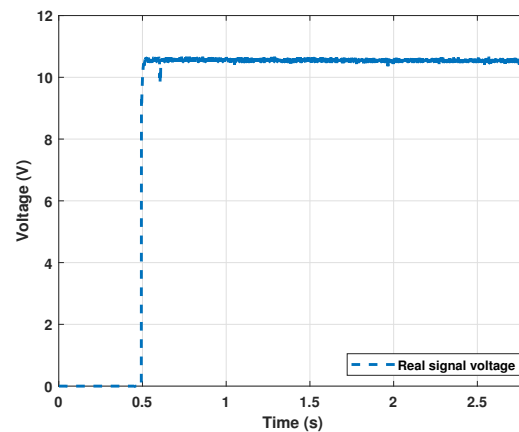


Figure 14. Step voltage signal used.

Figure 15 shows the current and velocity signals. Due to the acquisition conditions, the speed signal must be filtered. For this, the Chebyshev type 1 digital filter is used. Excessive noise is characteristic of the signal acquisition system and the sensor used. Improving the acquisition hardware would give better signals. However, this is not always possible.

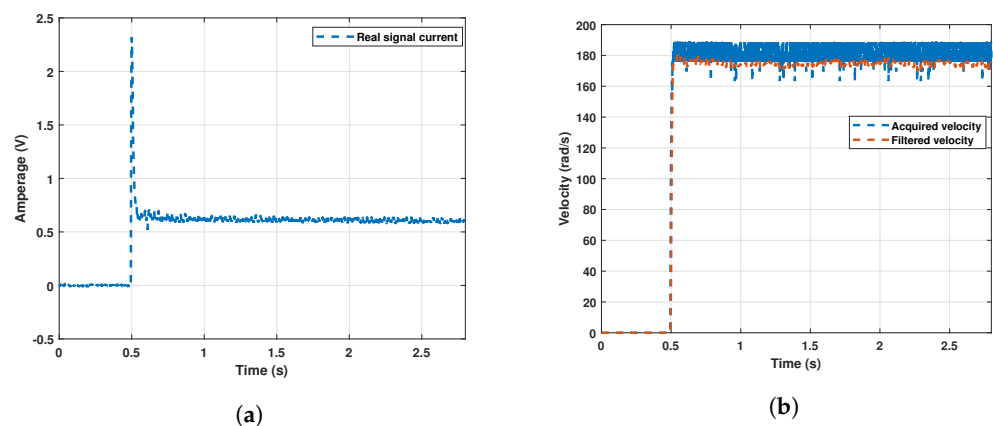


Figure 15. Sensed velocity and current signals for Motor 1. (a) Current signal of Motor 1; (b) Velocity signal of Motor 1.

With these signals, the dynamic response relations are calculated. For estimating the final current and the final speed, the last 20 values are averaged since the signals present minor variations, unlike the simulated signals. Again, these variations are attributed to the hardware. The performance of each algorithm can be seen in Figure 16.

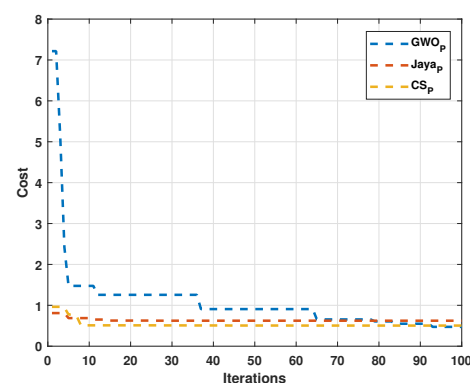


Figure 16. Performance of the proposed algorithms with experimental signals.

Figure 17 displays the comparison between the current signal and the speed signal calculated for the three modified algorithms with respect to the acquired signal.

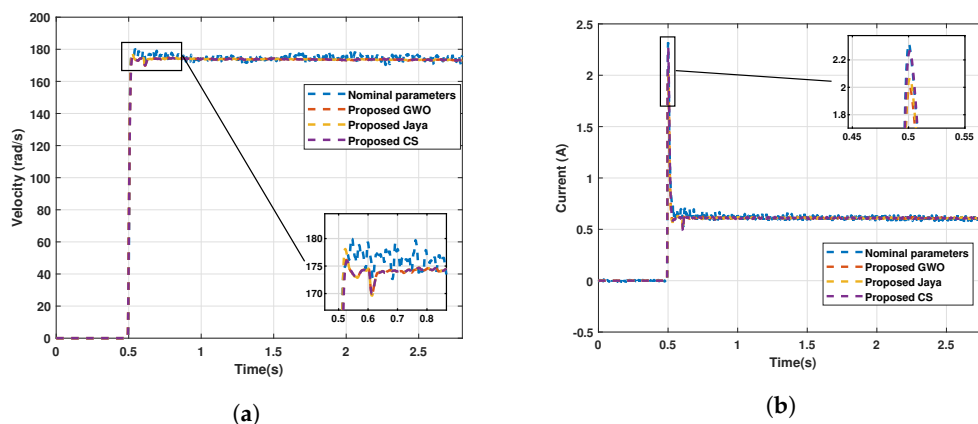


Figure 17. Comparison between the three modified algorithms for experimental signals. (a) Comparison between velocities estimates and acquired velocity; (b) Comparison between currents estimates and acquired current.

The results of experimental signals acquired with noise suggest that the modified algorithms can correctly estimate the parameters to recreate the current and velocity signals.

6. Conclusions

This article presents the use of dynamic response relations as search constraints in metaheuristic algorithms used as parametric estimators. The three relationships were used, two in the steady-state and one in the transient-state. It was implemented in three different algorithms with two plants to test the validity of the use of relations. The three relationships founded were used, two in the steady-state and one in the transient state. The proposed method is adaptable to any system with a stable response to the step input, and has at least one relationship in dynamic response. Therefore, this method developed in DC motors can be extended to systems with similar models.

DC motor modeling is a widely studied topic. In particular, the parameter identification with metaheuristic algorithms has shown satisfactory results with multiple techniques. However, the study with the dynamic relationships used as search restrictions has not been studied and most of the improvement techniques are oriented to the optimization of hyperparameters (see Table 1). The relations were implemented in three different algorithms with two plants to test the validity of their implementation.

In general terms, the results and the improvement in each algorithm depend on the initial values that the algorithm takes. Being an algorithm that starts from random values, a precise improvement cannot be quantified. Nevertheless, the results show an evident improvement in each algorithm, especially in the first iterations, where the error is reduced in a smaller number of iterations due to the use of dynamic response relations. Therefore, a lower error and a faster convergence can be observed for all algorithms. The trend holds for different motors, although the performance of the GWO is higher with the M1 than with the M2.

Metaheuristic algorithms with similar or equivalent parameters and with no specific parameters were used to reduce the effect of selecting parameters in different algorithms, achieving a more fair comparison. Reducing iterations and initial errors can help in the implementation in multiple systems since the main problem with metaheuristic algorithms is their high iterative degree, which complicates their implementation.

Metaheuristic algorithms with similar or equivalent parameters and that did not have specific parameters were used to reduce the effect of selecting parameters in different algorithms, achieving a more fair comparison. Reducing iterations and initial errors can

help its implementation in multiple systems since the main problem with metaheuristic algorithms is their high iterative degree, which complicates their implementation.

Finally, it should be noted that the metaheuristic algorithms start from random populations, thus the performance can vary according to the initial values that it takes. Four different tests were carried out in two motors to avoid this bias in the three original and modified algorithms. In total, forty-eight tests were developed to verify that cost reduction and iteration reduction are maintained despite the variation of the initial population and the search parameters. Future work should aim to make the tests with the adjustment of hyperparameters and measure execution times with the objective of parallel online implementations.

Author Contributions: Conceptualization, O.R.-A.; methodology, O.R.-A.; software, O.R.-A. and J.R.-R.; validation, O.R.-A.; formal analysis, O.R.-A., J.M.Á.-A. and A.G.-C.; investigation, O.R.-A.; resources, O.R.-A.; data curation, O.R.-A. and A.G.-C. writing—original draft preparation, O.R.-A. and A.G.-C.; writing—review and editing, O.R.-A. and J.R.-R.; visualization, O.R.-A.; supervision, O.R.-A., J.R.-R., A.G.-C. and J.M.Á.-A.; project administration, O.R.-A. and J.R.-R.; funding acquisition, O.R.-A. and J.M.Á.-A. 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: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to acknowledge the support of Universidad Politécnica de Querétaro in the production of this work. This research was done with the help from Universidad de Málaga and Autonomus University of Queretaro.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

CS_O	Original Cuckoo Search Algorithm
WGO_O	Original Gray Wolf Optimizer
$Jaya_O$	Original Jaya Algorithm
CS_P	Proposed Cuckoo Search Algorithm for motor parameter estimation
WGO_P	Proposed Gray Wolf Optimize for motor parameter estimation
$Jaya_P$	Proposed Jaya Algorithm for motor parameter estimation
GA	Genetic algorithm
DC	Direct current
ess	Steady-stade error
R	Armature resistance
L	Armature self-inductance
T_L	Load torque
K_m	Mechanical constant
K_e	Electrical constant
B	Friction coefficient
J	Moment of inertia
K	The equivalent value of both the electrical and the mechanical constant.
$\tau(t)$	Motor torque
$E(t)$	Back electromotive force
$\omega(t)$	Angular velocity of rotor

$v(t)$	Voltage
$i(t)$	Current
v_{ss}	Voltage in steady-state
i_{ss}	Current in steady-state
ω_{ss}	Angular velocity in steady-state
$\max(i)$	Maximum current
$\omega(t_{\max(i)})$	Time in which the maximum current occurs
f	fitness/cost function
Ub	Upper boundary
Lb	Lower boundary
i	Current response to the step of motor
i_s	Current response to the step of the parameters proposed in a simulation
ω	Angular velocity response to the step of motor
ω_s	Angular velocity response to the step of the parameters proposed in a simulation
$\max(\frac{d\omega}{dt})$	Maximum angular acceleration

References

1. Yang, S.; Zhou, S.; Hua, Y.; Zhou, X.; Liu, X.; Pan, Y.; Ling, H.; Wu, B. A parameter adaptive method for state of charge estimation of lithium-ion batteries with an improved extended Kalman filter. *Sci. Rep.* **2021**, *11*, 5805. [[CrossRef](#)] [[PubMed](#)]
2. Mehmood, A.; Chaudhary, N.I.; Zameer, A.; Raja, M.A.Z. Novel computing paradigms for parameter estimation in Hammerstein controlled auto regressive auto regressive moving average systems. *Appl. Soft Comput.* **2019**, *80*, 263–284. [[CrossRef](#)]
3. Nampoothiri, M.G.H.; Vinayakumar, B.; Sunny, Y.; Antony, R. Recent developments in terrain identification, classification, parameter estimation for the navigation of autonomous robots. *SN Appl. Sci.* **2021**, *3*, 480. [[CrossRef](#)]
4. Sun, J.; Chai, Y.; Su, C.; Zhu, Z.; Luo, X. BLDC motor speed control system fault diagnosis based on LRGF neural network and adaptive lifting scheme. *Appl. Soft Comput.* **2014**, *14*, 609–622. [[CrossRef](#)]
5. Amin, S.R.; Amador-Jiménez, L.E. Backpropagation Neural Network to estimate pavement performance: Dealing with measurement errors. *Road Mater. Pavement Des.* **2017**, *18*, 1218–1238. [[CrossRef](#)]
6. Shah, A.K.; Adhyaru, D.M. Parameter identification of PWARX models using fuzzy distance weighted least squares method. *Appl. Soft Comput.* **2014**, *25*, 174–183. [[CrossRef](#)]
7. Rojas, C.; Toth, R.; Hjalmarsson, H. Sparse estimation of polynomial and rational dynamical models. *IEEE Trans. Autom. Control* **2014**, *59*, 2962–2977. [[CrossRef](#)]
8. Serradilla, F.; Cañas, N.; Naranjo, J.E. Optimization of the Energy Consumption of Electric Motors through Metaheuristics and PID Controllers. *Electronics* **2020**, *9*, 1842. [[CrossRef](#)]
9. Dey, N. *Advancements in Applied Metaheuristic Computing; Advances in Data Mining and Database Management (2327-1981)*; IGI Global: Hershey, PA, USA, 2017.
10. Huang, C.; Li, Y.; Yao, X. A Survey of Automatic Parameter Tuning Methods for Metaheuristics. *IEEE Trans. Evol. Comput.* **2020**, *24*, 201–216. [[CrossRef](#)]
11. Rodríguez-Abreo, O.; Garcia-Guendulain, J.M.; Hernández-Alvarado, R.; Flores Rangel, A.; Fuentes-Silva, C. Genetic Algorithm-Based Tuning of Backstepping Controller for a Quadrotor-Type Unmanned Aerial Vehicle. *Electronics* **2020**, *9*, 1735. [[CrossRef](#)]
12. Thammarat, C.; Puangdownreong, D. CS-based optimal $\pi\lambda d\mu$ controller design for induction motor speed control system. *Int. J. Electr. Eng. Inform.* **2019**, *11*, 638–661. [[CrossRef](#)]
13. Połap, D. An adaptive genetic algorithm as a supporting mechanism for microscopy image analysis in a cascade of convolution neural networks. *Appl. Soft Comput.* **2020**, *97*, 106824. [[CrossRef](#)]
14. Anh, H.P.H.; Son, N.N.; Van Kien, C.; Ho-Huu, V. Parameter identification using adaptive differential evolution algorithm applied to robust control of uncertain nonlinear systems. *Appl. Soft Comput.* **2018**, *71*, 672–684. [[CrossRef](#)]
15. Wu, Z.; Du, C. The Parameter Identification of PMSM Based on Improved Cuckoo Algorithm. *Neural Process. Lett.* **2019**, *50*, 2701–2715. [[CrossRef](#)]
16. Ogata, K. *Modern Control Engineering; Instrumentation and Controls Series*; Prentice Hall: Englewood Cliffs, NJ, USA, 2010.
17. Gutierrez-Villalobos, J.; Rodriguez-Resendiz, J.; Rivas-Araiza, E.; Mucino, V. A review of parameter estimators and controllers for induction motors based on artificial neural networks. *Neurocomputing* **2013**, *118*, 87–100. [[CrossRef](#)]
18. Torres-Salinas, H.; Rodríguez-Reséndiz, J.; Estévez-Bén, A.A.; Cruz Pérez, M.A.; Sevilla-Camacho, P.Y.; Perez-Soto, G.I. A Hands-On Laboratory for Intelligent Control Courses. *Appl. Sci.* **2020**, *10*, 9070. [[CrossRef](#)]
19. Cruz-Miguel, E.E.; Rodríguez-Reséndiz, J.; García-Martínez, J.R.; Camarillo-Gómez, K.A.; Pérez-Soto, G.I. Field-programmable gate array-based laboratory oriented to control theory courses. *Comput. Appl. Eng. Educ.* **2019**, *27*, 1253–1266. [[CrossRef](#)]
20. Beltran-Carbajal, F.; Tapia-Olvera, R.; Valderrabano-Gonzalez, A.; Yanez-Badillo, H.; Rosas-Caro, J.; Mayo-Maldonado, J. Closed-loop online harmonic vibration estimation in DC electric motor systems. *Appl. Math. Model.* **2021**, *94*, 460–481. [[CrossRef](#)]
21. Sands, T. Nonlinear-Adaptive Mathematical System Identification. *Computation* **2017**, *5*, 47. [[CrossRef](#)]
22. Poon, J.; Jain, P.; Spanos, C.; Panda, S.K.; Sanders, S.R. Fault Prognosis for Power Electronics Systems Using Adaptive Parameter Identification. *IEEE Trans. Ind. Appl.* **2017**, *53*, 2862–2870. [[CrossRef](#)]

23. Rodríguez-Abreo, O.; Hernandez-Paredes, J.M.; Rangel, A.F.; Fuentes-Silva, C.; Velásquez, F.A.C. Parameter Identification of Motors by Cuckoo Search Using Steady-State Relations. *IEEE Access* **2021**, *9*, 72017–72024. [[CrossRef](#)]
24. Ren, L.; Lin, G.; Zhao, Y.; Liao, Z. Smart Collaborative Performance-Induced Parameter Identification Algorithms for Synchronous Reluctance Machine Magnetic Model. *Sustainability* **2021**, *13*, 4379. [[CrossRef](#)]
25. Rahman, M.S.; Lukman, G.F.; Hieu, P.T.; Jeong, K.I.; Ahn, J.W. Optimization and Characteristics Analysis of High Torque Density 12/8 Switched Reluctance Motor Using Metaheuristic Gray Wolf Optimization Algorithm. *Energies* **2021**, *14*, 2013. [[CrossRef](#)]
26. Amiri, M.S.; Ibrahim, M.F.; Ramli, R. Optimal parameter estimation for a DC motor using genetic algorithm. *Int. J. Power Electron. Drive Syst.* **2020**, *11*, 1047–1054. [[CrossRef](#)]
27. Accetta, A.; Cirrincione, M.; Pucci, M.; Sferlazza, A. A Saturation Model of the Synchronous Reluctance Motor and its Identification by Genetic Algorithms. In Proceedings of the 2018 IEEE Energy Conversion Congress and Exposition, ECCE 2018, Portland, OR, USA, 23–27 September 2018; pp. 4460–4465.
28. Kim, J.W.; Kim, S. Parameter identification of induction motors using dynamic encoding algorithm for searches (DEAS). *IEEE Trans. Energy Convers.* **2005**, *20*, 16–24. [[CrossRef](#)]
29. Liu, Z.H.; Li, X.H.; Wu, L.H.; Zhou, S.W.; Liu, K. GPU-Accelerated Parallel Coevolutionary Algorithm for Parameters Identification and Temperature Monitoring in Permanent Magnet Synchronous Machines. *IEEE Trans. Ind. Inform.* **2015**, *11*, 1220–1230. [[CrossRef](#)]
30. Juang, C.F. A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2004**, *34*, 997–1006. [[CrossRef](#)]
31. Qiu, Y.; Kang, Q.; Wang, L.; Wu, Q. A Parameter Optimization Method for dq Axis Inductance Model of Synchronous Reluctance Motors Considering Cross-Coupling Magnetic Saturation. *Diangong Jishu Xuebao/Trans. China Electrotech. Soc.* **2017**, *32*, 85–92.
32. Huynh, D.; Dunnigan, M. Advanced particle swarm optimisation algorithms for parameter estimation of a single-phase induction machine. *Int. J. Model. Identif. Control* **2012**, *15*, 227–240. [[CrossRef](#)]
33. Srivastava, A.; Das, D.K.; Rai, A.; Raj, R. Parameter Estimation of a Permanent Magnet Synchronous Motor using Whale Optimization Algorithm. In Proceedings of the 2018 Recent Advances on Engineering, Technology and Computational Sciences (RAETCS), Allahabad, India, 6–8 February 2018; pp. 1–6. [[CrossRef](#)]
34. Balamurugan, K.; Mahalakshmi, R. Analysis of Brushless DC Motor Using Deep Neural Network and BAT Algorithm. In *Advances in Smart System Technologies*; Suresh, P., Saravanakumar, U., Hussein Al Salameh, M.S., Eds.; Springer: Singapore, 2021; pp. 51–60.
35. Biswas, S.; Saha, D.; De, S.; Cobb, A.D.; Das, S.; Jalaian, B.A. Improving Differential Evolution through Bayesian Hyperparameter Optimization. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; pp. 832–840. [[CrossRef](#)]
36. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
37. Hameed, I.A.; Bye, R.T.; Osen, O.L. Grey wolf optimizer (GWO) for automated offshore crane design. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–6. [[CrossRef](#)]
38. Venkata Rao, R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **2016**, *7*, 19–34. [[CrossRef](#)]
39. Yang, X.S.; Deb, S. Engineering Optimisation by Cuckoo Search. *Int. J. Math. Model. Numer. Optim.* **2010**, *1*, 330–343. [[CrossRef](#)]
40. Panda, M.; Das, B. Grey Wolf Optimizer and Its Applications: A Survey. In Proceedings of the 3rd International Conference on Microelectronics, Computing and Communication Systems, Ranchi, India, 12–13 May 2018; Nath, V., Mandal, J.K., Eds.; Springer: Singapore, 2019; pp. 179–194.