electronics

Review

Parameters Identification of a Permanent Magnet DC Motor: A Review

Mohamad Farid Fazdi and Po-Wen Hsueh *

Department of Mechatronics Engineering, National Kaohsiung University Science and Technology, No. 1 University Rd., Yanchao District, Kaohsiung City 824, Taiwan; faridfan31@gmail.com
* Correspondence: hpowen@nkust.edu.tw; Tel.: +886-7-6011000 (ext. 32212)

Abstract: Since permanent magnet direct current (DC) motors are mainly used in various industrial automation applications, the demand for electric motors is increasing rapidly. However, in the mass production of electric motors, often, only random inspections are used to check the specifications and performance of electric motors. For manufacturing or engineering application staff to have a more thorough understanding of the characteristics of the motor, it is necessary to conduct a full or quick inspection during the production process to ensure the quality of the electric motor. Based on this, this literature review reveals several methods and algorithms often used to estimate DC motor parameters, given the importance of knowing the parameters of the DC motor and the lack of research on estimating the parameters of the DC motor.

Keywords: DC motor; least squares method; differential evolution; particle swarm optimization methods; cuckoo search; metaheuristic; parameter estimation

Citation: Fazdi, M.F.; Hsueh, P.-W. Parameters Identification of a Permanent Magnet DC Motor: A Review. *Electronics* **2023**, *12*, 2559. https://doi.org/10.3390/electronics12122559

Academic Editors: Liang Yan and Teng Zhou

Received: 3 April 2023
Revised: 23 May 2023
Accepted: 31 May 2023
Published: 6 June 2023

1. Introduction

Permanent magnet direct current (DC) motors (also known as brushed DC motors) have been widely employed in numerous industries and applications, ranging from robotics and automation to automotive systems. The global brushed DC motor market is expected to reach USD 8742.20 million by 2029 at a CAGR of 5.30% during the forecast period, as shown in Figure 1 [1]. The brushed DC motor is an excellent choice for low-torque applications because it can change pace or speed with variable speed control options, including below- and above-rated speed [2]. Examples of brushed DC motors are car robots used for exploration, surveying, or mapping purposes, such as MER-B and MER-1 for exploitation on the planet Mars; robot vacuum cleaners; and toy robots. Brushed motors are often used in industries with high torque requirements at low speeds, such as a printing press, spinning drive, or agitator. Nevertheless, age and accompanying wear have changed the characteristics of DC motors. As a result, adjusting the motor characteristics as the motor matures can improve the DC motor model's accuracy [3].

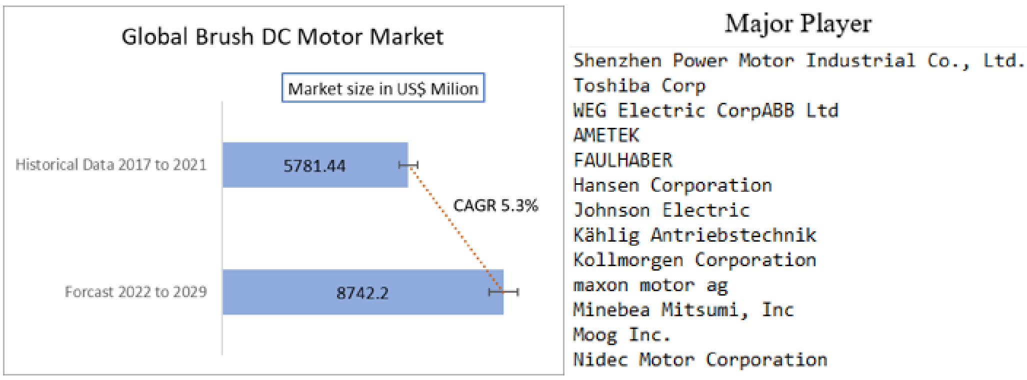

Figure 1. Brushed DC motors market analysis in 2021.

The strategies of DC motor parameter estimation are widely discussed in many previous studies and are employed by various algorithms, such as the curve fitting method [4], the constraint optimization methods [5], the inverse problem methodology with the conjugate gradient and regularization method [6], the regression method [7], the evolutionary process for parameter estimation [8,9], and the power series expansion approach for estimating electrical and mechanical time constants and frictional torque for DC motors [10]. Moreover, the least-squares-based approach is also of great interest in determining the estimation parameters of DC motors [11–17], and the differential evolutions method to determine DC motor parameters has been recently evaluated [18,19].

Additionally, the heuristic methods and metaheuristic algorithm had been proposed and integrated into the DC motor parametric estimation [20,21]. Architecture-intelligent algorithms have been valued in recent years due to their versatility and adaptability to various problems [22]. For instance, the analysis of parametric motors has been explored with metaheuristic algorithms [23,24], and the interest-pollination-based metaheuristic optimization routine with step input excitation has been presented [25]. Further, the more accurate estimation of motor parameters was assessed using different bio-inspired optimization algorithms, such as particle swarm optimization methods [26,27], ant colonies, artificial bee colonies [28], and whale and bat algorithms [29,30]. Moreover, the advantages of combining two metaheuristic algorithms were also explored [31]. Finally, a number of authors, including those of [32], investigated the estimator-based parallel processing employed by this technique online. Author [33] also developed a technique for adaptive learning that reduces system uncertainty and calculates DC motor settings to quicken the training of neural networks live. The motor transfer function technique [34] and the moment method [35] require several experiments to identify DC motor parameters effectively. The method in Ref. [36] requires the input voltage and angular position measurement of a DC servo motor to simultaneously determine viscous friction and motor inertia using an open-loop algebra method. The distribution-based offline parameter identification approach [37] converts a system of linear differential equations into linear algebraic equations. This method uses discrete time data and identifies continuous time model parameters. However, this method may be sensitive to the sampling rate of motor speed and current. The method presented in [38] optimizes a PI controller for digital DC motor control and parameter estimation. The cuckoo search algorithm [39,40] initially takes random numbers to obtain the back EMF constant, armature inductance, and rotor inertia values. Then, the result of the back EMF constant is substituted into the steady-state equation formula to obtain the Armature resistance and Friction coefficient. Because it speeds up processing, this trait is helpful in projects that simulate dynamic models. Numerous studies on the same subject, including [41–43], explicitly employ the cuckoo search method to enhance motor control. The approach is employed in [44] to tune the recurrent neural network hyperparameters automatically.

This article presents the literature review and describes popular methods and algorithms for estimating DC motor parameters. The rest of the work is organized as follows. Section 2 describes DC motor dynamic response and parameter estimation, reviewing the methods and algorithms based on the most popular categorizations mentioned above. Section 3 is the discussion. Section 4 presents the conclusions.

2. DC Motor Dynamic Response and Parameter Estimation

The dynamic response of a DC motor is its ability to modify its speed or torque in reaction to changes in the input signal fast and precisely. The electrical and mechanical properties of a DC motor govern its dynamic reaction. The motor's dynamic reaction is influenced mainly by its electrical time constant, mechanical time constant, and inertia. The time it takes for the electrical circuit of the motor to stabilize at 63.2% of its final value following a step change in the input signal is known as the electrical time constant. The resistance and inductance of the motor's windings determine it. The time required for the mechanical system of the motor to reach 63.2% of its final value following a step change in the output torque is known as the mechanical time constant. The rotor and load inertia of the motor and any friction or damping in the system all contribute to this. The motor's inertia also has an impact on its dynamic responsiveness. A motor with low inertia will react more rapidly, whereas a motor with high inertia will take longer to establish a steady-state speed or torque.

The advantage of the dynamic response is that it allows the motor to respond to changes in input or load quickly and with high precision. This allows the motor to change speed or torque instantly and accurately, making it essential in applications requiring rapid changes, high control, and adaptability to changing operating conditions. The dynamic response also allows for more in-depth measurement and analysis of the motor's response to changes, which is essential in maintaining, repairing, and optimizing DC motor performance. Maintaining system stability and avoiding oscillations or overshoots in the motor's output need a quick and precise dynamic reaction. Additionally, a well-tuned control system can aid in lowering energy usage and enhancing the motor's general effectiveness.

The dynamic relationships between the DC motor equations are briefly discussed in this section. The dynamic reaction of the motor may be divided into three stages: acceleration, steady state, and deceleration, as shown in Figure 2 [45]. The variables that make up the DC motor mathematical model are described in Table 1.

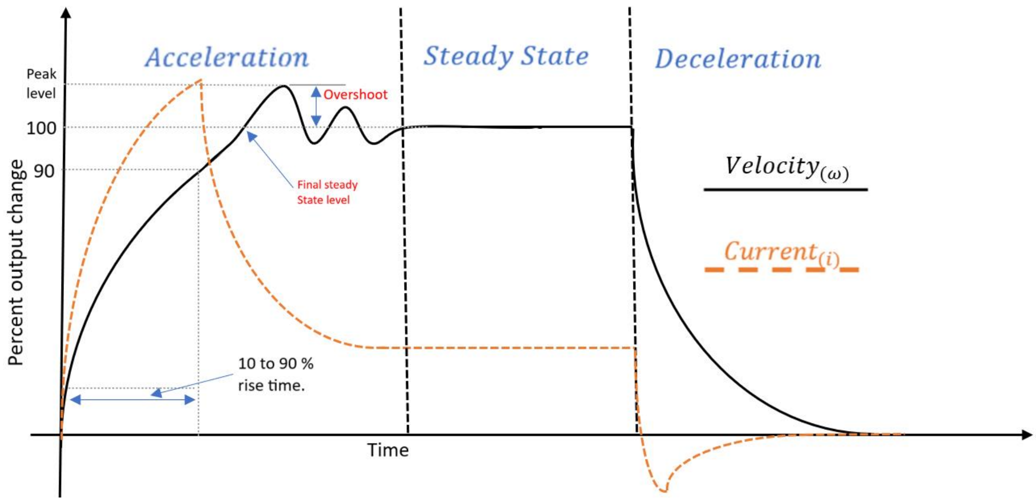

Figure 2. The dynamic responses.

Table 1. Variables used in the DC motor's mathematical model.

Parameter	Description
ν(t)	Applied voltage to the motor (V)
τ(t)	Motor's produced torque (Nm)
ω(t)	Rotor angular velocity (rad/s)
E_a(t)	Voltage in the Back EMF (V)
i	Current consumed by the motor (A)
T_L	Torque at load (Nm)
R	Armature resistance (Ω)
L	Armature inductance (H)
K_t	The mechanical constants' equal values
K_e	Back EMF
K	The mechanical and electrical constants' equal values
B	Coefficient of friction ($\frac{\text{Kgm}^2}{\text{s}^2}$)
J	Moment of inertia (Nm)

The acceleration state of a DC motor refers to its ability to change its speed in response to a change in the input signal. It specifically relates to how quickly the motor speeds up when the input signal moves from a lower value to a higher value. A DC motor's dynamic response and acceleration state are closely connected, and the motor's electrical and mechanical time constants, inertia, and load torque all play a role. As the input signal changes, a motor in a fast acceleration condition can react swiftly and quickly attain its new steady-state speed.

The steady state of a DC motor refers to the point at which the motor's speed and torque have stabilized and are no longer changing in response to changes in the input signal. A DC motor's dynamic response governs how rapidly it may regain a steady state following a change in the input signal. A motor with a slower reaction will take more time to stabilize, whereas one with a better dynamic response can quickly attain a steady-state speed or torque following a change in the input signal. Furthermore, a well-tuned control system can aid in reducing oscillations and overshoot of the motor's output during the transition to a steady state, resulting in a smoother and more stable operation.

The deceleration state of a DC motor refers to its ability to slow down and stop in response to a change in the input signal. It specifically refers to the rate at which the input signal's value shifts from higher to lower, causing the motor's speed to drop. A DC motor's dynamic reaction is directly tied to its deceleration state, determined by its electrical and mechanical time constants, inertia, and load torque. As the input signal changes, a motor with a fast deceleration condition may react swiftly and halt quickly.

2.1. DC Motor Mathematical Model of DC Motor

DC motors are hybrid systems made up of both mechanical and electrical components. Therefore, electrical and mechanical equations allow for observing the motor's dynamic behavior, as seen in Equations (1) and (2).

$$\nu(\mathrm{t}) = iR + L\frac{di}{dt} + E_a(t) \tag{1}$$

$$\tau(\mathrm{t}) = J\frac{d\omega(t)}{dt} + B\omega(t) + T_L \tag{2}$$

The two connection equations discussed in Equations (3) and (4) are connected to the previous set of equations.

$$E_a(t) = K_e\omega(t) \tag{3}$$

$$\tau(\mathrm{t}) = K_t i \tag{4}$$

By substituting Equations (3) and (4) into Equations (1) and (2) and assuming that the motor is without load (T_L = 0), we obtain Equations (5) and (6), and their representation can be observed in Figure 3 [40]. According to the results of Equations (5) and (6), Figure 4 displays the motor model's block diagram.

$$\nu(\mathrm{t}) = iR + L\frac{di}{dt} + K_e\omega(t) \tag{5}$$

$$K_t i = J\frac{d\omega(t)}{dt} + B\omega(t) \tag{6}$$

2.2. Least Squares Method

The least squares method was developed in 1795 by Karl Friedrich Gauss, who also used it for astronomical calculations. He proposed that the most likely values are also the most suitable for the unknowable but desired parameters. According to his definition, "The value of the unknown quantities that has the greatest chance of being correct is the one for which the accuracy-measuring numbers are multiplied by the sum of the squares of the discrepancies between the observed and calculated values". Since then, several technical issues have been solved using the least squares technique. Numerous analyses of its qualities were conducted, numerical methods for various applications were suggested, and the technique was changed to meet the required specifications. The same attempt can be seen in the controlled system parameter estimation area. In recent years, the issue of recognizing a dynamic process has drawn much attention. The numerous methods created for collecting and evaluating input–output data vary from the most basic deterministic procedures to beautiful numerical and statistical strategies based on the findings of optimum estimation theory. As a result, the least squares approach has significantly improved and grown in popularity in parameter estimation.

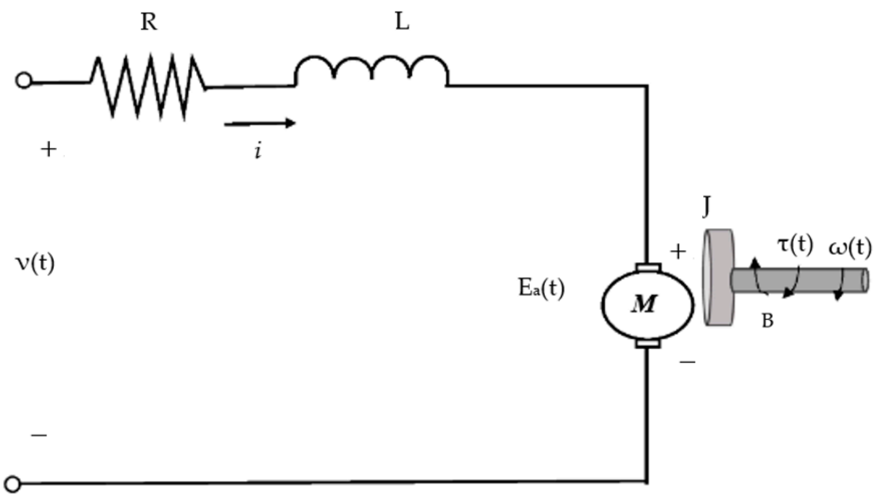

Figure 3. A DC motor's schematic diagram.

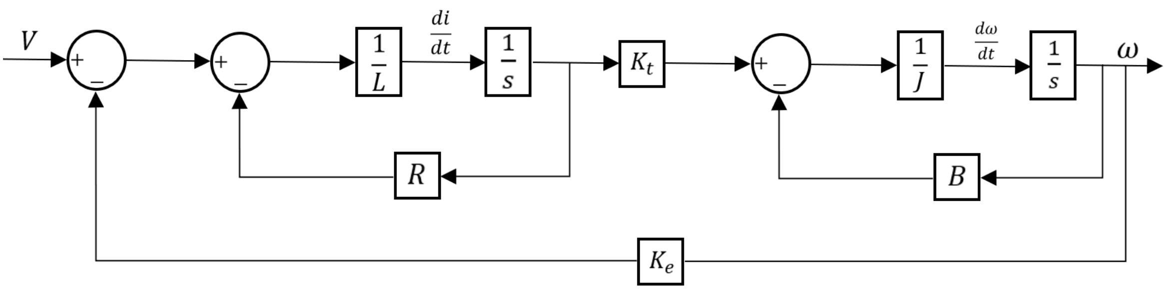

Figure 4. Block diagram of the motor model.

Ref. [11] used the Simulink design optimization method to estimate unknown PMDC motor parameter values using pattern search and nonlinear least squares. The PMDC motor dynamic equation was used to create a PMDC motor Simulink model and calculate the initial estimate of the motor parameters, which were used as initial values in the Simulink design optimization method.

Using Equations (1)–(4), the relationship between K_e and K_t can be determined. The back emf constant of the motor and the torque constant of the motor will be described in Equations (7) and (8).

$$K_e = \frac{Ea(t)}{\omega(t)} \tag{7}$$

$$K_t = \frac{\tau(t)}{i} \tag{8}$$

By assuming that the electromagnetic losses are equal to zero, the connection between these constants may be found. This shows that the mechanical power and the electrical power that the back emf voltage dissipates in the armature are equivalent. Equation (9) defines the relationship between mechanical power (P_m) and electrical power (P_e):

$$P_m = \tau(\mathrm{t})\omega(t) \tag{9}$$

From Equations (7) and (8) above, we obtain Equation (10). The result implies $K_e = K_t$.

$$\frac{Ea(t)}{\omega(t)} = \frac{\tau(\mathrm{t})}{i} \tag{10}$$

The motor parameter values that result from the use of the nonlinear least square algorithm are considered accurate results because the output response from the Simulink modal is almost symmetrical with the actual measured response of the motor, which makes the results obtained using the nonlinear least squares algorithm more accurate, as described in Table 2.

Table 2. The parameters that were computed using NLS and PS methods.

Parameter	Value	Nonlinear Least Squares (NLS)	Pattern Search (PS)
R	1.107	1.0591	0.27302
L	0.120016	0.1	0.0028285
K	0.02497621	0.03728	0.018262
B	0.0007815	0.0011448	0.00078149
J	0.000121	0.0009	0.00012102

Ref. [12] describes estimating dc motor parameters in steady state and deceleration. It is explained that the value of $K_e = K_t$, so there are four unknown parameters in the steady-state interval Armature resistance (R), the mechanical constants equal values (K_t), torque loss (T_0), and Friction coefficient (B). In this article, two sets of voltage inputs, V_{ss1} and V_{ss2}, are used as measurements to improve the accuracy of parameter estimation. Any unknown parameters can be solved using the least squares method described in Equation (11).

$$\begin{bmatrix} i_{ss1} & \omega_{ss1} & 0 & 0 \\ 0 & i_{ss1} & -\omega_{ss1} & -1 \\ i_{ss2} & \omega_{ss2} & 0 & 0 \\ 0 & i_{ss2} & -\omega_{ss2} & -1 \end{bmatrix} \begin{bmatrix} R_0 \\ K_t \\ B \\ T_0 \end{bmatrix} = \begin{bmatrix} V_{ss1} \\ 0 \\ V_{ss2} \\ 0 \end{bmatrix} \tag{11}$$

In the deceleration interval, the current equals zero, and the motor slowly stops. At this time, the motor inertia calculation formula can be explained by the Equation (12).

$$J\frac{d\omega(t)}{dt} = -B\omega - T_0 \tag{12}$$

This article uses the open source Python program to develop a parameter measurement system for permanent magnet DC motors. Among them, the signal measurement system built on the Python platform captures the motor drive's voltage, current, and speed signals and integrates the signal data. Enter MATLAB R2021a to estimate the acceleration parameters and the least squares method and propose a method to judge the acceleration and deceleration intervals. The results of the acceleration method are compared with the measurement using the least squares method. The result is that using the least squares method has more stable and complete results for each DC motor parameter described in Table 3.

Table 3. Parameter table obtained by using acceleration method and least squares method.

Parameter	Value	Acceleration Method	Least Squares Method
R	2.8	None	0.7636
L	0.003	None	0.00076356
K_e	0.2311	0.2159	0.2115
K_t	0.23	0.2159	0.2115
B	0.00019	0.0001181	0.0001157
J	0.00015	0.1465	0.0014
T_0	0.04	0.9442	0.9248

2.3. Metaheuristics Algorithms

Over the past few years, there has been a surge in the utilization of metaheuristic algorithms. The primary reason for this is the ease of implementation and adaptability to diverse problems as expounded in reference [24]. Hence, metaheuristic algorithms based on the population are frequently employed in the academic literature. The categorization of metaheuristic algorithms can be delineated into four principal groups based on the inspiratory source utilized to generate novel solutions. These four categories are evolutionary algorithms (EA), swarm intelligence algorithms, physics-based algorithms, and human-based algorithms, as detailed in reference [46]. It is commonly posited within the scholarly literature that the impetus behind the advancement of Evolutionary Algorithms (EA) can be attributed to the influential framework provided by Darwin's seminal theory of evolution, which emphasizes the notion of the preservation of the "fittest" individuals in the maintenance of species survival. This study employs genetic procedures, specifically, crossover, mutation, and selection, in order to generate improved quality offspring solutions. The swarm intelligence algorithm is rooted in the emulation of the aggregate actions exhibited by animals or insects, more specifically, in their pursuit of locating food sources and securing potential partners for the propagation of their species. Intelligent behavior, characterized by decentralized decision making and informed by the examination of local information and interaction with the search environment, can be exhibited by members of this flock. The tracing operators of algorithms that are based on human activity have the ability to imitate various human behaviors such as thinking, learning, speaking, and teaching. A taxonomy of metaheuristic algorithms based on their source of inspiration and notable examples associated with each branch of MSA is presented in Figure 5. In the present study, a review was conducted with a focus on three specific population-based metaheuristic algorithms, namely, differential evolution, particle swarm optimization, and cuckoo search. The selection of these algorithms was based on their pertinence to examining the implications of utilizing dynamic connections within the context of population-based metaheuristic algorithms.

2.3.1. Differential Evolution (DE)

DE is a population-based direct stochastic optimization technique initially developed by Storn and Price [70]. It is easy, powerful, and simple, qualities that make it appealing for numerical optimization over continuous search space. Compared to evolutionary algorithms, DE takes a more greedy and less stochastic approach to problem resolution. DE combines straightforward arithmetic operators with the traditional crossover, mutation, and selection operators to evolve from a randomly generated beginning population to a final solution. In addition, the way that DE implements the mutation process differs significantly from other evolutionary algorithms. The vector differentials between the members of the current population are used in DE mutation operation to determine the level and direction of perturbation that will be applied to each subject of the mutation operation.

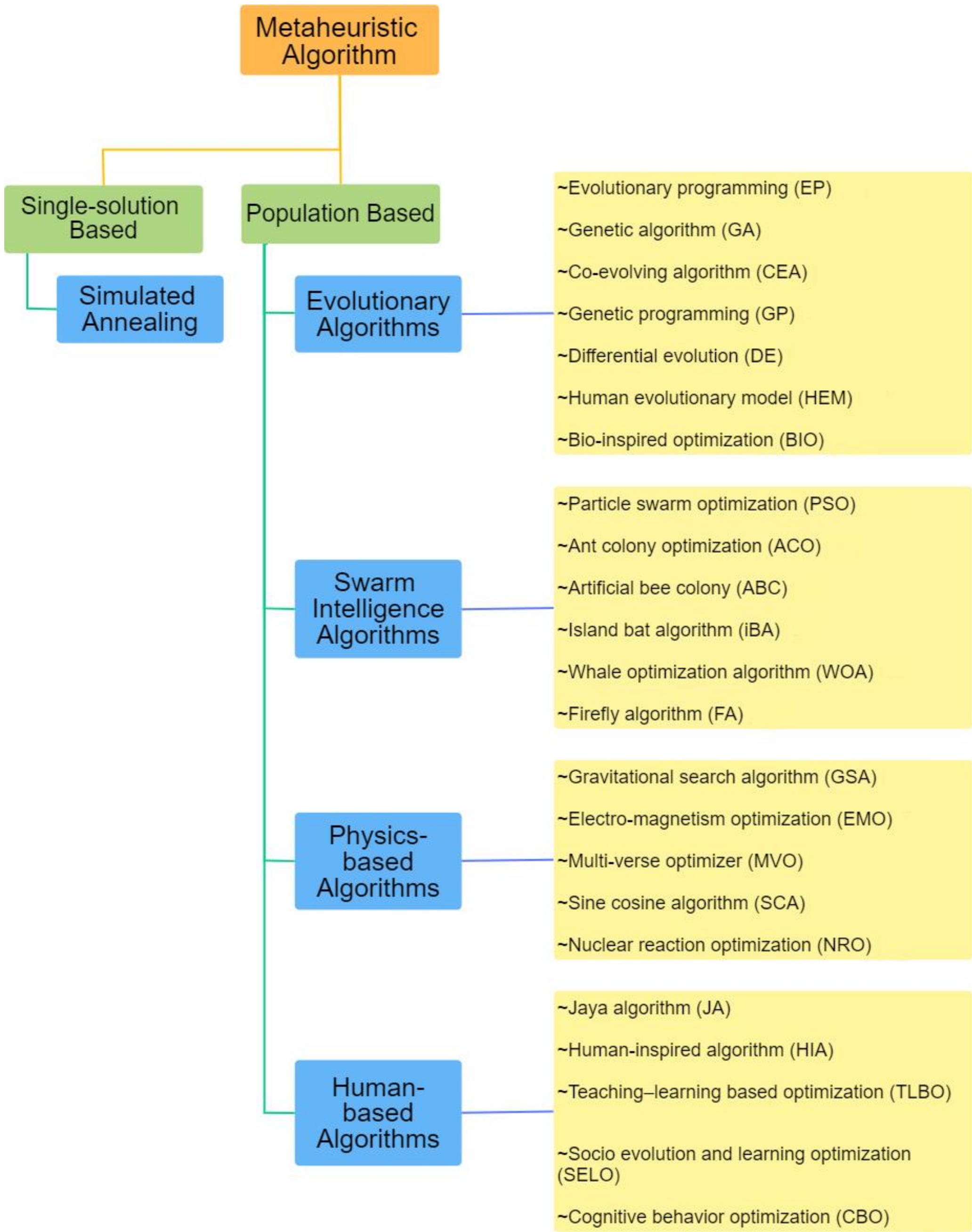

Figure 5. Taxonomy of metaheuristic algorithms [47–69].

Scholarly article [18] enumerates several methodologies employed in optimization, particularly the Genetic Algorithm (GA), Differential Evolutions (DE) with two strategies (DE/rand/1/exp, DE/best/1/bin), Teaching–Learning-Based Optimization (TLBO), and Artificial Bee Colony (ABC). This article presents an analysis of the variation in simulation time of the motor model and memory assistance, utilizing three distinct approaches to minimize the overall computational time. The diagrammatic representation of the DC motor is depicted in Figure 6, along with the operative device that may or may not have a correlation with the motor. The potential association between the item in question and the motor system is uncertain and warrants further investigation. The specification of the drive's parameter requirements is contingent upon various factors. The proposed methodology can be applied to DC motors featuring distinct excitation systems such as parallel excitation or permanent magnet configurations.

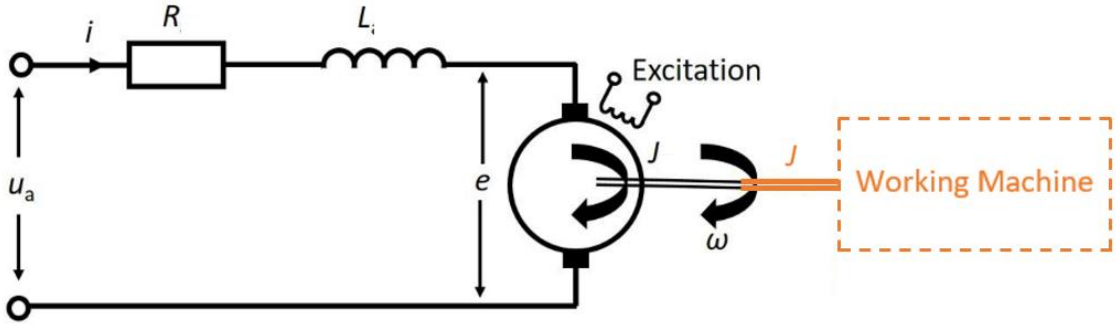

Figure 6. Presentation of the drive-in schematic form, which consists of a working machine and a DC motor with independent excitation.

In this article, the motor stroke response can be simulated using Equations (1) and (2). Equation (2) is converted into Equation (13) which is described below:

$$\tau(t) - T_L = J\frac{d\omega(t)}{dt} \tag{13}$$

Equations (1) and (2) are combined because the induced voltage e relies on the motor speed and the motor torque $\tau(t)$ depends on the motor current. In Equations (14) and (15), e and $\tau(t)$ are written.

$$e = c_m\omega \tag{14}$$

$$\tau(t) = c_m i \tag{15}$$

Since $c_m = k_m\Phi$, where k_m is the DC motor constant and the magnetic flux Φ is considered to be constant in the instance of the particular motor, c_m is taken to be constant (referred to as the motor constant in the continuation). The load T_L given in (2) can be broken down further, as shown in (16).

$$T_L = T_{la} + T_{lb}\omega + T_{lc}\omega^2 \tag{16}$$

The components T_{la} and T_{lb}, respectively, stand in for Coulomb and viscous friction when there is no load from the working machine, while T_{lc} stands in for any air resistance in the fan along the motor axis. In the event that the working machine creates a load, the variables T_{la}, T_{lb}, and T_{lc} indicate friction, air resistance at the fan at the motor axis, and the combined working machine load. Equations (17) and (18) are created by putting the following Equations (14)–(16) into (1) and (2).

$$U_a = iR + L\frac{di}{dt} + c_m\omega \tag{17}$$

$$c_m i - \left(T_{la} + T_{lb}\omega + T_{lc}\omega^2\right) = j\frac{d\omega}{dt} \tag{18}$$

Results show that Differential Evolution (DE)/rand/1/exp is the most powerful method described in Table 4. The results are the same for each run within the scope of the individual test data. TLBO is only slightly worse, but computation time is two times longer. The results obtained using DE/best/1/bin and ABC are also acceptable. Only the results obtained using GA are too bad. Based on that, DE/rand/1/exp can be the best method for the problem discussed. The weakness of this article is that it does not consider the dynamic properties of torsion, mass with torsional stiffness, and damping effects.

Table 4. Mean values of calculated parameters.

Parameter	Method					
	Use Value (Simulation)	**GA**	**DE/Rand/1/Exp**	**DE/Best/1/Bin**	**TLBO**	**ABC**
R	42.5	37.34	42.5	43.58	42.5	42.48
L	8×10^{-2}	5.692×10^{-2}	8×10^{-2}	8.116×10^{-2}	8×10^{-2}	8.002×10^{-2}
c_m	0.4781	0.6782	0.4781	0.4773	0.4781	0.4781
J	2×10^{-5}	8.302×10^{-5}	2×10^{-5}	1.978×10^{-5}	2×10^{-5}	2.004×10^{-5}
T_{la}	1×10^{-2}	1.358×10^{-2}	1×10^{-2}	1.981×10^{-2}	0.998×10^{-2}	0.736×10^{-2}
T_{lb}	3.27×10^{-5}	2.033×10^{-5}	3.270×10^{-5}	3.026×10^{-5}	3.260×10^{-5}	2.577×10^{-5}
T_{lc}	8.55×10^{-8}	4.778×10^{-8}	8.55×10^{-8}	10.98×10^{-8}	8.850×10^{-8}	11.34×10^{-8}

2.3.2. Particle Swarm Optimization (PSO)

The behavior of a swarm of living objects, such as birds and other flies, is mimicked by the bio-inspired metaheuristic search approach known as particle swarm optimization [71]. The standard particle swarm optimization is known to attract suboptimal solutions that are only sometimes the best solution [71,72]. As a result, the search process quickly converges. The particles continue searching within a confined and smaller area of the search space after they are caught at such local optimal solutions. All particles seek locally in certain areas, making it difficult or impossible to escape from such local optimum search ranges. Once the particles have prematurely converged, they may continue to do so until they reach a solution that is very close to the others. As a result, the personal and overall best may be limited to that tiny area of the search field. As a result, the algorithm will not be able to execute the exploration process, which will improve the exploitation process. A typical PSO randomly starts a swarm of particles in the solution search space with a normal distribution and specified upper and lower limit ranges. Each particle has a velocity vector that directs its subsequent travel while being measured against a predetermined goal function to obtain the optimal answer [71,73]. Each particle's velocity is adjusted following its personal best solution (*pBest*) and overall best solution (*gBest*). A particle finds the best solution is *pBest*, while the entire swarm finds the best solution is *gBest* [72,73].

A particle may quickly become stuck in a local optimum in the typical PSO, which might converge to a decent solution but has the issue of premature convergence. This is especially true in a significant problem search space with numerous local optimal solutions. This limits the particles' ability to discover new regions of the search space. The introduction and manipulation inertia weight factor, the inclusion of diversity in the swarm, and the hybridization of an algorithm with another metaheuristic algorithm, such as genetic algorithms, are just a few of the solutions that have been suggested in the literature [74]. These techniques enable the particles to avoid being trapped in a local optimum and improve the global searching process.

Ref. [26] compared the modified PSO method to the original PSO algorithm. PSO adaptations include Chaos-Initialized PSO (CI PSO), Adaptive Inertia Weight Factor PSO (AIWF PSO), and constricted with linearly decreasing inertia weight PSO (CIW PSO). Computer modeling and system parameter estimation were performed using MATLAB/Simulink software. This article presents a DC motor system with the two-mass model (2MM) described in the Figure 6 and will be simplified using Equations (19)–(21).

$$J_1\dot{\omega_1} = T_m - T_s - b(\omega_1 - \omega_2) \tag{19}$$

$$J_2\dot{\omega_2} = -T_s + b(\omega_2 - \omega_1) - T_L \tag{20}$$

$$T_s = K_s(\omega_1 - \omega_2) \tag{21}$$

In Figure 7, the motor is linked to the load inertia through the implementation of a flexible shaft. The variable J_1 signifies the inertia of the motor, while J_2 denotes the inertia of the load. Flexible couplings are frequently employed in order to facilitate the connection of loads, thus permitting a particular degree of shaft misalignment. This particular mechanism is observable in various systems such as wind turbines and robot arms that employ flexible joints, assuming that the flexible coupling exhibits greater stiffness relative to the flexible center shaft. In this scenario, a simplified model of the system may be employed, in which solely two masses are interlinked by a pliant shaft, with the consideration of their inherent damped effect. Equation (22) explains the transfer function of the system, which takes motor torque as input and produces the inertial angular velocity of the motor as an output.

$$G_1(s) = \frac{\omega_1(s)}{T_m} = \frac{J_2 s^2 + K_s + bs}{(J_1 J_2 s^3) + (J_1 + J_2)bs^2 + (J_1 + J_2)K_s s} \tag{22}$$

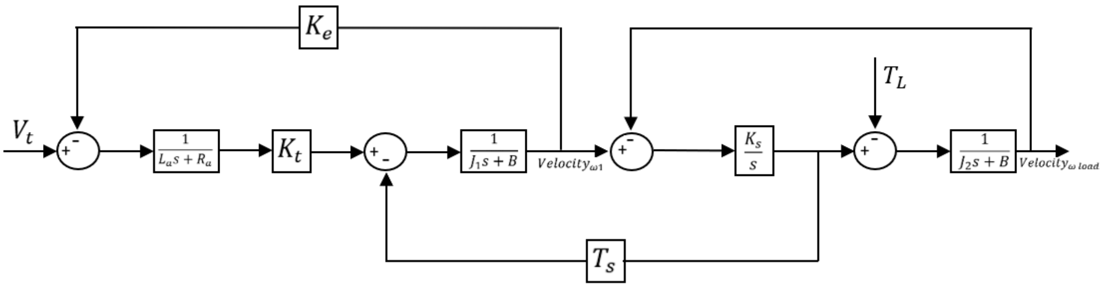

Figure 7. Block diagram of the 2MM.

The Equation (23) represents the task of identifying the system model parameters for the 2MM as an optimization problem utilizing a minimal cost function.

$$minJ = \frac{1}{N}\sum_{i=0}^{n}[ModelOutput(i) - MeasuredOutput(i)]^2 \tag{23}$$

The result is that the Constricted with Linearly Decreasing PSO (CIW PSO) outperforms all algorithms with its ability to get out of the local optimal, converge to values very close to the real ones, and obtain the best cost function, as described in Table 5.

Table 5. Percentage error for the estimate parameter by algorithm.

Parameter	Value	STD PSO	Error	CIW PSO	Error	AIWF PSO	Error	CI PSO	Error
J_1	0.00274	0.0034	25.18%	0.00271	0.89%	0.0029	8.73%	0.00275	0.56%
J_2	0.00256	0.0018	27.05%	0.00258	0.957%	0.00239	9.33%	0.00254	0.61%
K_s	43.4560	41.129	5.35%	43.49	0.093%	43.233	0.45%	41.7134	4.01%
b	0.0550	0.1	150.0%	0.04036	0.924%	0.03852	9.66%	0.04755	18.88%

In reference to Ref. [27], a novel approach for determining the parameters of independently excited DC motors is presented utilizing the Chaotic Initialized Particle Swarm Optimization (CIPSO) algorithm. The process of parameter estimation is transformed into an optimization problem through the utilization of an objective function. The strategy that has been presented holds significant importance in the precise estimation of motor parameters compared to the standard particle swarm optimization technique. This is evidenced by the low mean squared error observed between the actual speeds and the estimated speeds. The DC motor is represented by means of a transfer function in the context of academic writing. In this study, the optimal estimation of five important parameters of a DC motor, namely, the moment of inertia, viscous friction, electromotive force constant, resistance, and inductance, was performed through the application of CIPSO. The generation of the initial population swarms was accomplished through the implementation of a stochastically generated marquee map. The estimated parameters are subjected to a comparison with the true values alongside the parameters obtained through the standard particle swarm optimization technique. The CIPSO method offers the distinct advantages of providing relatively precise parameter estimates with an insignificant mean squared error. In this article, the findings presented in Table 6 indicate the superior performance of the CIPSO algorithm, compared to the SPSO algorithm, in relation to both mean square error (MSE) values and the accuracy of the estimations for the motor parameters. The MSE value attained by CIPSO, which is much lower than the MSE achieved by SPSO, is 1.399×10^{-16}. The CIPSO-estimated parameters and the real motor parameters are quite near to each other.

Table 6. Actual and estimated parameters with MSE.

Parameter	Actual	CIPSO	SPSO
J	0.01	0.0102	0.0110
B	0.1	0.1	0.104
K	0.01	0.0101	0.014
R	1	1.007	1.0901
L	0.5	0.503	0.508
MSE		1.399×10^{-16}	2.080×10^{-12}

2.3.3. Cuckoo Search Optimization (CSO)

The CSO algorithm exhibits a comparatively complex nature, unlike the remaining two algorithms. In contrast to the aforementioned algorithms, similarities to the Genetic Algorithm (GA) are evident in this approach, as with the previous one. Notably, however, convergence is demonstrated in a reduced number of iterations. The CSO method constitutes a bioinspired algorithm that draws inspiration from the peculiar reproductive strategy of cuckoo birds, which involves laying eggs in the nests of other bird species for subsequent care and rearing. In contrast to the remaining two algorithms, the CSO method comprises specific parameters. The determination of certain parameters is a complex subject of inquiry in academia. The body of scholarly works have recommended a proportion of 25% [75].

Ref. [39] introduced a modified cuckoo search algorithm as an effective parametric estimation tool for optimizing motor performance. The charge function is derived from the discrepancy between the actual and desired current and velocity measurements, which arise upon applying a step input voltage to the motor. The present study entailed a comparison of the modified metaheuristic cuckoo search algorithm with the Steiglitz–McBride method and the standard cuckoo search technique.

By utilizing Equations (1)–(4), it is possible to formulate a differential equations system predicated upon the motor's current and speed. In this manner, a load-free motor configuration ($T_L = 0$) is considered, and substitutions are made by utilizing (3) and (4) on (1) and (2), correspondingly. Equations (24) and (25) represent the consequent outcome of this procedure.

$$\frac{dI(t)}{dt} = \frac{v(t) - Ri(t) - K_e\omega(t)}{L} \tag{24}$$

$$\frac{d\omega(t)}{dt} = \frac{K_m I(t) - B\omega(t)}{J} \tag{25}$$

Equation (24) represents the current equation, whilst Equation (25) pertains to the velocity equation. The two variables under consideration, namely, velocity and current, are deemed to be quite feasible in terms of measurement. Nonetheless, the values of the six undetermined variables are contingent upon the type of motor utilized. Conversely, it may be posited that electrical constants exhibit magnitudes that are typically commensurate with those of mechanical constants [2]. Consequently, the issue may be simplified to acquiring five key parameters.

This article exclusively employs the steady-state equation for determining the parameters. A steady state refers to a condition in which the variables characterizing the system remain constant over time. In regard to regulation, the state is defined as stabilized when the magnitude of the output remains within 2% of the ultimate value of the response to the step input. Thus, in order to derive distinct correlations, an analysis of the system's response to the step input should be conducted. By solely considering the steady state, Equations (7) and (8) can be utilized to derive the subsequent association.

$$R = \frac{V_{ss} - K\omega_{ss}}{I_{ss}} \tag{26}$$

$$B = \frac{KI_{ss}}{\omega_{ss}} \tag{27}$$

In the present context, the parameter denoted by K serves as a measure of the magnitude of both the electrical and mechanical constants. I_{ss} denotes the steady-state current, while V_{ss} represents the voltage exerted in steady-state conditions, also known as the step magnitude. Additionally, ω_{ss} signifies the velocity attained during steady-state conditions. It is contended that Equations (26) and (27) are applicable exclusively to the motor reaction to step inputs in a state of equilibrium.

Numerous scholarly investigations have demonstrated that the Integral Absolute Error (IAE) along the trajectory is a fitness function of notable efficacy. The motor is explicated through a set of differential equations. Hence, it is crucial to consider the inaccuracies present in both the current and velocity variables. Prior studies have demonstrated the suitability of employing the Euclidean distance as a form of fitness function when dealing with two vectors. Specifically, the original proposed fitness function, denoted as (28), involves the use of estimated values for I_{ss} and ω_{ss}.

$$fitness = \frac{1}{\sqrt{\sum(I - I_s)^2 + \sum(\omega - \omega_s)^2}} \tag{28}$$

The outcomes demonstrate a noteworthy enhancement over the conventional cuckoo search algorithm and exhibit superior performance compared to the Steiglitz–McBride technique. The efficacy of the modified cuckoo search algorithm in comparison with the standard cuckoo search algorithm has been demonstrated through results tabulated in Table 7. Notably, the modified cuckoo search algorithm adheres to the steady-state relationship, which is frequently invalidated by the application of random values estimated by the standard cuckoo search algorithm. The tabulated data for the performance of the two algorithms is presented in Table 7. The root means square error (RMSE) for velocity and current is computed for each methodology. The algorithm developed by Steiglitz and McBride demonstrates an RMSE value of 2.1542 for velocity and an RMSE value of 0.0296. The modified cuckoo search method exhibits an RMSE of 0.8562 for velocity and an RMSE of 0.0242

Table 7. Comparing nominal parameters, parameters acquired using the original cuckoo method, and parameters obtained using the modified cuckoo algorithm.

Parameter	Nominal Value	Steiglitz–McBride		Modify CSO	
		Value	Error	Value	Error
R	3.1363	3.0031	4.44%	3.0112	3.99%
K	0.048774	0.0477	2.25%	0.049203	0.88%
L	0.01307	0.013556	3.72%	0.01144	12.41%
J	9×10^{-6}	9.0011×10^{-6}	0.01%	8.55×10^{-6}	4.99%
B	1.69×10^{-4}	1.7458×10^{-4}	16.35%	1.705×10^{-4}	0.02%

In contrast, it offers marginally superior outcomes than the Steiglitz–McBride algorithm. The modified cuckoo search's shortcoming, despite the fact that this approach is an enhanced heuristic and metaheuristic method, is that it is only possible to use the modified cuckoo search with DC motors.

Ref. [40] employed a test that illustrates the use of dynamic response relationships as search constraints in a metaheuristic algorithm employed as a parametric estimator, and it was conducted with three algorithms (Gray Wolf Optimizer, Jaya Algorithm, and cuckoo search algorithm). This article compares the three original algorithms with three modified algorithms. Not only are they compared with the original algorithms, but the calculations in this article also use two motors for comparison.

The steady state is employed as a search constraint in article [40], which is similar to article [39] in that it estimates the DC motor parameters solely in the transient state. As a result of this connection, the search algorithm's random parameters are used less frequently, and it requires fewer iterations to provide usable results.

The system is considered to be in a steady state when its output has stabilized and its reaction varies over time by less than 2%. The current and speed remain consistent in this regard. As a result, both the velocity derivative and the current derivative are zero. The similarity in magnitude between the electrical and mechanical constants is a well-known phenomenon. K is therefore utilized exactly in both cases.

When the system's output varies with respect to time, this is known as a transient. Even though it is known that the current has a direct proportionate impact on the velocity derivative, the relationship in this phase is not readily visible. This relationship causes the maximum current value to cause the current derivative's maximum value, as explained in Equation (29).

$$K_{max(i)} = J_{max}\left(\frac{d\omega}{dt}\right) + B\omega\left(t_{max(i)}\right) \quad (29)$$

Since B is already defined as a function of K, removing J only leaves it specified as a function of K, as can be shown from Equation (29); the parameter J can be a function of both B and K.

$$J = \frac{b\omega\left(t_{\max(i)}\right) - K_{max(i)}}{max\left(\frac{d\omega}{dt}\right)} \quad (30)$$

In this article, the fitness/cost function is the Euclidean distance between the sum of the errors in flow and the sum of the errors in speed, explained in Equation (31).

$$fitness = \sqrt{\sum(I - I_s)^2 + \sum(\omega - \omega_s)^2} \quad (31)$$

The absolute percentage error for each test concerning the nominal parameters was calculated to assess the algorithm performance's correctness. Tables 8 and 9 for Motor 1 and Motor 2 describe the findings. The numerical and tabular results demonstrate improvements when comparing the modified algorithms to the originals. The focus of the findings that follow will be the changed algorithms. The observed findings show that the updated method converges more quickly and adjusts the cost curves in the initial rounds. Given that the adjustment speed had been preset, the search ended when the cost function reached a value of less than 0.1. The drawback of this article is that it was only tested in an accelerated state and a steady state, even though measuring a DC motor according to changes in speed has three stages: acceleration, steady state, and deceleration.

The beginning values are those that each algorithm uses to determine the outcomes and the improvement. An exact improvement cannot be measured because the method starts with random numbers. Nonetheless, the findings demonstrate a clear improvement in each algorithm, particularly in the first iterations, where the application of dynamic response relations causes the error to be decreased in fewer iterations. As a result, all algorithms show decreased error and quicker convergence. The pattern persists across different motors, although the GWO_m performs better with Motor 1 than with Motor 2.

Table 8. Motor 1 test's absolute percentage inaccuracy.

Test	Algorithm	R%	K%	L%	J%	B%
Test 1	GWO_o	1.557	0.355	6.721	2.317	0.182
	GWO_m	1.473	0.173	0.591	0.397	0.173
	$Jaya_o$	15.081	3.801	55.18	12.917	0.34
	$Jaya_m$	0.012	0.001	0.21	0.225	0.001
	CSO_o	0.252	0.029	0.068	0.29	0.029
	CSO_m	0.008	0.001	0.041	0.027	0.001
Test 2	GWO_o	0.262	0.161	4.03	0.198	0.077
	GWO_m	0.653	0.076	4.476	0.3	0.076
	$Jaya_o$	5.623	0.396	55.18	20.54	0.899
	$Jaya_m$	0.013	0.001	0.194	0.225	0.001
	CSO_o	0.198	0.023	1.084	0.226	0.023
	CSO_m	0.022	0.002	0.069	0.098	0.002
Test 3	GWO_o	1.445	0.02	3.555	0.508	0.391
	GWO_m	1.355	0.159	1.391	0.383	0.159
	$Jaya_o$	20.539	0.572	55.18	168.091	12.048
	$Jaya_m$	0.014	0.002	0.3	0.225	0.002
	CSO_o	3.137	0.376	9.907	3.56	0.376
	CSO_m	0.004	0.001	0.212	0.215	0.001
Test 4	GWO_o	13.269	2.785	53.598	66.302	0.375
	GWO_m	1.233	0.141	0.437	0.082	0.141
	$Jaya_o$	63.585	21.087	57.78	55.245	35.513
	$Jaya_m$	0.151	0.018	0.191	0.241	0.018
	CSO_o	0.369	0.043	5.891	3.551	0.043
	CSO_m	0.241	0.028	0.269	0.251	0.028

Table 9. Motor 2 test's absolute percentage inaccuracy.

Test	Algorithm	R%	K%	L%	J%	B%
Test 1	GWO_o	29.525	4.733	61.008	70.75	5.774
	GWO_m	0.244	0.054	5.024	0.755	0.054
	$Jaya_o$	9.779	2.48	3.253	4.529	5.228
	$Jaya_m$	0.087	0.02	0.667	0.829	0.02
	CSO_o	0.203	0.045	0.086	0.264	0.045
	CSO_m	0.025	0.005	0.024	0.023	0.005
Test 2	GWO_o	5.792	1.448	19.533	6.69	1.554
	GWO_m	2.773	0.64	18.245	1.454	0.64
	$Jaya_o$	9.55	1.637	28.856	17.807	0.466
	$Jaya_m$	0.06	0.013	0.703	0.823	0.013
	CSO_o	0.307	0.069	0.632	0.459	0.069
	CSO_m	0.053	0.012	0.025	0.013	0.012
Test 3	GWO_o	11.298	1.53	67.539	192.484	2.66
	GWO_m	0.936	0.206	8.692	0.601	0.206
	$Jaya_o$	6.186	0.08	65.769	162.739	6.26
	$Jaya_m$	0.121	0.027	0.637	0.836	0.027
	CSO_o	0.291	0.065	10.56	5.329	0.065
	CSO_m	0.028	0.006	0.551	0.102	0.006
Test 4	GWO_o	4.496	0.966	12.593	12.692	0.737
	GWO_m	2.004	0.436	4.146	0.37	0.436
	$Jaya_o$	35.942	14.237	161.4	28.31	6.117
	$Jaya_m$	0	0	0.665	0.809	0
	CSO_o	4.296	0.909	13.716	11.149	0.909
	CSO_m	0.113	0.025	0.035	0.556	0.025

3. Discussion

This section discusses in detail the comparison of each method and algorithm for estimating DC motor parameters and comparing the results of each study.

3.1. Least Squares Method

The six measurements above can be divided into two parts: measurements using simulations in MATLAB Simulink and direct measurements using real DC motors, which are described in Table 10.

Table 10. Motor measurement with simulation or actual motor.

Parameter	Simulink	Actual Motor
1. Parameter Estimation of a Permanent Magnets DC Motor [11]	✓	
2. Improvement and Implementation of Model Identification for Permanent Magnet DC Motors [12]		✓

Ref. [11] has the drawback of not having comparative data between the value manufacturer and parameter estimates, which are fully explained in Ref. [12]. With this value comparison, the reader can find out how effectively this method is used to solve the problem of estimating DC motor parameters. Article [12] also explains in detail how to obtain DC motor parameter estimates.

3.2. Differential Evolution

Article [18] examines measurements for estimating DC motor parameters using differential evolution, using MATLAB Simulink to create DC motor simulations. Both have the same advantages by comparing parameter estimation results with several algorithms. The advantages of differential evolution in Ref. [18] obtain the best results but have a weakness. Namely, the calculation time is twice as long as the other algorithms. Another drawback in Ref. [18] is that it does not consider dynamic torsional properties, mass with torsional stiffness, or damping effects, which could be the topic of the following journal. The strength of this article is that it considers the constant, linear, and quadratic load sections, although it does not cover all the physical phenomena during motor starting.

Future research will likely use different evolution to estimate DC motor parameters due to the many articles cited each year, described in Figure 8 [76]. The application of this algorithm hybridization is also widely used, among others, in Table 11.

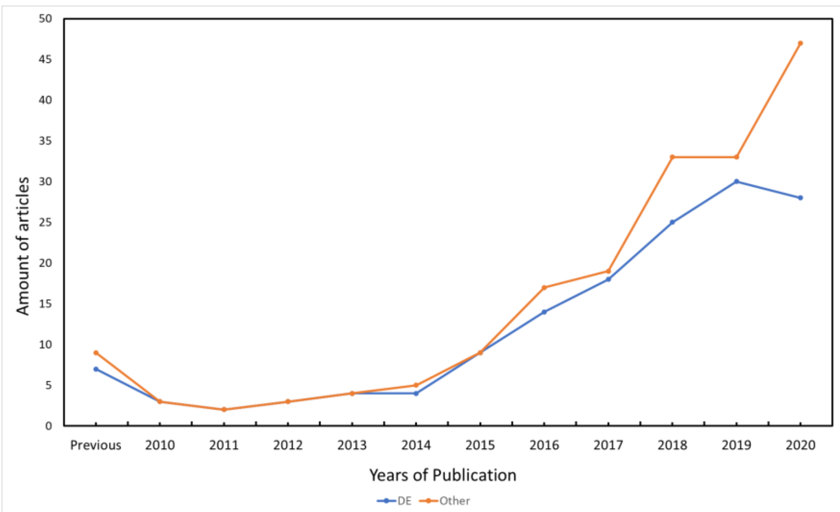

Figure 8. Number of all cited articles.

Table 11. Hybridizations of DE algorithm with other AI algorithms.

Different Evolution with	Years
PSO (Particle swarm optimization)	2016 [77], 2018 [78], 2019 [79], 2020 [80,81], 2021 [82]
CS (Cuckoo Search)	2016 [83], 2018 [84], 2019 [85,86], 2020 [87]
ABC (Artificial bee colony)	2016 [88], 2017 [89], 2018 [90], 2019 [91], 2020 [92]
GA (Genetic algorithm)	2016 [93], 2017 [94], 2018 [95]
ACO (Ant colony optimization)	2017 [96], 2018 [97], 2019 [98]

3.3. Particle Swarm Optimization

The two articles [26,27] regarding measurements for estimating DC motor parameters using particle swarm optimization use MATLAB Simulink to identify DC motor parameters. The two articles explain the same flow and how to obtain the estimated DC motor parameters, depicted in Figure 9.

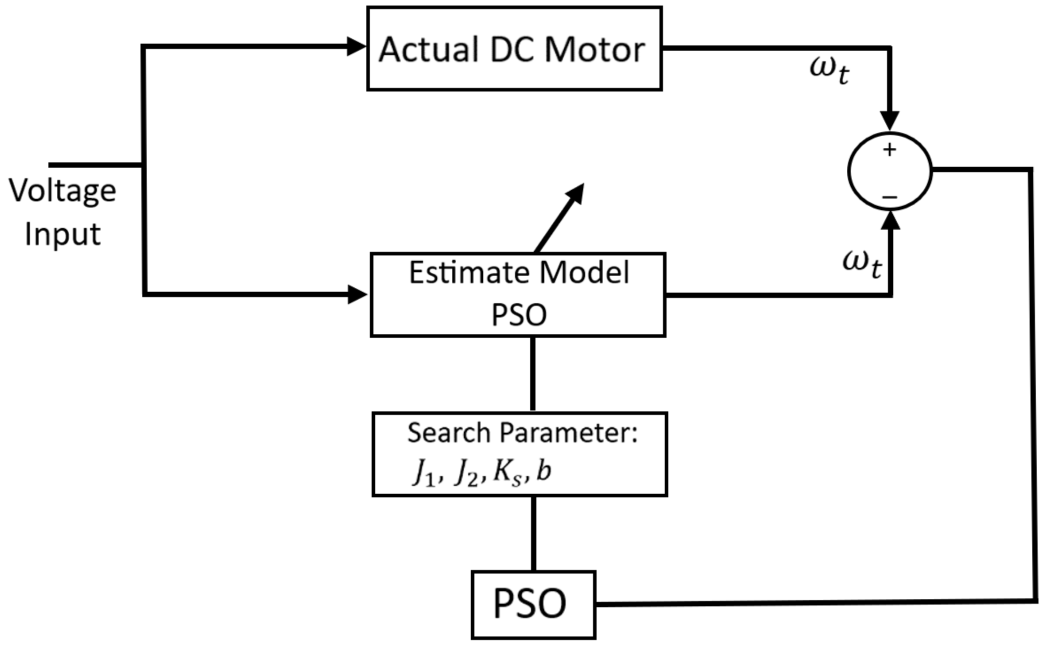

Figure 9. The process of estimating model system parameters with particle swarm optimization.

Even though it uses the same flow, article [27] obtains slightly better results than article [26]. This difference can be due to the two articles using different DC motors. In article [26], the authors modified particle swarm optimization (PSO), whereas, in an article [27], the authors used chaotic initialized particle swarm optimization (CIPSO) described as a finite nonlinear system with a deterministic dynamic behavior that has ergodic and stochastic properties [99].

Considering that there is little research on the estimation of DC motor parameters using particle swarm optimization here, the authors suggest that in the future, we can combine chaotic initialized particle swarm optimization (CIPSO) with constricted particle swarm optimization. Because chaotic initialized particle swarm optimization can improve the global exploration process for the parameter search space by allowing the particle to fluctuate above the search space, using inertia weight methods will improve the exploration process.

3.4. Cuckoo Search Optimization

From a review of two articles [39,40], both use Simulink for DC motor simulation. The two articles have their respective advantages in terms of results. The advantage of article [39] is that there is a more detailed measurement of the DC motor parameters estimation results and it compares the RMSE of the current and velocity between modifications of CSO, standard CSO, and Steiglitz–McBride. The strength of article [40] is the modification of three algorithms (Gray Wolf Optimizer, Jaya Algorithm, and cuckoo search algorithm) to prove that the modifications made to the cuckoo search algorithm in article [39] can be extended to various metaheuristics. The modified algorithm achieved better results than the three standard algorithms above to obtain estimates of DC motor parameters. The drawback of the two articles above is that the modified algorithm can only determine the estimated DC motor parameters and cannot determine other types of motors. Measurements are only carried out in the steady-state position.

In the future, it is hoped that many will apply the cuckoo search optimization algorithm considering that there are still many variants of cuckoo search optimization that have not been used to estimate the DC motor parameters depicted in Figure 10 [100], and we can combine cuckoo search optimization with other algorithms such as hybrid cuckoo search optimization with a genetic algorithm.

3.5. Quantitative Comparison of Computational Costs

The computational cost of parameter estimation in DC motors is affected by several factors, including the dimension of the parameter space to be estimated, as a more significant number of parameters increases the search space and generally leads to an increase in computational cost for the optimization algorithm; the evaluation time required to calculate the fitness or objective function, which involves simulating motor behavior and can be time consuming; and the convergence speed of the optimization algorithm, as faster convergence reduces the number of iterations required and, consequently, lowers the overall computational cost.

3.5.1. Least Squares Method

The computational cost of the least squares method depends on the optimization algorithm used, such as Gauss–Newton or Levenberg–Marquardt. The least squares method is often efficient for parameter estimation in smaller-sized problems. The computational cost is relatively low compared to the metaheuristic optimization algorithms.

3.5.2. Differential Evolution (DE)

DE generally has moderate to high computational costs. It utilizes a complex search strategy involving trial vector generation and crossover operations. DE may require more function evaluations and iterations to converge, especially for higher-dimensional parameter spaces. The computational cost increases with the dimension of the problem and the number of iterations required.

3.5.3. Particle Swarm Optimization (PSO)

PSO usually has a moderate computational cost. It relies on the movement of particles through the search space to find optimal parameter values. PSO can converge relatively quickly, but the computational cost increases with larger swarm sizes and longer simulation times to evaluate the fitness function. The number of iterations required affects the computational cost.

3.5.4. Cuckoo Search Optimization (CSO)

CS often has a lower computational cost compared to DE and PSO. It uses a random search strategy which can efficiently explore the parameter space. However, CS may require more iterations to converge than PSO and DE, which can impact the overall computational cost. The evaluation time of the fitness function remains a significant factor.

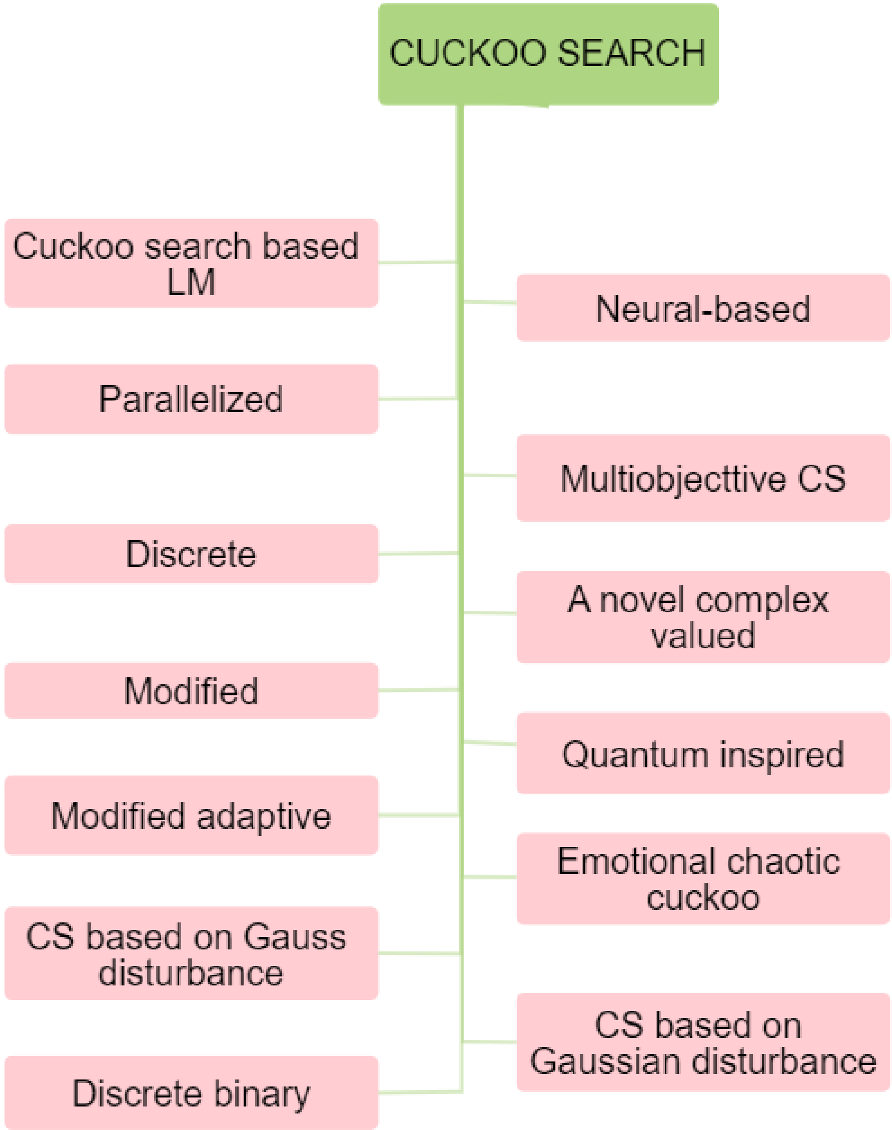

Figure 10. Variant of cuckoo search.

4. Conclusions

This paper reviews several methods and algorithms used to estimate the excess parameters of DC motors from the least squares method. This method has more applications for estimating DC motor parameters than metaheuristic algorithms due to the ease of calculating using the least squares method. Judging from the two articles that use this method, article [12] has advantages compared to similar articles using the least squares method because article [12] uses actual motor measurements and reasonably explains how to obtain each parameter estimate.

The advantage of differential evolution is that there are many studies on this algorithm, as seen from the fact that its use continues to increase yearly. There are also many studies on this algorithm in combination with several other algorithms to achieve optimal results.

The advantages of particle swarm optimization are the speed in estimating compared to other algorithms and the ease of its relation with low limitations on the environment and the objective function. There is still enough space for research on estimating DC motor parameters.

The advantage of using cuckoo search is its simplicity, and in the future, more variants should be used, as described in Figure 10, to estimate DC motor parameters.

The advantage of using complex metaheuristic methods is that they are necessary for handling large search spaces, global optimization, non-differentiable or discrete functions, flexibility, limited problem-specific information, and balancing exploration–exploitation trade-offs in complex optimization problems, surpassing the limitations of simple methods. In the future, this metaheuristic algorithm will be even more used and developed, considering that metaheuristic algorithms can be hybridized with various methods and algorithms.

The three metaheuristic algorithms above also show good results because they are close to the parameter values of the DC motor manufacturers. Hybridizing these three metaheuristic algorithms to estimate DC motor parameters can optimize the results and existing iterations, considering that each algorithm has advantages and disadvantages.

Author Contributions: Conceptualization, M.F.F. and P.-W.H.; methodology, M.F.F.; validation, M.F.F. and P.-W.H.; formal analysis, M.F.F.; investigation, M.F.F.; resources, P.-W.H.; data curation, M.F.F.; writing—original draft preparation, M.F.F.; writing—review and editing, M.F.F. and P.-W.H.; project administration, P.-W.H.; funding acquisition, P.-W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Council of Taiwan, under grant number MOST 111-2221-E-992-084.

Data Availability Statement: Data sharing not applicable.

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

References

1. Maximize Market Research. Global Brush DC Motors Market—Industry Analysis and Forecast (2020–2027). 2021. Available online: https://www.maximizemarketresearch.com/market-report/global-brush-dc-motors-market/71704/ (accessed on 22 May 2023).
2. Batool, A.; Ain, N.U.; Amin, A.A.; Adnan, M.; Shahbaz, M.H. A comparative study of DC servo motor parameter estimation using various techniques. *Automatika* **2022**, *63*, 303–312. [CrossRef]
3. Usman, H.M.; Mukhopadhyay, S.; Rehman, H. Permanent magnet DC motor parameters estimation via universal adaptive stabilization. *Control Eng. Pract.* **2019**, *90*, 50–62. [CrossRef]
4. Wu, W. DC Motor Parameters Identification Using Speed Step Response. *Model. Simul. Eng.* **2012**, *2012*, 189757. [CrossRef]
5. Adewusi, S. Modeling and Parameters Identification of a DC Motor Using Constraint Optimization Technique. *IOSR J. Mech. Civ. Eng.* **2016**, *13*, 46–56.
6. Hadef, M.; Mekideche, M.R. Parameter identification of a separately excited dc motor via inverse problem methodology. *Turk. J. Electr. Eng. Comp. Sci.* **2009**, *17*, 99–106. [CrossRef]
7. Shanmuga, N.B.; Mythile, A.; Pavithra, S.; Nivetha, N. Parameter Identification of a DC Motor. *Int. J. Sci. Technol. Res.* **2020**, *9*, 2.
8. 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]
9. Dupuis, A.; Ghribi, M.; Kaddouri, A. Multiobjective genetic estimation of DC motor parameters and load torque. In Proceedings of the 2004 IEEE International Conference on Industrial Technology, 2004. IEEE ICIT '04., Hammamet, Tunisia, 8–10 December 2004; Volume 3, pp. 1511–1514. [CrossRef]
10. Wu, W. DC motor identification using speed step responses. In Proceedings of the IEEE American Control Conference (ACC), Baltimore, MD, USA, 30 June–2 July 2010; pp. 1937–1941.
11. Awoda, M.L.; Ramzy, S.A. Parameter Estimation of a Permanent Magnets DC motor. *Iraqi J. Electr. Electron. Eng.* **2019**, *15*, 28–36. [CrossRef]
12. Lin, Y.-H. Improvement and Implementation of Model Identification for Permanent Magnet DC Motors. Master's Thesis, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan, 2020.
13. Arshad, S.; Qamar, S.; Jabbar, T.; Malik, A. Parameter Estimation of a DC Motor Using Ordinary Least Squares and Recursive Least Squares Algorithms. In Proceedings of the 8th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 12–13 December 2010. [CrossRef]
14. Li, M.; Ma, Y. Parameter Identification of DC Motor based on Compound Least Square Method. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 1107–1111. [CrossRef]
15. Ivanov, D.V.; Sandler, I.L.; Chertykovtseva, N.V.; Mitroshin, D.I.; Ivanova, O.S.; Kormakov, A.A. Identification of Parameters of DC Motor of Independent Excitation by Noisy Data. In Proceedings of the 2021 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russia, 10–12 November 2021; pp. 194–198. [CrossRef]

16. Gao, D.; Wu, S.; Yu, J.; Wang, M.; Wang, Y. Parameter identification of DC motor model based on improved dynamic forgetting factor recursive least squares method. In Proceedings of the 2022 IEEE 8th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), Melaka, Malaysia, 26–28 September 2022; pp. 282–286. [CrossRef]
17. Hsueh, P.-W.; Tseng, C.-M.; Lin, Y.-H. Improvement of Model Identification for Permanent Magnet DC Servo Motors. In Proceedings of the 18th International Conference on Automation Technology (Automation 2021), Online, 6–8 July 2021.
18. Jesenik, M.; Hamler, A.; Trbušić, M.; Trlep, M. The Use of Evolutionary Methods for the Determination of a DC Motor and Drive Parameters Based on the Current and Angular Speed Response. *Mathematics* **2020**, *8*, 1269. [CrossRef]
19. Rodríguez-Molina, A.; Villarreal-Cervantes, M.G.; Aldape-Pérez, M. Optimal Adaptive Control of a DC Motor Using Differential Evolution Variants. In Proceedings of the 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Honolulu, HI, USA, 31 July–4 August 2017; pp. 283–288. [CrossRef]
20. Sands, T. Nonlinear-Adaptive Mathematical System Identification. *Computation* **2017**, *5*, 47. [CrossRef]
21. 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]
22. Serradilla, F.; Canas, N.; Naranjo, J.E. Optimization of the Energy Consumption of Electric Motors through Metaheuristics and PID Controllers. *Electronics* **2020**, *9*, 1842. [CrossRef]
23. 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]
24. 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]
25. Puangdownreong, D.; Hlungnamtip, S.; Thammarat, C.; Nawikavatan, A. Application of flower pollination algorithm to parameter identification of DC motor model. In Proceedings of the IEEE International Electrical Engineering Congress (IEECON), Pattaya, Thailand, 8–10 March 2017; pp. 1–4.
26. Hafez, I.; Dhaouadi, R. Parameter Identification of DC Motor Drive Systems using Particle Swarm Optimization. In Proceedings of the 2021 International Conference on Engineering and Emerging Technologies (ICEET), Istanbul, Turkey, 27–28 October 2021; pp. 1–6. [CrossRef]
27. Mughal, M.A.; Khan, M.; Shah, A.A.; Almani, A.A. DC Motor Parameter Estimation Using Chaotic Initialized Particle Swarm Optimization. In Proceedings of the International Conference on Electromechanical Control Technology and Transportation (ICECTT), Chongqing, China, 19–21 January 2018. [CrossRef]
28. Sankardoss, V.; Geethanjali, P. PMDC motor parameter estimation using bio-inspired optimization algorithms. *IEEE Access* **2017**, *5*, 11244–11254. [CrossRef]
29. 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.
30. 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.
31. Tabish, M.; Kalam, A.; Zayegh, A. Robot DC Servo Motor Parameters Estimation in a closed loop Using BAT Optimisation Algorithm. In Proceedings of the 2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Swat, Pakistan, 24–25 July 2019; pp. 1–5. [CrossRef]
32. 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]
33. Rubaai, A.; Kotaru, R. Online identification and control of a DC motor using learning adaptation of neural networks. *IEEE Trans. Ind. Appl.* **2000**, *36*, 935–942. [CrossRef]
34. Da Silva, M.F.; Bastos, F.F.; Da Silva Casillo, D.S.; Casillo, L.A. Parameters identification and analysis of brushless direct current motors. *IEEE Lat. Am. Trans.* **2016**, *14*, 3138–3143. [CrossRef]
35. Hadef, M.; Bourouina, A.; Mekideche, M. Parameter identification of a DC motor via moments method. *Iran. J. Electr. Comput. Eng.* **2008**, *7*, 159–163.
36. Mamani, G.; Becedas, J.; Feliu-Batlle, V.; Sira-Ramirez, H. Open-loop algebraic identification method for a DC motor. In Proceedings of the IEEE European Control Conference (ECC), Kos, Greece, 2–5 July 2007; pp. 3430–3436.
37. Sendrescu, D. Parameter identification of a DC motor via distribution-based approach. In Proceedings of the 17th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, 27–30 August 2012; pp. 17–22.
38. Galijašević, S.; Mašić, Š.; Smaka, S.; Akšamović, A.; Balić, D. Parameter identification and digital control of speed of a permanent magnet DC motor. In Proceedings of the 23rd IEEE International Symposium on Information, Communication and Automation Technologies (ICAT), Sarajevo, Bosnia and Herzegovina, 27–29 October 2011.
39. 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]
40. 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. [CrossRef] [PubMed]

41. Narula, N.S.; Bhatnagar, A.; Kumar, V.; Rana, K.P.S. A teacher learning based optimization approach to tune backstepping controller for a single-link flexible-joint manipulator. In Proceedings of the 2nd International Conference on Communication, Control and Intelligent Systems (CCIS), Mathura, India, 18–20 November 2016; pp. 159–164.
42. Singh, K.S.M.J.; Elamvazuthi, I.; Shaari, K.Z.K.; Lima, F.V. PID tuning control strategy using cuckoo search algorithm. In Proceedings of the 2015 IEEE Student Conference on Research and Development (SCOReD), Kuala Lumpur, Malaysia, 13–14 December 2015; pp. 129–133.
43. Singh, K.S.M.J.; Elamvazuthi, I.; Shaari, K.Z.K.; Perumal, N. Development of PID controller tuning tool based on cuckoo search algorithms. In Proceedings of the IEEE 3rd International Symposium in Robotics and Manufacturing Automation (ROMA), Kuala Lumpur, Malaysia, 28 August–1 September 2017; pp. 1–5.
44. Srivastava, D.; Singh, Y.; Sahoo, A. Auto tuning of RNN hyperparameters using cuckoo search algorithm. In Proceedings of the 12th International Conference on Contemporary Computing (IC), Noida, India, 8–10 August 2019; pp. 1–5.
45. Wu, C.Y.; Tasi, M.C.; Mao, S.H. Characteristics measurement of direct-drive brushless DC motors without using dynamometers. In Proceedings of the 2009 International Conference on Electrical Machines and Systems, Tokyo, Japan, 15–18 November 2009; pp. 1–6. [CrossRef]
46. Jamal, A.; Rahman, M.T.; Al-Ahmadi, H.M.; Ullah, I.; Zahid, M. Intelligent intersection control for delay optimization: Using meta-heuristic search algorithms. *Sustainability* **2020**, *12*, 1896. [CrossRef]
47. Schaffer, J.D.; Roebber, P.J.; Evans, C. Development and evaluation of an evolutionary programming-based tropical cyclone intensity model. *Mon. Weather. Rev.* **2020**, *148*, 1951–1970. [CrossRef]
48. Zahir, A.; Alhady, S.S.N.; Wahab, A.; Ahmad, M. Objective functions modification of GA optimized PID controller for brushed DC motor. *Int. J. Electr. Comput. Eng.* **2020**, *10*, 2426. [CrossRef]
49. Özcan, E.; Drake, J.H.; Altıntaş, C.; Asta, S. A self-adaptive multimeme memetic algorithm co-evolving utility scores to control genetic operators and their parameter settings. *Appl. Soft Comput.* **2016**, *49*, 81–93. [CrossRef]
50. Mehr, A.D.; Nourani, V.; Kahya, E.; Hrnjica, B.; Sattar, A.M.; Yaseen, Z.M. Genetic programming in water resources engineering: A state-of-the-art review. *J. Hydrol.* **2018**, *566*, 643–667. [CrossRef]
51. Li, S.; Gu, Q.; Gong, W.; Ning, B. An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models. *Energy Convers. Manag.* **2020**, *205*, 112443. [CrossRef]
52. Montiel, O.; Castillo, O.; Melin, P.; Díaz, A.R.; Sepúlveda, R. Human evolutionary model: A new approach to optimization. *Inf. Sci.* **2007**, *177*, 2075–2098. [CrossRef]
53. Rahman, I.; Mohamad-Saleh, J. Hybrid bio-Inspired computational intelligence techniques for solving power system optimization problems: A comprehensive survey. *Appl. Soft Comput.* **2018**, *69*, 72–130. [CrossRef]
54. Yu, H.; Tan, Y.; Zeng, J.; Sun, C.; Jin, Y. Surrogate-assisted hierarchical particle swarm optimization. *Inf. Sci.* **2018**, *454*, 59–72. [CrossRef]
55. Uthayakumar, J.; Metawa, N.; Shankar, K.; Lakshmanaprabu, S. Financial crisis prediction model using antcolony optimization. *Int. J. Inf. Manag.* **2020**, *50*, 538–556. [CrossRef]
56. Chen, J.; Yu, W.; Tian, J.; Chen, L.; Zhou, Z. Image contrast enhancement using an artificial bee colony algorithm. *Swarm Evol. Comput.* **2018**, *38*, 287–294. [CrossRef]
57. Al-Betar, M.A.; Awadallah, M.A. Island bat algorithm for optimization. *Expert Syst. Appl.* **2018**, *107*, 126–145. [CrossRef]
58. Chen, H.; Yang, C.; Heidari, A.A.; Zhao, X. An efficient double adaptive random spare reinforced whale optimization algorithm. *Expert Syst. Appl.* **2019**, *154*, 113018. [CrossRef]
59. Li, M.; Zhao, H.; Weng, X.; Han, T. Cognitive behavior optimization algorithm for solving optimization problems. *Appl. Soft Comput.* **2016**, *39*, 199–222. [CrossRef]
60. Feng, Z.-K.; Liu, S.; Niu, W.-J.; Li, S.-S.; Wu, H.-J.; Wang, J.-Y. Ecological operation of cascade hydropower reservoirs by elite-guide gravitational search algorithm with Levy flight local search and mutation. *J. Hydrol.* **2020**, *581*, 124425. [CrossRef]
61. Xing, B.; Gao, W.-J. *Electromagnetism-like Mechanism Algorithm, Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 347–354. [CrossRef]
62. Taser, A.E.; Guney, K.; Kurt, E. Circular antenna array synthesis using multiverse optimizer. *Int. J. Antennas Propag.* **2020**, *2020*, 3149826. [CrossRef]
63. Neggaz, N.; Ewees, A.A.; Elaziz, M.A.; Mafarja, M. Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst. Appl.* **2019**, *145*, 113103. [CrossRef]
64. Chung, Y.; Char, I.; Neiswanger, W.; Kandasamy, K.; Nelson, A.O.; Boyer, M.D.; Kolemen, E.; Schneider, J. Offline contextual bayesian optimization for nuclear fusion. *arXiv* **2020**, arXiv:2001.01793.
65. Wang, H.; Wang, W.; Zhou, X.; Sun, H.; Zhao, J.; Yu, X.; Cui, Z. Firefly algorithm with neighborhood attraction. *Inf. Sci.* **2017**, *382*, 374–387. [CrossRef]
66. Luu, T.V.; Nguyen, N.S. Parameters extraction of solar cells using modified JAYA algorithm. *Optik* **2019**, *203*, 164034. [CrossRef]
67. Wagan, A.I.; Shaikh, M.M. A new metaheuristic optimizationalgorithm inspired by human dynasties with an application to the wind turbine micrositing problem. *Appl. Soft Comput.* **2020**, *90*, 106176. [CrossRef]
68. Shukla, A.K.; Singh, P.; Vardhan, M. An adaptive inertia weight teaching-learning-based optimization algorithm and itsapplications. *Appl. Math. Model.* **2020**, *77*, 309–326. [CrossRef]

69. Askari, Q.; Younas, I.; Saeed, M. Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowl.-Based Syst.* **2020**, *195*, 105709. [CrossRef]
70. Storn, R.; Price, K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
71. Nápoles, G.; Grau, I.; Bello, R. Constricted Particle Swarm Optimization based Algorithm for Global Optimization. *Polibits* **2012**, *46*, 5–11. [CrossRef]
72. Sengupta, S.; Basak, S.; Peters, R.A. Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives. *Mach. Learn. Knowl. Extr.* **2018**, *1*, 157–191. [CrossRef]
73. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
74. Hefny, H.A.; Azab, S.S. Chaotic particle swarm optimization. In Proceedings of the 2010 The 7th International Conference on Informatics and Systems (INFOS), Cairo, Egypt, 28–30 March 2010; pp. 1–8.
75. Yang, X.S.; Deb, S. Engineering Optimisation by Cuckoo Search. *Int. J. Math. Model. Numer. Optim.* **2010**, *1*, 330–343. [CrossRef]
76. Ahmad, M.F.; Isa, N.A.M.; Lim, W.H.; Ang, K.M. Differential evolution: A recent review based on state-of-the-art works. *Alex. Eng. J.* **2022**, *61*, 3831–3872. [CrossRef]
77. Moharam, A.; El-Hosseini, M.A.; Ali, H.A. Design of optimal PID controller using hybrid differential evolution and particle swarm optimization with an aging leader and challengers. *Appl. Soft Comput.* **2016**, *38*, 727–737. [CrossRef]
78. Zhang, J.-H.; Zhang, Y.; Zhou, Y. Path planning of mobile robot based on hybrid multi-objective bare bones particle swarm optimization with differential evolution. *IEEE Access* **2018**, *6*, 44542–44555. [CrossRef]
79. Wang, S.; Li, Y.; Yang, H. Self-adaptive mutation differential evolution algorithm based on particle swarm optimization. *Appl. Soft Comput.* **2019**, *81*, 105496. [CrossRef]
80. Boks, R.; Wang, H.; Bäck, T. A modular hybridization of particle swarm optimization and differential evolution. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion (2020), Cancún, Mexico, 8–12 July 2020; pp. 1418–1425.
81. Dash, J.; Dam, B.; Swain, R. Design and implementation of sharp edge FIR filters using hybrid differential evolution particle swarm optimization. *AEU-Int. J. Electron. Commun.* **2020**, *114*, 153019. [CrossRef]
82. Choi, Z.C.; Ang, K.M.; Lim, W.H.; Tiang, S.S.; Ang, C.K.; Solihin, M.I.; Juhari, M.R.M.; Chow, C.E. Hybridized Metaheuristic Search Algorithm with Modified Initialization Scheme for Global Optimization. In Proceedings of the Advances in Robotics, Automation and Data Analytics: Selected Papers from ICITES 2020, Bangkok, Thailand, 13–14 May 2021; Springer: Cham, Switzerland; Volume 1350, p. 172. [CrossRef]
83. Mlakar, U.; Potočnik, B.; Brest, J. A hybrid differential evolution for optimal multilevel image thresholding. *Expert Syst. Appl.* **2016**, *65*, 221–232. [CrossRef]
84. Lin, H.; Siu, S.W. A hybrid cuckoo search and differential evolution approach to protein–ligand docking. *Int. J. Mol. Sci.* **2018**, *19*, 3181. [CrossRef] [PubMed]
85. Zhang, Z.; Ding, S.; Jia, W. A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems. *Eng. Appl. Artif. Intell.* **2019**, *85*, 254–268. [CrossRef]
86. Chi, R.; Su, Y.; Qu, Z.; Chi, X. A hybridization of cuckoo search and differential evolution for the logistics distribution center location problem. *Math. Probl. Eng.* **2019**, *2019*, 7051248. [CrossRef]
87. Xia, J.; Zheng, L. A hybrid algorithm based on cuckoo search and differential evolution for numerical optimization. *Computing* **2020**, *4*, 1–8. [CrossRef]
88. Gao, W.-F.; Huang, L.-L.; Wang, J.; Liu, S.-Y.; Qin, C.-D. Enhanced artificial bee colony algorithm through differential evolution. *Appl. Soft Comput.* **2016**, *48*, 137–150. [CrossRef]
89. Zhou, J.; Yao, X. DE-caABC: Differential evolution enhanced context-aware artificial bee colony algorithm for service composition and optimal selection in cloud manufacturing. *Int. J. Adv. Manuf. Technol.* **2016**, *90*, 1085–1103. [CrossRef]
90. Haohao, C.; Xiaohui, Q.I.; Sen, Y. Obstacle Avoidance and Path Planning for Quadrotor based on Differential Evolution-Artificial Bee Colony Algorithm. *J. Phys. Conf. Ser.* **2018**, *1087*, 022030. [CrossRef]
91. Najari, S.; Gróf, G.; Saeidi, S.; Gallucci, F. Modeling and optimization of hydrogenation of CO2: Estimation of kinetic parameters via Artificial Bee Colony (ABC) and Differential Evolution (DE) algorithms. *Int. J. Hydrogen Energy* **2019**, *44*, 4630–4649. [CrossRef]
92. Zorarpacı, E.; Özel, S.A. Differentially private 1R classification algorithm using artificial bee colony and differential evolution. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103813. [CrossRef]
93. Trivedi, A.; Srinivasan, D.; Biswas, S.; Reindl, T. A genetic algorithm–differential evolution based hybrid framework: Case study on unit commitment scheduling problem. *Inf. Sci.* **2016**, *354*, 275–300. [CrossRef]
94. Thakshaayene, J.; Kavitha, D. Unit commitment using hybrid genetic algorithm with differential evolution. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017; pp. 1–6.
95. Li, Y.; Wang, S.; Hong, X.; Li, Y. Multi-objective task scheduling optimization in cloud computing based on genetic algorithm and differential evolution algorithm. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 4489–4494.
96. Rahmat, N.; Aziz, N.; Mansor, M.; Musirin, I. Optimizing economic load dispatch with renewable energy sources via differential evolution immunized ant colony optimization technique. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2017**, *7*, 2012. [CrossRef]

97. Zhang, X.; Xue, Y.; Lu, X.; Jia, S. Differential-evolution-based coevolution ant colony optimization algorithm for Bayesian network structure learning. *Algorithms* **2018**, *11*, 188. [CrossRef]
98. Xie, X.; Xu, K.; Wang, X. Cloud computing resource scheduling based on improved differential evolution ant colony algorithm. In Proceedings of the 2019 International Conference on Data Mining and Machine Learning (2019), Beijing, China, 8–11 November 2019; pp. 171–177.
99. Tian, D. Particle Swarm Optimization with Chaos-based Initialization for Numerical Optimization. *Intell. Autom. Soft Comput.* **2018**, *24*, 331–342. [CrossRef]
100. Fister, I.; Yang, X.S.; Fister, D.; Fister, I. Cuckoo Search: A Brief Literature Review. In *Cuckoo Search and Firefly Algorithm. Studies in Computational Intelligence*; Yang, X.S., Ed.; Springer: Cham, Switzerland, 2014; Volume 516. [CrossRef]