Application of recent nature-inspired meta-heuristic optimisation techniques to small permanent magnet DC motor parameters identification problem

Yannis L. Karnavas

Abstract: In this study, an attempt is made to find an effective solution on the small direct current (DC) motor’s parameters identification problem by employing two recently introduced stochastic nature-inspired techniques. The first one is called ‘salp swarm algorithm’ (SSA) while the second one is called ‘anti-lion optimiser’ (ALO). These algorithms have been inspired by the interaction strategy between the ocean salps and the ants and ant-lions, respectively, in nature. To appraise the effectiveness of these algorithms, a DC motor model has been appropriately implemented and its performance is evaluated by using speed step responses, while all motor parameters are treated as unknown and therefore search variables. Integral of the square of the error (ISE), integral absolute error and integral in time of absolute error have been adopted as objective functions for the algorithms’ evaluation. In order to judge the acceptability of these algorithms, the simulation results are compared with those of another one similar technique, namely the ‘grey wolf optimiser’ (GWO). The obtained results reveal very satisfactory performance and confirm that the examined SSA, ALO and GWO algorithms can identify accurately the DC motor parameters and can be applied effectively to the specific problem. Another finding is that SSA combined with ISE criterion seems to be the most appropriate technique among the three algorithms.

1 Introduction

It is known that there is a large number of every day’s applications where small permanent magnet DC (PMDC) motors are used. These applications cover not only household use but also extend to many other areas such as medical or industrial equipment, whereas wide range of rotational speeds as well as high precision are required. Some examples include hand-held precision tools, portable battery-operated devices, radio-controlled models, pump equipment, fans, blowers, drills, robotic mechanisms/actuators, shakers and stirrers, security cameras’ pan and tilt drives and many more. Moreover, the cost reduction on high energy rare earth magnets over the last few years has given even greater impetus to the use of the specific motors.

Additionally, in some cases there are low friction and low inertia mass requirements and therefore rapid acceleration/ deceleration and fast motor response times are exhibited. The thermal behaviour of these motors is also satisfactory and thus makes them the best choice especially when combined with a well-designed control system. For the latter, however, it is necessary for the designer to know the motor parameters with as much as possible accuracy, since the design of the respective controllers is sensitive to high tolerances.

The parameters that fully cover the description of the mathematical model of a permanent magnet DC motor are six, namely (i) the armature resistance, (ii) the armature inductance, (iii) the inertia moment of the rotor, (iv) the coefficient of friction/viscosity, (v) the torque coefficient and (vi) the coefficient of the back-electromotive force (back-emf). Most of the manufacturers either do not provide information about these parameters (considering them to be a sensitive industrial information), or give approximate values with a wide range of tolerance (especially in relatively inexpensive motor models).

On the other hand, there is no easy experimental way by which the parameters of a DC motor can be accurately evaluated, for example, in a laboratory using simple equipment. It therefore becomes apparent that the development and/or application of techniques and strategies/methods that would assist in this direction is always necessary. Moreover, since the values of these parameters in small DC motors are relatively small – in order of magnitude – to be measured accurately by an experimental apparatus, the usefulness of other approaches is evident. Once the motor parameters are determined, the next step in designing the appropriate controller for the application becomes easy and straightforward. Another feature which arises after the identification of the DC motor parameters is to analyse the motor performance over its full operational range through simulations and investigate its behaviour in extreme conditions, which would be hazardous if the actual motor was used.

General system identification methods have been applied to DC motor model identification, e.g. [1, 2]. Other methods like the algebraic identification method [3, 4] and the recursive least square method [5] have also been used. An expansion of the latter method has also been reported in parameter estimation of a permanent magnet synchronous motor [6]. A method based on block-pulse function series and neural networks was used to estimate the parameters of the continuous-time model of a PMDC [7], while adaptive Tabu-search method was adopted in [8] to identify the parameters of a separately excited DC motor. Another approach utilising no-load speed step responses have been proposed in [9]. Other methods include quantised speed measurements [10], and commercial software solutions like MATLAB parameter estimation/identification toolboxes [11, 12]. At the same time, efforts employing evolutionary based techniques such as genetic algorithm (GA) [13] and non-dominated sorting genetic algorithm [14], reported results in the same problem. In [15], identification of PMDC motor parameters for an electric wheelchair application using GA was proposed.

In some cases, uncertain model were obtained from transfer functions identified in several working points chosen from the I/O steady characteristics under minimum and maximum adjustable loads, e.g. [16]. In [17], a methodology was described based on a variational approach combined with worst-case analysis and Monte Carlo simulation to determine the impact of a DC motor parameters variation on the system behaviour. Also, some works have focus in determining accurately only some of the required parameters. For example, in [18], a friction compensation and
It should be noted here that the paper does not intend to argue with the effectiveness of other parameter estimation techniques mentioned before, neither to propose a certain technique out of those examined here. Instead, the motivation in this work is clearly the investigation of the applicability of the examined nature-inspired meta-heuristic algorithms as an alternative method for the specific problem, which may be used in conjunction with other well-established ones.

The work is organised as follows. The mathematical modelling of the PMDC motor as utilised here is given in Section 2. The description of the problem statement as well as the selection of the objective functions are presented in Section 3. The main features of the three nature-inspired techniques optimisation are given in Section 4. In Section 5, the identification strategy is analysed, the relevant results from the application of the three techniques along with the corresponding comments and discussion are given. Finally, the work concludes in Section 6.

2 Brief review of small DC motor characteristics and mathematical modelling

2.1 General characteristics

Small PMDC motors differ from other types of motors in many ways. Fig. 1 depicts different views of a disassembled such motor. When the motor is assembled the only parts which are projected out of the frame (housing) are the shaft and the power supply terminals. In Fig. 1a, the brush cap has been removed and the two carbon brushes are shown which – under operation – are connected to the supply terminals. At the same figure, a three-pole rotor with its winding is shown installed inside the motor housing. Also, two permanent magnets surround the rotor in order to provide constant magnetic flux while the rotor is running. In Fig. 1b, the rotor has been pulled out, exposing the thin stator iron laminations and the necessary commutator. Metal pins are usually used to hold the permanent magnets in place. Finally, the rotor may or may not be supported by one or two ball bearings at the ends of its shaft so it can be rigidly affixed to the outside motor housing and to rotate freely with low friction.

2.2 Mathematical modelling

The typical equivalent circuit of a permanent magnet DC motor is shown in Fig. 2. The source voltage ($V_s$) is applied to the terminals of the armature winding which is characterised by its ohmic resistance ($R_a$) and its inductance ($L_a$). At any time instant during motor's operation, an armature current ($i_a$) is imposed to the winding and an induced electromotive force is developed ($E_a$), which is proportional to the rotational speed ($\omega_m$) – or equivalently to the rate of change in shaft position ($\theta$) – and the magnetic field's flux ($\Phi$). Considering that the motor has been designed in such a way that its operating range is related to the linear portion of the stator/rotor ferromagnetic material's magnetisation curve, this proportionality is expressed through the back-emf coefficient ($k_e$). Thus, from this part of the equivalent circuit, the following well-known equations are derived:

$$V_s = E_a + i_a R_a + L_a \frac{di_a}{dt}$$

(1)

where

$$E_a = k_e \omega_m$$

(2)

and

$$\omega_m = \frac{d\theta}{dt}$$

(3)

In physical terms, the back-emf represents the transition between the electrical and the mechanical power. Since $E_a$ is less than $V_s$ due to the voltage drop across the armature winding, the
By combining (1)–(3) and (4)–(7), the following differential equations are obtained:

\[ T_{em} = k_i s = T_{out} + T_L + T_f \]  

where

\[ k_i = K \Phi \]  

\[ T_{out} = J_m \frac{d\omega_m}{dt} \]  

and

\[ T_f = B_m \omega_m \]  

By combining (1)–(3) and (4)–(7), the following differential equations are obtained:

\[ L_s \frac{d}{dt} i_s(t) + R_s i_s(t) = V_i(t) - k_t \frac{d}{dt} \theta(t) \]  

\[ J_m \frac{d^2}{dt^2} \theta(t) + J_m \frac{d^2}{dt^2} \theta(t) + B_m \frac{d}{dt} \theta(t) = k_i s \]  

Now, the Laplace transform of (8) and (9) yields to the following transfer function:

\[ \frac{\theta(s)}{V(s)} = \frac{k_i}{J_m L_s + (J_m R_s + L_s B_m) \cdot s + (k_t k_i + R_s B_m) \cdot s} \]  

where it can be seen that the DC motor is represented as a single-input single-output system with input variable the supply voltage and output variable the rotor position or its derivative, i.e. speed. From (10) it is also clear that in order to fully identify the motor model, there are six parameters that need to be known. The block diagram representing the above transfer function is shown in Fig. 3, whereas an experimental (blue line) speed-step response is depicted in Fig. 4. Under constant input voltage and zero initial conditions, the first part of the curve in Fig. 4 is the no-load characteristic (the motor accelerates and reaches the no-load speed), while the second part is the load characteristic (a load torque is applied at \( t = 0.1 \) s and the motor slows down until reaches the load speed).

Based on the above, the approach followed here comprises of three steps: (i) a step speed response under no-load (start-up) and under load torque disturbance is considered using the actual motor under determination, (ii) a model is built as the ‘unknown’ model to be fitted, either in transfer function form or in state-space form and (iii) the parameters \( R_s, L_s, J_m, B_m, k_t \) and \( k_i \) are estimated through the applied algorithms and the model is evaluated using certain criteria, performing a number of iterations for a given number of search agents. The procedure is repeated several times. The goal is to minimise at any time instant the error between the response to be fitted and the actual response. From the obtained results, mean absolute error and the standard deviation are mainly considered.

### 3 Optimisation problem formulation

Without loss of generality, the problem set out above is an optimisation problem under constraints [25, 26]. Mathematically can be described as follows: let \( f \) be an objective function defined in the search space \( Q \subseteq \mathbb{R}^n \). Also, let \( C \) and \( Z \) be the set of feasible solutions and the set of infeasible solutions, respectively, where \( C \subseteq Q \) and \( Z = Q - C \). Additionally, the search variables domains are limited by their upper (ub) and lower (lb) bounds, i.e. the search space is defined as an \( n \)-dimensional rectangle in the set \( \mathbb{R}^n \). Each vector \( x \in C \) is called a feasible solution, otherwise an infeasible one. Thus, the problem is denoted as

\[ \text{minimise} \quad f(x), \quad x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \]  

\[ x_{lb} \leq x_j \leq x_{ub}, \quad j = 1, 2, \ldots, n \]  

\[ C = \left\{ x | g_k(x) \leq 0; \quad h_j(x) = 0; \quad k = 1, 2, \ldots, L; \quad j = L + 1, \ldots, P \right\} \]  

whereas the feasible set \( C \) can be defined also by a set of additional \( P \geq 0 \) constraints.

#### 3.1 Objective (fitness) function selection

Usually one fitness function to which an optimisation algorithm is applied is used in similar experiments. Here, three objective functions have been employed in order to evaluate the applied methods. Considering that the error (between the two signals according to Fig. 4) can be expressed as

\[ e(t) = n_{\text{experimental}}(t) - n_{\text{simulation}}(t) \]  

then, the first objective function is the integral of the square of the error (ISE) which penalises large errors more than the small ones. The second objective function is the integral absolute error (IAE) which penalises all errors equally regardless of “direction” (sign). Finally, the third objective function is the integral in time of absolute error (ITAE), which penalises long duration transients and in some cases can be more selective than the ISE [27]. Thus, the formulation of the problem is now completed like
In 2014, the GWO algorithm was presented by Mirjalili et al. [22]. His first application was to train a multi-layer perceptron network. The results were compared with those of GAs, evolution strategy, ant-colony optimisation and PSO and it was found that GWO is superior to them. Recently, the first attempt of GWO application to the problem of determining DC motor parameters has been presented by Karnavas and Chasiotis [28]. The results were compared with GA and found that the GWO performed better. Mirjalili et al. [22] claimed that the superiority of GWO is based on the fact that there is no need for specific input parameters along with the complexity-free inherent characteristic. A recent literature review about their application up to now can be found in [29].

The GWO algorithm mimics the leadership hierarchy as well as the mechanism by which the grey wolves (Canis lupus of Canidae family) hunt in nature (Fig. 5). Grey wolves usually live in 5–12 member packs and are at the top of the food chain in the area where they live. At the same time, they follow a strict social hierarchy. Four types of grey wolves were adopted to simulate this hierarchy: alpha (α), beta (β), delta (δ) and omega (ω). Alpha wolves correspond to the top of the hierarchy, i.e. leaders (they give the optimal solution to the optimisation problem). Beta wolves are the next level and actually help alphas in making decisions as well as commanding those of the lower levels. The omega wolves refer to the weakest subgroup of the pack and are submitted to the other subgroups. Finally, members who cannot be described as alpha, beta or omega are the delta wolves. The latter can be – for example – scouts or caretakers of the pack. Besides the social hierarchy, an additional characteristic of GWO is the group hunting. The hunting phases are (i) searching for and approaching the prey, (ii) encircling prey and (iii) attacking prey. The mathematical implementation of these phases is actual the mean of performing optimisation in a complex problem. The mathematical expressions used by GWO are formulated here as follows:

$$X_{\text{wolf}}(k + 1) = X_{\text{prey}}(k) - A \cdot D$$

(16)

where

$$D = [C \cdot X_{\text{prey}}(k) - X_{\text{wolf}}(k)]$$

$$A = 2a \cdot r_1 - a, \quad C = 2r_2$$

In (16), $k$ is the current iteration, $X_{\text{wolf}}$ and $X_{\text{prey}}$ are the grey wolves’ position vector and the prey’s position vector, respectively. Moreover, $A$, $C$ are coefficient vectors, $r_1$, $r_2$ are arbitrary vectors ($[r_1], [r_2] \in [0, 1]$), $a$ is a vector linearly reduced from 2 to 0 during iterations (approaching prey mechanism). Especially for the vector $A$, it can take values in the range $[-a, a]$, so when $|A| < 1$ the wolves are forced to attack the prey, while when $|A| > 1$ the wolves are forced to stand off the prey and possibly find a fitter one.

The effect of (16) can be understood considering a three-dimensional search space like in Fig. 6. If a grey wolf is considered in the position of $(x, y, z)$ and the prey lie in the position $(x', y', z')$ then different positions around the prey can be reached with respect to the current position by adjusting the vectors $A$ and $C$. In this adjustment, the random vectors $r_1$ and $r_2$ play a vital role in the algorithm performance. This is translated as the method’s exploitation and exploration ability. In any iteration, the wolves’ position update is performed by (see (17)). In Fig. 7, the way by which a search agent (wolf) updates its position in a 2D search space is shown. It can be observed that the final position would be in a random place within a circle which is defined by the positions of alpha, beta and delta in the search space. Finally, an illustration of the GWO algorithm used here – in pseudo-code form – is depicted in Fig. 8.

### 4.1 Grey wolf optimiser

In 2014, the GWO algorithm was presented by Mirjalili et al. [22]. His first application was to train a multi-layer perceptron network. The results were compared with those of GAs, evolution strategy, ant-colony optimisation and PSO and it was found that GWO is superior to them. Recently, the first attempt of GWO application to the problem of determining DC motor parameters has been presented by Karnavas and Chasiotis [28]. The results were compared with GA and found that the GWO performed better. Mirjalili et al. [22] claimed that the superiority of GWO is based on the fact that there is no need for specific input parameters along with the complexity-free inherent characteristic. A recent literature review about their application up to now can be found in [29].

The GWO algorithm mimics the leadership hierarchy as well as the mechanism by which the grey wolves (Canis lupus of Canidae family) hunt in nature (Fig. 5). Grey wolves usually live in 5–12 member packs and are at the top of the food chain in the area where they live. At the same time, they follow a strict social hierarchy. Four types of grey wolves were adopted to simulate this hierarchy: alpha (α), beta (β), delta (δ) and omega (ω). Alpha wolves correspond to the top of the hierarchy, i.e. leaders (they give the optimal solution to the optimisation problem). Beta wolves are the next level and actually help alphas in making decisions as well as commanding those of the lower levels. The omega wolves refer to the weakest subgroup of the pack and are submitted to the other subgroups. Finally, members who cannot be described as alpha, beta or omega are the delta wolves. The latter can be – for example – scouts or caretakers of the pack. Besides the social hierarchy, an additional characteristic of GWO is the group hunting. The hunting phases are (i) searching for and approaching the prey, (ii) encircling prey and (iii) attacking prey. The mathematical implementation of these phases is actual the mean of performing optimisation in a complex problem. The mathematical expressions used by GWO are formulated here as follows:

$$X_{\text{wolf}}(k + 1) = X_{\text{prey}}(k) - A \cdot D$$

(16)

where

$$D = [C \cdot X_{\text{prey}}(k) - X_{\text{wolf}}(k)]$$

$$A = 2a \cdot r_1 - a, \quad C = 2r_2$$

In (16), $k$ is the current iteration, $X_{\text{wolf}}$ and $X_{\text{prey}}$ are the grey wolves’ position vector and the prey’s position vector, respectively. Moreover, $A$, $C$ are coefficient vectors, $r_1$, $r_2$ are arbitrary vectors ($[r_1], [r_2] \in [0, 1]$), $a$ is a vector linearly reduced from 2 to 0 during iterations (approaching prey mechanism). Especially for the vector $A$, it can take values in the range $[-a, a]$, so when $|A| < 1$ the wolves are forced to attack the prey, while when $|A| > 1$ the wolves are forced to stand off the prey and possibly find a fitter one.

The effect of (16) can be understood considering a three-dimensional search space like in Fig. 6. If a grey wolf is considered in the position of $(x, y, z)$ and the prey lie in the position $(x', y', z')$ then different positions around the prey can be reached with respect to the current position by adjusting the vectors $A$ and $C$. In this adjustment, the random vectors $r_1$ and $r_2$ play a vital role in the algorithm performance. This is translated as the method’s exploitation and exploration ability. In any iteration, the wolves’ position update is performed by (see (17)). In Fig. 7, the way by which a search agent (wolf) updates its position in a 2D search space is shown. It can be observed that the final position would be in a random place within a circle which is defined by the positions of alpha, beta and delta in the search space. Finally, an illustration of the GWO algorithm used here – in pseudo-code form – is depicted in Fig. 8.

### 4.2 Ant-lion optimiser

The next nature-inspired meta-heuristic technique adopted here was proposed in 2015 and it was inspired by the ant-lions and ants interaction [23]. In that study, 19 different mathematical problems were solved plus three engineering problems. Since then, there...
were many applications of the ALO in several engineering problems (e.g. in [30, 31]) which demonstrated its capabilities. However, ALO has never been used in the DC motor parameter modelling for the entrapment of the ants is given by

\[ X(k) = [0, cs(2r(1) - 1), cs(2r(2) - 1), \ldots, cs(2r(k) - 1), \ldots, cs(2r(N) - 1)] \]

(20)

where \( N \) is the number of search variables, \( cs \) is the ants position (search agents) and \( k \) is the current iteration. At the same time, the ants position – in the search space – and their corresponding fitness are given by

\[
P_{\text{ant}} = \begin{bmatrix}
\alpha_{1,1} & \alpha_{1,2} & \ldots & \alpha_{1,d} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha_{n,1} & \alpha_{n,2} & \ldots & \alpha_{n,d}
\end{bmatrix}
\]

\[
F_{\text{ant}} = \begin{bmatrix}
f_1(k([\alpha_{1,1}, \alpha_{1,2}, \ldots, \alpha_{1,d}])) \\
\vdots \\
f_k(k([\alpha_{n,1}, \alpha_{n,2}, \ldots, \alpha_{n,d}]))
\end{bmatrix}
\]

where \( d \) is the number of search variables, \( n \) is the ants population (search agents) and \( k \) is the current iteration. At the same time, the ants position – in the search space – and their corresponding fitness are given by

\[
P_{\text{d}} = \begin{bmatrix}
\alpha_{1,1} & \alpha_{1,2} & \ldots & \alpha_{1,d} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha_{n,1} & \alpha_{n,2} & \ldots & \alpha_{n,d}
\end{bmatrix}
\]

\[
F_{\text{d}} = \begin{bmatrix}
f_1(k([\alpha_{1,1}, \alpha_{1,2}, \ldots, \alpha_{1,d}])) \\
\vdots \\
f_k(k([\alpha_{n,1}, \alpha_{n,2}, \ldots, \alpha_{n,d}]))
\end{bmatrix}
\]

As the ants move randomly by nature, their movement (Fig. 10) is modelled by a random walk technique which can be expressed by

\[
X = \frac{(X(k) - \alpha) + (\delta(k) - \gamma(k))}{(\beta - \alpha)}
\]

(22)

\[
\gamma(k) = c(k)R, \delta(k) = d(k)R
\]

(23)

where \( \alpha \) and \( \beta \) are the minimum and maximum values, respectively, for the ith variable, \( \gamma \) and \( \delta \) are the minimum and maximum value, respectively, of the ith variable at the iteration \( k \), \( c \) and \( d \) are vectors with the minimum and maximum values of the search variable at iteration \( k \), and \( R = (wk/\text{Iter}) \) is a ratio which models the increment/decrement of the ant’s position distance from the trap \( w \) a constant and \( \text{Iter} \) the total iterations. Continuing, the modelling for the entrapment of the ants is given by

\[
y_{\text{ant}}(k) = al(k) + \gamma(k), \delta_{\text{ant}}(k) = al(k) + \delta(k)
\]

(24)
where $a_i$ and $\delta_i$ are vectors including the minimum and maximum values of the search variables for the $i$th ant at iteration $k$ and all is the position of the selected $l$th ant-lion at the same iteration. Moreover, the elitism feature is modelled also in ALO. Elitism is an important aspect of the evolutionary algorithms which allows them to keep the best solution during the optimisation process. Thus, the mechanism which is considered in order to generate new positions for the ants is based on the elite ant-lion (the best ant-lion found up to a certain iteration) and the average of the elite and the roulette wheel selected ant-lion random walks, i.e.

$$a_i(k) = \frac{(R_{selected,al}(k) + R_{base,al}(k))}{2}$$

(25)

At the final step of ant-lion hunting, an ant falls into the bottom of the trap and is caught by the ant-lion’s jaws. The ant-lion pulls the ant into sand and consumes it. In the ALO algorithm, catching the prey occurs when an ant’s fitness function becomes better than its corresponding ant-lion. In this situation, the ant-lion changes its position to the position of the hunted ant. This process can be mathematically expressed as

$$a_i(k) = a_i(k) \quad \text{if} \quad f(a_i(k)) > f(a_i(k))$$

(26)

Finally, an illustration of the ALO algorithm in pseudo-code form used here is depicted in Fig. 11 while a further detailed description can be found in [32].

### 4.3 Salp swarm algorithm

The last algorithm adopted in this work was shown in the literature in 2017 inspired by salps behaviour [24]. Despite the very short time since it was introduced, applications of SSA have already been reported (e.g. [33, 34]). However, this is the first time where SSA is used in an electrical engineering problem.

Salps belong to the family of Salpidae and have transparent barrel-shaped body. Their tissues are highly similar to jelly fishes. Essentially, salps are gelatinous zooplankton owing complex nervous and digestive systems. It is important species in the food chain since 200 other species eat them. Salps form chains of clones. These chains then break up and each salp gives birth to more salps. Their movement is accomplished, by pumping the water through their body as a propulsion mean to move forward. Fig. 12a depicts the shape of a salp, while in Fig. 12b a salp chain formation is shown.

In mathematical and modelling terms, the salp swarm behaviour is simpler compared to the previous techniques presented, as it will be shown next. Firstly, it is assumed that the salp chain has one leader (the first salp in the formation) while all the rest of the salps are followers. Then, if an $n$-dimension search space is considered, i.e. $n$ is the number of search variables, the positions of the salps are stored in a matrix, let it be $X$. It is also considered that there is a target (let it be denoted $F$) which in nature is a food source. The relationship which SSA uses then is given by

$$X_{i,j} = \frac{F_j + c_1((ub_j - lb_j)v_i + lb_j)}{c_3 > 0}$$

(27)

In essence, (27) implies that the leader salp only updates its position with respect to the food source. Additionally, $X_{i,j}$ denotes the leader salp’s position in the jth dimension, $F_j$ represents the food source’s position, $ub_j$ and $lb_j$ indicate the upper and lower bound of the jth dimension, $c_1$ and $c_3$ are random numbers uniformly distributed in the interval of $[0, 1]$ which are dictating if the next position will be towards negative or positive infinity and $c_1$ is a coefficient which depends on the current ($k$) and the total number of iterations ($L$), i.e.

$$c_1 = 2e^{-\frac{k}{L}}$$

(28)

Next, the Newton’s law of motion is utilised for the modelling of the position updating mechanism of the follower salps

$$X_{i,j} = \frac{1}{2}a_i + v_{i,j}$$

(29)

where $v_i$ is the initial speed, $a$ is a speed ratio

$$a = \frac{v_i}{v_0}, \quad v_j = X - X_0$$

As time ($t$) in optimisation is the iteration, by considering $v_0 = 0$, (29) can be written as

$$X_{i,j} = \frac{1}{2}(X_{i,j} + X_{i,j+1})$$

(30)

The pseudo-code of the SSA algorithm implemented here is depicted in Fig. 13.

### 5 Study cases, results and discussion

#### 5.1 General remarks

Table 1 shows the functional characteristics and parameters of the actual DC motor under consideration [35]. According to the manufacturer, this is a small DC motor with no-load and full-load speeds of 2500 and 19,200 rpm, respectively (the blue curve in Fig. 4 shows the experimental speed response for this motor). Such a response can be obtained in a laboratory environment through a data acquisition system connected to a computer, which records the motor speed. Apart from that, the only equipment needed is a constant voltage power supply and a speed sensor (optical encoder). Also, Table 2 shows the common parameters of the three nature-inspired algorithms used here. Each algorithm ran 20 times for 500 iterations each with a population of 60 search agents. This was repeated for each of the three criteria that were adopted as objective functions. In each run, the best, worst and average values for each of the six search parameters were stored, as well as the objective function.

![Fig. 11](http://creativecommons.org/licenses/by/3.0/)

**Fig. 11** Simplified illustration of ALO algorithm used here in script pseudo-code

![Fig. 12](http://creativecommons.org/licenses/by/3.0/)

**Fig. 12** Sea salps in nature
(a) Individual salp, (b) Salp chain formation (salps swarm)
5.2 Results and discussion

The results obtained are presented and analysed next. In Fig. 14, the algorithm performance (in logarithmic scale) with respect to their ability to converge is presented. In particular, each sub-graph shows the best, worst and average convergence curves from the 20 independent algorithm executions. The classification was made based on the criterion examined (Fig. 14a for ISE, Fig. 14b for IAE and Fig. 14c for ITAE), because the order of magnitude of the result of each criterion is – by mathematical definition – different.

In addition, for easy comparison reasons, the scale of the algorithm used here the GWO marginally reaches a value in the order of 10−4. The classification was made based on the criterion examined (Fig. 14a for ISE, Fig. 14b for IAE and Fig. 14c for ITAE), because the order of magnitude of the result of each criterion is – by mathematical definition – different. In addition, for easy comparison reasons, the scale of the y-axis in each criterion is the same. A general remark is that all algorithms exhibit a converging capability, even though the performance of some of them cannot be judged very satisfactory as it can be seen later.

With respect to the ISE criterion it is noted that SSA and ALO can minimise the objective function by up to an order of 10−5 while the GWO marginally reaches a value in the order of 10−2.

Similarly, the average convergence curves present an analogous behaviour. The corresponding performances for the IAE criterion show that the objective function reaches (at best) a value at an order of 10−3 and 10−4 for SSA and ALO while for GWO this value is about 10−1. Regarding the ITAE criterion, 10−3 and 10−4 are the orders of magnitude for SSA and ALO, while for GWO is low again at an order of 10−8.

Another feature that is observed is as follows: the convergence of the SSA follows for a certain number of iterations a slight downward trend and after a point it decreases abruptly. On the other hand, the convergence of ALO follows a relatively constant rate downward trend across the whole range of iterations. Finally, GWO abruptly drops during a small number of iterations initially, and then almost gets its final value, indicating that it may be inadequate to explore further the possible solution space for the problem.

Results in tabular form are presented next. Tables 3–5 show the SSA performance regarding the DC motor parameters for each criterion examined, respectively. The second column of each table is the actual parameter value according to the manufacturer, while the third, fourth and fifth columns are the best, worst and average values found by the applied algorithm among the 20 independent runs

\[
\epsilon(\%) = \left( \frac{P_{\text{ave}} - P_i}{P_{\text{ave}}} \right) \times 100, \quad P_{\text{ave}} = \frac{1}{10} \sum_{j=1}^{10} P_j
\]

The sixth column is the mean percentage of the error for each parameter – through the 20 runs – which has been calculated using (31) where \( P_i \) is the exact value of the parameter to be found, \( P_{\text{ave}} \) is the estimated value of the parameter in a certain run and \( P_{\text{ave}} \) is the mean value of all the \( P_i \) values. Finally, the last column of the tables shows the standard deviation of the best estimated value for each parameter and for each independent run.

The contents of the tables follow the same layout for the rest of them (Tables 6–11) for easy reference, as it will be presented later.

Hence, it can be observed from Table 3–Table 5 (bold values) that for four out of the six search variables (i.e. \( R_e, L_a, J_m \) and \( k_t \)), SSA gave the smallest error for ISE as objective function, while the same is given when IAE is used. Specifically, the accuracy range for motor winding resistance (\( R_e \)) is 3.2–5.6%, for the winding inductance (\( L_a \)) 2.2–4.9%, for the moment of inertia (\( J_m \)) (8 × 10−5)–0.2%, for the friction coefficient (\( B_a \)) 1.1–2.8%, for the torque coefficient (\( k_t \)) 3.2–5.6% and for the back-emf coefficient (\( k_e \)) 0.14–0.27%. It should be noted however here that all of the aforementioned accuracies (even the worst ones) are very satisfactory and considered fully acceptable from a technical point of view. Finally, it seems that IAE maybe not suitable for performance criterion when SSA is adopted.

The corresponding results for ALO performance are given in Table 6–Table 8. Here, for three out of the six search variables (i.e. \( R_e, L_a, k_t \)) IAE performed better as objective criterion, while for the mechanical parameters (\( J_m, B_a \)) IAE seems to be better. Back-emf coefficient (\( k_e \)) presented minimum error when ISE was used but it is comparable with this found using IAE, so it can be said that IAE performed equally well in estimating \( k_e \). The accuracy range for motor winding resistance (\( R_e \)) is 2.6–4.2%, for the winding inductance (\( L_a \)) 0.3–3.2%, for the moment of inertia (\( J_m \)) (8 × 10−5)–2.3 × 10−3%, for the friction coefficient (\( B_a \)) 2–5.3% for the torque coefficient (\( k_t \)) 2.6–4.2% and for the back-emf coefficient (\( k_e \)) 0.09–0.14%.

Also, it seems that ISE is not adequate in case of ALO.

The last three tables (Tables 9–Table 11) refer to the GWO performance. Here it can be seen that four out of the six search variables (i.e. \( R_e, J_m, k_t \) and \( k_e \)), are estimated with the minimum error using the ISE criterion again (as in SSA). Viscous friction coefficient (\( B_a \)) smallest error obtained by IAE, while winding inductance (\( L_a \)) by ITAE. The parameter estimation accuracy ranges obtained by GWO are: for motor winding resistance (\( R_e \)) is

---

Table 1

| Parameter                      | Symbol | Unit   | Value |
|-------------------------------|--------|--------|-------|
| nominal voltage               | \( V_s \) | Volts  | 1.5   |
| nominal current               | \( I_n \) | mA     | 290   |
| no-load current               | \( I_0 \) | mA     | 30    |
| stall current                 | \( I_{\text{max}} \) | mA     | 380   |
| nominal torque                | \( T_n \) | mNm    | 17    |
| stall torque                  | \( T_{\text{max}} \) | mNm    | 24    |
| nominal speed                 | \( n_n \) | rpm    | 2500  |
| no-load speed                 | \( n_0 \) | rpm    | 19,100|
| terminal resistance           | \( R_e \) | \( \Omega \) | 3.9 |
| terminal inductance           | \( L_a \) | \( \mu H \) | 12   |
| torque constant               | \( k_t \) | mNm/A  | 0.69  |
| back-EMF constant             | \( k_c \) | mV/min\(^{-1}\) | 0.072 |
| rotor inertia                 | \( J_m \) | g \cdot cm\(^2\) | 0.01  |
| damping constant              | \( B_n \) | mNm/\( \text{min}^{-1}\) | 1/78,224 |

---

Table 2

| Algorithm parameter          | Value |
|-------------------------------|-------|
| no. of search variables      | 6     |
| iterations                   | 500   |
| agent population             | 60    |
| independent runs             | 20    |
| criteria examined            | 3     |

---

Initialize the salp population \( x_i \) \((i = 1, 2, \ldots, n)\) considering \( a_b \) and \( a_l \)

while \( \text{end condition is not satisfied} \)

Calculate the fitness of each search agent (salp)

\( F = \text{the best search agent} \)

Update \( c_i \) by Eq. (20)

for each salp \( (x_i) \)

\( i = (i+1) \)

else

Update the position of the leading salp by Eq. (27)

end if

end for

Amend the salps based on the upper and lower bounds of variables

return \( F \)

Fig. 13 Simplified illustration in script pseudo-code form of SSA algorithm used here
Fig. 14 Convergence (fitness functions) variation of SSA, ALO and GWO through iterative process (for easy comparison, sub-figures for each criterion have common y-axis.)
(a) For ISE criteria, (b) For IAE criteria, (c) For ITAE criteria

Table 3 SSA performance based on ISE criterion (through 20 independent runs)

| Quantity | Exact value | Best | Worst | Mean | Mean error, % | Standard deviation |
|----------|-------------|------|-------|------|---------------|--------------------|
| $R_a$, $\Omega$ | 3.9 | 3.9330 | 5.0680 | 4.0291 | 3.2040 | 0.4822 |
| $L_a$, $\mu H$ | 12 | 12.1090 | 15.1660 | 12.2780 | 2.2640 | 1.4397 |
| $J_m$, $gcm^2$ | 0.01 | 0.0099 | 0.0100 | 0.0099 | $8.29 \times 10^{-6}$ | $2.31 \times 10^{-7}$ |
| $B_m$, $min^{-1}/mNm$ | 78,224 | 79,903.78 | 101,219.77 | 77,309.30 | 1.1831 | 8642.25 |
| $k_c$, $mNm/A$ | 0.69 | 0.6958 | 0.8966 | 0.7128 | 3.2040 | 0.0853 |
| $k_e$, $mV/min^{-1}$ | 0.072 | 0.0721 | 0.0731 | 0.0718 | 0.1588 | 0.0005 |
| $f_1$ (ISE) | $4.81 \times 10^{-7}$ | 0.0292 | 0.0037 | | | |
| Sim. time, min | — | 86.53 | 102.16 | 94.79 | — | — |
### Table 4 SSA performance based on IAE criterion (through 20 independent runs)

| Quantity | Exact value | Best | Worst | Mean | Mean error, % | Standard deviation |
|----------|-------------|------|-------|------|---------------|-------------------|
| $R_a, \Omega$ | 3.9 | 3.8798 | 5.0537 | 4.1351 | 6.8683 | 0.3900 |
| $I_v, \mu$H | 12 | 11.8953 | 15.3693 | 12.3111 | 2.5269 | 1.3766 |
| $J_m, \text{gcm}^2$ | 0.01 | 0.0100 | 0.0102 | 0.0100 | 0.0028 | $8.49 \times 10^{-6}$ |
| $B_m, \text{min}^{-1}/\text{mNm}$ | 78.224 | 78.430.42 | 98.930.91 | 77.345.43 | 1.1359 | 7928.94 |
| $k_v, \text{mNm/A}$ | 0.69 | 0.6864 | 0.8941 | 0.7316 | 5.6861 | 0.0690 |
| $k_e, \text{mV/min}^{-1}$ | 0.072 | 0.0720 | 0.0730 | 0.0719 | 0.1447 | $4.91 \times 10^{-4}$ |
| $f_s, \text{IAE}$ | $7.04 \times 10^{-4}$ | 6.1824 | 1.4469 | — | — | — |
| Sim. time, min | — | 103.85 | 134.02 | 118.81 | — | — |

### Table 5 SSA performance based on ITAE criterion (through 20 independent runs)

| Quantity | Exact value | Best | Worst | Mean | Mean error, % | Standard deviation |
|----------|-------------|------|-------|------|---------------|-------------------|
| $R_a, \Omega$ | 3.9 | 3.8602 | 4.6821 | 4.0355 | 3.3590 | 0.3575 |
| $I_v, \mu$H | 12 | 12.1102 | 15.4603 | 12.6255 | 4.9546 | 1.4186 |
| $J_m, \text{gcm}^2$ | 0.01 | 0.0099 | 0.0102 | 0.0100 | 0.0013 | $3.31 \times 10^{-7}$ |
| $B_m, \text{min}^{-1}/\text{mNm}$ | 78.224 | 78.007.60 | 97.167.66 | 76.046.17 | 2.8638 | 8131.28 |
| $k_v, \text{mNm/A}$ | 0.69 | 0.6829 | 0.8283 | 0.7139 | 3.3584 | 0.0632 |
| $k_e, \text{mV/min}^{-1}$ | 0.072 | 0.0719 | 0.0729 | 0.0718 | 0.2717 | 0.0005 |
| $f_s, \text{ITAE}$ | $3.50 \times 10^{-8}$ | 0.0575 | 0.0077 | — | — | — |
| Sim. time, min | — | 135.21 | 165.57 | 150.70 | — | — |

### Table 6 ALO performance based on ISE criterion (through 20 independent runs)

| Quantity | Exact value | Best | Worst | Mean | Mean error, % | Standard deviation |
|----------|-------------|------|-------|------|---------------|-------------------|
| $R_a, \Omega$ | 3.9 | 3.9934 | 5.0699 | 3.7925 | 2.8339 | 0.9741 |
| $I_v, \mu$H | 12 | 12.0193 | 15.5973 | 12.4238 | 3.4118 | 2.3575 |
| $J_m, \text{gcm}^2$ | 0.01 | 0.0099 | 0.0100 | 0.0100 | 0.0013 | $8.92 \times 10^{-5}$ |
| $B_m, \text{min}^{-1}/\text{mNm}$ | 78.224 | 79.323.48 | 101.689.33 | 80.489.31 | 2.8144 | 16,806.89 |
| $k_v, \text{mNm/A}$ | 0.69 | 0.7065 | 0.8970 | 0.6709 | 2.8338 | 0.1723 |
| $k_e, \text{mV/min}^{-1}$ | 0.072 | 0.0720 | 0.0731 | 0.0719 | 0.0992 | 0.0009 |
| $f_s, \text{ISE}$ | $3.50 \times 10^{-8}$ | 0.0575 | 0.0077 | — | — | — |
| Sim. time, min | — | 57.23 | 88.39 | 68.92 | — | — |

### Table 7 ALO performance based on IAE criterion (through 20 independent runs)

| Quantity | Exact value | Best | Worst | Mean | Mean error, % | Standard deviation |
|----------|-------------|------|-------|------|---------------|-------------------|
| $R_a, \Omega$ | 3.9 | 4.2729 | 5.0699 | 3.8012 | 2.5987 | 0.9953 |
| $I_v, \mu$H | 12 | 12.0817 | 15.5955 | 12.3957 | 3.1928 | 2.5904 |
| $J_m, \text{gcm}^2$ | 0.01 | 0.0100 | 0.0102 | 0.0100 | 0.0023 | $15.127.38$ |
| $B_m, \text{min}^{-1}/\text{mNm}$ | 78.224 | 77.044.87 | 101.669.25 | 79.867.14 | 2.0573 | 15.127.38 |
| $k_v, \text{mNm/A}$ | 0.69 | 0.6264 | 0.8970 | 0.6709 | 2.8338 | 0.1723 |
| $k_e, \text{mV/min}^{-1}$ | 0.072 | 0.0720 | 0.0731 | 0.0719 | 0.0992 | 0.0009 |
| $f_s, \text{IAE}$ | $2.22 \times 10^{-6}$ | 0.0002 | 7.1810^{-5} | — | — | — |
| Sim. time, min | — | 58.82 | 134.15 | 112.07 | — | — |

### Table 8 ALO performance based on ITAE criterion (through 20 independent runs)

| Quantity | Exact value | Best | Worst | Mean | Mean error, % | Standard deviation |
|----------|-------------|------|-------|------|---------------|-------------------|
| $R_a, \Omega$ | 3.9 | 4.2729 | 5.0699 | 3.8012 | 2.5987 | 0.9953 |
| $I_v, \mu$H | 12 | 12.0817 | 15.5955 | 12.3957 | 3.1928 | 2.5904 |
| $J_m, \text{gcm}^2$ | 0.01 | 0.0100 | 0.0102 | 0.0100 | 0.0023 | $8.53 \times 10^{-7}$ |
| $B_m, \text{min}^{-1}/\text{mNm}$ | 78.224 | 79.296.35 | 101.192.53 | 82.628.59 | 5.3306 | $14,748.09$ |
| $k_v, \text{mNm/A}$ | 0.69 | 0.7559 | 0.8970 | 0.6725 | 2.5987 | 0.1761 |
| $k_e, \text{mV/min}^{-1}$ | 0.072 | 0.0720 | 0.0731 | 0.0721 | 0.1479 | 0.0009 |
| $f_s, \text{ITAE}$ | $2.22 \times 10^{-6}$ | 0.0002 | 7.1810^{-5} | — | — | — |
| Sim. time, min | — | 58.82 | 134.15 | 112.07 | — | — |
1.8–6.7%, for the winding inductance ($L_m$) 0.2–8.9%, for the moment of inertia ($J_m$) ($1.4 \times 10^{-6}$–0.02%), for the friction coefficient ($B_m$) 2.3–3.4%, for the torque coefficient ($k_t$) 1.8–6.7% and for the back-emf coefficient ($k_e$) 0.09–0.44%. It can be commented from the above that GWO does not perform quite well and for the back-emf coefficient ($k_e$). Also, IAE seems to be inappropriate choice as the problem’s objective function when GWO is adopted.

Table 9: GWO performance based on ISE criterion (through 20 independent runs)

| Quantity | Exact value | Best | Worst | Mean | Mean error, % | Standard deviation |
|----------|-------------|------|-------|------|---------------|-------------------|
| $R_m$, $\Omega$ | 3.9 | 3.0228 | 5.0254 | 3.8288 | 1.8598 | 1.0387 |
| $L_m$, $\mu$H | 12 | 12.3209 | 15.6000 | 13.1829 | 8.9734 | 1.9883 |
| $J_m$, gcm$^2$ | 0.01 | 0.0099 | 0.0100 | 0.0100 | $1.46 \times 10^{-4}$ | $4.70 \times 10^{-6}$ |
| $B_m$, min$^{-1}$/mNm | 78.224 | 77.765.49 | 99.008.97 | 80.958.49 | 3.3776 | 11.757.55 |
| $k_t$, mNm/A | 0.69 | 0.5348 | 0.8970 | 0.6774 | 1.8573 | 0.1837 |
| $k_e$, mV/min$^{-1}$ | 0.072 | 0.0719 | 0.0730 | 0.0720 | 0.0953 | $7.42 \times 10^{-4}$ |
| $f_I$, ISE | — | 0.0292 | 3.6195 | 0.7731 | — | 0.8442 |
| Sim. time, min | — | 57.97 | 90.73 | 74.54 | — | — |

Table 10: GWO performance based on IAE criterion (through 20 independent runs)

| Quantity | Exact value | Best | Worst | Mean | Mean error, % | Standard deviation |
|----------|-------------|------|-------|------|---------------|-------------------|
| $R_m$, $\Omega$ | 3.9 | 3.0047 | 5.0254 | 3.8288 | 1.8598 | 1.0387 |
| $L_m$, $\mu$H | 12 | 11.6469 | 15.4216 | 12.3171 | 2.5745 | 2.1344 |
| $J_m$, gcm$^2$ | 0.01 | 0.0100 | 0.0100 | 0.0100 | 0.0999 | 0.2000 |
| $B_m$, min$^{-1}$/mNm | 78.224 | 79.225.05 | 101.451.96 | 76.467.61 | 2.2969 | 13.467.51 |
| $k_t$, mNm/A | 0.69 | 0.5317 | 0.8970 | 0.7111 | 2.9744 | 0.1852 |
| $k_e$, mV/min$^{-1}$ | 0.072 | 0.0720 | 0.0731 | 0.0717 | 0.3597 | 0.0008 |
| $f_I$, ITAE | — | 0.0347 | 3.6195 | 0.7731 | — | 0.8442 |
| Sim. time, min | — | 57.97 | 90.73 | 74.54 | — | — |

Table 11: GWO performance based on ITAE criterion (through 20 independent runs)

| Quantity | Exact value | Best | Worst | Mean | Mean error, % | Standard deviation |
|----------|-------------|------|-------|------|---------------|-------------------|
| $R_m$, $\Omega$ | 3.9 | 3.7481 | 5.0317 | 4.1825 | 6.7542 | 1.0187 |
| $L_m$, $\mu$H | 12 | 12.0274 | 14.9451 | 11.9669 | 0.2759 | 1.5223 |
| $J_m$, gcm$^2$ | 0.01 | 0.0099 | 0.0100 | 0.0100 | 0.0138 | $1.07 \times 10^{-5}$ |
| $B_m$, min$^{-1}$/mNm | 78.224 | 77.587.02 | 101.069.35 | 75.590.51 | 3.4838 | 13.378.51 |
| $k_t$, mNm/A | 0.69 | 0.6630 | 0.8970 | 0.7399 | 6.7525 | 0.1802 |
| $k_e$, mV/min$^{-1}$ | 0.072 | 0.0719 | 0.0731 | 0.0716 | 0.4470 | 0.0008 |
| $f_I$, ITAE | — | 0.0347 | 3.6195 | 0.7731 | — | 0.8442 |
| Sim. time, min | — | 57.97 | 90.73 | 74.54 | — | — |

6 Conclusions

In this paper, a parameter identification approach of a PMDC motor was performed, by employing three recent nature-inspired meta-heuristic techniques. For this purpose, both the no-load and full-load speed responses were utilised and no simplifications made about the motor torque and back-emf coefficients which were treated unequal as this is true in practice. Three objective functions were examined and several simulations were conducted for each case. The general conclusion is that all of the algorithms can only, may not be a ‘safe’ way to argue about the robustness of the proposed meta-heuristic techniques to the specific problem. We remind here though, that all obtained errors are technically accepted and potentially a motor controller designer can use the results effectively.

On the other hand, the standard deviation measure helps significantly towards this direction. By observing Fig. 15b, it is seen that the SSA gives the smaller values for every each one of the motor parameters compared to ALO and GWO. This also happens if ISE is used as an objective function. The same is observed for the IAE and IAET with the exception of the motor inertia parameter ($J_m$).

Thus, it can be concluded that the SSA outperforms ALO and GWO in terms of the standard deviation of the problem search parameters, when accompanied by the ISE criterion. For completing the comparison, it is also seen that neither of ALO and GWO can take the ‘second’ place, since the corresponding performances vary.
potentially be successfully used in this or similar problems, because they succeeded to converge giving acceptable parameter values. It was also found that the percentage mean error cannot be safely used as an estimation metric. On the contrary, the estimated parameters standard deviation is proven as a more robust measure. Based on the total obtained results, the SSA presented the better performance compared to ALO and GWO, and seems to be a very promising optimisation technique for the DC motor parameter identification and relative problems. Nevertheless, the accurate characterisation of the DC motor allows, in turn, the ability to predict its performance over a wide range of operating conditions towards the easier design of relevant robust controllers.

7 References

[1] Unbehauen, H., Rao, G.P.: ‘A review of identification in continuous-time systems’, Annu. Rev. Control, 1998, 22, pp. 145–171
[2] Basilio, J.C., Moreira, M.V.: ‘State-space parameter identification in a second control laboratory’, IEEE Trans. Educ., 2004, 47, (2), pp. 204–210
[3] Mamani, G., Becedas, J., Bartle, V.C.: ‘On-line fast algebraic parameter and state estimation for a DC motor applied to adaptive control’. Proc. World Congress on Engineering, London, UK, 2–4 July 2008
[4] Becedas, J., Mamani, G., Felius, V.: ‘Algebraic parameters identification of DC motors: methodology and analysis’, Int. J. Syst. Sci., 2010, 41, (10), pp. 1241–1255
[5] Kmeta, R., Antic, S., Stujanovic, D.: ‘Recursive least square method in parameters identification of DC motors models’, Facta Univ., 2005, 18, (3), pp. 467–478
[6] Yousefi, I., Ghanbari, M.: ‘Parameter estimation of permanent magnet synchronous motor using orthogonal projection and recursive least squares combinatorial algorithm’, Math. Probl. Eng., 2015, 1, pp. 1–7
[7] Liu, X.Q., Zhang, H.Y., Liu, J., et al.: ‘Fault detection and diagnosis of permanent-magnet DC motor based on parameter estimation and neural network’, IEEE Trans. Ind. Electron., 2000, 47, (5), pp. 1021–1030
[8] Udomsuk, S., Arcerak, K.-L., Arcerak, K.-N., et al.: ‘Parameters identification of separately excited DC motor using adaptive tabu search technique’. Proc. Int. Conf. on Advances in Energy Engineering, Kottayam, India, 16–17 October 2010, pp. 48–51
[9] Wu, W.: ‘DC motor parameter identification using speed step responses’, Model. Simul. Eng., 2012, 2012, pp. 1–5
[10] Obiedat, M.A., Wang, L.Y., Lin, F.: ‘Real-time parameter estimation of PMDC motors using quantized sensors’, IEEE Trans. Veh. Technol., 2013, 62, (7), pp. 2977–2986
[11] Erdal, H., Dogan, B., Buldoo, A.: ‘Real-time parameter estimation, calibration and simulation of a DC motor’, Tech. Technol. Educ. Manage., 2011, 6, (3), pp. 606–614
[12] Galijasevic, S., Masic, S., Smaka, S., et al.: ‘Parameter identification and digital control of speed of a permanent magnet DC motors’. Proc. XXIII Int. Symp. Information, Communication and Automation Technologies, Sarajevo, Bosnia and Herzegovina, 27–29 October 2011, pp. 1–7
[13] Nolan, R., Pillay, P., Haque, T.: ‘Application of genetic algorithms to motor parameter determination’. Proc. IEEE Industry Applications Society Annual Meeting, Denver, CO, USA, 2–6 October 1994, pp. 47–54
[14] Dupuis, A., Gheibi, M., Kaddouri, A.: ‘Multiojective genetic estimation of DC motor parameters and load torque’. Proc. IEEE Int. Conf. on Industrial Technology (ICIT), Hammamet, Tunisia, 8–10 December 2004, vol. 3, pp. 1511–1514
[15] Sankardoss, V., Geethanjali, P.: ‘Parameter estimation and speed control of a PMDC motor used in wheelchair’, Energy Proc., 2017, 117, pp. 345–352
[16] Čapković, R., Kozaković, A., Minar, M., et al.: ‘Robust QFT-based control of the DC motor laboratory model’. Proc. of Cybernetics & Informatics Conf. (K&I), Velke Karlovice, Czech Republic, 29 January–1 February 2020, pp. 1–6
[17] Siala, H., Mhenni, F., Barkallah, M., et al.: ‘Variational approach for mechatronic tolerancing: application to a DC motor’, Mech. Ind., 2020, 21, (2), pp. 1–11
[18] Pillai, B.M., Suthakorn, J.: ‘Motion control applications: observer based DC motor parameters estimation for novices’, Int. J. Power Electron. Drive Syst., 2019, 10, (1), pp. 195–210
[19] Wolpert, D.H., Macready, W.G.: ‘No free lunch theorems for optimization’, IEEE Trans. Evol. Comput., 1997, 1, (1), pp. 67–82
[20] Sakhivel, V.P., Bhuvaneswari, R., Subramanian, S.: ‘Multi-objective parameter estimation of induction motor using particle swarm optimisation’, Eng. Appl. Artif. Intell., 2010, 23, (3), pp. 302–312
[21] Mughal, M., Khan, M., Shah, A., et al.: ‘Parameter estimation of DC motor using chaotic initialized particle swarm optimisation’. Proc. 3rd Int. Conf. on Electromechanical Control Technology and Transportation, Chongqing, China, 19–21 January 2018, vol. 1, pp. 391–395
[22] Mirjalili, S.: ‘Ant lion optimizer’. Eng. Comput., 2015, 31, (4), pp. 469–488
[23] Mirjalili, S.: ‘The ant lion optimizer’. Adv. Eng. Softw., 2015, 83, pp. 80–98
[24] Bonnans, J.F., Gilbert, J.C., Lemarechal, C., et al.: ‘Numerical optimization: theoretical and practical aspects’. (Springer-Verlag, Berlin Heidelberg, Germany, 2006)
[25] Karnavas, V.L., Psachaloglou, T.P., Mouroutos, S.G., et al.: ‘Application of salp swarm algorithm for DC motor parameter estimation in an industry 4.0 control systems IoT framework’. Proc. of 12th Int. Conf. on Developments in eSystems Engineering (DeSe), Kazan, Russia, 7–10 October 2019, pp. 778–783

Fig. 15 Comparative bar plots for the performance of the algorithms examined
(a) Normalised absolute mean error for ISE, IAE and ITAE criteria, (b) Normalised standard deviation for ISE, IAE and ITAE criteria
[27] Marlin, T.E.: ‘Process control: designing processes and control systems for dynamic performance’ (McGraw-Hill, New York, NY, USA, 1995), pp. 288–299

[28] Karnavas, Y.L., Chasiotis, I.D.: ‘PMDC coreless micro-motor parameters estimation through grey wolf optimizer’. Proc. XXII Int. Conf. on Electrical Machines (ICEM), Lausanne, Switzerland, 4–7 September 2016, pp. 865–870

[29] Faris, H., Aljarah, I., Al-Betar, M.A., et al.: ‘Grey wolf optimizer: a review of recent variants and applications’, Neural Comput. Appl., 2017, 30, (2), pp. 413–435

[30] Ali, E.S., Abd Elazim, S.M., Abdelaziz, A.Y.: ‘Optimal allocation and sizing of renewable distributed generation using ant lion optimization algorithm’, Electr. Eng., 2016, 100, (1), pp. 99–109

[31] Rajan, A., Jeevan, K., Malakar, T.: ‘Weighted elitism based ant lion optimizer to solve optimum VAr planning problem’, Appl. Soft Comput., 2017, 55, pp. 352–370

[32] Mani, M., Bozorg-Haddad, O., Chu, X.: ‘Ant Lion Optimizer Algorithm’, in Bozorg-Haddad, O. (ed.): ‘Advanced optimization by nature-inspired algorithms, studies in computational intelligence series’ (Springer Nature, Singapore Pte Ltd, Singapore, 2018), pp. 105–116

[33] Hussien, A.G., Hassanien, A.E., Houssein, E.H.: ‘Swarming behaviour of SALPS algorithm for predicting chemical compound activities’. Proc. 8th IEEE Int. Conf. on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 5–7 December 2017, pp. 315–320

[34] Ibrahim, H.T., Mazher, W.J., Ucan, O.N., et al.: ‘Feature selection using salp swarm algorithm for real biomedical datasets’, Int. J. Comput. Sci. Netw. Secur., 2017, 17, (12), pp. 13–20

[35] ‘Faulhaber DC-Micromotors 0615S series datasheet’. Available at https://www.faulhaber.com/en/products/series/0615s/ (last access on 25 May 2019)