Turk J Elec Eng & Comp Sci,
Vol.17, No.1, 2009, © TÜBİTAK
doi:10.3906/elk-0608-6

Softcomputing Identification Techniques of Asynchronous Machine Parameters: Evolutionary Strategy and Chemotaxis Algorithm

Nouri BENAÏDJA
Department of Electronics, Faculty of Engineer Sciences, University of Constantine Zarzara,
25000, Constantine, ALGERIA
e-mail: benaidja@yahoo.fr

Abstract

Softcomputing techniques are receiving attention as optimisation techniques for many industrial applications. Although these techniques eliminate the need for derivatives computation, they require much work to adjust their parameters at the stage of research and development. Issues such as speed, stability, and parameters convergence remain much to be investigated. This paper discusses the application of the method of reference model to determine parameters of asynchronous machines using two optimisation techniques. Softcomputing techniques used in this paper are evolutionary strategy and the chemotaxis algorithm. Identification results using the two techniques are presented and compared with respect to the conventional simplex technique of Nelder and Mead. Discussion about the chemotaxis algorithm as the most promising optimisation technique is presented, giving its advantages and disadvantages.

Key Words: *Asynchronous machine, Identification, Optimization, Softcomputing techniques, Evolutionary strategy, Chemotaxis algorithm.*

1. Introduction

Due to its simple and sturdy structure, the asynchronous machine has become an evitable part of the modern industrial drive system. Recently many techniques have been developed to control the asynchronous machine as efficiently as direct current machine. These techniques, however, rely on the accuracy of the machine parameters which are known to vary under different operating conditions. The use of incorrect parameters in controllers can result in error and improper dynamic behaviours [1]. Therefore, having accurate parameters of an asynchronous machine becomes essential to accomplish the desired dynamic performances under different operating conditions. However, it is well known that the rotor time constant is necessary for tuning the controller. Rotor parameter variations are well known and neglecting the variations can lead to poor performing control schemes. The accuracy of the estimated rotor flux is greatly influenced by the value of rotor resistance used for control. Rotor resistance may vary up to 100% due to rotor heating and recovering this information with a temperature model or a temperature sensor is not desirable. In addition rotor resistance can change significantly with rotor frequency due to skew/proximity effect in machines with double-cage and deep-bar rotors. It has also been shown that neglecting the iron loss resistance in control schemes results in a detuning of the drive. That is, neglecting the iron loss parameter inherently leads to error between the reference torque and the actual torque in the torque control scheme. More recently it has been shown that changes in the magnetizing parameters are critical for establishing self-excitation in self-excited induction generators. The stator leakage reactance is also known to vary with current [2].

In the literature, many works are devoted to softcomputing techniques for electric machines identification and control [3–7].

Bose et al. [4] have described fuzzy and nonfuzzy approaches for online stator-resistance estimation of an induction motor, where the resistance value is derived from stator-winding temperature estimation as a function of stator current and frequency through an approximate dynamic thermal model of the machine. The results of the estimation have been used in a stator-flux-oriented, sensorless, vector-controlled induction motor drive. Resistance pattern can be stored in a look-up table. A difficulty of the look-up table method is that it tends to give reduced resistance resolution unless the table is very large.

Min et al. [5] used the fuzzy estimation for tuning the stator resistance in direct torque control of induction machines. The fuzzy resistance estimation should provide better performance than the PI estimation.

A neural net-based inverse model of an induction motor is given in [6].

Benaïdja and colleagues [7] used genetic algorithms for the estimation of electrical and mechanical parameters of an induction motor. Evolutionary strategy is compared to the particle swarm algorithm for determining the parameters of linear and nonlinear models of two asynchronous machines in [8]; the best results are given by evolutionary strategy in 99% of the cases. Algorithmic parameters are determined by trial-and-error. Initial conditions are not studied. This study focused on finding the best parameters, but not on determining the best algorithm, to achieve this task. The later is typically the objective in a comparative study of the algorithms aiming at identification of machine parameters.

Among others, softcomputing techniques can be applied to neural networks and fuzzy logic paradigms, neighbourhood techniques, evolutionary techniques and bionics. The class of neighbourhood techniques encloses, essentially, simulated annealing and tabu search. Genetic algorithms, evolutionary programming, and evolutionary strategy are classified under the category of evolutionary techniques. In the class of bionics we find the chemotaxis algorithm, particle swarm, and immune system.

While the chemotaxis algorithm is known as an emergent optimisation technique, evolutionary strategy is generally preferable to genetic algorithms for solving problems that deal with the optimisation of functions of real numbers. Its main advantage over simulated annealing lies in its adaptive step length. Evolutionary strategy and chemotaxis algorithm have fundamental commonality: they are highly parallelized, they each evolve the random variation, reproduction and competition. These form the essential essence of evolution, and one these four processes are in place, whether in nature or in a computer, evolution is an evitable outcome [9].

In this work we propose two softcomputing techniques, evolutionary strategy and chemotaxis algorithm, for off-line identification of asynchronous machine parameters under no load conditions. In order to study the methods' behaviour toward the changes of the start time t_0, we choose the simplex technique of Nelder and Mead [10] as a reference.

2. Identification Method

The identification method of the reference model is based on the minimization of a performance criterion, generally a weighted cost function (objective function) C. Hence, experimental and computed outputs are used by optimization algorithm to adjust machine parameters iteratively. The procedure continues until there is no appreciable improvement in the objective function value. A block diagram of the general iterative loop with accompanying criterion and successive optimization forming the reference model is shown in Figure 1.

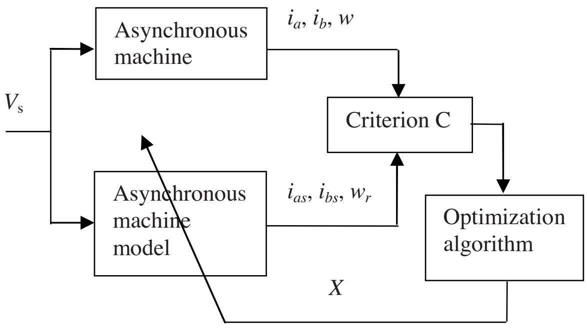


Figure 1. Block diagram of the identification method of reference model.

The main parts of the method are presented in the subsequent sections of this monograph.

2.1. Asynchronous machine model

Several models are considered suitable for asynchronous machines [11]. We select the most convenient for the vector control. Neglecting the magnetic saturation and stray losses, the mathematical model of asynchronous machine, referred to $\alpha\beta$ axes fixed with the stator, can be expressed by the following equations:

$$\dot{i}_s = \left[-R_e i_s + \left(I/\tau_r - w_r J\right)\phi_r + V_s\right]/L_e, \tag{1}$$

$$\dot{\phi}_r = \frac{L_s - L_e}{\tau_r} i_s - \left(I/\tau_r - w_r J\right)\phi_r, \tag{2}$$

$$w_r = \left(-f_v w_r + p\left(T_e - T_l - T_d\right)\right)/J_m, \tag{3}$$

$$T_e = p i_s^T J \phi_r, \tag{4}$$

$$T_d = f_d sgn\left(w_r\right), \tag{5}$$

$$V_s = \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix}, \quad i_s = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}, \quad \phi_r = \begin{bmatrix} \phi_{r\alpha} \\ \phi_{r\beta} \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

where V_s, i_s, Φ_r represent the stator voltage vector, stator current vector, and transformed rotor flux vector, respectively. T_e, T_l, w_r denote the electromagnetic torque, load torque, and rotor frequency, respectively. J_m, f_v, and f_d denote the rotor inertial moment, viscous friction coefficient, and dry friction coefficient, respectively. The equivalent resistance R_e, equivalent inductance L_e, and rotor time constant τ_r are related to the machine parameters as shown below:

$$R_e = R_s + R_r L_m^2/L_r^2, \quad L_e = L_s - L_m^2/L_r, \quad \tau_r = L_r/R_r, \tag{6}$$

where R_s and R_r are the stator resistance and rotor resistance, respectively. L_s, L_r, and L_m represent the stator self-inductance, rotor self-inductance, and magnetizing inductance, respectively, and p is the number of pole pairs.

The vector of machine parameters to be identified is: $X = [R_e L_e \ \tau_r L_s f_v f_d J_m]$. The number of simple cage three-phase asynchronous machine parameters is, then, $n = 7$.

Fourth order Runge Kutta method is used to determine the states of machine model, i.e. transformed rotor flux and stator currents, throughout a given interval between t_0 and $t_0 + \Delta t_f$. However, the initial conditions of these states at t_0 are required. Since $i_{\alpha s}$ and $i_{\beta s}$ are to be measured, one needs to calculate only $\Phi_{\alpha r}(t_0)$ and $\Phi_{\beta r}(t_0)$. Assuming that the machine is initially ($t = 0$) unexcited, it is easy to show that

$$\phi_r(t_0) = \int_0^{t_0} V_s dt - \left(R_e - \frac{L_s - L_e}{\tau_r}\right) \int_0^{t_0} i_s dt - L_e i_s(t_0). \tag{7}$$

The integrated quantities in (7) are to be calculated using the aforementioned integration techniques and are simply stored in memory for future reference, for example, a procedure of estimation of flux.

2.2. Cases of study

In order to test optimisation techniques, two three-phase asynchronous machines are used:

Machine-1 [12]: Machine-1 has the following characteristics:

The power rating $P = 0.63$ kw, supply voltage $V_s = 380$ V, nominal velocity $\Omega_r = 2900$ rpm, nominal power factor $\cos\phi_n = 0.737$, frequency $f = 50$ Hz, number of poles pairs $p = 1$. The machine is supplied by a reduced balanced system of sinusoidal voltages of an rms value $V_{seff} = 88$ V, load torque $T_l = 0$. The start time $t_0 = 0.71$ s.

Preliminary electrical parameters are obtained by the nameplate method [13]:

$$\sigma = \frac{L_e}{L_s} = \frac{1-\cos\varphi_n}{1+\cos\varphi_n}, w_g = w_n - \frac{2\pi p \Omega_r}{60}, \tau_r = \frac{1}{\sqrt{\sigma} w_g}$$

$$L_f = \frac{\sqrt{\sigma} V_s}{I_{sn} w_n}, L_r = L_f \frac{1-\sigma}{\sigma}, L_s = L_f + L_r \tag{8}$$

where σ is the dispersion coefficient, w_g is the slip pulsation.

By assuming the stator resistance $R_s = R_r$, subtraction of (6) from (8) leads to the preliminary values of electrical parameters: $R_{ep} = 8.612\ \Omega$, $L_{ep} = 39.2$ mH, $\tau_{rp} = 25.7$ ms, $L_{sp} = 259.2$ mH.

The mechanical parameters are determined separately, mostly by the slowing down method.

The search intervals of electrical and mechanical parameters $\Delta x_i = [x_{min,i} x_{max,i}]$, $i = 1, 2, \ldots, n$, are selected such that simulation variables are in the feasible region:

$\Delta R_e = [8.2\ 22.836]$, $\Delta L_e = [L_{ep}/3\ L_{ep}]$, $\Delta\tau_r = [\tau_{rp}/3\ 0.024]$,

$\Delta L_s = [L_{sp}/3\ 0.264]$, $\Delta f_v = [0\ 0.002]$, $\Delta f_d = [0\ 0.1]$, $\Delta J_m = [0.01\ 0.02]$.

Machine-2 [14]: Machine-2 has the following characteristics:

Power rating $P = 5.22$ kw, supply voltage $V_s = 220$ V, nominal velocity $\Omega_r = 1767$ rpm, frequency $f = 60$ Hz, number of poles pairs $p = 2$. The machine is supplied by a balanced system of sinusoidal voltages of the peak value $V_s = 380$ V, load torque $T_l = 0$. The start time is $t_0 = 0.05$ s.

Preliminary electrical parameters are obtained by classical tests (using DC step supply) [15]:

$R_{ep} = 1.835\ \Omega$, $L_{ep} = 40.3$ mH, $\tau_{rp} = 321$ ms, $L_{sp} = 175.2$ mH;

and

$\Delta R_e = [11.5\ 17.8777]$, $\Delta L_e = [0.01\ 0.0239]$, $\Delta\tau_r = [\tau_{rp}/3\ 3\tau_{rp}]$, $\Delta L_s = [0.55\ 1.4163]$,

$\Delta f_v = [0\ 0.0045]$, $\Delta f_c = [0\ 0.2872]$, $\Delta J_m = [0.001\ 0.01]$.

We notice that classical tests (no-load test and locked-rotor test) do not give good approximation for electrical parameters; except for rotor time constant.

2.3. Performance Criterion

In [16], Alonge et al. have tested using two performance criterions, quadratic criterion and absolute criterion. However, the results obtained with quadratic criterion are slightly better than those given by absolute criterion. At the time of our study, we have observed that both performance criterions give the same results. In order to lighten the representation in term of digits, the reported results are referred to absolute criterion

$$C = \sum_{j=1}^{K} [k_a \left|(i_{aj} - i_{asj})\right| + k_b \left|(i_{bj} - i_{bsj})\right| + k_\omega \left|(\omega_j - \omega_{rj})\right|], \tag{9}$$

where i_{aj}, i_{bj}, and w_j denote the measured variables; i_{saj}, i_{sbj}, and w_{rj} denote the variables obtained by simulation. These quantities are useful for the vector control. k_a, k_b, k_w are weights.

In [16] the weights $k_a = 1$, $k_b = 1$, $k_w = 0.5$, 1, 2 are tested, the best results are obtained with $k_w = 0.5$. By examining the absolute values of measured variables; this value seems to be an equitable weighting of currents and velocity. Then, we choose the values of weights: $k_a = 1$, $k_b = 1$, $k_w = 0.5$.

2.4. State of the art

2.4.1. Evolutionary strategy (ES)

Evolutionary strategy is an optimization method, inspired from the notion of biological evolution by means of natural selection [17]. The evolutionary strategy procedure can be summarized in the following steps.

a) Initialize population

The initial population is carried out picking μ individuals at random from the search space $\Delta X = [\Delta R_e \Delta L_e \Delta\tau_r \Delta L_s \Delta f_v \Delta f_c \Delta J_m]^T$; where the search intervals are determined according to the case of study. The initial parent population, $\{a^k\} = \{(x,\sigma)^k\}$ for $k = 1,2,\ldots,\ \mu$ is defined by

$$x_i^k = x_{Min,i} + rand\,(x_{Max,i} - x_{Min,i}), \tag{10}$$

$$\sigma_i^k = \left| x_i^k - \left(x_{Min,i} + \frac{x_{Max,i} - x_{Min,i}}{2} \right) \right| \frac{1}{\sqrt{n}}, \tag{11}$$

where$x^k = [R_e^k L_e^k \tau_r^k L_s^k \ f_v^k f_c^k J_m^k]$ is the vector of means and σ^k is the vector of standard deviations of the individual a^k. *rand* denotes a random variable of uniform distribution in the interval [0,1] that is sampled for each parameter i ($i = 1,2,\ldots,n$) which the upper and lower boundaries are $x_{max,i}$ and $x_{min,i}$ respectively. The environment delivers a quality metric (fitness value c^k) for each search point k ($k = 1,2,\ldots,\mu$).

b) Recombination

The recombination mechanism allows for mixing of parental information while passing it to their descendants. Many schemes of recombination exist and are being used in evolutionary strategies. The strategy called generalized intermediate panmictic recombination consists in taking one individual from the parent population and holding it fixed while other parents are chosen to recombine with it. The number of parents chosen for recombination, with each individual, is equal to ρ. The process of recombination is mathematically expressed as:

$$x_i' = x_{R,i} + U_i\,(0,1)\,(x_{Q_i,i} - x_{R,i})\,, \tag{12}$$

$$\sigma_i' = \sigma_{R,i} + U_i\,(0,1)\,(\sigma_{Q_{i*},i} - \sigma_{R,i})\,, \tag{13}$$

where R and Q denote two randomly chosen individuals from the parent pool. Q_i and Q_{i*} denote that the parent Q is to be sampled for each parameter individually. $U_i(0,1)$ is an uniform random number in the range [0 1]. The number of the candidate parents for the next generation is then: $\lambda = \rho \cdot \mu$.

c) Mutation

According to the biological observation that offspring are similar to their parents and that small changes occur more often than large ones, the mutation operator for the i^{th} parameter expressed in mathematical terms is defined as:

$$x_i' = x_i + \sigma_i' N_i\,(0,1)\,, \tag{14}$$

$$\sigma_i' = \sigma_i \exp\left[\tau' N\,(0,1) + \tau N_i\,(0,1)\right], \tag{15}$$

where $N(0,1)$ denotes a random variable of normal distribution with zero mean and standard deviation 1, sampled just once per recombination procedure. Analogously, $N_i(0,1)$ denotes a random variable of normal distribution with zero mean and standard deviation 1 which is to be sampled for each parameter individually. The values of τ and τ ' appear to be rather robust and they can be picked as:

$$\begin{cases} \tau \propto \left(\sqrt{2\sqrt{n}}\right)^{-1} \\ \tau' \propto \left(\sqrt{2n}\right)^{-1} \end{cases}, \tag{16}$$

where the proportionality constants are usually one. To guarantee a minimum of variation in the mutation of the parameters, all standard deviations σ_i are required to remain above a certain threshold. This minimum deviation should be expressed as a percentage of the parameter value, that is,

$$\sigma_i \geq \varepsilon\,|x_i|\,. \tag{17}$$

d) Selection

Since the offspring population is larger than the parent population ($\lambda > \mu$), a mechanism has to be implemented that allows us to select which offspring will conform the new parent population. Mainly, we have to distinguish between so-called elitist and non-elitist selection operators.

More refined selection methods have been introduced, out of which, the fitness-based-reinsertion combined with the elitist selection is recommended. In this method, the λ offsprings generated through recombination and mutation from the parents are ranked in terms of their fitness and the best μ - γ ($\gamma < \mu$) are selected to become part of the next parent generation. The remaining γ slots in the parent population to complete the μ individuals required are filled by the best parents of the older generation, which are retained so that the previous information is not completely lost in one evolution step.

This technique presents in intermediate scenario between elitist and non-elitist selection and it aims to capture the best of both worlds, allowing the new offsprings to substitute most of the old parents but keeping the best of the former to enrich the genetic pool.

e) Termination

Steps b to d are repeated until there is no appreciable improvement in the objective function value.

2.4.2. Chemotaxis algorithm (CA)

The biologically inspired optimization technique of chemotaxis algorithm is proposed by analogy to the way bacteria react to chemoattractants in concentration gradients.

There is a great diversity of strategies for foraging and survival at the bacterial level [18]. Particularly, an E-coli bacterium is conspicuous by his thermotaxis and phototaxis capabilities.

E-coli bacteria swarm foraging can be modelled for optimization by the following steps.

a) Initialization

A population of μ bacteria is randomly and uniformly created. Each bacterial cell is defined by its position (vector of machine parameters) $x^i = [R_e^i L_e^i \tau_r^i L_s^i f_v^i \ f_c^i J_m^i]$ and his nutrient surface (objective function value C).

b) Chemotaxis

Define chemotaxis step to be a tumble followed by a tumble or a tumble followed by a run. The new location of the cell is then:

$$x^i(k+1) = x^i(k) + c(k)\,\Phi^i(k)\,, i = 1, 2, \ldots, \mu, \tag{18}$$

where $c(k)$ is the size of the step taken in the random direction specified by unit length random direction $\Phi(k)$.

The cell-to-cell signalling, for the i^{th} cell, is represented as

$$C_{cc}\left(x,x^{i}\right) = \sum_{\substack{k=1\\k\neq i}}^{s}\left[-d_{attract}\exp\left[-w_{attract}\sum_{j=1}^{n}\left(x_{j}^{k}-x_{j}^{i}\right)^{2}\right]\right] \tag{19}$$

$$+\sum_{\substack{k=1\\k\neq i}}^{s}\left[-h_{repellent}\exp\left[-w_{repellent}\sum_{j=1}^{n}\left(x_{j}^{k}-x_{j}^{i}\right)^{2}\right]\right], \tag{20}$$

where $d_{attract}$ and $w_{attract}$ denote the depth and width of the attractant signal, respectively, and $h_{repellent}$ and $w_{repellent}$ denote the height and width of the repellent.

The swarming effect is then released by the health function (instead of objective function C):

$$C_{h}\left(x^{i}\right) = C\left(x^{i}\right) + C_{cc}\left(x,x^{i}\right). \tag{21}$$

Bacterium cell make N_s runs before a tumble. The tumble is anticipated exactly when a degradation of objective function value is observed. So that the cells will try to find nutrients, avoid noxious substances, and at the same time try to move toward other cells but not too close to them.

c) Reproduction

After N_c chemotactic steps, a reproduction step is taken.

Let N_r be the number of reproduction steps to be taken. For convenience, we assume that μ is a positive even integer. Let: $\mu_r = \mu/2$, the number of population members who have had sufficient nutrients so that they will reproduce (split in two) with no mutations. For reproduction, the population is sorted order of ascending accumulated health (higher accumulated health represents that a bacterium did not get as many nutrients during its lifetime of foraging and hence is not as healthy and thus unlikely to reproduce), then the μ_r lest healthy bacteria die and the other S_r healthiest bacteria each split into two bacteria, which are placed at the same location.

d) Elimination dispersal event

Let N_{ed} be the number of elimination-dispersal events and, for each elimination dispersal event, each bacterium in the population is subjected to elimination-dispersal (death, then random placement of a new bacterium at a random location on the optimization domain) with probability p_{ed}.

e) Termination

Steps b to d are repeated until there is no appreciable improvement in the objective function value.

3. Identification Results

Identification is performed in visual C++ 6 compiler running in 1.7-GH Pentium-based-PC. Sampling frequency is 2 kHz.

Block diagram of the experimental data acquisition is given in Figure 2. SYNCHR block generates a 12 kHz synchronisation signal which acts on a digital input of an A/D converter and constitutes the time base for PC external operations. More precisely, when the above signal commutes from low to high level the PC acquires signal. Voltage signals in the range [-10, 10] V are acquired by means of three resistive voltage dividers (P1, P2, P3), realised with resistors having accuracy equal to 0.1%. Two stator currents are acquired by means of two Hall transducers (H1, H2) which generate voltage signals in the range [-5, 5] V. DC tachometer generates a voltage proportional to the velocity which, by means of a calibrated resistive voltage divider (P4), is converted into a signal in the range [-10, 10] V, and then acquired.

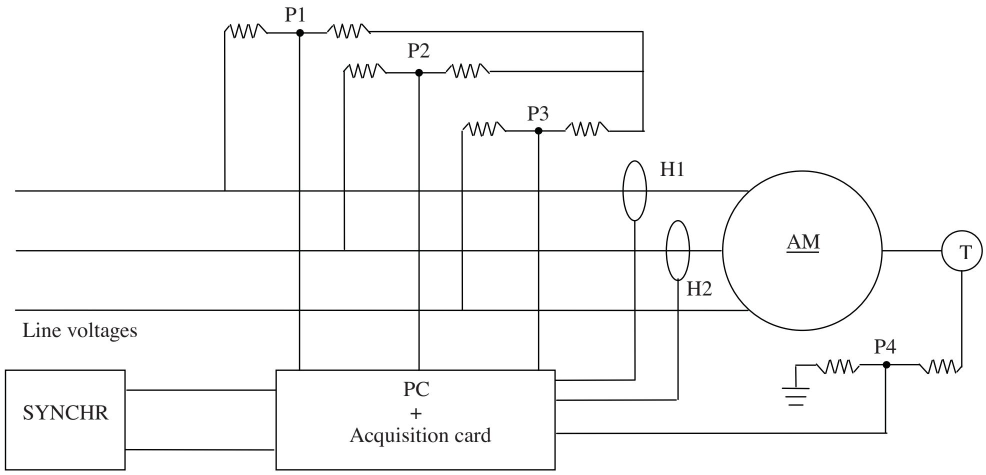


Figure 2. Basic scheme of the experimental equipment.

The results of identification of Machine-1 and Machine-2 using different optimisation techniques are represented in Table 1 and Table 2, respectively.

Table 1. Results of parameter identification via Machine-1.

	Nelder & Mead	Evolutionary Strategy	Chemotaxis Algorithm
equivalent resistance R_e (Ω)	18.25367	8.200000	8.200000
equivalent inductance L_e (H)	0.040094	0.039004	0.039000
rotor time constant τ_r (s)	0.024731	0.024004	0.024346
tator self-inductance L_s (H)	0.150972	0.228760	0.229000
viscous friction coefficient f_v (Nms)	0.000876	0.000000	0.000010
dry friction coefficient f_d (Nm)	0.001604	0.000000	0.000010
rotor inertial moment J_m (Nms^2)	0.015701	0.010001	0.010000
Cost$C$	470.7807	376.4795	376.0273
$Time(s)$	2	10	81

It is observed that for Machine-1, where the value of the start time t_0 is high corresponding to the steady state, the results given by the chemotaxis algorithm are better than those given by the evolutionary strategy and Nelder and Mead method in term of the best cost over the population at the time of convergence. Their resulting electrical parameters are close to those given by the nameplate method. The difference lies in the values of mechanical parameters.

Table 2. Results of parameter identification via Machine-1.

	Nelder & Mead	Evolutionary Strategy	Chemotaxis Algorithm
equivalent resistanceR_e (Ω)	12.77727	11.50000	11.50000
equivalent inductance L_e (H)	0.010010	0.010000	0.039000
rotor time constant τ_r (s)	0.282843	0.344114	0.963000
tator self-inductance L_s (H)	0.900482	0.807952	0.550000
viscous friction coefficient f_v (Nms)	0.003796	0.000734	0.000000
dry friction coefficient f_d (Nm)	0.232700	0.069929	0.000000
rotor inertial moment J_m (Nms^2)	0.006924	0.006066	0.001650
Cost$C$	2362.479	2376.203	2370.640
$Time(s)$	< 1	1	8

For Machine-2, where the value of the start time t_0 is low; corresponding to the transient regime, the best results in term of the value of objective function are given by the simplex technique.

Conversely to the analytical techniques, whose high number of experimental magnitudes (equal or up to 200 [19]) are required to obtain stable results, much less number of experimental data are needed for the convergence of the tested population based techniques. This observation is in accordance with the results obtained with genetic algorithm [20].

3.1. Initial population

Ten uniformly distributed initial populations of Machine-1 and Machine-2 parameters are tested for each optimisation technique. Biased populations of

ten different distributions of populations concentrated in the half of the search intervals of all the parameters ($1/2n$ of the search space);

ten different distributions of populations concentrated in the quarter of the search intervals of all the parameters ($1/4n$ of the search space),

are also tested. It has been observed that, generally, evolutionary strategy gives better results with biased (or even concentrated) initial population than with uniformly distributed initial population. This is due to the extrapolation capability of the technique. Conversely, the chemotaxis algorithm gives better results with uniformly distributed initial population than with biased initial population; because bacterium cells have tendency rather to gathering.

The maximum variations from each parameter of the individual that give the low value of objective function x_{ibest} $(i = 1, 2, \ldots, n)$ could be calculated as:

to the lower limit:

$$dx_i = \frac{x_{ibest} - x_{i\min}}{x_{ibest}} 100, \tag{22}$$

to the higher limit:

$$Dx_i = \frac{x_{i\max} - x_{ibest}}{x_{ibest}} 100. \tag{23}$$

Variations obtained are shown in Table 3 and Table 4 for machine-1 and Machine-2, respectively.

Table 3. Variation of the resulting Machine-1 parameters vs. distribution of the initial population.

	Nelder & Mead	Evolutionary Strategy	Chemotaxis Algorithm
equivalent resistance, variation to the lower boundary dR_e	47	0	0
equivalent resistance, variation to the upper boundary DR_e	33	29	2
equivalent inductance variation to the lower boundary dL_e	0	0	0
equivalent inductance variation to the upper boundary DL_e	86	18	0
rotor time constant, variation to the lower boundary $d\tau_r$	0	0	1
rotor time constant, variation to the upper boundary $D\tau_r$	84	100	0
viscous friction coefficient, variation to the lower boundary df_v	46	28	43
viscous friction coefficient, variation to the upper boundary Df_v	34	13	1
dry friction coefficient, variation to the lower boundary df_d	19	0	0
dry friction coefficient, variation to the upper boundary Df_d	56	72	50
rotor inertial moment, variation to the lower boundary dJ_m	0	0	0
rotor inertial moment, variation to the upper boundary DJ_m	93	90	0
Cost, maximal variation dC	1	20	14

We notice that the chemotaxis algorithm is the most robust against changes of distribution of the initial population. However, with evolutionary strategy and for high value of the start time, electrical parameters converge toward the higher limits of the search intervals, where the decrease of the value of objective function is induced. Besides, with evolutionary strategy, and for low value of the start time, the important variations of the mechanical parameters do not cause noticeable variation of the value of objective function; the problem of sensitivity comes up in this case.

It is worth mentioning that there are almost no variations of equivalent resistance and equivalent inductance, versus changes of the distribution of initial population, as they induce the greediness of the algorithms.

3.2. Tuning of parameters of optimisation techniques

3.2.1. Nelder and Mead simplex method (NM)

Ten initial values are allotted to the algorithm, the best results are recorded.

Machines and computed outputs obtained with the parameters giving the lower values of objective functions with start times t_0 of 0.71 s and 0.05 s for Machine-1 and Machine-2, respectively, are shown in Figures 3.

Table 4. Variation of the resulting Machine-1 parameters vs. distribution of the initial population.

	Nelder & Mead	Evolutionary Strategy	Chemotaxis Algorithm
equivalent resistance, variation to the lower boundary dR_e	80	0	0
equivalent resistance, variation to the upper boundary DR_e	10	0	0
equivalent inductance variation to the lower boundary dL_e	100	0	0
equivalent inductance variation to the upper boundary DL_e	0	0	0
rotor time constant, variation to the lower boundary $d\tau_r$	59	50	50
rotor time constant, variation to the upper boundary $D\tau_r$	20	25	3
viscous friction coefficient, variation to the lower boundary df_v	60	50	1
viscous friction coefficient, variation to the upper boundary Df_v	31	29	1
dry friction coefficient, variation to the lower boundary df_d	16	100	0
dry friction coefficient, variation to the upper boundary Df_d	80	0	0
rotor inertial moment, variation to the lower boundary dJ_m	9	100	0
rotor inertial moment, variation to the upper boundary DJ_m	81	0	17
Cost, maximal variation dC	34	50	2

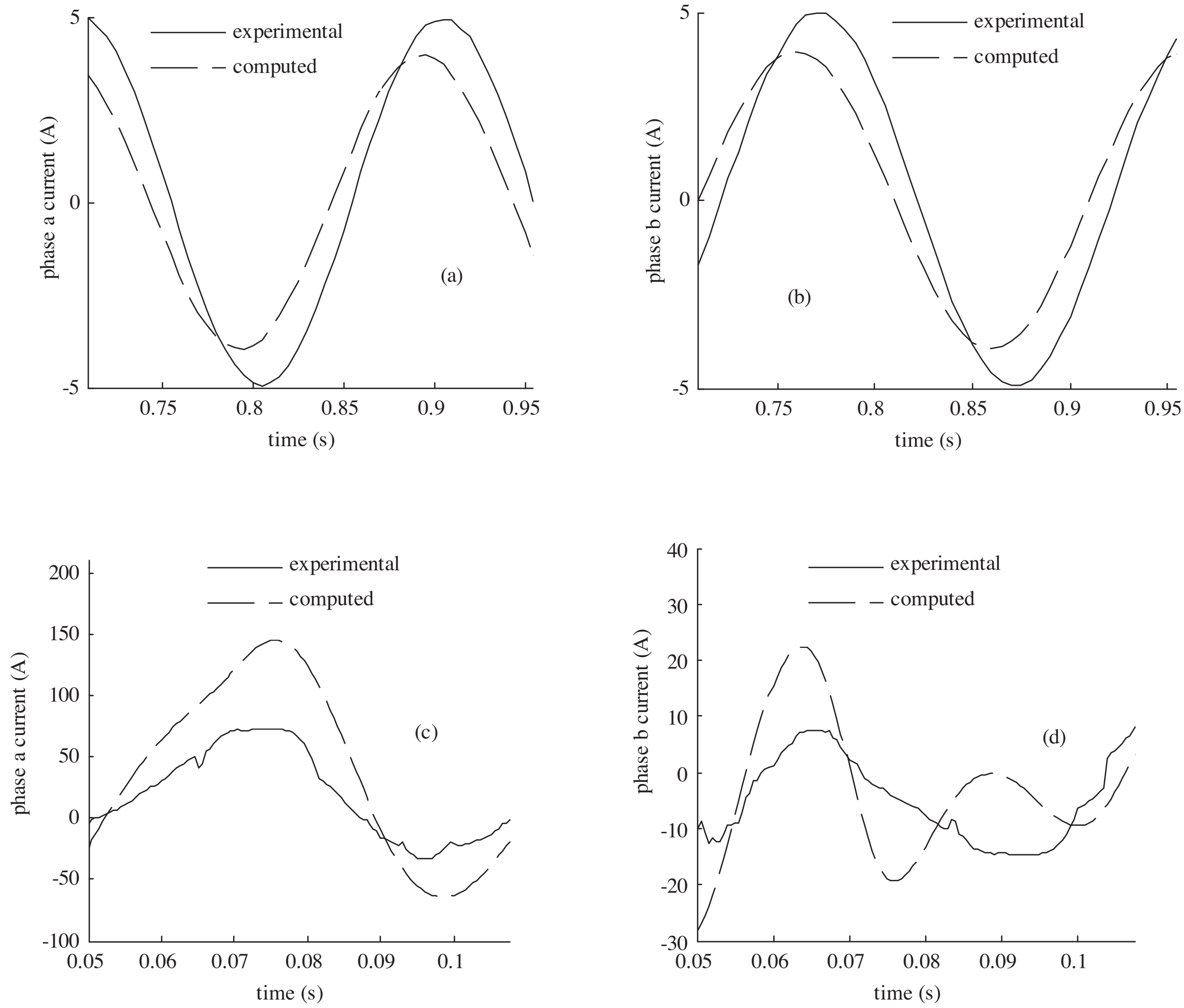
Although it is known that the Nelder and Mead method does not work well in many situations, no problem of numerical stability was shown.

3.2.2. Evolutionary strategy (ES)

The algorithm of evolutionary strategy has been tested using several combinations: populations of different sizes (μ = 10–90, with a resolution of 10), and number of parents chosen for recombination ($\rho = 1, \ldots, \rho = 10$, $\rho = 20, \ldots, \rho = 100$, logarithmic scaling).

Machines and computed outputs obtained with the parameters giving the lower values of objective functions with start times t_0 of 0.71 s and 0.05 s for Machine-1 and Machine-2, respectively, are shown in Figures 4.

For Machine-1, the value of the start time t_0 is high. It is observed that the most accurate parameters are given with an intermediate size of the population $\mu = 100$, essentially with uniformly distributed initial population. Biased initial populations required larger size $\mu = 200$, which is in accordance with [21]. The number of parents chosen for recombination ρ change from 3 to 60. The values of the deviations σ_i^k ($i = 1, 2, \ldots, n$; $k = 1, 2, \ldots, \mu$) evolve quickly during the optimisation process and they will fine-tune themselves, making the choice of their initial values not of critical importance.



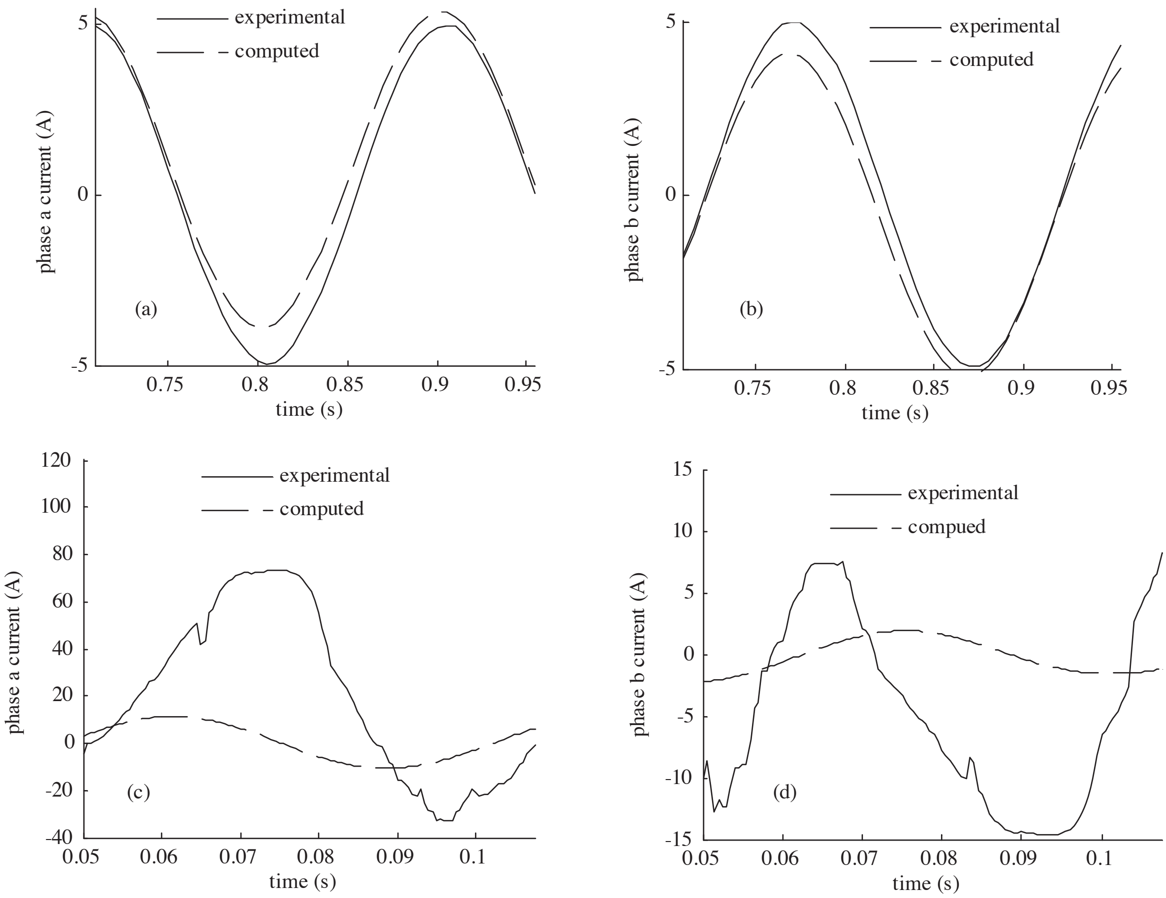
Figures 3. Machines' currents resulting from identification using Nelder and Mead method: (a), (b) currents from Machine-1; (c), (d) currents from Machine-2.

For Machine-2, the value of the time t_0 is low, and the choice of the population size μ and number of parents chosen for recombination ρ become more significant. However, it is observed that the quality of solution improve with the population size μ until a certain value changing from 20 to 70, according to the distribution of initial population. Poor results appear upwards from this population size. Optimal parameters of the algorithm are obtained with a number of parents chosen for recombination ρ proportional to the population size μ: high number of parents ρ with high population size μ, and vice-versa.

It is worth mentioning that an elitism number γ up to 3 causes the best cost to decrease considerably, since there are not enough new individuals in the population.

3.2.3. Chemotaxis algorithm (CA)

Machines and computed outputs obtained with the parameters giving the lower value of objective function with start times t_0 of 0.71s and 0.05s for Machine-1 and Machine-2, respectively, are shown in Figures 5.



Figures 4. Machines' currents resulting from identification using evolutionary strategy: (a), (b) currents from Machine-1; (c), (d) currents from Machine-2.

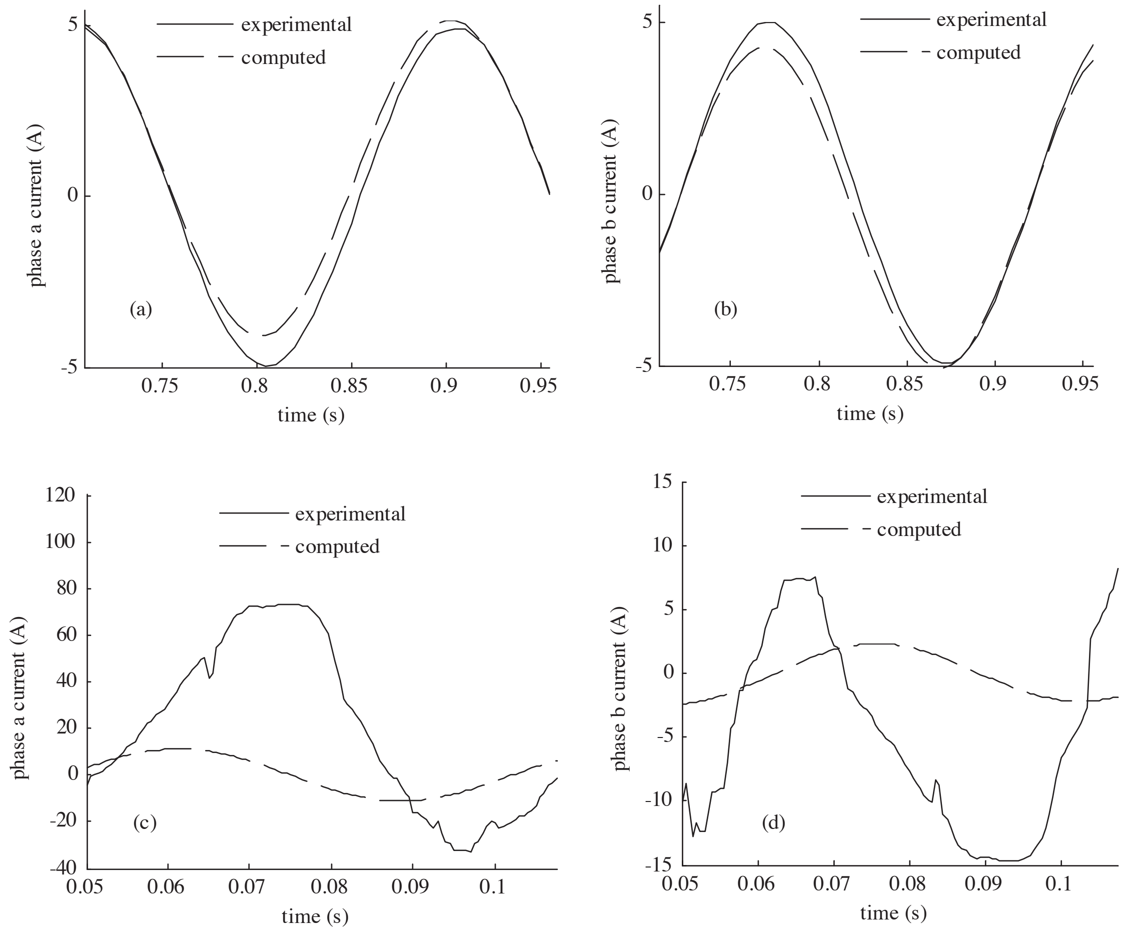
Usually the population size μ is chosen to have a large value, but it has to be kept within a reasonable level to prevent excessive enumeration [22]. Algorithm performance is greatly affected by the characteristics of swarming. Indeed, a small number of runs before a tumble N_s makes foraging a random superficial search through the search landscape. When the number of runs N_s increases, bacterial cells penetrate along the nutrient profile and, thus, the value of cost function improves. If the value of the number of runs N_s is too high, the algorithm could become unstable [23], in the sense that the performance criterion begins oscillating around some local minimum, which make the parameters not settle to their final values. The optimal value of the number of runs N_s ranges between 5 and 15; according to initial population.

The step sizes are:

for Machine-1: c = [0.01 0.01 0.002 0.002 0.0002 0.002 0.002];

for Machine-2: c = [0.05 0.0025 0.025 0.005 0.0005 0.025 0.00025].

It is interesting to note the relative behaviour of the number of tumbles before a reproduction step N_c and the best cost over the population at the time of convergence, for Machine-2. When the number of chemotactic steps N_c is below 10, the best cost increases by 2%. If the number of chemotactic steps N_c is chosen up to 10, the best cost increases by 7%. On average, bacteria find the best path within 10 tumbles; after this, bacteria follow less nutrient landscapes without sharing memory of the previous information.



Figures 5. Machines' currents resulting from identification using chemotaxis algorithm: (a), (b) currents from Machine-1; (c), (d) currents from Machine-2.

The number of reproduction steps before an elimination-dispersal step N_r reflects the anxiousness of the environment. A large value for the number of reproduction steps N_r did not improve performances of the algorithm, which converge slowly, due to presence of predators. Predator action originates in chaotic attractors [24]. However, the poor information offered by Machine-1 causes a large life ratio and, hence, the number of reproduction steps N_r in the surroundings of 50. This value ranges between 3 and 5 for Machine-2.

Start Here Next It is observed that high values for the depth of the attractant released by cell $d_{attract}$, as a low value of the height of the repellent$h_{repellent}$, improve the cost function value and reduce the execution time. Then we can conclude that the social behaviour of bacteria foraging contributes to the enhancement of the performances of the algorithm. In [25], the author suggests the initial value of the function representing the combined cell-to-cell signalling C_{cc} in the range of 10% of the value of the health function C_h. This is done by adjusting the widths $w_{attract}$ and $w_{repellent}$. Following this recommendation, it is observed that, as time goes on, bacteria are gathered and, hence, health function C_h is asymptotically reduced to the cost function C. The best results are given with $d_{attract} = 10^{-3.5}$, $h_{repellent} = 20 \cdot 10^{-3.5}$, for machine-1, $d_{attract} = 10^{-9.8}$, $h_{repellent} = 20 \cdot 10^{-9.8}$ for machine-2, and $w_{attract} = 0.2$, and $w_{repellent} = 1$ for both machines.

Bearing in mind that healthiest bacteria are not obligatorily the fittest bacteria, fluctuations nonetheless appeared in the best cost. In the aim to eliminate the resulting fluctuations, an elitism procedure is adopted. It consists of substituting one of the two worst elements resulting from the elimination-dispersal operator by the best element of the preceding generation. Since the elitism procedure has taken place after the elimination-dispersal step, the whole composition of the population of cells will not be altered.

4. Conclusion

This paper investigates the effect of the start time on the performances of optimisation techniques to develop the highest possible benefit from the information of experimental magnitudes issued from the machine.

Various results were found for two asynchronous machines with different rating powers and for different start times. For a high value of the time t_0, i.e. close to permanent state, the chemotaxis algorithm is the greedier. The large inrush of currents at start-up, corresponding to a low value of the time t_0, permits the chemotaxis algorithm to exhibit more mature performances; and the tuning of the parameters of both population-based optimisation techniques become very easy.

References

[1] R. Krishnan and A. S. Baharwaj, "A review of parameter sensitivity and adaptation in indirect vector controlled induction motor drive system", IEEE Transactions on Power Electronics, Vol. 6, No. 4, pp. 695–703, October 1991.

[2] D. J. McKinnon, D. Seyoum and C. Grantham, "Improved parameter characteristics for induction machine analysis and control", Proceedings of the Australasian Universities Power Engineering Conference, AUPEC2003, 28 Septembre-1 Octobre 2003, Christchurch, New Zealand.

[3] K. B. Bose, "fuzzy logic and neural networks", IEEE Industry Applications Magazine, pp. 57–63, May/June 2000.

[4] K. B. Bose and N. R. Patel, "Quazy fuzzy estimation of stator résistance of induction motor", IEEE Transactions on Power Electronics, Vol. 13, No. 3, pp. 401–409, May 1998.

[5] S. Min, M. E. Elbuluk, and D. S. Zinger, "PI and fuzzy estimators for tuning the stator resistance in direct torque control of induction machines", IEEE Transactions on Power Electronics, Vol. 13, No. 2, pp. 279–287, March 1998.

[6] P. Idowu, "Robust neural net-based inverse model identification of an induction motor", Electric Machines and Power Systems, Vol. 27, No. 5, pp. 513–525, May 1999.

[7] N. Benaïdja, D. Slimani, and N. Khenfer, "Genetic algorithms identification of induction motor parameters", Proceedings of the International Aegean Conference on Electrical Machines and Power Electronics (ACEMP2001), IEEE Industry Applications Society, Kuşadasi, Turkey, pp. 295–298, June 2001.

[8] R. K. Ursem and P. Vadstrup, "Parameter Identification of Induction Motors using Stochastic Search Algorithms". Applied Soft Computing, Vol. 4, No. 1, pp. 49–64, 2004.

[9] D. B. Fogel, Handbook of evolutionary computation, IOP Publishing LTD and Oxford University Press, 1997.

[10] J. A. Nelder and R. Mead, "A simplex method for function minimization", computer journal, 7, pp. 308–313, 1965.

[11] F. Córcoles, J. Pedra, and M. Salichs, “Numerical identifiability of the parameters of induction machines”, Proceedings of the International Conference on electrical Machines (ICEM2000), ESPOO, Finland, pp. 9–13, August 2000.

[12] M. Biriescu, V. Groza, V. Cretu, M. Mot, and G. Modescu, “Testing of electrical machines using a data acquisition and processing system”, Proceedings of the Workshop of electrical machines parameters, Napoca, Romania, pp. 31–34, May 2001.

[13] H. Razik and A. Rezzoug, “Identification of electrical parameters of an induction motor, a comparison between R.L.S and genetic algorithm”, International Conference on electrical Machines (ICEM2000), ESPOO, Finland, pp. 1433–1437, August 2000.

[14] S. Moon, A. Keyhani, and S. Pillutla, “Nonlinear neural network modelling of induction machine”, IEEE Transactions on Control Systems Technology, Vol.7, pp. 203–211, March 1999.

[15] A. B. Proca and A. Keyhani, “Induction motor parameter identification from operating data for electric drive operation”, Proceedings of the18th DASC, St. Louis, Missouri, October 1999.

[16] F. Alonge, F. D’ippolito, and F. M. Raimondi, “Parameter identification of induction motors: least squares vs. genetic algorithms”, International Workshop on Motion Control, Grenoble, France, pp. 1112–1122, September 1998.

[17] T. P. Runarsson, “An Asynchronous Parallel Evolution Strategy”, International Journal of Computational Intelligence and Applications, Vol. 3, No. 4, pp. 381-394, 2003.

[18] S. D. Müller, J. Marchetto, S. Airaghi, and P. Koumoutsakos, “Optimisation based on bacterial chemotaxis”, IEEE Transactions on Evolutionary Computation, Vol. 6, No. 1, pp. 16–29, February 2002.

[19] J. P. Da Cunha, G. Champenois, and J. D. Gabano, “Comparison of parameter estimation algorithms for continuous time models applied to D. C. motor thermal supervision”, *Electric Machines and Power Systems* 1998, Vol. 26, pp. 839–854, 1998.

[20] Ž. Ferkov and L. Zboray, “Contribution to parameter identification of induction motor by genetic algorithm” *Acta Electrotechnica et Informatica*, Vol. 5, No. 2, 2005.

[21] S. D. Müller, N. Hansen, and P. Koumoutsakos, “Increasing the serial and the parallel performance of the CMA-evolution strategy with large populations”, Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature (PPSN VII), Berlin, Germany, pp. 422–431, 2002.

[22] K. M. Passino, “Biomimicry of bacterial foraging for distributed optimisation and control”, IEEE Control Systems Magazine, Vol. 22, No. 3, pp. 52–67, June 2002.

[23] V. Gazi and K. M. Passino, “Stability analysis of swarms”, IEEE Transactions on Automatic Control, Vol. 48, No. 4, pp. 692–697, April 2003.

[24] C. A. Gaizares and W. Rosehart, “Bufirication analysis of induction motor loads for voltage collapse studies”, Proceedings of the north American power symposium (NAPS), MIT, pp. 559–565, November 1996.

[25] L. Yanfei and K. M. Passino, “Biomimicry of social foraging behavior for distributed optimization: models, principles, and emergent behaviors”, Journal of Optimization Theory and Applications, Vol. 115, No. 3, pp. 603–628, December 2002.