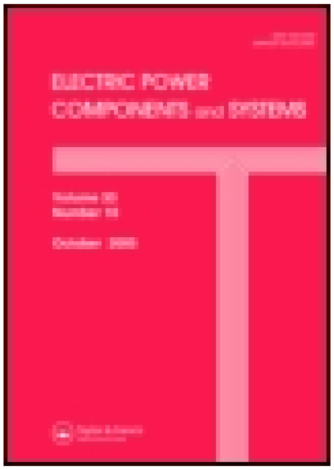


Electric Power Components and Systems
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uemp20

Identification of Asynchronous Machine Parameters by Evolutionary Techniques

N. Benaïdja [a] & N. Khenfer [a]

[a] Department of Electronics, Faculty of Engineer Sciences , University of Setif , Setif, Algeria

Published online: 23 Feb 2007.

Electric Power Components and Systems, 34:1359–1376, 2006

ISSN: 1532-5008 print/1532-5016 online
DOI: 10.1080/15325000600748897

Identification of Asynchronous Machine Parameters by Evolutionary Techniques

N. BENAÏDJA
N. KHENFER

Department of Electronics
Faculty of Engineer Sciences
University of Setif
Setif, Algeria

Four evolutionary techniques (scatter search, evolutionary programming, ant colony, and particle swarm algorithms) were used for off-line identification of three phase asynchronous machine parameters. Optimization techniques were then tested on two distinct machines. In order to evaluate how much good the achieved machines parameters obtained, experimental and simulation input-output behaviors are presented for each method. The performances in term of objective function and convergence time prove the effectiveness of this class of optimization methods.

Keywords asynchronous machine, identification, optimization, evolutionary techniques

1. Introduction

The AC asynchronous machine is clearly the choice drive for a significant percentage of all industrial machine applications. The choice undoubtedly has much to do with its universally acknowledged qualities, such as reliability, high efficiency and physical robustness. Hence, tasks commonly reserved for highly controllable, more expensive and less rugged DC machines are being turned over to the asynchronous machines. Unlike the DC machine, the AC asynchronous machine is highly non-linear and machine parameters are temperature-sensitive and influenced by the magnetic saturation properties of the core. Field-oriented control, direct torque control, direct self-control and other advanced drives are now frequently applied in high-performance control systems where quick and accurate response is essential. The drive scheme relies heavily on the model assumed for the machine [1]. Two basic approaches are used by the authors for the identification of asynchronous machine parameters: On-line methods, adapting the parameters in a recursive fashion [2–6], and off-line (batch) techniques, which rely on statistical curve fitting to the measured data under specific conditions. These methods are either analytics (determinists) [7–9] or heuristics (approximates) [9, 10]. Neural network, which could

Manuscript received in final form on 14 March 2006.
Address correspondence to Mr. Nouri Benaïdja, Electronics Dept., Faculty of Engineer Sciences, Cité Maabouda, Setif, 19000, Algeria. E-mail: benaidja@yahoo.fr

be employed as on-line or off-line identification paradigms, can be considered as a third category [11, 12].

In [2], authors formulate the problem as a least squares identification. The merits of the least squares method is its independency from a special tests such as the locked rotor test or the no-load test and the measure of the uncertainties in the estimation procedure. Unfortunately, the parameters estimates depend on second derivatives of the current signals, which may be very noisy. Bose et al. [3] describe 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. Min et al. [4] used the fuzzy estimation for tuning the stator resistance in direct torque control of induction machines. The fuzzy resistance estimation should a better performance than the PI estimation. In [5], the parameters are estimated from transient data using a constrained optimization algorithm. The parameters are mapped to the operating conditions using polynomial functions and artificial neural networks. In [6], three analytical methods (Gauss-Newton, Marquardt, and conjugate gradient) are used for the identification of a very saturated machine over its useful range of slip variations. However, it is observed that the three of them yield the same solution, since the functions are very flat around their minimum, and the Newton and Marquardt methods are faster near the solution, but the gradient method is faster when one is far from the solution. Bounekhla et al. [7] used the Hooke and Jeeves algorithm for minimizing the objective function in order to obtain the machine electrical and mechanical parameters. Wamkeue and colleagues [8] used an approach of time-domain estimation procedure in computing double squirrel-cage induction generator parameters from saturated electromechanical line-to-line short-circuit test at machine terminals in operating conditions. Chung [9] used the genetic algorithms for asynchronous machine identification. The superiority of the genetic algorithms over the least squares method is illustrated in [10]. In [11], the maximum likelihood algorithm is used for the estimation of the parameter sets of the nonlinear model at various operating conditions. Then the nonlinear model parameters are represented by the feed forward neural networks. Moreover, a neural net-based inverse model of an induction motor is given in [12].

In the present article, a class of evolutionary optimization methods is proposed for asynchronous machine identification. Training data were generated under no load conditions. The influence of starting time of identification (t_0) is discussed for the identification using genetic algorithms in [10]. Accordingly, we extend the study of the influence of the time t_0 to the class of evolutionary methods. Firstly, the identification method of reference model which incorporates the optimization algorithm into the identification algorithm is presented. Next, a comparative study of optimization methods' performances is carried out. Finally, a conclusion that illustrates advantages and drawbacks of every optimization method concludes the article.

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). 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.

The block diagram of the identification method of reference model is presented in Figure 1. The main components of the identification method of reference model are described in the remaining parts of this section.

2.1. Asynchronous Machine Model

The mathematical model of asynchronous machine referred to $\alpha\beta$ axes fixed with the stator can be expressed by the following equations [13]:

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

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

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

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

$$T_d = f_d \operatorname{sgn}(w_r) \tag{5}$$

where

$$\tau_r = \frac{L_r}{R_r}, \qquad R_e = R_s + \frac{R_r L_m^2}{L_r^2}, \qquad L_e = L_s - \frac{L_m^2}{L_r},$$
$$V_s = \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix}, \qquad i_s = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}, \qquad \phi_r = \begin{bmatrix} \phi_{r\alpha} \\ \phi_{r\beta} \end{bmatrix}, \tag{6}$$
$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

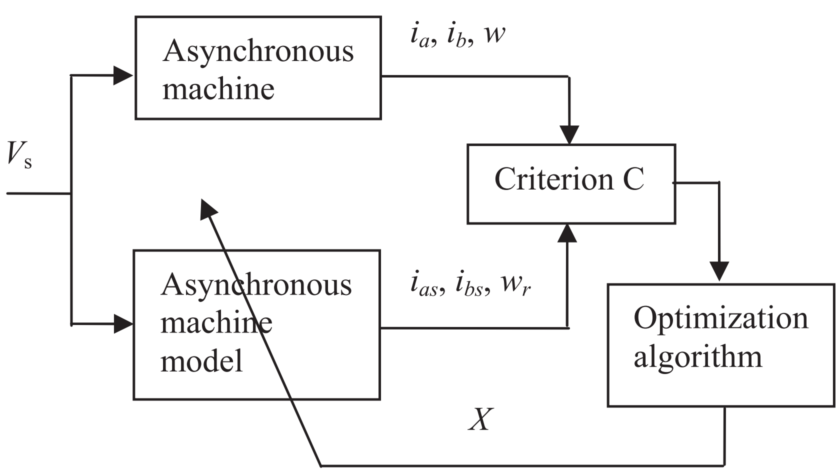


Figure 1. Block diagram of the identification method of reference model.

V_s	stator voltage vector (V)
i_s	stator currents vector (A)
ϕ_r	scaled rotor flux vector (Wb)
R_s	stator resistance (Ω)
R_r	rotor resistance (Ω)
R_e	scaled equivalent resistance (Ω)
L_s	stator self inductance (H)
L_r	rotor self inductance (H)
L_m	magnetizing inductance (H)
L_e	scaled equivalent inductance (H)
τ_r	rotor time constant (s)
T_e	electromagnetic torque (Nm)
T_l	load torque (Nm)
w_r	rotor frequency (rd/s)
J_m	rotor inertial moment (Nms^2)
f_v	viscous friction coefficient (Nm)
f_d	dry friction coefficient (Nms)
p	number of poles pairs
n	number of parameters of three-phase simple cage asynchronous machine, $n = 7$.

Equations (1)–(5) are solved by fourth order Runge Kutta method. Since the rotor flux cannot be measured in normal production motors, the behavior of the system in term of stator current vector and velocity is expressed through scaled rotor flux vector. Simulation starts with the time of standstill $t = 0$, where the values of currents, flux, and rotor angular speed are nil. Objective function is computed only upwards the time t_0. The vector of machine parameters to be identified is then: $X = [R_e \;\; L_e \;\; \tau_r \;\; L_s \;\; f_v \;\; f_d \;\; J_m]$.

2.2. Asynchronous Machine

In order to test optimization methods, we use two three-phase asynchronous machines:

Machine 1 [14]. Machine 1 has the following characteristics: Power rating $P = 0.63$ kw, supply voltage $V_s = 380$ V, nominal angular speed $\Omega_r = 2900$ rpm, power factor $\cos\varphi_n = 0.737$, frequency $f = 50$ Hz, number of poles pairs $p = 1$. The machine is supplied by a reduced balanced system of sinusoidal voltages of $V_{seff} = 88$ V (rms value). The starting time $t_0 = 0.71$ s, load torque $T_l = 0$. Preliminary electrical parameters are obtained by the nameplate method [15]:

$$\sigma = \frac{L_e}{L_s} = \frac{1 - \cos\varphi_n}{1 + \cos\varphi_n}, \qquad w_g = w_n - \frac{2\pi p \Omega_r}{60}, \qquad \tau_r = \frac{1}{\sqrt{\sigma} w_g},$$
$$L_f = \frac{\sqrt{\sigma} V_s}{I_{sn} w_n}, \qquad L_r = L_f \frac{1 - \sigma}{\sigma}, \qquad L_s = L_f + L_r. \tag{7}$$

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

By assuming the stator resistance $R_s = R_r$, a simple handling of Eqs. (6) and (7) 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.

As the use of mechanical parameters in drives is a critical task [16, 17], it is of a caution practice to choose the search intervals of mechanical parameters as large as possible. Then, the search intervals of electrical and mechanical parameters are selected such that simulation variables are in the feasible region: $\Delta R_e = [8.2\ \ 3R_{ep}]$, $\Delta L_e = [L_{ep}/3\ \ 3L_{ep}]$, $\Delta\tau_r = [\tau_{rp}/3\ \ 3\tau_{rp}]$, $\Delta L_s = [L_{sp}/3\ \ 0.264]$, $\Delta f_v = [0\ \ 0.002]$, $\Delta f_d = [0\ \ 0.1]$, $\Delta J_m = [0\ \ 0.02]$.

Machine 2 [10]. Machine 2 has the following characteristics: Power rating $P = 5$ HP, supply voltage $V_s = 220$ V, nominal angular speed $\Omega_r = 1750$ rpm, frequency $f = 60$ Hz, number of poles pairs $p = 2$. The machine is supplied by a balanced system of sinusoidal voltages of $V_s = 380$ V (peak value). The starting time $t_0 = 0.05$ s, load torque $T_l = 0$. Preliminary electrical parameters are obtained by classical tests (using DC step supply) [18]:

$$R_{ep} = 0.626\ \Omega, \qquad L_{ep} = 13.6\ \text{mH}, \qquad \tau_{rp} = 321\ \text{ms}, \qquad L_{sp} = 72.1\ \text{mH}.$$

And, $\Delta R_e = [R_{ep}/1.5\ \ 1.5R_{ep}]$, $\Delta L_e = [L_{ep}/1.5\ \ 1.5L_{ep}]$, $\Delta\tau_r = [\tau_{rp}/1.5\ \ 1.5\tau_{rp}]$, $\Delta L_s = [L_{sp}/1.5\ \ 1.5L_{sp}]$, $\Delta f_v = [0.0007\ \ 0.00165]$, $\Delta f_d = [0.042\ \ 0.094]$, $\Delta J_m = [0.02\ \ 0.046]$.

2.3. Performance Criterion

As performance criterion, we choose the following cost functions:

For machine 1:

$$C = \sum_j k_a|(i_{aj} - i_{saj})| + k_b|(i_{bj} - i_{sbj})| + k_w|(w_j - w_{rj})| \tag{8}$$

For machine 2:

$$C = \sum_j k_a|(i_{\alpha j} - i_{s\alpha j})| + k_b|(i_{\beta j} - i_{s\beta j})| + k_w|(w_j - w_{rj})| \tag{9}$$

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

In [10], Alonge et al. have tested using the weights $k_a = 1$, $k_b = 1$, and $k_w = 0.5$, $k_w = 1$, $k_w = 2$. In this work the weights k_a, k_b, k_w are tested in the range 0.1–10. Since a rigid relation exists between the currents i_a, i_b and i_α, i_β, respectively, the identification process is not affected by the choice of the currents' reference.

2.4. Optimization Techniques

2.4.1. Scatter Search (SS). Scatter search algorithm has been developed by Glover [19]. It is based on the combination of decision rules and the interpolation in convex regions. By explicitly underlying the intensification-diversification principle, we considered it as a platform for the other methods of the class. Scatter search procedure can be summarized in the following steps:

1. A diversification generation that serves to generate a collection of diverse trial solutions x_i.
2. An improvement that is used for transforming a trial solution into one or more enhanced trial solutions. We used Nelder and Mead Simplex method as an improvement method [20].
3. A reference set update, in order to build and maintain a reference set, with a length of b, consisting of the $b1$ best solutions, in term of objective function, and the $b2$ diverse solutions, in term of Euclidean distance to the best solutions.
4. A subset generation that consists of generating the subsets that will be used for creating new solutions with the solution combination method. We are satisfied by building $(b^2 - b)/2$ subsets composed of 2 elements of reference set.
5. A solution combination that uses the subsets generated with the subset generation method to combine the elements in each subset with the purpose of creating new trial solutions. It consists of finding linear Combinations of subsets elements by application of the following rules: By assuming x' and x'' reference solutions:

$$\begin{cases} C1\colon x = x' - d \\ C2\colon x = x' + d \\ C3\colon x = x'' + d \end{cases} \tag{10}$$

   where $d = rand.(x'' - x')/2$ and *rand* is the random number in the range (0, 1).
   —If x' and x'' are elements of $b1$, then generate 4 solutions by applying $C1$ and $C3$ once and $C2$ twice.
   —If only one of x' and x'' is a member of $b1$, then generate 3 solutions by applying $C1$, $C2$, and $C3$ once.
   —If neither x' nor x'' is a member of $b2$, then generate 2 solutions by applying $C2$ once and randomly choosing between $C1$ and $C3$.
6. The search continues in a loop that consists of applying the solution combination method followed by the improvement method and the reference update method. This loop terminates when the reference set does not change and all the subsets have already been subjected to solution combination method. At this point, the diversification generation method is used to construct a new reference set and the search continues.

2.4.2. Evolutionary Programming (EP). The biologically inspired algorithm of evolutionary programming has been invented in 1962 [21]. It can be summarized in the following steps:

1. Initialization: The initial population of m control variables, indicating the machine parameters are selected randomly from the set of uniformly distributed control variables ranging over their upper $x_{\max}$ and lower $x_{\min}$ limits, indicating the feasible region of search. The fitness score f is obtained according to the environment and the objective function:

$$f = \frac{1}{1 + C} \tag{11}$$

2. Statistics: The maximum fitness $f_{\max}$, minimum fitness $f_{\min}$, and the sum of fitness of this generation are calculated.

3. Mutation: Each selected parent P_i is mutated and added to its population following the rule:

$$P_{i+m,j} = P_{i,j} + N\left(0, \beta(x_{\max} - x_{\min})\frac{f_i}{f_{\max}}\right) \qquad j = 1, 2, \ldots, n. \tag{12}$$

   where n is the number of decision variables in an individual, corresponding to the number of machine parameters. $N(\mu, \sigma^2)$ represents a Gaussian random variable with mean μ and variance σ^2. β is the mutation scale that could be adaptively decreased during generations.

   If the individual $P_{i+m,j}$ exceeds the upper or lower limits, the value of upper or lower limit is attributed to $P_{i+m,j}$.
4. Competition: k elitist individuals, which have the best fitness, are kept as the parents for the next generation. Other individuals in the combined population of size ($2m$) have to compete with each other to get their chances for the next generation:

   —The value of the weight w_i of the ith individual is calculated by the following competition:

$$w_i = \sum_{t=1}^{N} w_{i,t} \tag{13}$$

   where N is the competition number calculated randomly, and

$$w_{i,t} = \begin{cases} 1 & \text{if } rand < \dfrac{f_i}{f_r + f_i} \\ 0 & \text{elsewhere} \end{cases} \tag{14}$$

   where f_r is the fitness of the randomly selected rth individual and *rand* is the random number in the range (0, 1).

   —The ($2m$) individuals are ranked according to their weight w_i. The first ($m-k$) individuals are selected along their corresponding fitness f_i to be the bases for the next generation.
5. Steps 1 to 4 are repeated until no improvement of fitness $f_{\max}$ is showed.

2.4.3. Ant Colony Algorithm (AC). Ethnologists' studies reveal that blind ants had the capability to establish shortest route paths from their nest to food source thinks to pheromone trail deposit and redeposit. Initially, ants choose any path with equal probability. Nevertheless, the ant following the shortest path will be the first to reach the source food and back early to the nest. New ants will see some pheromone trail on the shorter path and will choose it with higher probability than the longer paths. Pheromone will be released on the chosen path making it even more attractive for the subsequent ants. This stigmatic behavior has inspired Dorigo to propose the algorithm called "ant system AS," which is applied to the combinatorial optimization problems [22]. An analogy between the stochastic descent technique and the ant colony algorithm is presented in [23] that shows the similarity of the two methods. Accordingly, the continuous optimization problem of asynchronous machine identification can be formulated as following:

1. Initialization: Before each run, the initial population of m ants is generated randomly within the feasible region. All the ants possess the same initial quantity of pheromone: $\tau_i = \tau_0$.

2. Autocatalytic process (positive feedback): Neighborhood identifies moves that lead from one solution to the next. For asynchronous machine, the neighborhood is defined by incrementing or decrementing one decision variable at once. The neighborhood's cardinal is then: $N = 14$. Each ant F examines his immediate neighborhood and crawls toward the chosen ant j according to the probability:

$$p_j = \frac{[\tau_j]^\alpha [C_j]^\beta}{\sum_{i=1}^{N} [\tau_i]^\alpha [C_j]^\beta} \tag{15}$$

where τ_k denotes the pheromone quantity of the ant k, C_k denotes the value of objective function of the ant k, α is the randomness coefficient, and β is the greediness coefficient.

Decision variables become:

$$\begin{cases} x_j^i(k+1) = x_j^i(k) + L^i & \text{if } x_F^i > x_j^i \\ x_j^i(k+1) = x_j^i(k) - L^i & \text{if } x_F^i < x_j^i \\ x_j^i(k+1) = x_j^i(k) & \text{if } x_F^i = x_j^i \end{cases} \qquad i = 1, 2, \ldots, n \tag{16}$$

where L_i is the size of the crawl step of the parameter x^i.

3. Trail evaporation: The pheromone of the ant j is updated as following:

$$\tau_j(k+1) = \rho \tau_j(k) + \Delta\tau_j(k) \tag{17}$$

where ρ is the evaporation coefficient.

$\Delta\tau_j(k)$ represents the contribution of the ant j to the reinforcement of the movement of ant F. We select:

$$\Delta\tau_j(k) = QC_j(k) \tag{18}$$

where Q is the intensification coefficient.

4. Nest displacement: At this step, all the ants are evaluated and pheromone trail updated. Old ants die, which the acquired information is inherited by new ants. At each step, each agent leaves a sign of its activity that changes the probability with which the decisions will be made in the future; in such a way, the path

$$l = \sum_{i=1}^{N-1} C(x^i) \tag{19}$$

where

$$N \leq \prod_{j=1}^{n} \left(\frac{L_j}{\Delta x_j} \right)$$

is minimized, where L_j is the size of the search interval of the parameter j, and Δx_j is the step size of the movement of the parameter j. The process continues until the objective function has not improved.

2.4.4. Particle Swarm (PS). Particle swarm algorithm has been originally designed by Kennedy in 1995 [24]. It mimics chorography and social behavior of some creatures foraging through D-dimensional space. Firstly, m particles $x_i(x_{i1}, x_{i2}, \ldots, x_{in})$, of initial speed v_i and initial position s_i, are randomly and uniformly created. Particle position is located by the parameters set of the machine. At every iteration, each particle share memory of its previous best position s_{best} and the previous best position among the group g_{best}. The combined positions serve to adjust velocity along each dimension:

$$v_i(k+1) = w_i v_i(k) + c1 + c2 \qquad i = 1, 2, \ldots, n \tag{20}$$

with w_i: weight, $c1 = C1.rand.(s_{besti} - s_i(k))$, denotes the cognitive component, $c2 = C2.rand.(g_{besti} - s_i(k))$, denotes the social component, *rand*: random number in the range (0, 1), and $s_i(k)$ is the position of the particle i at the iteration k.

The new position is then:

$$s_i(k+1) = s_i(k) + v_i(k+1) \qquad i = 1, 2, \ldots, n \tag{21}$$

The objective function is then evaluated with the new parameters s_i. The previous step is repeated until there is no improvement of g_{best}.

3. Identification Results

Identification is performed in visual C++ 6 compiler running in 1.7-GH Pentium-based-PC. Sampling frequency is 2 KHz. Voltage signals are acquired by means of three resistive voltage dividers, realized with resistors having accuracy equal to 0.1%. Two stator currents are acquired by means of two Hall transducers that 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, is converted into a signal in the range $[-10, 10]$ V, and then acquired.

Several combinations of objective function weights are tested. For both machines, the more accurate results are given by the following: $k_a = k_b = 1$, $k_w = 0.5$. The parameters resulting from the application of different methods to the identification of machine 1 and machine 2 are represented in Tables 1 and 2, respectively. The second column indicates the parameters obtained by the steepest descent method.

For machine 1, we notice that with the same computation time as ant colony algorithm, the value of objective function resulting from the application of scatter search is low. However, with the same value of objective function as particle swarm, the computation time of the identification using scatter search is the quarter. The best results among the four evolutionary techniques are given by the application of evolutionary programming technique. The worst results are given by the application of ant colony algorithm. Electrical parameters obtained by different techniques are close to those given by the nameplate method. The non-compound electric parameter L_s converges toward the lower limit when the steepest descent is applied; whereas it converges toward the upper limit in the opposite direction of the gradient when the evolutionary techniques are applied. A local minimum is then avoided. The four evolutionary techniques had not succeeded outperforming the steepest descent technique.

Conversely, for machine 2, the best results that compromise objective function value and computation time are given by ant colony algorithm. Electrical parameters obtained by different techniques, namely the equivalent resistance R_e for scatter search, are far

Table 1
Parameters resulting from identification of machine 1

	SD	SS	EP	AC	PS
R_e (Ω)	16	8.71717	8.31012	8.2	8.2
L_e (mH)	39	65	39	46.24	39.2075
τ_r (ms)	24	24.9211	29	33.35	29
L_s (mH)	110	264	264	218.132	264
f_v (Nms)	0.002	0.00096	0.00028	64.10^{-8}	0.000258
f_d (Nm)	0.1	0.00015	0.017544	0.000057	0.033282
J_m (Nms^2)	0.01	0.01	0.010026	0.01183	0.014727
C	300.025	414.966	381.666	425.4367	414.349
Time (s)	<1	32	10	32	8

Table 2
Parameters resulting from identification of machine 2

	SD	SS	EP	AC	PS
R_e (Ω)	0.417257	1.18182	0.60517	0.774146	0.630684
L_e (mH)	20.44868	4.54415	9.08830	3.1332	6.81623
τ_r (ms)	481.5	123.702	214	182.385	214.002
L_s (mH)	48.067	200.966	108.15	130.644	105.5545
f_v (Nms)	0.000733	0.00165	0.00094	0.002818	0.000947
f_d (Nm)	0.0936	0.1565	0.06019	0.073775	0.069035
J_m (Nms^2)	0.02073	0.05142	0.02073	0.053283	0.021424
C	3789.27	1434.99	2941.18	1575.002	2581.503
Time (s)	<1	69	27	18	<1

from those given by DC step supply method. Hence, optimization techniques give advantages over the steepest descent only if identification starting time t_0 is low.

From Tables 1 and 2, it is clear that the resulting costs are not nil; hence, the optimal solution is not reached. On the other hand, it is well known that softcomputing techniques settle for a sub-optimal solution within the allotted amount of time. Being so, and taking into account the fact that the parameters resulting from asynchronous machine identification change with the test conditions, overfitting of the machine parameters is avoided. Thus, it is not surprising that there is little correlation between experimental and computed results in Figures 2–5. The same observation comes when noise is considered.

When the ant colony algorithm is applied, it is observed that for machine 1, the pheromone quantities of different ants are proportional to their costs, making the probabilities that the decisions will be made in the future less significant. Hence, ant colony algorithm is reduced to a pure stochastic descent. For machine 2, the pheromone quantities of the ants are nonlinear functions of their costs. As is well known, the change of different parameters of the machine is coherent (like of the movement of foots of ants [25]). Then, when the time t_0 is low, the inrush information of the currents is included into the autocatalytic process for the guidance of the search. When the evolutionary programming is applied, the changes of machine parameters resulting from the mutation operator are quazy-independent, which is more convenient for the poor information offered by the high values of the time t_0. A large variability of mechanical parameters, essentially f_v and f_d, is observed even if large values are allotted to the weight k_w.

3.1. Steepest Descent Method (SD)

The search intervals of machines parameters are regularly sampled and different combinations of initial values of the parameters are tested. The parameters obtained are dependant on the initial values. The steepest descent method is a single-solution based technique, objective function does not compute, then execution time is very low. Steepest descent method is very sensitive toward noise.

3.2. Scatter Search (SS)

Output resulting from the identification of machine 1 and machine 2 with starting times t_0 of 0.71 s and 0.05 s, respectively, and using scatter search are shown in Figure 2. For creating a balanced initial population, usually a large population size is chosen. Bearing in mind that reference set is continuously updated, low values are attributed to reference set size. An increase of reference set size causes computation time to dramatically increase without improving objective function values. In addition, an intermediate value seems to be convenient to the Cartesian distance to other parameters of reference set from which a parameter is considered as new. When this value decreases, the algorithm could be trapped to a local minimum.

Many parameters of scatter search algorithm are tested. For both machines, the best results are given by population size of 30, the length of the reference set of the best solutions $b_1 = 5$, the length of the reference set of the most distant from the best solutions $b_2 = 4$, Cartesian distance to other parameters of reference set from which a parameter is considered as new and added to reference set is 0.1 for machine 1 and 0.01 for machine 2.

3.3. Evolutionary Programming (EP)

Output resulting from the identification of machine 1 and machine 2, with starting times t_0 of 0.71 s and 0.05 s, respectively, and using evolutionary programming algorithm are shown in Figure 3. It is worth noticing that, when a low value of mutation scale β increase computation time, a high value of mutation scale β might conduct the algorithm to search in infeasible regions. It has also been noted that a high value of elitist number k can alter the selection process of the algorithm. Around 10% of population size seems to be a reasonable value for the elitist number k. The best results are observed when the mutation scale β is chosen linearly decreasing from 0.1 to 0.01 for machine 1, from 0.005 to 0.001 for machine 2; and for both machines the number of elitist individuals $k = 10$, the population size $m = 100$.

3.4. Ant Colony Algorithm (AC)

Output resulting from the identification of machine 1 and machine 2 with starting times t_0 of 0.71 s and 0.05 s, respectively, and using ant colony algorithm are shown in Figure 4. Ant colony algorithm seems well working with intermediate values of colony size.

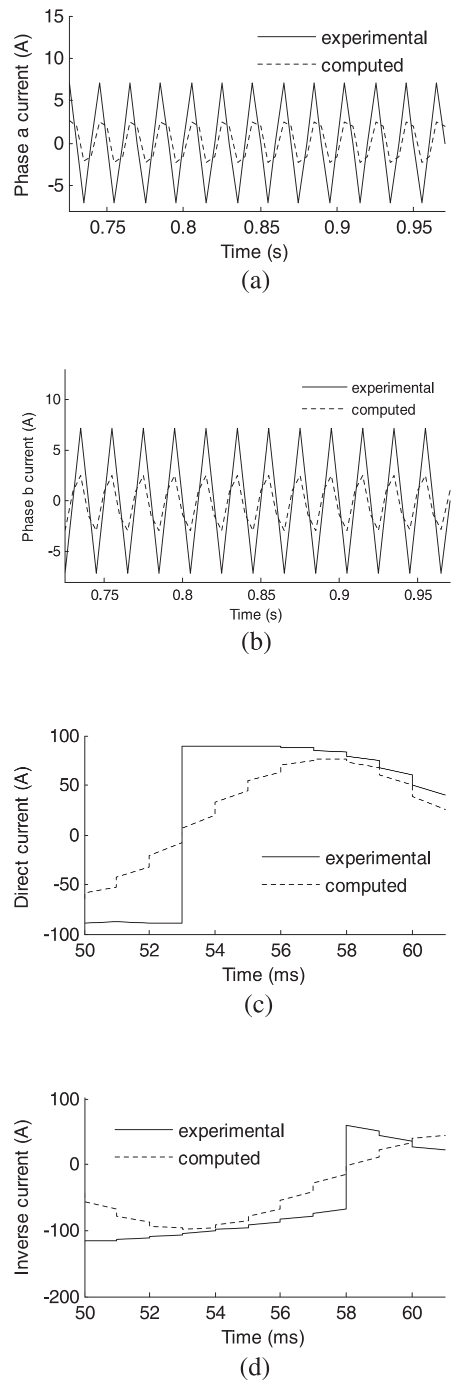


Figure 2. Machines' outputs resulting from scatter search identification; (a), (b) machine 1 currents; (c), (d) machine 2 currents.

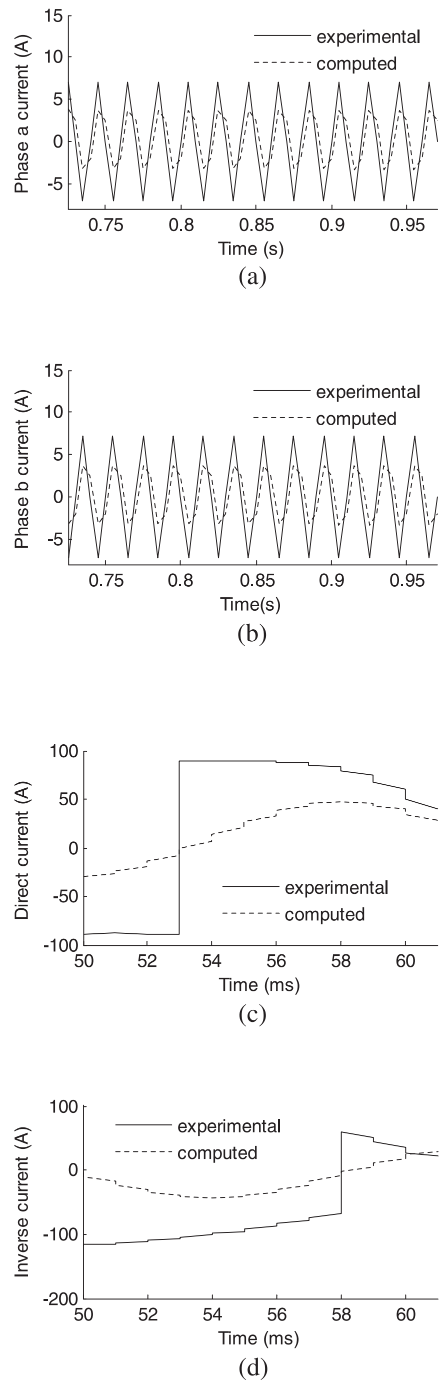
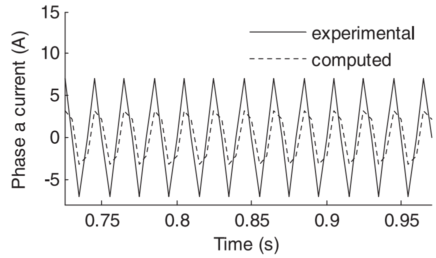
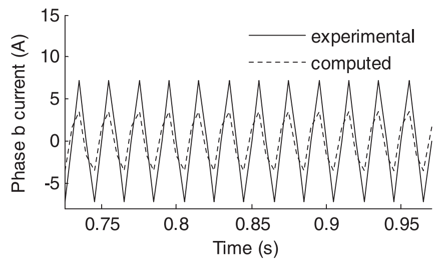


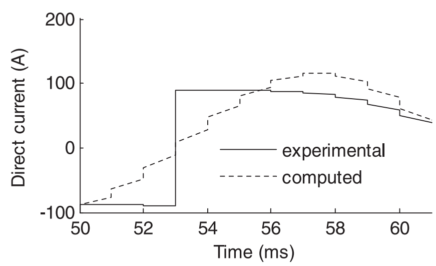
Figure 3. Machines' outputs resulting from evolutionary programming identification; (a), (b) machine 1 currents; (c), (d) machine 2 currents.



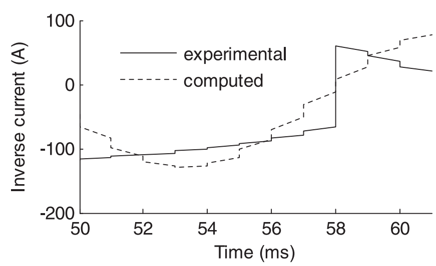
(a)



(b)

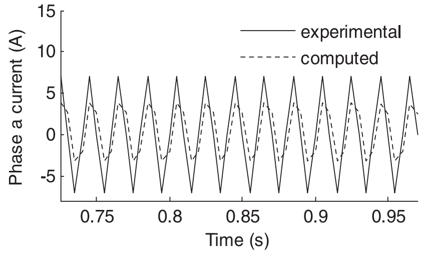


(c)

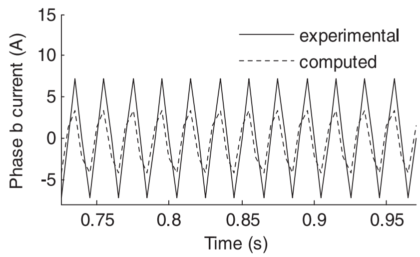


(d)

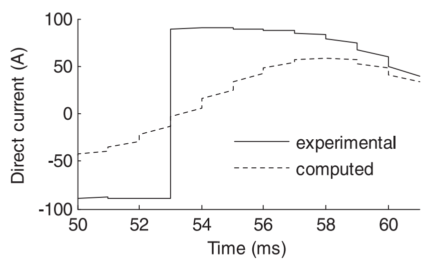
Figure 4. Machines' outputs resulting from ant colony algorithm identification; (a), (b) machine 1 currents; (c), (d) machine 2 currents.



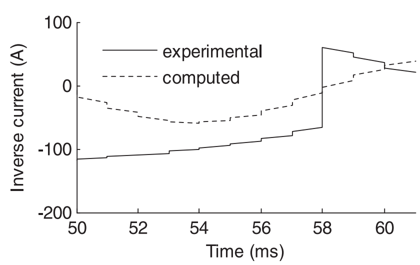
(a)



(b)



(c)



(d)

Figure 5. Machines' outputs resulting from particle swarm identification; (a), (b) machine 1 currents; (c), (d) machine 2 currents.

Indeed, it is observed that for a colony size over 30, the best objective function value decreases; a lot of ants pursue further search in a limited amount of food, in the sense that conflicting behavior appears instead of cooperation. When high values of randomness coefficient α are tried, stagnation behavior is observed (all the ants follow the same path). The impact of the choice of the initial quantity of pheromone τ_0 and the intensification coefficient Q is negligible.

For both machines, ant colony algorithm parameters that give the best results are the following: population size $m = 30$, evaporation coefficients $\rho_i = 0.98$, $i = 1, 2, \ldots, 7$, greediness coefficient $\beta = 2$, randomness coefficient $\alpha = 0.5$, initial pheromone quantity $\tau_0 = 100$, and the intensification coefficient $Q = 10000$. The sizes of step of the ants' crawl are: for machine 1: $L = [0.05\ \ 0.005\ \ 0.001\ \ 0.001\ \ 0.0001\ \ 0.001\ \ 0.001]$, and for machine 2: $L = [0.005\ \ 0.0025\ \ 0.005\ \ 0.0025\ \ 0.00025\ \ 0.0025\ \ 0.0025]$.

3.5. *Particle Swarm (PS)*

Output resulting from the identification of machine 1 and machine 2 with starting times t_0 of 0.71 s and 0.05 s, respectively, and using particle swarm optimization are shown in Figure 5. The weights w_i are the essential components of particle swarm algorithm. High values for weights w_i might cause the algorithm to escape interesting regions, while with low weights w_i the algorithm tend to converge poorly. This is due to particles inertia. The best results are shown when the weights w_i are chosen linearly decreasing from 0.4 to 0.1 for both machines and the coefficients:

$$C_i1 = 0.5, \qquad i = 1, 2, \ldots, 7.$$

$$C_i2 = 0.5, \qquad i = 1, 2, \ldots, 7.$$

4. Conclusion

Four methods belonging to evolutionary optimization class are applied to asynchronous machine identification. All the methods have proven their numerical stability and their robustness toward initial conditions. The choice of methods parameters is not trivial, but no longer affected by machine parameters. Any parameter set of different optimization techniques does not give the best results. Consequently, careful work should be devoted to the selection of algorithms parameters. Methods performances are dependant on the starting time t_0. The large inrush of current at startup, corresponding to a low value of the time t_0, permits the ant colony algorithm to exhibit performances superior to the other methods. When the value of the time t_0 is high, i.e., close to permanent state, evolutionary programming algorithm gives advantage over the other methods. Slightly scatter search and ant colony algorithms have close performances toward time t_0 changes. Scatter search is the slowest algorithm; this is due to its overlapped loops and reference set exhaustion. Conversely, particle swarm algorithm is the faster, thanks to its efficient simplicity. Identification of mechanical parameters remains a hard task.

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. J. Stephan, M. Bodson, and J. Chiasson, "Real time estimation of the parameters and fluxes of induction motors," *IEEE Transactions on Industry Applications*, Vol. 30, No. 3, pp. 746–759, May/June 1994.
3. 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.
4. 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.
5. A. B. Proca and A. Keyhani, "Identification of variable frequency induction motor models from operating data," *IEEE Transactions on Energy Conversion*, Vol. 17, No. 1, pp. 24–31, March 2002.
6. P. T. Lagonotte, H. Al Miah, and M. Poloujadoff, "Modelling and identification of parameters of saturated induction machine operating under motor and generator conditions," *Electric Machines and Power Systems*, Vol. 27, No. 2, pp. 107–122, February 1999.
7. A. Bounekhla, M. E. Zaïm, A. Rezzoug, and A. A. Ahmed, "The Hooke and Jeaves algorithm approach in the identification of the induction machine parameters," *ICEM2000, ESPOO*, Finland, pp. 222–228, August 2000.
8. R. Wamkeue and I. Kamwa, "Saturated electromechanical transients based maximum likelihood identification of double-cage induction generator parameters," *ICEM2000, ESPOO*, Finland, pp. 286–290, August 2000.
9. P. Y. Chung, M. Dolen, and R. D. Lorenz, "Parameter identification of induction machines by continuous genetic algorithms," *ANNIE2000*, St. Louis, pp. 1–13, November 2000.
10. F. Alonge, F. D'ippolito, and F. M. Raimondi, "Parameter identification of induction motors: Least squares vs. genetic algorithms," *IWMC*, Grenoble, France, pp. 1112–1122, September 1998.
11. 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.
12. 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.
13. P. C. Krause, *Analysis of Electric Machinery*, New York: McGraw-Hill Book Company, 1986.
14. M. Biriescu, V. Groza, V. Cretu, M. Mot, and G. Modescu, "Testing of electrical machines using a data acquisition and processing system," *Workshop of Electrical Machines Parameters*, Napoca, Romania, pp. 31–34, May 2001.
15. H. Razik and A. Rezzoug, "Identification of electrical parameters of an induction motor, a comparison between R.L.S and genetic algorithm," *ICEM2000, ESPOO*, Finland, pp. 1433–1437, August 2000.
16. K. B. Lee, "Disturbance observer that uses radial basis function networks for the low speed control of a servo motor," *IEE Proc.-Control Theory Applications*, Vol. 152, No. 2, pp. 118–124, March 2005.
17. G. Xu, C. Chen, and Q. Sun, "Analyzing the influence of induction motor inertia on power system low frequency oscillation," *Electrical Power Components and Systems*, Vol. 33, No. 5, pp. 551–562, May 2005.
18. A. B. Proca and A. Keyhani, "Induction motor parameter identification from operating data for electric drive operation," *18th DASC*, St. Louis, Missouri, p.no. 0-7803-5749-3/99/$10.00, CD Proceeding, October 1999.
19. F. Glover, M. Laguna, and R. Marti, "A template for scatter search and path relinking," in *Artificial Evolution*, J. K. Hao, E. Lutton, E. Ronald, and M. Schoenauer (Eds.), Germany: Springer, pp. 3–51, 1998.
20. J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, Vol. 7, pp. 308–313, 1965.
21. L. J. Fogel, "Autonomous automata," *Industrial Research*, Vol. 4, pp. 14–19, 1962.

22. M. Dorigo, M. Maniezzo, and A. Colorni, “The ant system: Optimization by a colony of cooperating agents,” *IEEE Transactions on Systems Man and Cybernetics, Part B*, Vol. 26, No. 1, pp. 1–13, July 1996.
23. N. M. Meuleau and M. Dorigo, “Ant colony optimization and stochastic descent,” Technical Report. No. TR/IRIDIA/2000-36, Institut de Recherches Interdisciplinaires et de Développements en Intelligence Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA), Université Libre de Bruxelles, Belgium, December 2000.
24. J. Kennedy and W. Spears, “Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator,” *IEEE ICEC*, Anchorage, Alaska, pp. 78–83, May 1998.
25. E. E. Vallejo and F. Ramos, “Evolving insect locomotion using cooperative genetic programming,” *MICAI2000*, Mexico, pp. 170–181, 2000.