Application of Evolutionary Algorithms with Adaptive Mutation to the Identification of Induction Motor Parameters at Standstill

Joanna M. Lis, Teresa Orlowska-Kowalska, Senior Member, IEEE
Wroclaw University of Technology, Institute of Electrical Machines, Drives and Measurements, Wroclaw, Poland teresa.orlowska-kowalska@pwr.wroc.pl, joanna.lis@pwr.wroc.pl

Abstract—In this paper the application of evolutionary algorithms to the identification of induction motor equivalent circuit parameters at standstill is presented. In order to improve the time efficiency of the identification procedure, the adaptive mutation mechanism is introduced to the evolutionary algorithm. Few versions of the adaptive mutation mechanisms are investigated and evaluated in simulations. By employing the adaptive mutation the significant reduction of the processing time has been obtained while the required accuracy of the algorithm has not been deteriorated. The results of simulation are verified and confirmed in the experimental tests.

Keywords—induction motor, identification, evolutionary algorithm, adaptive mutation, simulated annealing

I. INTRODUCTION

The popularity of induction motor drives is still growing due to the high reliability and low cost of induction motors. The power electronic and microprocessor devices are being constantly improved, while their cost is decreasing. Therefore high performance control methods and techniques for induction motor are used in different drive applications. Among those techniques the sensorless approach based on the application of the state observers or other estimators for the state variables reconstruction is recently very popular. State observers are based on the mathematical model of the induction motor and thus are sensitive to motor parameters values. Therefore the application of sensorless drives involves proper identification of the induction motor parameters with regard to accuracy and quality of the control.

Methods for the induction motor (IM) parameters estimation can be classified into four categories [4]:
- parameter calculation from motor construction data,
- parameter estimation based on steady-state motor models,
- frequency-domain parameter estimation,
- time-domain parameter estimation.

In case of the last type of methods, time-domain motor measurements are performed and model parameters are adjusted to match the model response to these measurements. Few authors proposed evolutionary algorithms (EA) for the identification [1], [2], [6], [9]-[11]. In the most cases simple genetic algorithm (GA) was applied [1], [2], [6], [10], [11]. The proposed EA operated in the binary-coded domain. However, the problem is of continuous nature, therefore transforming continuous solutions into binary strings was necessary in order to apply those algorithms. Yet the locality of the problem landscape might suffer from digitization, and small changes in a discrete solution may lead to large changes in its continuous form. In other words a small value in Hamming distance (syntactic information) does not always imply a small variation in their fitness interpretation (semantic information). The performance of the simple EA operating in binary-coded and real-coded domain was investigated and compared in [11]. The test showed that solving the problem in binary-coded domain instead of the real-coded one is particularly unfavorable in case of the considered problem of the induction motor parameters’ identification.

Application of the classical binary coded GA introduces some unfavorable changes to the optimized objective function and results in intensification of its multimodal character, which can be to a large extend neglected by appropriate formulation of the problem i.e. using the presented mathematical model, identifying the physical parameters instead of their arithmetical combination and employing the algorithm working with the real coded individuals. Nevertheless the GA proposed for instance in [2] is capable of dealing with the difficult multimodal functions. In other words this approach enables solving the problem despite the fact that it introduces significant complications due to changing the character of the objective function from relatively simple to a challenging one. Thus the difference in those approaches can be mainly observed in the execution time. The algorithm, which works with the real coded individuals, is capable of finding the satisfactory solution in significantly shorter time than the one with binary coding [12]. It should not however lead to a conclusion on the superiority of the real coded EA, because the both algorithms actually solve completely different optimization tasks (the real coded version of the objective function obviously does not correspond to the binary coded one). The real coded version appears to be very advantageous from the optimization’s point of view, while the binary one obviously places a greater challenge before the genetic (binary coded) algorithm.

The identification procedure presented in this paper is performed at standstill and is based on the reconstruction of stator current response to the forced stator voltage using EA. The objective function is defined as the mean squared error between the computed and the experimental data. Few versions of EA with adaptive mutation were...
applied to solve the considered optimization problem. Their performances were investigated in simulations and compared. Comparing the effectiveness of different EA with classical and adaptive mutation, the focus was also on the algorithm’s time efficiency, because in sensorless drive the problem of parameter identification in the initial start-up of the drive is crucial and should not be a time consuming. The results of simulations were verified in experiments.

II. THE IDENTIFICATION OF Induction Motor PARAMETERS AT STANDSTILL

A. Formulation of the Problem

The proper initial identification of the IM equivalent circuit parameters is required in sensorless drives in order to provide their smooth start and stable work. In most cases the initial identification has to be performed at standstill, due to the fact that the drive is directly coupled to the load machine. The inaccuracy of such identification can be compensated by suitable adaptive mechanisms of observers during the closed-loop system operation. Yet, only electrical parameters of IM can be determined in such manner.

The IM can be described by the following set of equations in the stationary reference frame (α-β) in p.u. system:

\[ u_s = r_s i_s + T_N \frac{d\Psi_r}{dt}, \]
\[ 0 = r_s i_s + T_N \frac{d\Psi_r}{dt} - j\omega\Psi_r, \]
\[ \Psi_s = x_s i_s + x_M i_r, \]
\[ \Psi_r = x_r i_s + x_M i_r, \]
\[ \frac{d\omega}{dt} = \frac{1}{T_s} x_s \left( \Im(\Psi_r \times i_s) - m_x \right). \]

where: \( u_s, i_s, i_r, \Psi_s, \Psi_r \) – stator voltage vector, stator and rotor current vectors, stator and rotor flux vectors, respectively, \( \omega \) – rotor speed, \( m_x \) – load torque, \( T_M \) – mechanical time constant, \( r_s, r_r \) – stator and rotor resistance, \( x_s, x_r \) – stator and rotor reactance, \( x_M \) – magnetizing reactance, \( T_s = 1/2f_{50} \).

At standstill (\( \omega = 0 \)) the above equations can be rearranged as follows:

\[ \frac{d\Psi_r}{dt} = \frac{1}{T_N} (u_s - r_s i_s), \]
\[ \frac{di_r}{dt} = \frac{1}{T_N} \left( x_r \Psi_r - (r_r x_r + r_s x_s) i_s + x_s u_s \right). \]

The stator windings should be connected in such manner, that the rotating electromagnetic field would not occur and the motor would remain not rotating. The values of the space vectors of the voltage, current, and stator flux in the equations (6), (7) are real values if the motor is fed by DC voltage applied to the stator winding connected as it is shown in Figure 1.

For such connection of the stator windings the following relationship can be deduced:

\[ U_{sa} = \frac{2}{3} U_{dc}, \]
\[ U_{sb} = \frac{1}{3} U_{dc}. \]

Taking into consideration the definition of the complex space vector for the three-phase AC machine, the following equations can be formulated including (8),(9) [11]:

\[ u_s = \frac{2}{3} (u_{sa} + a u_{sb} + a^2 u_{sc}) = u_{sa} = \frac{2}{3} U_{dc}, \]
\[ i_s = \frac{2}{3} (i_{sa} + a i_{sb} + a^2 i_{sc}) = i_{sa}, \]

where: \( a = e^{j\frac{2\pi}{3}} \), \( a^2 = e^{j\frac{4\pi}{3}} \).

The simulation model was developed on the basis of equations (6)–(11). Taking advantage of that model the stator current response was calculated. The minimized objective function was defined by the sum of the squares of the differences between the experimental \( i(j)_{\text{meas}} \) and calculated \( i(j)_{\text{est}} \) current response.

\[ f = \sum_{j=1}^{N} \left[ i(j)_{\text{meas}} - i(j)_{\text{est}} \right]^2 \]

where \( N \)–the number of the current samples.

Five parameters are identified: stator and rotor resistances \( (r_s, r_r) \), stator and rotor inductances \( (x_s, x_r) \) and mutual inductance \( (x_M) \).

B. Evolutionary Algorithms

Evolutionary algorithms are the optimization methods, founded on the principles of Darwinian natural selection [3]. At present numerous variations of EA are applied to many technical disciplines thus it is hard to describe a typical algorithm of such kind in detail. In general, EA search the space of alternative solutions to the given optimization problem, by evolving a population of candidate solutions (individuals) over generations, to produce better solutions. The new candidate solutions are created in each algorithm’s iteration by means of the variation operators. In the EA presented in this paper the population is concentrated [5] and moves through the search space in the direction of the consecutive objective function’s optima, in contrast to the classical GA, in which the solutions placed in any location of the search space can be obtained.
in each iteration. The basic idea of such a search algorithm is to iteratively improve a single solution by looking in its neighborhood and choose the most promising adjacent solution as a new candidate.

The algorithms presented in this paper use only the mutation operator - a classical widely used Gaussian mutation and a special kind adaptive mutations. In case of the presented version of the EA the crossover operator, which is typical for the GA and the modifications of the EA arising from that basis, is not applied. The crossover operator, however, performing very well in the case of the genetic algorithm and being the main variation operator concerning this approach, is not advantageous as far as the presented version is concerned. Despite the fact that the crossover operator would have a significantly different effect, when applied to a real coded individuals, the reason for neglecting the crossover operation is more on the side of the algorithm’s strategy than the coding mode.

In case of the population that moves concentrated through the search space, the differences in the characteristics of the two given ancestral individuals are in general not large. Therefore the crossover defined either similarly as in GA - as the recombination of the individuals genome (i.e. some characteristics would be inherited from one predecessor and some from the other), or as the mean of the characteristics, would rather be a stabilizing factor acting in opposition to mutation operator. Defining the crossover operator in such manner that it would ensure the populations’ diversity is of course possible also in case of the real coded algorithm. It would however result in total change of the algorithms’ strategy and would reduce it to some version of the genetic algorithm.

The other type of the evolutionary operators is selection, which realizes the “survival of the fittest” concept, i.e. biases the search towards high-quality solutions by letting the individuals with higher fitness value exercise the right to breed their descendants into the next generation. In case of the presented approach the inaccuracy of the selection scheme is the main factor providing the continuous evolutionary approach with the ability to cope with multimodal functions. The inaccuracy of the selection scheme enables the population (which exhibits a poor diversification) to escape the area of the local optimum’s gravity by temporary deterioration of the populations’ fitness value. That mechanism is more effective if the algorithm works with small populations. Therefore in case of this approach the small populations are advantageous in contrast to the GA [5], while in case of simple unimodal functions this is obvious due to the fact that good results can be obtained even with the aid of few individuals.

All the investigated algorithms work with small population of eight real-coded individuals. The performance of the EA of such kind applied to the IM equivalent circuit parametric identification at standstill has been investigated in advance and the results of those investigations are presented in [11]. The EA, which preformed best in that task has been chosen as the basic algorithm, to which the adaptive mutation scheme was then introduced. Few versions of EA with adaptive mutation have been tested in this research and compared to the performance of the EA with classical Gaussian mutation, which constituted the basis for the further modifications.

Each individual in the algorithms is represented by the vector of five characteristics corresponding to the values of five identified parameters. The fitness function is defined by the sum of squares of the differences in the computational current responses of the model, with the values of induction motor parameters equal the values of individual’s characteristics, and experimental current response (12). The objective function is minimized.

The vector of starting parameters for the algorithm can be the values of parameters estimated from the nominal data of the IM. The value of individuals characteristics are in the range of 5 times bigger to 5 times smaller the value of the starting parameters. The specific boundaries of the characteristics were taken into account, i.e. each parameter of the motor equivalent circuit must be greater than zero and the mutual inductance \(x_{ij}\) must be smaller than the stator and rotor inductance \((x_s, x_r)\).

In the presented algorithms the only variation operator was mutation (applying the crossover operation is pointless in the accepted approach). In the basic EA the mutation is a small normally distributed random number.

While introducing the adaptive mutation scheme to the EA algorithm the focus was on improving the algorithm’s time-efficiency. In spite of the fact that EA is a relatively fast computational algorithm, it is still desirable to reduce the processing time in the view of the industrial implementation of the IM parameters identification procedure. Since the basic EA is an algorithm with greedy selection scheme (yet sufficient in case of the considered problem as the previous investigations revealed [11]) and is working with small population of individuals, the processing time mainly depends on the mutation operator. It is obvious that for such a type of an algorithm large mutation value is advantageous when the population is far from the optimum and it should decrease when the population approaches the vicinity of the optimum. However, it also has to be taken into consideration that in the experimental application the reference function is disturbed by noise, whose filtration is not always possible and the mutation value should not be stiffly and significantly decreased.

Few versions of the strategies for changing the mutation value during the algorithms execution were tested:

- EALA (Evolutionary Algorithm with Linear Adaptation) – the evolutionary algorithm EA with linear mechanism of adapting the mutation,
- EAEA (Evolutionary Algorithm with Exponential Adaptation) – the evolutionary algorithm EA with exponential scheme of adapting the mutation,
- EATA (Evolutionary Algorithm with Threshold Adaptation) – the evolutionary algorithm EA with threshold adaptive mutation
- EASA (Evolutionary Algorithm with Simulated Annealing) – the evolutionary algorithm EA with the adaptive mutation mechanism by means of classical Simulated Annealing algorithm.

In the case of EALA, the mutation changes during the algorithm’s execution according to the formula:

\[ \sigma_{i+1} = b \frac{\sum_{k=0}^{n} f_k \left( \sigma^i \right)}{n} \]  

(13)

where: \( \sigma \) – the normal distribution variance, \( b \) – a constant, \( n \) – the number of individuals in the population, \( f \) – the individual’s fitness.
The strategy for changing the mutation in EAEA is as follows:

\[ v^{(i+1)} = c \exp \left( \frac{1}{n} \sum_{x=1}^{n} f(x) \right), \quad (14) \]

where \( c \) – a constant and other symbols are as in (13).

The EATA algorithm proceeds the following adaptation scheme: the mutation value is decreased in each generation by some given coefficient. If the collective fitness of the population’s generation decreases comparing to its fitness obtained with the previous mutation value (notice that the fitness function is minimized), then the new mutation value is accepted otherwise it is rejected. The fitness value can, however, grow if the sum of population fitness does not decrease for few generations.

In the EASA the adaptive mutation mechanism is performed by means of a classical simulated annealing algorithm [7], [8]. The objective function for the simulated annealing algorithm is the sum of the population’s fitness in each generation. In each generation changes are introduced to the mutation value. If the sum of fitness values of all individuals corresponding to this mutation decreases, the change is always accepted. Otherwise it is either accepted or rejected with the probability typical for the simulated annealing algorithm:

\[ p \left( \Delta \sum f_i(x') \right) = \exp \left( -\frac{\Delta \sum f_i(x')}{k_B T} \right) \quad (15) \]

where: \( T \) – the temperature profile, \( k_B \) – a constant, the other symbols are as in (13), (14).

The flow diagram of EASA algorithm is shown in Fig. 2.

The stop condition was identical in case of all the algorithms. The calculations were stopped when the average fitness function value of the population dropped below the small assumed threshold value.

III. SIMULATION TESTS

All the algorithms were tested in the same conditions. The reference model for the identification procedure was the mathematical model of the IM (6)–(11) commissioned with parameters obtained from the idle-running and short circuit tests for the SDChm 180M6/24 motor of 5.5kW. It has also been taken into account in simulations that the motor is fed from the inverter, as it is in the real applications. The algorithms were developed in C++ and executed in the regular PC computer.

The results of simulation tests are presented in Tables I–V. In each table the values of the assumed parameters of the IM equivalent circuit and the values of the parameters calculated by means of the respective EA as well as the absolute maximal, minimal error and standard deviation of the obtained results are presented. The time results, i.e. average, maximal, minimal computation time and the deviation are also given.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>SIMULATION RESULTS - ALGORITHM EA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference parameters</td>
<td>Identified parameters</td>
</tr>
<tr>
<td>( r ) [p.u.]</td>
<td>0.05743</td>
</tr>
<tr>
<td>( r ) [p.u.]</td>
<td>0.03694</td>
</tr>
<tr>
<td>( x ) [p.u.]</td>
<td>2.10666</td>
</tr>
<tr>
<td>( x ) [p.u.]</td>
<td>2.09721</td>
</tr>
<tr>
<td>( x_d ) [p.u.]</td>
<td>2.02170</td>
</tr>
<tr>
<td>Time results</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>61.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>SIMULATION RESULTS - ALGORITHM EALA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference parameters</td>
<td>Identified parameters</td>
</tr>
<tr>
<td>( r ) [p.u.]</td>
<td>0.05743</td>
</tr>
<tr>
<td>( r ) [p.u.]</td>
<td>0.03694</td>
</tr>
<tr>
<td>( x ) [p.u.]</td>
<td>2.10666</td>
</tr>
<tr>
<td>( x ) [p.u.]</td>
<td>2.09721</td>
</tr>
<tr>
<td>( x_d ) [p.u.]</td>
<td>2.02170</td>
</tr>
<tr>
<td>Time results</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>13.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>SIMULATION RESULTS - ALGORITHM EAEA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference parameters</td>
<td>Identified parameters</td>
</tr>
<tr>
<td>( r ) [p.u.]</td>
<td>0.05743</td>
</tr>
<tr>
<td>( r ) [p.u.]</td>
<td>0.03694</td>
</tr>
<tr>
<td>( x ) [p.u.]</td>
<td>2.10666</td>
</tr>
<tr>
<td>( x ) [p.u.]</td>
<td>2.09721</td>
</tr>
<tr>
<td>( x_d ) [p.u.]</td>
<td>2.02170</td>
</tr>
<tr>
<td>Time results</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>19.8</td>
</tr>
</tbody>
</table>
TABLE IV  
SIMULATION RESULTS – ALGORITHM EATA

<table>
<thead>
<tr>
<th>Reference parameters</th>
<th>Identified parameters</th>
<th>Absolute error [%]</th>
<th>Maximal absolute error [%]</th>
<th>Minimal absolute error [%]</th>
<th>Error s.d. [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_s ) [p.u.]</td>
<td>0.05743</td>
<td>0.012</td>
<td>0.131</td>
<td>0.000</td>
<td>0.030</td>
</tr>
<tr>
<td>( r_l ) [p.u.]</td>
<td>0.03694</td>
<td>0.018</td>
<td>1.866</td>
<td>2.606</td>
<td>0.062</td>
</tr>
<tr>
<td>( x_d ) [p.u.]</td>
<td>2.10666</td>
<td>0.036</td>
<td>12.788</td>
<td>0.018</td>
<td>0.283</td>
</tr>
<tr>
<td>( x_d ) [p.u.]</td>
<td>2.09721</td>
<td>0.020</td>
<td>1.848</td>
<td>7.492</td>
<td>0.018</td>
</tr>
<tr>
<td>( x_d ) [p.u.]</td>
<td>2.02170</td>
<td>0.043</td>
<td>9.591</td>
<td>1.279</td>
<td>0.018</td>
</tr>
</tbody>
</table>

Time results

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>Max</th>
<th>Min</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time [s]</td>
<td>17.5</td>
<td>29.0</td>
<td>5.0</td>
<td>5.7</td>
</tr>
</tbody>
</table>

Despite the fact that from the experimental application point of view the accuracy is not of the main concern, due to the fact that it is anyway limited by the simplifications in the assumed model and measurement noise, the deterioration in accuracy due to mutation mechanism should, however, be within the reasonable range. Nevertheless in the case of all tested algorithms the impact of the adaptive mutation on the accuracy was on the acceptable level - the average fitness function value dropped below the assumed small threshold.

In the algorithms evaluation the focus was on time efficiency. Application of all the investigated adaptive mutation mechanisms provided with the visible improvement of the computational time efficiency. Best results were obtained using EASA algorithm and thus the EASA algorithm was selected for the experimental application. Its performance in the experiment was compared with the performance of the basic EA algorithm.

IV. EXPERIMENTAL TESTS

The experimental setup consisted of: PC computer (with DS1103 PPC/DSP), the ST80X-2C induction motor, PWM inverter and LEM measurement converters. The setup scheme is shown in Fig.3.

The inverter was controlled to ensure the connection of the stator windings such as presented in Fig.1, and the step change of the stator voltage by means of PWM control. The idea of controlling the inverter is presented in Fig. 4 and the stator current responses, measured and calculated, are demonstrated in Fig. 5. The application worked in the harsh environment such as presented in Fig. 5, where the measured current response is shown.

The stator current responses calculated from the mathematical model (6)–(11), commissioned either with the motor parameters obtained by means of the idle-run and short circuit tests or by means of the EA identification procedure are also demonstrated in this figure.
Due to the fact that only the stator resistance $r_s$ can be easily and accurately measured, its value is given in the Table VI for the purpose of comparison. The other IM parameters can be determined by means of the short circuit and idle-running tests, but their values are rather rough, so were not placed in this table. As results from the Table VI, the identification of IM parameters using the modified EASA algorithm with adaptive mutation is much less time consuming comparing to the basic EA.

### V. CONCLUSIONS

Few versions of evolutionary algorithms with adaptive mutation for the identification of induction motor equivalent circuit (EALA, EAEA, EATA and EASA) were proposed. The performance of those algorithms as well as the performance of the algorithm with classical Gaussian mutation EA, was investigated in simulations. The focus was on the time efficiency.

Best results have been obtained for the EASA algorithm, which employed the simulated annealing algorithm as the mechanism for adapting the mutation. The performance of the algorithm EASA was also verified in experimental tests and compared with the performance of the basic algorithm EA. Both algorithms EA and EASA provided satisfactory results as far as the accuracy is concerned, but the identification procedure using EASA algorithm was significantly less time-consuming, what is particularly advantageous from the experimental application’s point of view.

### ACKNOWLEDGMENT

This research work was supported by the Ministry of Science and Higher Education (Poland) under Grant N510 039 31/2440 (2006-2008)

### REFERENCES


