A New Genetic Algorithm for Motor Parameter Estimation

ONOUFRIOS LADOUKAKIS, STEFANOS TSITMIDELIS and APHRODITE KTENA
TEI of Chalkis, Psahna, Euboea, 34400
GREECE

Abstract: - The genetic algorithms are being increasingly used in motor parameter estimation because they are based on simple models while offering high accuracy results, stable solutions and high convergence speeds. The proposed algorithm uses the binary tournament parent selection method, a dynamic population table with an elitist strategy and a special 6-point crossover operator. It has been tested against data from three different induction motors. The agreement between the theoretical and calculated values of the parameters is very good and superior to results existing in the literature.

Key-Words: - steady state technique, 6-point crossover, induction motors

1 Introduction
In the last two decades the demand for higher model accuracy in induction motors has increased due to the more stringent requirements in power control. The problem of induction motor parameter estimation has been addressed by several researchers both in the academic and industrial community and from several points of view. The classical electrotechnical techniques utilize the approximate equivalent circuit of the motor to extract its parameters. The low accuracy of these techniques has led to improved solutions such as using a more accurate equivalent circuit, or applying standstill data [2]. Recently, techniques employing neural networks and FEM have been used [3, 4], while combinatorial methods such as the application of genetic algorithms have been attracting the interest of researchers. In this latter case, the motor is modeled by a simple equivalent circuit described by a set of nonlinear equations which can be solved in a classical manner, as is the use of the Newton-Raphson method [5] which however presents convergence problems and is dependent on the initial conditions. Several researchers have addressed this problem by employing genetic algorithms in the solution of these nonlinear systems which have been proven to be more stable [6, 7]. In this paper, we use the method suggested in [7] to obtain the objective function with a new genetic algorithm and compare it against results using more classical approaches in genetic algorithm implementation [6] or neural network solutions [5].

2 The problem
The simplified Thévenin equivalent circuit, shown on Fig. 1, models the induction motor as a series circuit in terms of the equivalent stator resistance $R_{TH}$ and reactance $X_{TH}$ and the rotor resistance $R_2$ and leakage reactance $X_2$. A set of fault equations between the theoretical and actual values can be extracted regarding the full load torque, $T_{FL}$, the locked rotor torque, $T_{LR}$, and the breakdown torque, $T_{BD}$ (Eqs. 1-3):

$$F_1(R_{TH}, R_2, X_L) = \frac{V_{TH}^2 R_2}{\alpha \left( R_{TH} + R_2 \right)^2 + X_L^2} - T_{LR} \quad (1)$$

$$F_2(R_{TH}, R_2, X_L) = \frac{V_{TH}^2 R_2}{\text{sec} \left( \frac{R_{TH} + R_2}{8} \right)^2 + X_L^2} - T_{FL} \quad (2)$$

Fig. 1 - Thévenin equivalent circuit of an induction motor.
For calculation simplicity purposes, the leakage reactances $X_{TH}$ and $X_2$ are combined into one term $X_L$ which is their sum. This is allowed because once the motor’s class is known, after the solution is reached, the rotor and stator reactances can be easily calculated. Also, in the above formulation, the shunt magnetizing reactance, $X_M$, which is the mutual inductance reactance due to the magnetic interaction of the stator and rotor fields is considered to be much larger than $X_1$ and $X_M + X_1 \gg R_1$ so that the following simplifying equations can be applied:

$$R_{TH} \approx R_1 \left( \frac{X_M}{X_1 + X_M} \right)^2$$  \hspace{1cm} (4)

$$X_{TH} \approx X_1$$  \hspace{1cm} (5)

The purpose now is to determine, by a suitable method, the parameters of the equivalent circuit shown in Fig.1 that minimize the fault values defined by Eqs 1-3. The method of genetic algorithms is chosen to accomplish this task mainly because of its stability and its ability to overcome local maxima.

### 3 The genetic algorithm method

Genetic Algorithms (GA) are global optimization algorithms and are used for global search purposes in applications. In the last two decades, researchers tend to find out ways of optimization that can be used to simulate specific processes in nature and the human thought. The artificial neural networks and the fuzzy logic systems also fall under this category. The genetic algorithms are based on principles inspired by the real mechanisms of Genetics and the development of genres that rule the natural biological systems.

The prevalent rule in nature in the selection procedure of individuals for reproduction is the survival of the fittest. Reproduction in such systems takes place when two parents mix their genetic stuff to obtain new individuals with characteristics of both of their parents. The recombination of the parents’ genetic features, with a small but not zero possibility of mutation due to outside environment reasons, is the way to give the possibility of new improved or worse characteristics in the created descendants. However, in accordance with the rule of the survival of the fittest, evolution comes always in the direction of continuous improvement of the population, making them more viable in the changes of environment.

Genetic algorithms attempt to simulate exactly these population dynamics, targeting not so much at the modeling and understanding of natural biological systems but at the search for and development of solutions in problems of optimization and learning. A typical GA is prescribed by five simple parameters: the starting population, the convergence criterion, the parents’ selection method, the mutation effect and the crossover cutting points. Every individual of the population is stored in a coded form as is the DNA coding scheme in chromosomes. This way, the GA is not “aware” of what is doing but it just does the job – and hopefully in the right, “natural” manner, according to the object function which encodes the description of the problem it is called to solve. Therefore, every solution is coded in a bit string symbol. The simplest code form, in this sense, is the binary set of symbols: (0, 1). The next step is to decide on the starting population, which will be stored in a single array, and on the loop process that will operate on the population measuring their ‘strength’ by a fitness function. The fitness function will then assign a ‘fitness value’ to each string. The solution is reached when through the advancement and mating of ‘strong’ strings with other ‘strong’ strings only one wins because it is the fittest. This loop varies from case to case depending on the programming technique applied to the algorithm.

Fig.2 shows the flow chart of the GA used in this work. The parent selection is accomplished using the binary tournament method instead of a proportional selection method that is more commonly used, like the roulette wheel method. Crossover and mutation are the main genetic operators. The crossover recombines parts of two individual parent strings in order to create a new individual child string which inherits characteristics from both parents. Crossover is the main operator responsible for solution search but it cannot produce information that does not already exist in the population, e.g. applying the crossover operator between two identical strings will give the same string again. The simplest use of this operator is a single-point crossover with a single crossing site. Improved crossover operators can also be used, such as a multi-point crossover, a uniform crossover, a shuffle crossover or a reduced surrogate crossover [5].

Mutation comes to complement crossover because it can produce new information within the population. This operator randomly inverses the state of a bit from 0 to 1 or 1 to 0. In that way, mutation changes symbols at random places of the new-born child individual, simulating the random mutations of
biologic chromosomes caused by environment effects. Even though it is not mandatory to apply crossover and/or mutation, the probability for crossover is around 0.8 and around 0.01 per bit for mutation.

![Diagram of the implemented GA](image)

### 4 The implementation of the GA

The GA method described in part 3 is now applied to the problem stated in part 2. The population chosen consists of 450 individuals randomly selected. The string used is shown in Fig. 3 and consists of the three parameters, $R_{TH}$, $R_2$ and $X_L$. Because the maximum value each parameter can assume is $10 \, \Omega$, with an accuracy of three decimal points, a 14-bit word length is needed, allowing for a maximum value of $16,384 \, \Omega$. Therefore, the string length is 42 bits. Then, a 2-point crossover is applied to each parameter which results in a 6-point crossover in total. The mutation effect is set to 2-7 bits per string while the convergence criterion is 50000 new individuals.

![String Diagram](image)

The objective function of the algorithm is the set of the three torque error functions (Eqs. 1-3) that describe the problem that must be solved. The error function that must be minimized in order to determine the three parameters of the induction motor is given as the sum of the squares of the torque error functions [5]:

$$
\varepsilon = F_1(\ldots)^2 + F_2(\ldots)^2 + F_3(\ldots)^2
$$

(6)

Then, the fitness function is defined as the inverse of the error function:

$$
\text{Fitness} = \frac{1}{\varepsilon}
$$

(7)

The fitness function is responsible for evaluating the ‘strength’ of each string. The aim of the genetic algorithm is to minimize the error, i.e. to maximize the fitness [5-7].

The population table, in the proposed GA, is dynamically refreshed according to the steady-state technique. This means that the population size is kept constant with the population being continuously refreshed by the fittest child-strings that are the offspring of the strongest parent mating procedure. This type of population table has been chosen because it is the result of a by default elitist procedure ensuring that no good solutions are going to be discarded. Moreover, it usually offers better convergence than the classical implementation of a typical generation-based GA.
5 Results and discussion
The performance of the new GA was tested against the known parameters of the three motors reported in [5] and [7] and the results were compared.

The parameters are summarized in Table 1.

<table>
<thead>
<tr>
<th>HP</th>
<th>rpm</th>
<th>V (Ω)</th>
<th>R1 (Ω)</th>
<th>R2 (Ω)</th>
<th>XL (Ω)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>175</td>
<td>230</td>
<td>0.43</td>
<td>0.30</td>
<td>2.51</td>
</tr>
<tr>
<td>50</td>
<td>170</td>
<td>460</td>
<td>0.08</td>
<td>0.22</td>
<td>0.60</td>
</tr>
<tr>
<td>50</td>
<td>177</td>
<td>230</td>
<td>0.26</td>
<td>0.18</td>
<td>2.41</td>
</tr>
</tbody>
</table>

The GA was applied to all three motors and the results are shown in Table 2. The last column in Table 2 shows the number of runs required for the convergence of the GA.

It can be easily seen that the agreement between the actual and the calculated values is very good and superior to either the classical GA implementation [6], the neural network approach or the Newton-Raphson method [5].

<table>
<thead>
<tr>
<th>HP</th>
<th>rpm</th>
<th>V (Ω)</th>
<th>R1 (Ω)</th>
<th>R2 (Ω)</th>
<th>XL (Ω)</th>
<th># of runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>175</td>
<td>230</td>
<td>0.43</td>
<td>0.30</td>
<td>2.51</td>
<td>98</td>
</tr>
<tr>
<td>50</td>
<td>170</td>
<td>460</td>
<td>0.08</td>
<td>0.22</td>
<td>0.60</td>
<td>73</td>
</tr>
<tr>
<td>50</td>
<td>177</td>
<td>230</td>
<td>0.26</td>
<td>0.18</td>
<td>2.41</td>
<td>9</td>
</tr>
</tbody>
</table>

5 Conclusion
In conclusion, the proposed GA, with the binary tournament selection method, the dynamic population table and the 6-point crossover operator offers some definite advantages over previous implementations. It is fast, reliable and yields more accurate solutions.

Acknowledgments
This work was jointly funded by the European Community Fund and the Hellenic Ministry of Education.

References: