XX IMEKO World Congress
Metrology for Green Growth
September 9–14, 2012, Busan, Republic of Korea

PERFORMANCE EVALUATION OF GENE EXPRESSION PROGRAMMING IN IMPEDANCE SPECTROSCOPY FOR SENSOR MODELLING

Pedro M. Ramos[1] *and Fernando M. Janeiro*[2]

[1]Instituto de Telecomunicações, Instituto Superior Técnico, UTL, Portugal, pedro.m.ramos@ist.utl.pt
[2]Instituto de Telecomunicações, Universidade de Évora, Portugal, fmtj@uevora.pt

Abstract: This paper describes the recent improvements in the use of gene expression programming and genetic algorithms in impedance spectroscopy, namely in sensor modelling. Insight is given in the inner workings of the gene expression programming while highlighting the proposed improvements. The performance of the improved algorithm is also analyzed. It is further validated by the successful application to measurements of a real sensor.

Keywords: impedance spectroscopy, impedance measurements, circuit identification, sensor characterization.

1. INTRODUCTION

Impedance spectroscopy has many uses in science, ranging from electrochemical [1] to biomedical applications [2], among others [3]. The modelling of sensors is an important field where impedance spectroscopy is used. The first step consists on measuring the impedance response of the sensor, for example using an impedance vector-analyzer or using a two-channel data acquisition system and applying a sine-fitting algorithm [4]. The sensor can be modelled from this data, but usually a priori knowledge of the physical principles and inner workings of the sensor is needed. Even when the equivalent circuit topology is known, the problem of finding the component values remains. This problem is usually addressed with the Complex Non-Linear Least Squares (CNLS) method [5] which has disadvantages such as the need for many measurements in a wide frequency range. Methods using bilinear transformations [6] or Genetic Algorithms (GA) [7] have been recently proposed to obtain the circuit component values when the circuit topology is known [8]. The search for a circuit topology that fits a measured impedance can be performed using Gene Expression Programming (GEP) [9], which was developed for the automatic creation of computer programs. A coding procedure for electric circuits in GEP was developed in [10] and was efficiently used, with GA, in impedance spectroscopy [11].

In this paper, the GEP implementation for electric circuit modelling is improved by the introduction of an automatic simplification routine that reduces the complexity of the circuit by combining elements of the same type that appear in series or parallel. The performance of GEP, as a method to identify impedance equivalent circuits, is analyzed from the point of view of evolution of fitness, type and number of possible circuits that can be modelled. Finally, the GEP+GA algorithm is applied to measurements of a viscosity sensor to evaluate its performance in real measurement conditions.

2. GENE EXPRESSION PROGRAMMING

Gene expression programming is an evolutionary algorithm that is used together with a genetic algorithm to find an equivalent circuit and respective parameters that fit the measured frequency response of a sensor or impedance. In GEP, a given electric circuit is encoded as an expression tree which can be translated into a linear gene. This is known as genotype/phenotype separation [9].

The binary tree includes nodes and leafs, where the nodes contain the operations series (+) and parallel (//) and the leafs contain the circuit components: resistances, inductors and capacitors (coded as 1, 2, and 3 respectively). The translation to the linear gene is done by traversing the tree from left to right in level-first fashion. Since each linear gene must generate a valid expression tree and electric circuit, the genes must have a head of size h and a tail of size $t = h + 1$ [9]. The head may contain operators and components, while the tail can only contain components.

In GEP, a population of N linear genes is randomly created according to the $t = h + 1$ rule (ensuring the validity of the tree and respective circuit). A genetic algorithm is then applied to each candidate circuit (gene) to find the circuit component values that minimize its fitting error relative to the measured impedance response. The genetic algorithm is needed due to the large search space and the existence of many local minima in the fitness function. The circuit genes are then ordered as function of fitness (lowest to highest fitting error) and a set of GEP operations are applied to create a new generation of circuits. The fittest circuits have a higher probability of propagating their characteristics to the new population. Therefore, on average, as the generations evolve the fitness of the population will improve. This is the well known *survival of the fittest* principle for population evolution. In addition, elitism is used and therefore the fittest gene of each generation always moves on to the next generation ensuring that the best element of each population never gets worse.

The GEP operations used to create a new generation are: replication, mutation, transposition and recombination [9]. In replication, genes are copied to a new intermediate population based on their fitness. Mutation is then applied to this population and consists on randomly changing a position on some genes (taking care to maintain the head/tail rule). Transposition consists on transferring part of a gene to an earlier position in the gene. Recombination is applied to pairs of genes by exchanging part of their code resulting in two offspring genes.

These operations are applied sequentially to the whole gene population and can greatly modify the tree and circuit topologies. Also, they are easily applied to the genes but would be difficult to apply to the expression trees, thus explaining the need to separate the genotype and phenotype of the circuit.

GA is then applied to the new population of circuit topologies to find the circuit component values and respective fitting error. This procedure is repeated until the fitting error drops below a previously defined threshold (which depends on the measurement noise) or the maximum number of iterations is reached, in which case the algorithm failed to find a suitable circuit.

The GEP+GA algorithm estimates and minimizes the fitness function

$$\varepsilon = \frac{1}{P}\sum_{p=1}^{P}\frac{\left|Z_m(\omega_p) - Z_{est}(\omega_p)\right|^2}{\left|Z_m(\omega_p)\right|^2} \qquad (1)$$

where P is the number of measurements, $Z_m(\omega_p)$ is the measured impedance and $Z_{est}(\omega_p)$ is the GEP+GA estimated impedance at angular frequencies ω_p.

When compared to the previous version of the algorithm [11], the main evolution of this current version is the inclusion of an automatic simplification routine. This simplification is done before the GA step and consists on sweeping the tree to find similar component elements connected either in series or in parallel and replace them with a single one. For example, for the tree represented in Fig. 1, L_1 and L_2 are of the same type, but they are not in parallel or in series with each other and thus can be simplified as is clear from the circuit also shown in Fig. 1. On the other hand, R_1 and R_2 are connected in series and should be simplified. The simplification algorithm eliminates R_2 and moves the parallel of R_3, L_2 up one level as shown in Fig. 2. Since, before the GA is applied, no values are assigned to the components, no further steps are needed. However, in order to ensure that the resulting gene is valid (which it must be, since it is a simplified version of an already valid gene), gene reconstruction is done by ordering all tree branches according to the depth of the last operation in each branch. This simplification makes the GA algorithm more effective because it reduces the number of degrees of freedom in the GA.

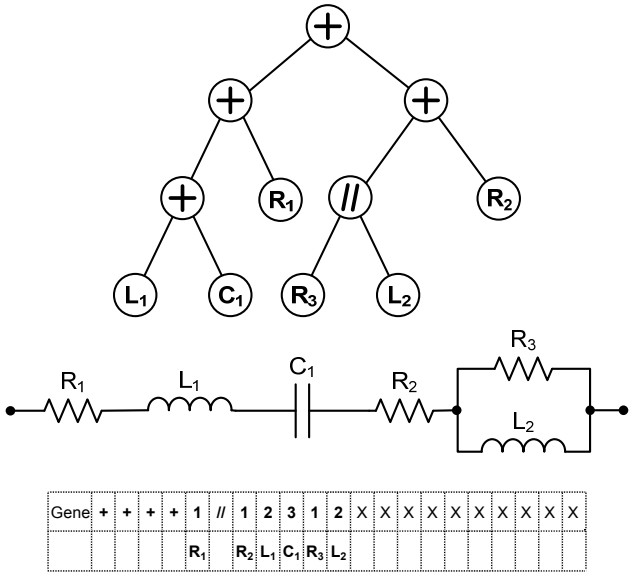

Fig. 1 – Example of a circuit, tree and gene before simplification. Notice that R_1 and R_2 are connected in series and can be replaced by a single resistor. X's represent the non-coding region of the gene.

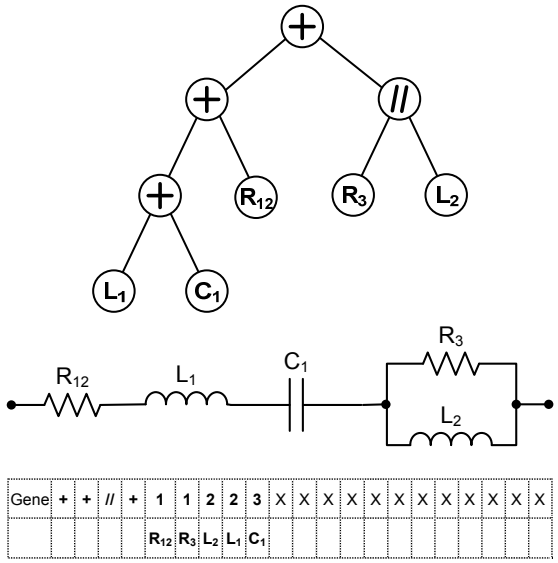

Fig. 2 – Simplification of the circuit/tree/gene of Fig. 1. Notice that the R_{12} has replaced R_1 and R_2 and thus reduced the effective length of the gene (it was 11 elements long in Fig. 1 and here, it is 9 elements long).

3. PERFORMANCE ANALYSIS

To analyze the performance of the proposed approach, the circuit represented in Fig. 3 was simulated for P = 1000 frequency points, equally spaced from 100 Hz up to 10 kHz. This circuit has a resonance near 5 kHz.

To take into account the uncertainty of measurements, random values were added to the impedance magnitude and phase according to Gaussian distributions with standard deviation of 0.08% and 0.05° respectively. These values are similar to the uncertainty values of a commercial impedance measurement device.

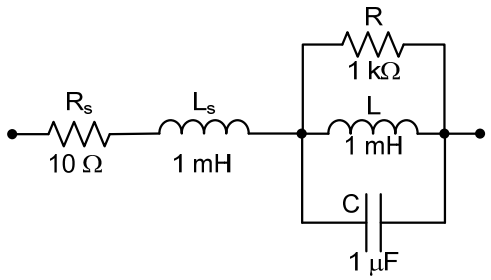

Fig. 3 – Circuit used to assess the performance of the GEP and GA combined algorithm.

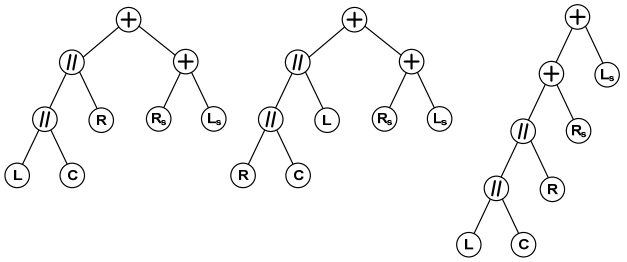

Fig. 4 – Three different trees that represent the circuit of Fig. 3 are shown. For this particular circuit, 9 different trees correspond to the same circuit.

For the circuit represented in Fig. 3 and even with the tree/gene rules detailed in section 2, there are 9 different trees/genes that represent the same circuit. Three of those are shown in Fig. 4. The threshold to stop the algorithm is set to 2×10^{-6}, the GEP population size is 20 and the maximum number of iterations is 50.

The GEP and GA procedure was repeated 1000 times and the results were registered and analyzed. In Fig. 5, one of the identified circuits is presented. The difference to Fig. 3 is the inductance L_3 which does not affect the impedance frequency response at the considered frequencies.

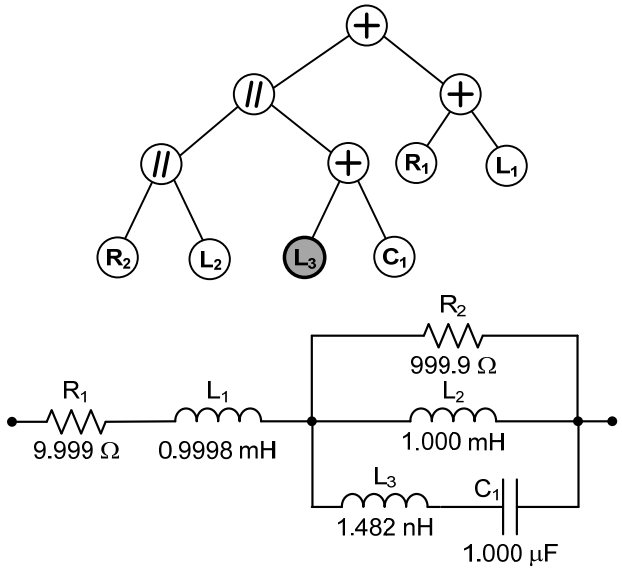

Fig. 5 – Tree and circuit of one of the circuits identified by the GEP+GA procedure. Notice that the only difference to the input circuit is the inclusion of L_3 which, with its value and at the considered frequencies, does not significantly influence the impedance frequency response.

In Fig. 6, the percentage of occurrence of the estimated effective gene length is shown. The algorithm has a tendency to overestimate the gene length (the correct gene length is 9) by adding additional components that do not significantly change the overall impedance response (see for example, the case shown in Fig. 5).

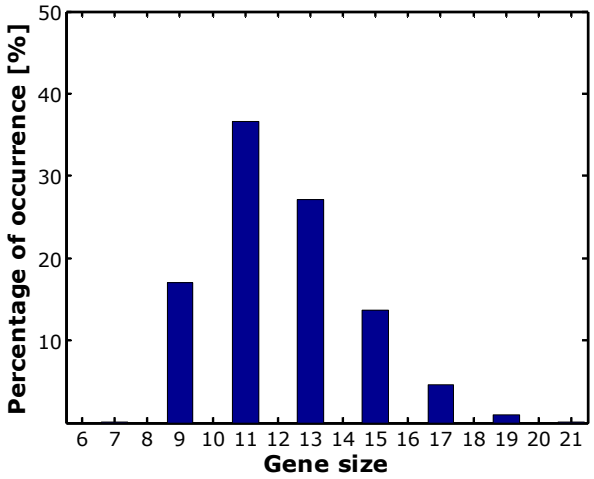

Fig. 6 – Percentage of occurrence of the gene size for 1000 runs of the GEP+GA algorithm.

The final fit error for the 1000 runs is depicted in Fig. 7. All of the cases where the final fit error is above the preset threshold (2×10^{-6}) correspond to non convergence situations. The algorithm converged in 95.4% of the runs. Notice that setting the threshold lower would not ensure better results due to the noise added to simulate measurement uncertainty.

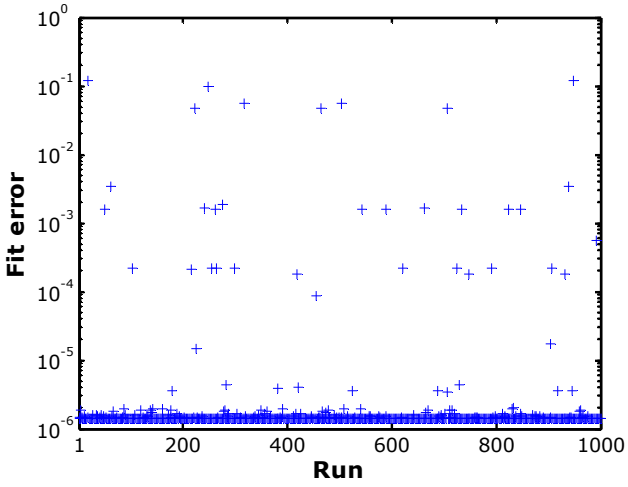

Fig. 7 – Final fit error of the 1000 runs. The algorithm converged in 95.4% of the runs.

The evolution of the fit error for four different runs is shown in Fig. 8. In run D, the algorithm did not converge because after the 50th iteration, the error was still above the preset threshold. In the other runs, the algorithm converged while following different error evolutions and also requiring different number of iterations to do so. The monotonicity is a direct result of the elitism used in the GEP procedure.

Although the algorithm converged in 95.4 % of the runs, it should be noted that the quantity of circuits that can be

represented is very, very large. As shown in Fig. 9, the number of possible combinations of a gene with $h = 10$ head elements is 1.73×10^{12}. Although all of these combinations are possible, they correspond (after the automatic simplification) to about 10^6 different valid circuits and in non-convergence situations, only 1000 circuits are actually tested at most (50 iterations and GEP population of 20).

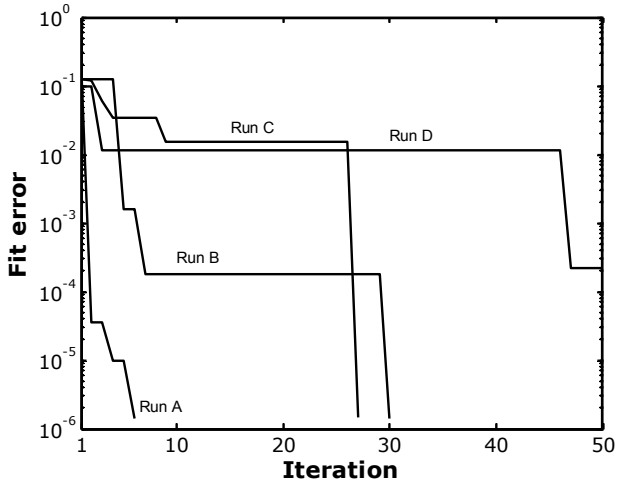

Fig. 8 – Evolution of the fit error as a function of the iteration for four different runs. In run D, the algorithm did not converge, while on the other three it reached the preset threshold (although not necessarily the correct circuit).

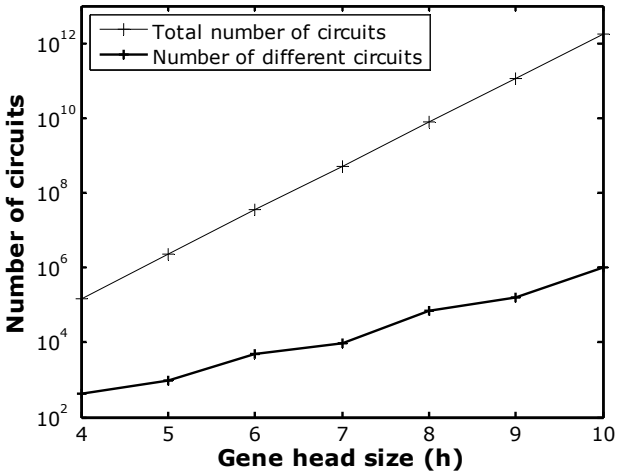

Fig. 9 – Number of circuits described by the gene as a function of the gene head size. The thin line represents the total number of combinations (5^h3^t) and the thicker line represents the actual number of different circuits (due to the tail elements not used in the actual gene coding and circuit simplifications).

4. SENSOR APPLICATION

In this section, the algorithm is applied to measurements of a vibrating wire viscosity sensor [12]. The vibrating wire is immersed in the liquid whose viscosity is to be measured. The viscosity can be obtained by measuring the resonance characteristics of the sensor which depend on the viscosity of the liquid. Since this sensor is based on electrochemistry is it necessary to include constant-phase elements in the list of circuit components. The equivalent impedance of the CPE is characterized by parameters Q and n according to

$$Z_{CPE}(\omega) = \frac{1}{Q\omega^n} e^{-\frac{\pi}{2}ni} . \quad (2)$$

Measurements were performed for liquid diisodecyl phthalate (DIDP) at 20ºC. The frequency response of the sensor, under these conditions, was obtained for $P = 21$ points in the range [0.5; 1.5] kHz, as shown in Fig. 10 (squares for the magnitude and circles for the phase). The resulting circuit and corresponding component values yielded the frequency response shown by lines in Fig. 10. There is an excellent agreement between the measurements and the GEP equivalent circuit impedance. The fitting error was, in this case, $\varepsilon = 0.000097\%$.

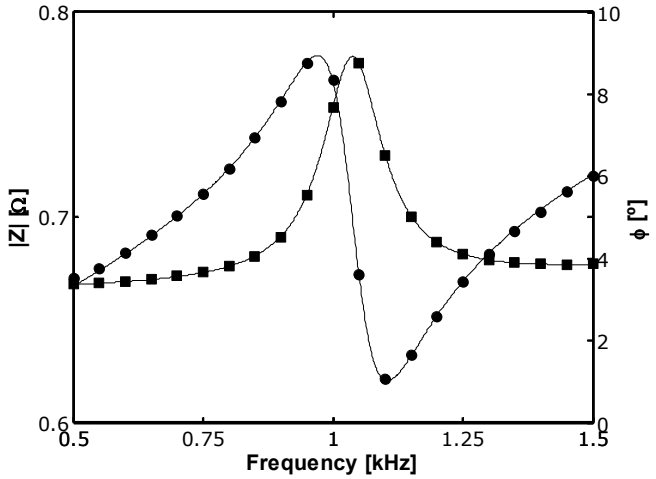

Fig. 10 – Measured impedance magnitude (squares) and phase (circles) of the viscosity sensor with DIDP at 20ºC. Lines represent the impedance magnitude and phase of the GEP estimated circuit.

The Nyquist diagram of the sensor measured impedance and that of the impedance obtained with the GEP is shown in Fig. 11.

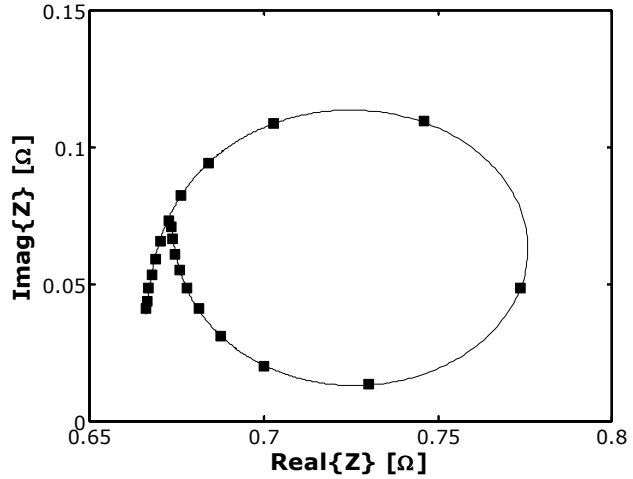

Fig. 11 – Nyquist diagram of the viscosity sensor measured impedance (squares) along with the impedance of the GEP equivalent circuit (line).

The equivalent circuit obtained by GEP is shown in Fig. 12 along with the expression tree that generated it. Resistances R_1, R_3 and the inductor L_2 model the connecting wires while the four component parallel circuit model the vibrating wire inside the viscous liquid.

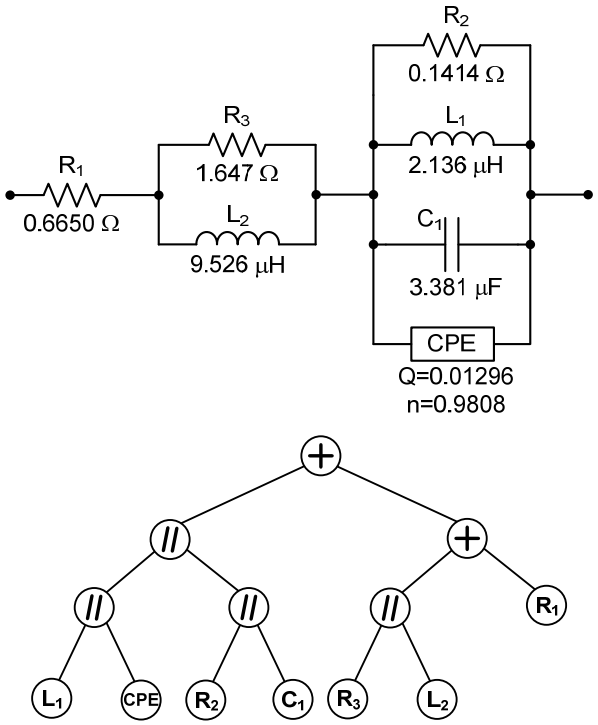

Fig. 12 – Equivalent circuit and corresponding expression tree obtained by GEP+GA algorithm for the viscosity sensor immersed in DIDP liquid.

5. CONCLUSIONS

In this paper, an improved version of gene expression programming for impedance spectroscopy was developed. The improvement consists on performing an automatic simplification of the circuit gene before searching for the circuit component values with a genetic algorithm. The simplification routine consists on combining the circuit components of the same type that are in series or parallel, thus reducing the complexity of the gene. The performance of the algorithm was analyzed as a function of gene length. It was also shown that the algorithm converges, within 50 iterations, in 95.4% of the cases. The proposed algorithm was then validated by the successful characterization of a vibrating wire viscosity sensor.

Work in a more extensive simplification procedure that will reduce the immense diversity of equivalent circuits for a given impedance frequency response is underway.

ACKNOWLEDGMENTS

This work was sponsored by the Fundação para a Ciência e Tecnologia

6. REFERENCES

[1] J. A. González, V. López, A. Bautista, E. Otero, X. R: Novoa, "Characterization of porous aluminum oxide films from a.c. impedance measurements", J. Appl. Electrochem., vol. 29, pp. 229-238, 1999.

[2] P. Arpaia, F. Clemente, and C. Romanucci, "An instrument for prosthesis osseointegration assessment by electrochemical impedance spectrum measurement," Measurement, vol. 41, no. 9, pp. 1040-1044, Nov. 2008.

[3] E. Barsoukov and J. R. Macdonald, Impedance Spectroscopy Theory, Experiment, and Applications: Wiley Interscience, 2nd ed., 2005.

[4] P. M. Ramos, A. C. Serra, "A new sine-fitting algorithm for accurate amplitude and phase measurements in two channel acquisition systems", Meas., vol. 41, pp. 135-143, 2008.

[5] J. R. Macdonald, J. Schoonman, and A.P. Lehnen, "The Applicability and Power of Complex Nonlinear Least Squares for the Analysis of Impedance and Admittance Data", J. Electroanal. Chem., no. 131, pp. 77-95, Jan. 1982.

[6] J. Hoja and G. Lentka, "Method using bilinear transformation for measurement of impedance parameters of a multielement two-terminal network," IEEE Trans. Instrum. Meas., vol. 57, pp. 1670-1677, 2008.

[7] F. M. Janeiro and P. M. Ramos, "Impedance Measurements using Genetic Algorithms and Multiharmonic Signals", IEEE Trans. Instrumen. Meas., vol. 58, pp. 383 388, 2009.

[8] F. M. Janeiro and P. M. Ramos, "Application of Genetic Algorithms for Estimation of Impedance Parameters of Two-Terminal Networks", in I²MTC 2009, Singapore, pp. 602-606, May 2009.

[9] C. Ferreira. Gene Expression programming: mathematical modeling by an artificial intelligence: Springer-Verlag, 2006.

[10] J. Yu, H. Cao, and Y. He, "A new tree structure code for equivalent circuit and evolutionary estimation of parameters", Chemometrics and Intelligent Laboratory Systems, vol. 85, pp. 27-39, 2007.

[11] F. M. Janeiro, P. M. Ramos, "Impedance circuit identification through evolutionary algorithms", in 18th Symposium IMEKO TC4 – Measurement of Electrical Quantities, Natal, Brazil, n.º 83980, Sept. 2011.

[12] F. M. Janeiro, J. Fareleira, J. Diogo, D. Máximo, P. M. Ramos, and F. Caetano, "Impedance Spectroscopy of a Vibrating Wire for Viscosity Measurements", in I²MTC 2010, Austin, USA, pp. 1067-1072, May 2010.