

AUTOMATIC ANALYTICAL MODELLING OF DYNAMIC CORROSION PHENOMENA BY ARTIFICIAL INTELLIGENCE TECHNIQUES

Pasquale Arpaia¹, Carmine Romanucci², Antonio Zanesco¹

¹. Dipartimento di Ingegneria, Università del Sannio, Corso Garibaldi 107, 82100 Benevento, ITALY. Ph : +39 0824 305804-17, Fax: +39 0824 305840, E-mail: arpaia-azanesco@unisannio.it

². Dipartimento di Ingegneria dei Materiali e della Produzione, Università di Napoli Federico II, Napoli, ITALY. Ph : +39 081 7177232, E-mail: romanuccicar@libero.it

Keywords: Automatic programming; Circuit modelling; Genetic algorithms; Impedance measurements.

I. INTRODUCTION

Corrosion rate and reaction kinetics of metal-electrolyte interface are surveyed experimentally by electrochemical impedance spectroscopy (EIS) and are modeled by an equivalent electric circuit [1]. Model building requires noticeable skill on the corrosion physics and is carried out by expert personnel [1]-[5]. Experts try to fit the experimental EIS Nyquist plot by three steps: (i) *modelling*: an equivalent circuit model for the examined corrosion phenomena is built on the basis of the measured EIS data and relevant electrochemical knowledge; (ii) *identification*: the parameter values of all components in the circuit are determined by curve-fitting, e.g. by software packages such as Z-ViewTM [6]; (iii) *interpretation*: parameters of the electrode reaction kinetics and physics are evaluated according to the model circuit structure and components. Among these steps, building a suitable model is the most crucial.

For complex corrosion phenomena, nonlinear least-square methods require the users to have abundant electrochemical professional knowledge and research experience, which limits their application range in some degree. Other problems arise from the sensitivity to initial values, the difficulty of computing the partial derivative expression of each parameter, and the great likelihood to trap into local optima.

The problem of equivalent electric circuit modeling and identification has been faced successfully through genetic programming [19]. In this evolutionary technique [12], possible solutions of the problem evolve such as a population of individuals in natural environment. The theoretical bases of evolutionary programming were coded by Holland [20] in genetic algorithms (GAs) concepts inspired by Darwin's theory of evolution.

Algorithm starts the search with a set of possible solutions (population), each one represented as a pool of chromosomes. Most promising solutions from one population are taken and used to form a new population. This is motivated by the hope that the new population will be better than the old one. Solutions which are then selected to form new solutions, their offspring, are selected according to their “fitness”: the more suitable they are to solve the problem, the more chances they have to reproduce. This is repeated until some conditions, e.g. number of populations or improvement of the best solution, is satisfied. Usually chromosomes in GAs are represented by real or binary number encoding solution characteristics. Genetic programming employs a different kind of chromosomes, represented by functions, constituting sub-models. Each function constitutes a simple block to build a more complex individual, i.e. a more suitable solution to the problem of analytical modelling.

However, though such an approach showed satisfying capabilities of scanning large solution spaces, GA involves an intrinsic resource waste [20]. Such as in natural evolution, a lot of useless solutions are created at each generation by the genetic alterations in the effort of finding the most suitable one. In recent soft computing applications, this problem has been faced by cultural algorithms (CAs) [21]-[24]: during the evolution, the information on the search advance acquired by most promising individual is shared with the entire population of potential solutions and stored also for next generations. This goal is achieved by means of a dynamic mechanism (*belief space*) for information selecting (*best solution acceptance*), processing (*knowledge update*), and diffusing (*evolution influence*).

In this work, a CA-based approach to the automatic modelling problem of corrosion EIS data is proposed in order to overcome efficiency and accuracy problems of Genetic Programming. A Cultural Program (Cultural Hybrid Evolutionary Modeling Algorithm for Corrosion, CHEMACorr), exploiting the belief-space structure to drive model evolution suitably, is proposed. In particular, in Section II, the proposed approach is detailed, and, in Section III, preliminary results of its application to simulated EIS corrosion data inside the framework of research projects [27]-[29] are discussed.

II. THE PROPOSED APPROACH

Building an equivalent circuit model for EIS-based corrosion analysis involves not only the definition of the circuit topology, but also the determination of the values of all its electric components. In particular, the impedance spectrum is quite sensitive to these values. Thus, appreciable results are obtained only if these values are found separately from the topology.

With this aim, modeling automation is approached by CHEMACorr at twofold CA-based levels: modeling and identification (Fig. 1). CHEMACorr processes the experimental data to be modeled in order to provide the equivalent circuit model best fitting them. A population of circuits (solution models) evolves according to CA-based evolutionary programming rules in order to define the equivalent-circuit model.

CHEMACorr exploits Gene-Expression Programming (GEP [18]) for defining the model in order to avoid the major drawback of traditional genetic programming: the superposition between spaces of *genotypes* (the specific genetic makeup of an individual, usually in the form of DNA) and *phenotypes* (either the individual total physical appearance and constitution or a specific manifestation of a trait, such as size or eye colour, that varies between individuals). In GEP, this drawback is overcome by fixed length chromosomes, where the genotype and the phenotype are separated. The individuals are encoded as symbolic strings of fixed length (the genotype), which are then expressed as non-linear expression trees of different sizes and shapes (the phenotype). With this structure, GEP

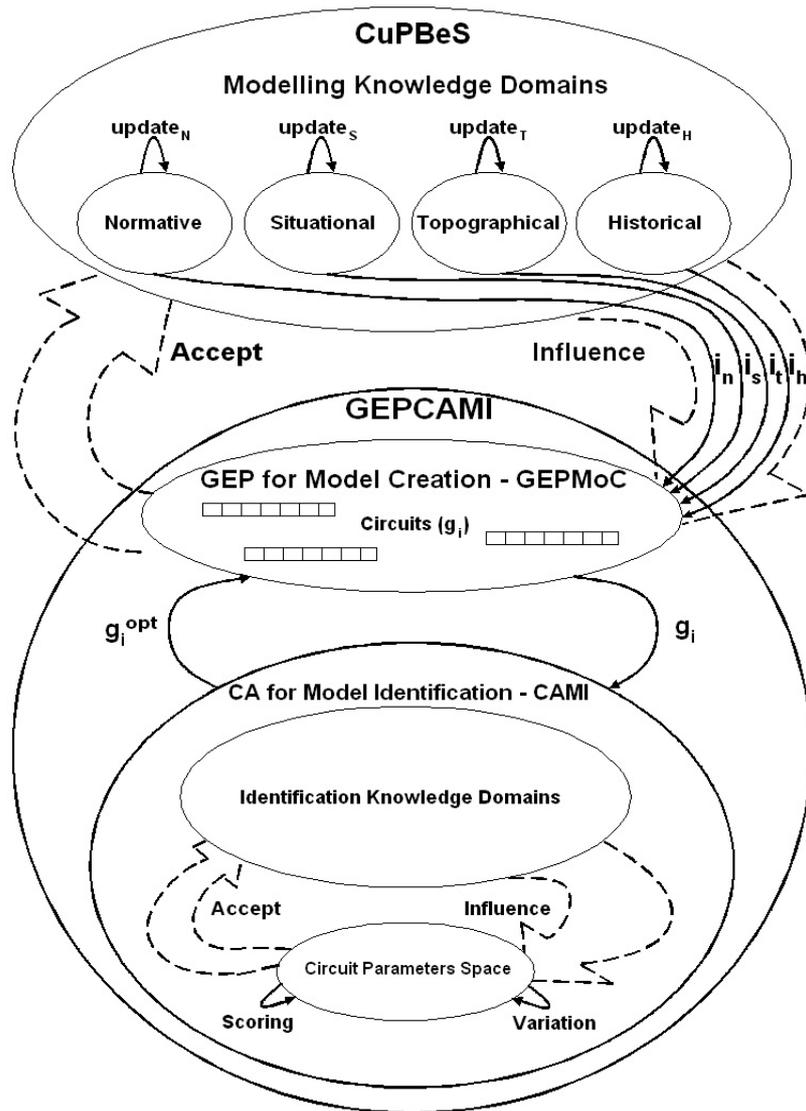


Fig. 1. CHEMACorr structure.

can use any genetic operator without restrictions, because any offspring consists in a syntactically correct program.

In CHEMACorr, the GA of the GEP block acts as the population space of a CA, with the evolution driven by a belief-space structure. Moreover, during the evolution for defining the model, a second CA for identifying the model candidates is exploited. Thus, with respect to state-of-the-art solutions, accuracy and efficiency performance is improved at twofold levels: (i) in modeling, by a belief-space structure, giving rise to a Cultural Program, and (ii) in identification, by using a CA.

Consequently, CHEMACorr is structured in two main blocks (Fig. 1): (i) the *GEPCAMI* (Gene-Expression Programming via Cultural Algorithms for Modeling and Identification), and (ii) the *CuPBeS* (Cultural Program Belief Space).

A. The *GEPCAMI*

This block is divided in two main parts (Fig. 1): (i) GEPMoC, Gene-Expression Programming for Model Creation, and (ii) CAMI, Cultural Algorithm for Model Identification. GEPMoC is based on a population of circuit models g_i . It evolves according to typical mutation functions of GEP, such as reproduction, elitism, one-point or multi-point

crossover, and so on [18]. Each circuit model is formalized by means of a fixed number of genes, each of them composed by a fixed number of chromosomes. The length of a gene is equal to $2h+1$ where h is the length of the head of the gene. A chromosome represents a *connection* or an *element* of a circuit model. The circuit connections are serial (coded as +), or parallel (coded as -). They link different elements such as resistance (R), inductance (L), capacity (C), and constant phase element (CPE). The impedance of a constant phase element CPE (w, n) is calculated as [1]:

$$Z_{\text{CPE}} = w \cdot (2\pi f)^{-n} \cdot \left[\cos\left(\frac{n\pi}{2}\right) - j \sin\left(\frac{n\pi}{2}\right) \right] \quad (1)$$

When n is 0.5, the equation is the Warburg impedance, caused by the semi-infinite diffusion process.

Thus, as an example, a simple circuit (Fig. 2a) is formalized by means of a binary tree (Fig. 2b), and is represented as a character string (Fig. 2c). Usually, GEP genes have non-coding regions downstream the gene's end [18] (“#” in Fig. 2c). A connection at the end of the individual allows different genes to be linked in order to build a more complex circuit. Therefore, for each individual of the population, an electrical circuit is defined and a corresponding Nyquist diagram can be plotted.

CAMI (Fig. 1) identifies each circuit model created by GEPMoC on the input experimental EIS data. The search for best-fit numerical values of circuit components is optimized by a CA [23]. With this aim, for each input circuit, a population of corresponding candidate numerical values is created (circuit parameter space of Fig. 1). For each identified circuit model, the following vector is computed:

$$N(i) = \sqrt{ErrRe(i)^2 + ErrIm(i)^2} \quad (2)$$

where i is the number of frequency points, and $ErrRe$ and $ErrIm$ are vectors composed by percentage errors between current and reference experimental spectra, on real and imaginary parts, respectively. Therefore, this population evolves by minimizing the fitness function $max(N)$.

After a fixed number of evolution cycles of the population, best identified models are extracted in order to be organized in knowledge domains. Their characteristics act on the population space with the use of mutation functions called *influences* (bottom *accept* function of Fig. 1); being mutated, the population space is ready for another evolutive cycle. The obtained best individual g_i^{opt} for each i -th electrical circuit is a vector composed by numerical values of the input circuit topology g_i .

The optimized parameter set g_i^{opt} is sent back to GEPMoC (bottom *influence* function in Fig. 1) and the evolution restarts. After a given number of times, GEPCAMI classifies identified model of equivalent circuits according to their best fitness. The *accept* function (top of Fig. 1) takes a given percentage of the best individuals, in order to extract the best characteristics for the CuPBeS block.

B. The CuPBeS

The CuPBeS block (Fig. 1) extracts characteristics of most promising solution candidates to the modeling problem in order to share them with all the individuals of new generations in the GEPCAMI. With this aim, it is divided into four knowledge domains: (i) *situational*, (ii) *topographical*, (iii) *historical*, and (iv) *normative* [23].

In the *situational* knowledge domain, the best model found in previous evolution cycles of GEPCAMI is stored. The influence function i_s (top of Fig. 1) mutates the GEPCAMI current population in order to obtain a new one according to the following rules:

- (i) two random individuals are chosen from the circuit population;

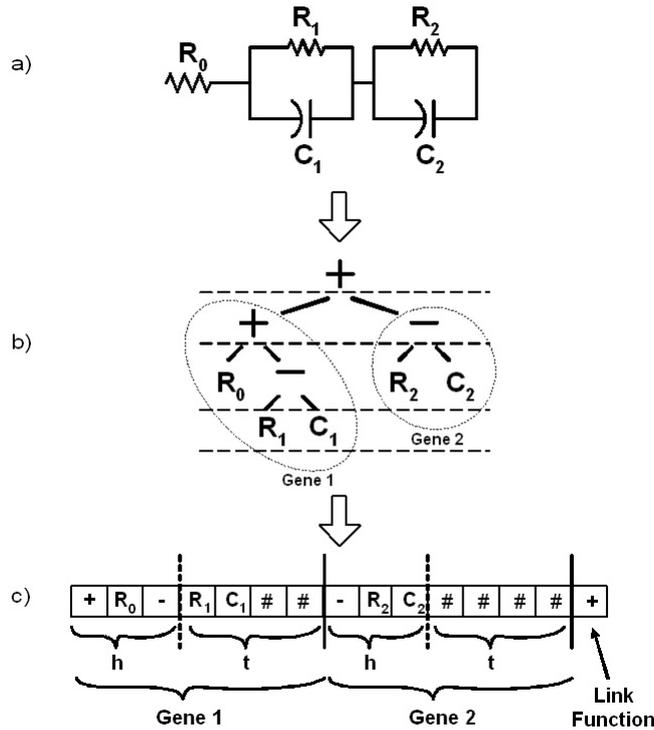


Fig. 2. Coding of a 2-genes individual: (a) model, (b) expression tree, and (c) character string of a circuit.

- (ii) each chromosome is compared with the Situational Knowledge homologue chromosome;
- (iii) if they are equals, the electric component is not changed and is stored in the new individual homologue chromosome of the population, while its parameter value is calculated with a differential evolution technique [23],[24];
- (iv) if they are different, the new electrical component and its numerical value are assigned randomly. The update function u_s compares the best individual with the current one and, if better, stores it in the domain.

In the *topographical* domain, a map of the solution search space is stored. The knowledge is organized as a 2-D matrix of circuits and a 3-D matrix of numerical values with their ranges. As for the situational domain, the function topographical influence i_t (top of Fig. 1) generates a population from the current one. Different circuits of the population are compared and their spaces are adjusted suitably by the differential evolution technique.

In the *historical* domain, best individuals, found in the past GEPCAMI evolution cycles are stored, in order to recover previous good candidate solutions. Moreover, by identifying position patterns of local optimal solutions, CHEMACorr is capable of avoiding them. In fact, if a model remains as the best one for a given number of GEPCAMI generations, CHEMACorr could be trapped in a local optimum, and this solution is stored in order to avoid it in future. By carrying out a fixed number of random mutation cycles, CHEMACorr tries to find another optimum solution and escapes from the local optimum. The structure of the historical knowledge is analogous as the topographical one. The influence function i_h (top of Fig. 1) increases the diversity of the population by injecting randomness to avoid local optimum traps. The update function u_h (top of Fig. 1) adds any local optimum found during the evolutionary process to a suitable list, with aims analogous as the topographical update.

In the *normative* domain, knowledge about electrochemical corrosion physics are stored as constraints on the space of possible solutions. Such a domain is created by coding suitably knowledge provided by expert personnel. The constraints limit the individual evolution by division of the infinite solution space in three zones: (i) feasible, (ii) semifeasible, and (iii) infeasible. After the accept function takes a subset of possible solutions, individuals feasible and semifeasible are preferred and stored in the other domains. The influence function i_n (top of Fig. 1) selects mutated populations in order to preserve only feasible and semifeasible solutions.

After influence mutations, individuals tournament selection is applied in order to choose a new evolving population. Each individual is randomly compared with others and, after a fitness based classement, feasible and semifeasible ones are stored as the new population.

III. SIMULATION RESULTS

Performance of the proposed approach was assessed by comparing the analytical model generated and identified automatically by CHEMACorr with a reference model of a simulated case study of corrosion.

The EIS simulation data set based on the equivalent electric circuit shown in Fig. 3a was employed to assess performance of CHEMACorr. Equivalent circuit is composed by typical parameters such as electrolyte resistance, the first resistance, and surface concentration of the adsorbed intermediate, the R and CPE parallel.

CHEMACorr was implemented in Matlab™ 7.0.1. The parameter settings were:

- (i) GEPMoC: $h = 8$; number of genes = 2; population size = 150; and CHEMACorr cycles = 10;
- (ii) CAMI: population size = 300; GA cycles = 20; CAMI cycles = 10; accept percentage = 20 %; crossover percentage = 90 %; and mutation percentage = 10 %.

Individuals composed by two genes were employed in order to face the complexity of the equivalent electrical circuit modeling.

The obtained mean percentage errors between CHEMACorr and reference spectra were computed as the mean error on the impedance real (MER) and imaginary (MEI) parts, and the mean error on the norm (MEN).

The corresponding equivalent circuit model is shown in Fig. 3b.

In Fig. 4, results of CHEMA run as a Nyquist diagram, compared with the reference EIS data, are shown. The best individual found with CHEMA run was:

$$++-R(987 \Omega)-R(8 \text{ k}\Omega)C(1 \mu\text{F})+R(2 \text{ k}\Omega)P(10^6, 0.6)L(11 \mu\text{H})$$

Error results were less than 1% for all kind of mean error (MER, MEI, and MEN) and demonstrate CHEMACorr to be able to model and identify a really good fitting circuit.

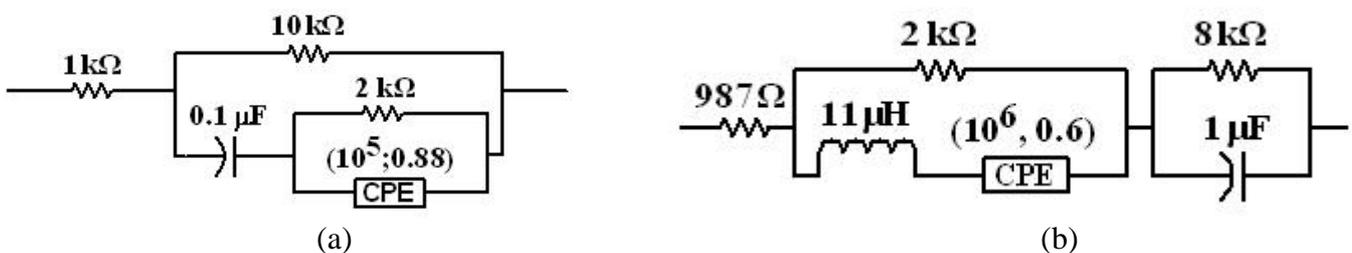


Fig. 3. Reference (a) and CHEMACorr (b) equivalent electric circuits.

Equivalent electric circuits could be quite different as for the topology as for numerical parameters. In fact, a Nyquist diagram can be produced by a lot of possible electrical circuit. Circuit in which changes are least, simplest, or closest to theoretical expectations should certainly preferred.

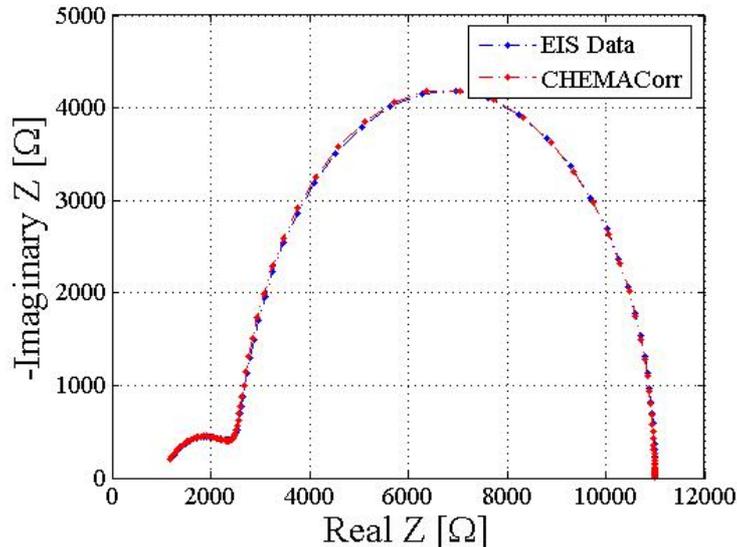


Fig. 4. CHEMACorr results compared with reference EIS data.

Generally, better modeling results could be achieved by the use of a bigger number of individuals and evolutionary cycles, but time computation arises excessively. The choice of a satisfying trade-off between these opposite constraints has to be analyzed. Time cost could be reduced by means of algorithm optimization, such as the tuning of evolutionary cultural pressure which is still under progress.

IV. CONCLUSIONS AND FUTURE WORK

An automatic analytical modeling of electrical circuit based on Cultural Program (CHEMACorr) was proposed for automatic diagnosis of electrochemical corrosion kinetics. CHEMACorr is capable of modelling automatically an equivalent electrical circuit by fitting a Nyquist diagram according to rules imposed by electrochemical corrosion phenomena. In particular, faster and better optimal solutions are found (i) in model definition, by means of a Belief Space (CuPBeS block) added to a Genetic Program to build a Cultural Program; (ii) in model identification, by means of a Cultural Algorithm for the model identification (CAMI block), by avoiding intrinsic resource waste typical of genetic algorithms; and (iii) in physical interpretation, by means of a normative domain in CuPBeS, filtering solutions without a physical electrochemical meaning.

At to date, CHEMACorr has been tested in simulation by showing promising performance in analytical modeling dynamic corrosion phenomena automatically. Future works will consider more comprehensive performance characterization as well as on-field data validation for several experimental case studies.

ACKNOWLEDGMENTS

This work was supported by twofold research projects of the Italian Regione Campania [27],[28], the latter one in cooperation with SCE srl of Pozzuoli, Italy, whose supports the authors gratefully acknowledge.

REFERENCES

- [1] Evgenij Barsoukov and J. Ross Macdonald, *Impedance Spectroscopy : Theory, Experiment, and Applications*, 2005.
- [2] Boukamp B. A., "Electrochemical impedance spectroscopy in solid state ionics: recent advances", Proceedings of the Annual Meeting of International Society of Electrochemistry, Dusseldorf, Germany, 15 - 20 September 2002, Volume 169, Issues 1-4, Pages 1-170.
- [3] Nelatury S. R., Singh P., "Equivalent circuit parameters of nickel/metal hydride batteries from sparse impedance measurements" *Journal of Power Sources*, Vol. 132, Issues 1-2, Pages 1-327 (May 2004).
- [4] Liaw B. Y., Nagasubramanian G., Jungst R. G., Doughty D. H., "Modeling of lithium ion cells—A simple equivalent-circuit model approach", *Solid State Ionics*, Volume 175, Issues 1-4, Pages 1-875 (November 2004).
- [5] Sgura I., Bozzini B., "Numerical issues related to the modelling of electrochemical impedance data by non-linear least-squares", *International Journal of Non-Linear Mechanics*, Volume 40, Issue 4, Pages 429-588 (May 2005).
- [6] www.scribner.com.
- [7] Gentilli P., Piazza F., Uncini A. "Evolutionary design of FIR digital filters with power-of-two coefficients" Proceedings of the First IEEE Conference on Evolutionary Computation, 1994, Piscataway, NJ. IEEE Press. Vol. I. pp. 110-114.
- [8] Kruiskamp M. W. "Analog design automation using genetic algorithms and polytopes" Eindhoven, 1996, The Netherlands: Data library Technische Universiteit Eindhoven.
- [9] Grimbleby J. B. "Automatic analogue network synthesis using genetic algorithms" Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA)" London: Institution of electrical Engineers. 1995. pp. 53-58.
- [10] Hemmi H., Hikage T., Shimoara K. "AdAM: A hardware evolutionary system" Proceedings of the First IEEE Conference on Evolutionary Computation. 1994. Piscataway, NJ: IEEE Press. Vol. I. pp. 193-196.
- [11] Mizoguchi J., Hemmi H., Shimoara K. "Production genetic algorithms for automated hardware design through an evolutionary process" Proceedings of the First IEEE Conference on Evolutionary Computation" 1994. Piscataway, NJ: IEEE Press. Vol. I. pp. 661-664.
- [12] Koza J. R., Bennett III F. H. , Andre D., and Keane M. A., *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann, San Francisco, 1999.
- [13] Koza J. R., "Automatic synthesis of both the topology and numerical parameters for complex structures using genetic programming", In Chakrabarti, Amaresh (editor). *Engineering Design Synthesis*. London: Springer 2002. Pages 319–337.
- [14] Koza J. R., Streeter M.J., Keane M. A., "Automated synthesis by means of genetic programming of human-competitive designs employing reuse, hierarchies, modularities, development and parameterized topologies", In Lipson, Hod, Antonsson, Erik K., and Koza, John R. (editors). *Computational Synthesis: From Basic Building Blocks to High Level Functionality: Papers from the 2003 AAAI Spring Symposium*. AAAI technical report SS-03-02. Pages 138–145.
- [15] Streeter M. J., Keane M. A., Koza J. R., "Automatic synthesis using genetic programming of both the topology and sizing for five post-2000 patented analog and mixed analog-digital circuits", In *Proceedings of the 2003 Southwest Symposium on Mixed-Signal Design*. Piscataway, NJ: IEEE Circuits and Systems Society. Pages 5–10.

- [16] Massey P., Clark J. A., Stepney S., “Evolving quantum circuits and programs through genetic programming”, In *Genetic and Evolutionary Computation Conference: GECCO 2004, Seattle, USA, June 2004*. LNCS 3103, pp 569–580. Springer, 2004.
- [17] Munoz D. G., Gustafsson O., Wanhammar L., “Evolution of filter order equations for linear-phase FIR filters using gene expression programming”, In *RVK 2005 RadioVetenskap och Kommunikation*, pages 679-682, Linkoping, Sweden, 2005.
- [18] C. Ferreira, “Gene Expression Programming: a New Adaptive Algorithm for Solving Problems”, *Complex Systems*, Vol. 13, Issue 2: 87-129, 2001.
- [19] H. Cao, J. Yu, L. Kang, “An Evolutionary Approach for Modeling the Equivalent Circuit for Electrochemical Impedance Spectroscopy”, *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003*, p. 1819-1825, IEEE 2003.
- [20] Holland, J. H., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975 (second edition: MIT Press, 1992).
- [21] R.G. Reynolds, W. Sverdluk, “Problem solving using cultural algorithms” 1994. *Proc. of the First IEEE Conf. on Evolutionary Computation*, 27-29 June 1994, pp. 645 – 650.
- [22] L. Spector, L. Sean, “Cultural transmission of information in genetic programming”, *Genetic Programming 1996: Proceedings of the First Annual Conference*, 209-214. Cambridge, MA: The MIT Press. P.
- [23] Arpaia, G. Lucariello, A. Zanesco, “Automatic Fault Isolation by Cultural Algorithms with Differential Influence”, *IEEE Trans. on Instrum. and Meas.*, 2005 – Submitted.
- [24] R. L. Becerra, C. A. Coello Coello, “Culturizing Differential evolution for constrained optimization”, *Proc. of the Fifth Mexican Int. Conf. in Computer Science, ENC2004*.
- [25] P. Arpaia, F. Clemente, A. Zanesco, “Low-invasive Diagnosis of Metallic Prosthesis Osseointegration by Electrical Impedance Spectroscopy”, *IEEE Trans. on Instrum. and Meas.*, 2005 – Submitted.
- [26] <http://www.solartron.com>.
- [27] Regione Campania (L.R. N.5, 28.03.2002), “Experimental characterization of dynamical phenomena of electrolytic corrosion in metallic materials by digital measurement techniques”.
- [28] Regione Campania (law 598/94), “Low-invasive measurement of prosthesis osseointegration by variable-polarization impedance tomography” (in Italian), 2004.