

**AN EVOLUTIONARY ALGORITHM FOR LINEAR SYSTEMS
IDENTIFICATION****UN ALGORITMO EVOLUTIVO PARA IDENTIFICACIÓN
DE SISTEMAS LINEALES**

**Ph.D. Edmary Altamiranda, Ing. Rodrigo Calderón
Ph.D. Eliezer Colina**

Universidad de Pamplona

*Instituto de Investigación y Desarrollo de Tecnologías Aplicadas IIDTA
{edmarya, rodcalderon, ecolina}@unipamplona.edu.co
Pamplona, Norte de Santander - Colombia*

Abstract: Abstract: This paper presents a systems identification method, for discrete time linear systems, based on an evolutionary approach, which allows achieving the selection of a suitable structure and the parameters estimation, using non conventional objective functions. This algorithm incorporates parametric crossover and parametric mutation along a weighted gradient direction (Tang and Wang, 1997). The performance of the proposed method is illustrated with computer simulations using ARX model structures, where parameters, model dynamical order and input-output delay values are estimated

Resumen: En este trabajo se presenta un algoritmo para identificación lineal de sistemas en tiempo discreto, basado en un enfoque evolutivo, el cual permite llevar a cabo la selección de una estructura apropiada basada en modelos tipo ARX y de la estimación de los parámetros, orden dinámico y retardo entrada/salida del sistema, utilizando funciones objetivos no convencionales. Este algoritmo incorpora operadores de cruce paramétrico y mutación paramétrica utilizando ascenso por gradiente (Tang and Wang, 1997). La eficiencia del algoritmo evolutivo propuesto es ilustrada a través de simulaciones computacionales..

Keywords: Systems identification, Linear systems, Evolutionary algorithms, Parameter Estimation.

1. INTRODUCTION

The identification of a dynamic process from a set of possibly noisy input-output data has been a classical problem in control engineering. System parametric identification is usually achieved in two steps: the first step is selecting a model family from

which a candidate model is produced by minimizing some error criterion. The second step is validating the identified candidate model in terms of verifying some performance indicators, such as noise independence, error auto-correlation, input/error correlation and real response following among others. If results do not satisfy some of the

performance indicators, an alternative model must be produced over a different model family. Least square methods have been traditionally used, for achieving parametric estimation, in systems identification due to their computational efficiency. Least square methods properties have been well analyzed by many authors, numerical procedures have been proposed for individual applications and modification from basic method have been performed according to specific requirements (Strejc, 1980). On the other hand, evolutionary approaches for system identification have been used in recent years, for making error criteria more exigible and for improving performance of system identification process in parameters estimation. The evolutionary method proposed allows the parametric estimation process using non conventional error criteria for linear systems and the calculation of optimal system order and system time delay, while minimizing the time involved in the validation step. The efficiency of the evolutionary method is illustrated by computer simulation results.

2. EVOLUTIONARY COMPUTATION

Evolutionary Computation methods are associated to computational models which use some mechanism related to the Evolution Theory and intended to the design and implementation of a variety of systems applications (Williams, 1966; Bäck and Schwefel, 1993; Fogel, 1995; Golberg, 1996; Pérez, n.d.; Bäck *et al.*, 1997; Fogel, 1998; Heitkotter and Beasley, 1998). The most important evolutionary computation contribution to problems solving methodology, has been the use of selection mechanisms of potential solutions and the construction of new candidates using characteristics from the recombination of existent individuals, similar to the organisms in natural evolution (Mayr, 1963; Rosen, 1967; Darwin, 1972; Stern, 1973; Lewontin, 1974; Darwin, 1985; Ridley, 1989; Futuyma, 1986; Maynard and Smith, 1989). The main objective associated to these techniques is to take advantage from some generic ideas behind evolutionary processes, in order to solve search and learning complex problems. Concrete implementations in the evolutionary computation area have been called Evolutionary Algorithms (Fogel, 1995; Fogel, 1998; Golberg, 1996; Bäck *et al.*, 1997; Heitkotter and Beasley, 1998; Pérez, n.d.). The generic purpose of these algorithms is to guide a stochastic search, making evolve a set of

structures and selecting iteratively the best of them. All these algorithms are characterized by their simplicity from the biological point of view, but at the same time are complex enough to provide robust and powerful search mechanisms. (Aguilar and Echeverría, 2001).

In Evolutionary Computation it is possible to distinguish five paradigms: (Fogel, 1995; Fogel, 1998; Golberg, 1996; Bäck *et al.*, 1997; Heitkotter and Beasley, 1998; Pérez, n.d.) a) Genetic Algorithms: A population of binary numbers or character strings evolve using a set of unitary and binary transformations and a selection process; b) Evolutionary Programs: A population of data structure evolve through a set of specific transformations and a process selection; c) Evolutionary Strategies: A population of real numbers is made evolve to find possible solutions of a numerical problem; d) Evolutionary Programming: The population is constituted by finite state machines which are subjected to unitary transformations; e) Genetic Programming: The population consists of programs which solve a specific problem. The objective is to make evolve the population in order to find the best program that solves the problem under study. The algorithm proposed in this paper, may be considered as an evolutionary strategy, since it is designed to solve parametric optimization problems, and it is based on some ideas presented in (Gao, 1999; Tang and Wang, 1997; Wang and Shen, 1999).

3. SYSTEM IDENTIFICATION METHOD FOR DISCRETE TIME LINEAR SYSTEMS

The ARX model structure may represent a wide variety of linear systems. For single input and single output (SISO) systems, the ARX model is represented by:

$$A(q^{-1})y(k) = B(q^{-1})q^d u(k) + e(k) \quad (1)$$

where

$$A(q^{-1}) = 1 + \sum_{i=1}^n (a_i q^{-i})$$

$$B(q^{-1}) = \sum_{j=1}^m (a_j q^{-j}) \quad (2)$$

The variables $y(k)$, $u(k)$ and $e(k)$ represent output, input and noise at time k , respectively. The system

order is $n \geq m$ and d is the dynamical system time delay.

3.1 Chromosome Coding for Parameters Estimation

The chromosome coding for a fixed ARX structure, is defined as a real values string containing the polynomials coefficients and the time delay corresponding to the ARX model. The chromosome structure, where each polynomial coefficient represents a chromosome gen, is illustrated in figure 1

The proposed chromosome coding facilitates the calculation of model parameters for different system orders n and time delay d . Chromosomes lengths mainly depend on n . System order n , time delay d and the order of polynomial $B(q^{-1})$, m , are chosen taking into account their contribution to the fitness function associated to each set of chromosomes with fixed structure.

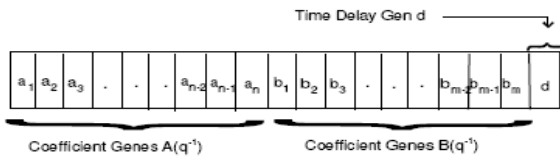


Fig. 1. Chromosome Structure for Parameters Estimation

Some parameters are required to create different sets of chromosomes: maximum system order n_{max} and maximum time delay d_{max} . All possible combinations will be considered to generate the sets of chromosomes with fixed structure, for example if $n_{max} = 2$ and $d_{max} = 2$, the different sets of chromosomes with fixed structure are illustrated in figure 2

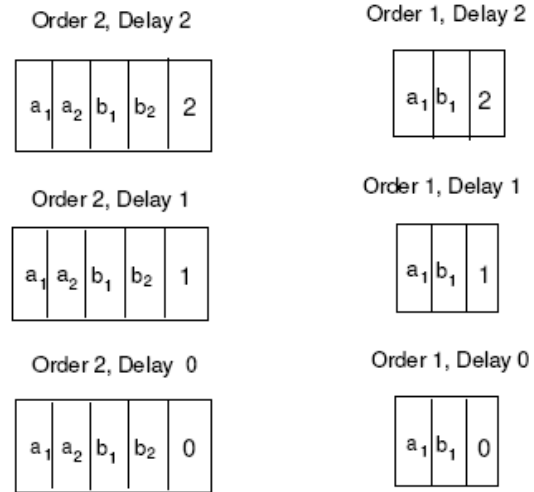


Fig. 2. Sets of Fixed Structure Chromosome

Without loss of generality, m has been considered to be, at most equal to n in order to generate the chromosomes lengths. Solutions associated to $m < n$ are included in these sets of chromosomes and solutions for which $a_n = 0$ are discarded.

4. EVOLUTIONARY ALGORITHM FOR PARAMETERS AND STRUCTURE ESTIMATION

In this section an evolutionary algorithm for Linear Systems Identification is proposed. This approach allows parameter estimation and finding a suitable structure for an ARX model. The algorithm uses parametric crossover and parametric mutation according to results presented in (Gao, 1999; Tang and Wang, 1997; Wang and Shen, 1999). Consider the following non conventional index which may be used as fitness function for the proposed algorithm

$$h(a_i, b_i, d) = \exp\left(- \sum_{l=k-p+1}^k \frac{(e_l)^2}{2} - \sum_{i=k-p+1}^k \frac{abs(de_l)}{p}\right) \tag{3}$$

where p is the number of data patterns, taken from the process under study over a time window $[k-p+1, k]$, which is long enough to be able to obtain crucial dynamics characteristics from the

inputoutput system data. This data is used to perform

the offline system identification procedure; where e_l is the error and de_l is the error derivative, corresponding to the l^{th} data pattern. Notice that the following equation holds true

$$A(q^{-1})y(k) - B(q^{-1})q^{-d}u(k) = e(k) \quad (4)$$

notice also that such fitness function h is affected by a_i , b_j and d . According to the fitness function definition, $h(a_i; b_j; d)$ must be maximized in order to minimize the estimation error. Therefore it is required to solve the following optimization problem

$$\text{Max}_{a_{ij} \in D_1; d \in D_2} (h(a_i, b_j, d)) \quad (5)$$

where $D_1 \subset \mathbb{R}$, and $D_2 \subset \mathbb{Z}^+$ (polynomial coefficients are represented with finite real values and the time delay is represented by a finite positive integer). Taking into account equations 4 and 5, the proposed algorithm may be written as follows:

Reproduction: Given the maximum system time delay and maximum dynamical system order, begin generating all possible sets of fixed structure for chromosomes. Then, start with an initial population of N individuals or chromosomes for each type of structure. Let $x_s(r)$ be the chromosome corresponding to the r^{th} generation associated to the s^{th} structure

$$x_s(r) = [a_{1s}(r), \dots, a_{n_s}(r), b_{1s}(r), \dots, b_{m_s}(r), d_s] \quad (6)$$

where for reproduction stage $r = 1$ and

$$x_s(1) = \zeta_s(1) \quad (7)$$

where $\zeta_s(1)$ is an array and the first $n+m$ elements constitute a random array with each element in $(-1, 1)$, selected to guarantee appropriate initial population values which improve the algorithm convergence. The last element value of $x_s(1)$, as well as the length of the chromosome, are given by the s^{th} generated structure.

Fitness Function Evaluation: The fitness function is evaluated for every population of individuals corresponding to each structure. The individuals are

sorted in a list, from the greatest fitness function value to the lowest, such that chromosomes with greater fitness function values go first. There will be one list for each structure.

Parametric Crossover: Let $x_s^i(r)$ be the chromosome corresponding to the i^{th} position in the list associated to the structure s^{th} , during the r^{th} generation; $i = 1..N$, the n_1 fitter individuals participate in the crossover operation. Let $x_s^i(r)$ and $x_s^j(r)$ be two individuals, then the parametric crossover results in:

$$\begin{aligned} \tilde{x}_s^i(r) &= x_s^i(r) + \sigma(x_s^i(r) - x_s^j(r)) \\ \tilde{x}_s^j(r) &= x_s^j(r) + \sigma(x_s^j(r) - x_s^i(r)) \end{aligned} \quad (8)$$

σ must be a very small positive number as compared to the magnitude order associated to individual elements. This operation is only achieved for the first $n + m$ chromosome elements. The element corresponding to the time delay d is generated with the chromosome fixed structure in the reproduction stage. Next these operation parents and new individuals are sorted according to their fitness function from the greatest value to the lowest value.

Parametric Mutation: The parametric mutation is achieved along a fitness function weighted gradient direction, based on the parametric mutation algorithm proposed in (Tang and Wang, 1997). This is the most important operator for the convergence of the proposed evolutionary algorithm. Additionally, it uses a penalty function for evaluating the solutions which violate the optimization problem constraints. According to the parametric algorithm proposed in (Tang and Wang, 1997), constraints of type $g_j(x) \leq 0$, $j = 1..J$, may be imposed on the optimization problem. For this case, one constraint, associated to the error tolerance ET_{\max} , may be imposed in order to accelerate the algorithm convergence $J = 1$.

$$\begin{aligned} g_1 &= \sum_{l=k-p+1}^k \frac{(e_l)^2}{2} \\ \sum_{i=k-p+1}^k \frac{\text{abs}((de_i))}{p} - ET_{\max} &\leq 0 \end{aligned} \quad (9)$$

All of N individuals are subjected to parametric mutation, according to the weighted gradient direction method which may be written as follows:

$$d(x) = \Delta h(x) - \sum_{j=1}^J ((\rho_j \Delta g_j)) \quad (10)$$

ρ_j corresponds to a penalty multiplier associated to each constraint. For this case $J=1$, then

$$d(x) = \Delta h(x) - (\rho_1 \Delta g_1) \quad (11)$$

In general these multipliers may be defined as

$$\rho_j = \begin{cases} 0, & g_j \leq 0 \\ \delta_j, & g_j \geq 0 \end{cases} \quad (12)$$

$$\delta_i = \frac{1}{g_{\max} - g_i(x) + \delta} \quad (13)$$

$$g_{\max}(x) = \max(g_j(x)), i = 1..J \quad (14)$$

where δ is a very small positive number. Let $x_s^i(r)$ corresponds to the i^{th} position in the list associated to the structure s^{th} , during the r^{th} generation; $i = 1..N$ then the parametric mutation operator may be expressed in the following way

$$\hat{x}_s^i(r) = x_s^i(r) + \beta d(x) \quad (15)$$

β must be a small positive number. A convergence proof, related to an evolutionary algorithm which uses only the parametric mutation method, may be found in (Tang and Wang, 1997). This operation is only achieved for the first $n + m$ chromosome elements, the time delay d is generated with the fixed structure in the reproduction stage.

Replacement: The least suitable individuals are replaced, with individuals that result from parametric crossover and/or parametric mutation processes, in order to keep constant the number N of population individuals.

Selection: The fittest individual, corresponding to each fixed structure, is selected and the chosen ones are sorted according their fitness function from the greatest to the lowest value. The first individual in

the final list corresponds to the best solution for the identification problem using ARXstructures.

5. SIMULATION RESULTS

The results presented in this section are associated to data from a Data Base for Systems Identification (DAISY), developed by SISTA: Signals, Identification, Systems Theory and Automation, which may be accessed from <http://www.esat.kuleuven.ac.be/sista/daisy/>. This Data Base provide data from different kind of processes. It has been selected an Industrial Process (Ethane-Ethylene Distillation Column), to illustrate the potentials of the proposed evolutionary algorithm, with data corresponding to, Outputs: Bottom Ethylene Composition and Bottom-Top Differential Pressure; Input: Ratio between Reflux and Feed Flow. The results are presented with linearized data.

Simulation 1

Input: Ratio between Reux and Feed Flow
Output: Bottom Ethylene Composition
The Arx Model polynomials and dynamic delay calculated, with the proposed algorithm, giving as input parameters: maximum delay 10, maximum order: 4 is the following:

$$\begin{aligned} A(q^{-1}) &= 1 - 1.3578q^{-1} + 0.3767q^{-2} \\ B(q^{-1}) &= -0.1263 + 0.0092q^{-1} + 0.06q^{-2} \quad (16) \\ d &= 2 \end{aligned}$$

Figures 3, 4 and 5 illustrate the real output vs. estimated output, residual analysis and zero-pole map respectively for the Arx model, calculated with the evolutionary algorithm.

Simulation 2

Input: Ratio between Reux and Feed Flow
Output: Bottom-Top Differential Pressure
The Arx Model polynomials and dynamic delay calculated, with the proposed algorithm, giving as input parameters: maximum delay 10, maximum order: 4 is the following:

$$\begin{aligned} A(q^{-1}) &= 1 - 0.9713q^{-1} \\ B(q) &= -0.1345 \quad (17) \\ d &= 8 \end{aligned}$$

Figures 6, 7 and 8 illustrate the real output vs. estimated output, residual analysis and zero-pole

map respectively for the Arx model, calculated with the evolutionary algorithm.

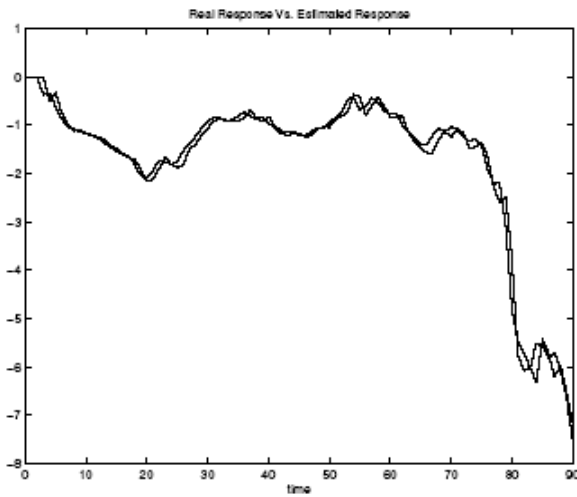


Fig. 3. Real Output Vs. Estimated Output (Bottom Ethylene Composition)

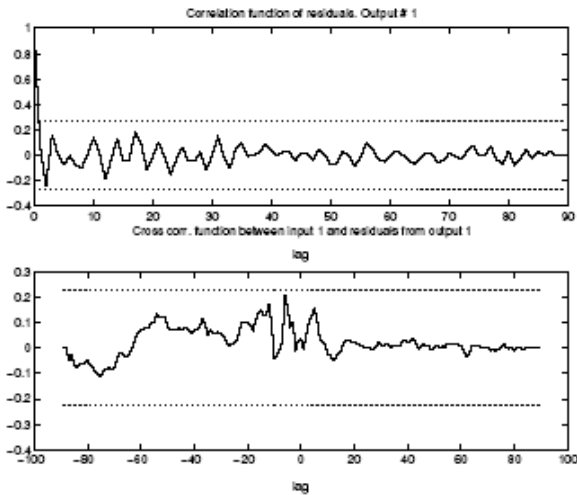


Fig. 4. Residuals Analysis Simulation 1

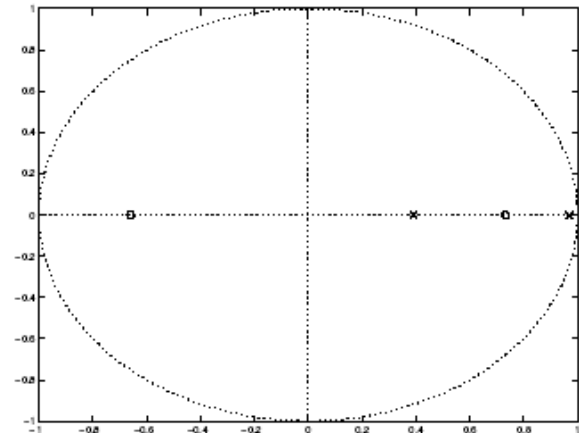


Fig. 5. Zero Pole Map Simulation 1

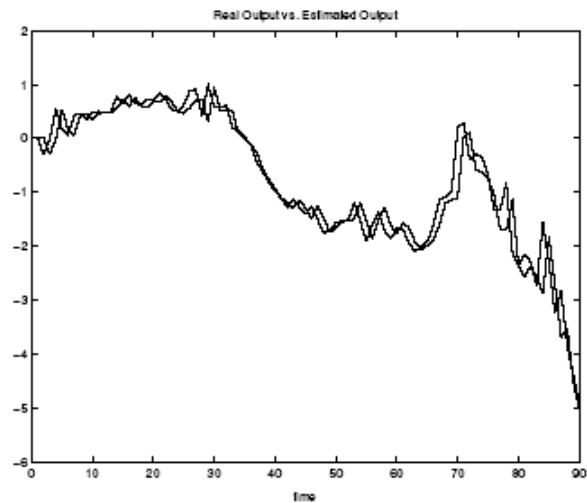


Fig. 6. Real Output Vs. Estimated Output (Bottom-Top Differential Pressure)

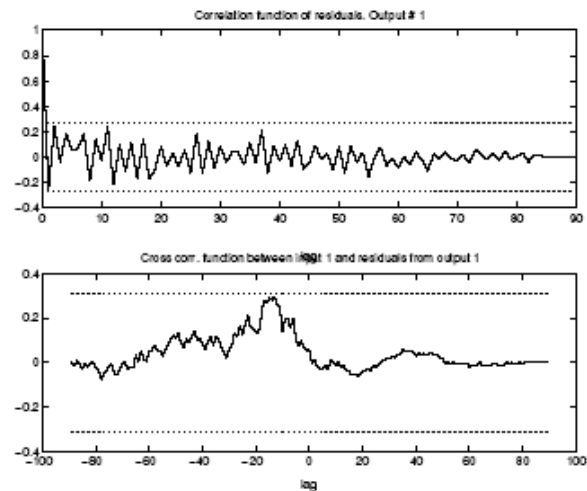


Fig. 7. Residuals Analysis Simulation 2

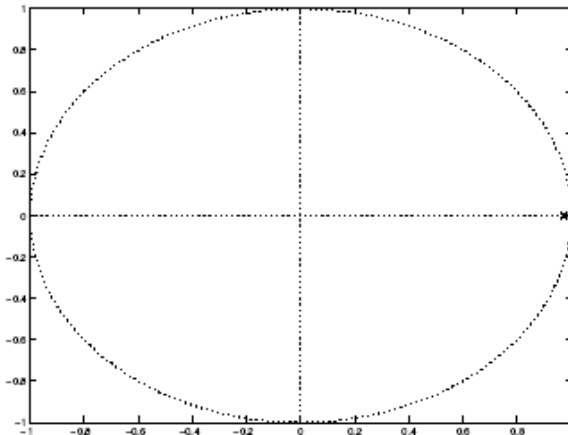


Fig. 8. Zero Pole Map Simulation 2

6. CONCLUSION

A new evolutionary algorithm for linear systems identification has been presented. The algorithm incorporates parametric crossover and parametric mutation along a weighted gradient direction. The evolutionary method proposed in this paper not only allows the parametric estimation process for a fixed model structure, with non conventional objective functions, but additionally, calculates an optimal system structure, while minimizing the time involved in the identification process validation step. The efficiency of the evolutionary method is illustrated by computer simulation results.

REFERENCES

- Aguilar, J. and F. Echeverría (2001). *Introducción a las Técnicas de Computación Inteligente*. 1 ed.. Universidad de los Andes, Editores: J.Aguilar y F. Echeverría. Mérida, Venezuela.
- Bäck, T. and H. Schwefel (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation* 1(1), 1-23.
- Bäck, T., U. Hammel and H. Shwefel (1997). Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation* 1(1), 3-17.
- Darwin, C. (1972). *Teoría de la Evolución*. Pennsula.
- Darwin, C. (1985). *The Origin of Species* (Original de su trabajo de 1859). Penguin

Classics.

- Fogel, D. (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press.
- Fogel, D. (1998). *Evolutionary Computation: The Fossil Record*. IEEE Press.
- Futuyma, D. (1986). *Evolutionary Biology*. Sinauer Assoc.
- Gao, F. (1999). A genetic algorithm and evolutionary programming hybrid strategy for structure and weight learning for multilayer feedforward neural networks. In: *14th Triennial IFAC World Congress*. Vol. K-3e-04-6. pp. 33-38.
- Golberg, D. (1996). Genetic and evolutionary algorithm come of age. *Communications of the ACM* 37(3), 113-119.
- Heitkotter, J. and D. Beasley (1998). *The Hitch Hiker's Guide to Evolutionary Computation*. Heitkotter and Beasley Ed.
- Lewontin, R. (1974) *The Genetic Basis of Evolutionary Change*. Columbia Univ. Press.
- Maynard, S. and M. Smith (1989). *Evolutionary Genetics*. Oxford Univ. Press.
- Mayr, E. (1963). *Animal Species and Evolution*. Harvard Univ. Press.
- Pérez, A. (n.d.). Introducción a la computación evolutiva. <http://www.etsimo.uniovi.es/pub/EC/EA/papers/intro-spanish.ps.gz>.
- Ridley, M. (1989). *The Problems of Evolution*. Oxford Univ. Press.
- Rosen, R. (1967). *Optimality Principles in Biology*. Butterworths.
- Stern, C. (1973). *El origen de la genética*. Alhambra.
- Strejc, V. (1980). *System Identification*. Tutorials presented at 5th ifac symposium on identification and system parameter estimation Least Squares Parameter Estimation, pp. 335-549. Pergamon Press, International Federation of Automatic Control. Great Britain.
- Tang, J. and D. Wang (1997). *A new genetic algorithm for nonlinear programming problems*. In: In Proc. 36th Conference on Decision and Control. pp. 4906-4907.
- Wang, J. and L. Shen (1999). *A new genetic synthesis of fuzzy control systems*. In: 14th Triennial IFAC World Congress. Vol. K-3e-04-4. pp. 21-26.
- Williams, C. (1966). *Adaptation and Natural Selection*. Princeton Univ. Press.