Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics - ICTAMI 2004, Thessaloniki, Greece

A GENETIC BASED APPROACH TO THE TYPE I STRUCTURE IDENTIFICATION PROBLEM

by
S.E. Papadakis, P. Tzionas, V.G. Kaburlasos,
J.B. Theocharis

Abstract. A novel approach to structure identification of type I is presented in this work. A simple genetic algorithm, enhanced with a local search operator is used to determine an optimum subset of features that actually affect the output of a system, from a set of candidate features selected intuitively. The problem of input selection, addressed in the bibliography as the type I structure identification problem, is considered to be the most important phase of the structure identification process. The fuzzy curve technique introduced by Lin and Cunningham [19] extended to fuzzy surface concept, is used as a fast modelling technique to built a coarse, model from a subset of the initial set of candidate inputs. In order to identify inputs that are interdependent, a genetic algorithm is used to guide the probing of more than one input simultaneously. The objective of the genetic algorithm is to determine the minimum number of inputs that, ideally, are necessary and sufficient to describe the system. Extensive simulation results on both artificial examples and real world applications were obtained in order to assess thoroughly the proposed approach.
Keywords: structure identification, feature selection, modelling, input selection, fuzzy curve, genetic algorithms.

Introduction

System identification from input-output data pairs has always been one of the most challenging topics of engineering. Recently, artificial intelligence techniques, including neural networks and neuro-fuzzy models have been successfully used, as universal model free estimators [5], [6], [11], for modelling, identification and control of ill-defined real world processes. The number of published works on novel cognitive models, with significant identification capabilities of complex physical systems in a black-box fashion, is huge. However, the determination of the 'proper structure' of a model, that is, a model with significant identification performance and low complexity remains active. Sugeno and Yasukawa [13] have organized the structure identification problem by dividing it into two major categories, named type I and type II. The type II refers to defining the architecture of the model and to the parameter identification process (training). This type is out of the scope of this paper. However, Sugeno states that the first type of structure identification process is one hundred times more important that the second one. Type I, is related to the selection of those inputs (features) that affect the output of a system significantly. It is subdivided into type Ia and type Ib subcategories. The type Ia is concerned with the definition of a set of candidate's inputs, which form the feature space of the system. It is realized entirely intuitively, requires human knowledge about the system to be identified and also, it is the starting gate to modelling. Frequently, the expert who decides on the feature space components is misplaced, overestimating the importance of several inputs and increasing the complexity of the model to be built. In many cases the set of features that actually affects the output of the system is a subset of the initial set. This work focuses on the structure identification problem of type Ib, that is, the selection of a small subset of features that, ideally, is necessary and sufficient to describe the target concept. The ultimate objective of feature selection is to obtain a feature space with a) low dimensionality, b) retention of sufficient information, c) enhancement of separability in feature space, for example in different categories by removing effects due to noisy features, and d) comparability of features among examples in same category [15].

Researchers have attempted input selection through varied means, such as statistical [7] ,[10] geometrical, information-theoretic measures [1], mathematical programming [2], among others.

In statistical analyses, forward and backward stepwise multiple regression (SMR) are widely used to select features, with forward SMR being used more often due to the lesser magnitude of calculations involved. The output here is the smallest subset of features resulting in a correlation coefficient value that explains a significantly large amount of the variance. Similarly in [12], the K–L transform was applied to the training examples to obtain the initial training vectors. Training is started in the direction of the major eigenvectors of the correlation matrix of the training examples. The main drawback of these approaches is that they searche for interdependent features in the input space ignoring the influence of each one on the output of the system.

In [16] genetic algorithms were used for feature selection by encoding the initial set of n features as n-element bit string with 1 and 0 representing the presence and absence respectively of features in the set. The authors used classification accuracy as the fitness function (for genetic algorithms, while selecting features) and obtained good neural network results compared to branch and bound and sequential search algorithms. However, later in was shown in [9] that classification accuracy may be a poor fitness function measure when searching for reducing the dimension of the feature set. Also, the time complexity of the method is overwhelming due to the training process required for each input combination.

Rough sets theory was also used to determine the degree of dependency of sets of attributes for selecting binary features. Features leading to a minimal preset decision tree, which is the one with minimal length of all paths from root to leaves, were selected. Similarly, best first search was used, stopping after a predetermined number of non-improving node expansions [15]. Similar algorithms such as the IDG take the positions of examples in the instance space to select features for decision trees. They limit their attention to boundaries separating examples belonging to different classes, while rewarding (penalizing) rules that separate examples from different (same) classes [15]. Decision trees generated using the proposed algorithm had better accuracy. The implied drawback of this hierarchical approach is that the structure of a decision tree is depended on the specific sequence of features to be tested apart from the features themselves.

Neural networks were also used to measure the contribution of individual input features to the output of the neural network [15]. These methodologies have to undergo the time-consuming training process of the network used to test every input combination.

The most popular feature selection methods in machine learning literature are variations of sequential forward search (SFS) and sequential backward search (SBS) [15]. SFS (SBS) obtains a chain of nested subsets of features by adding (subtracting) the locally best (worst) feature in the set. The serious weakness of this approach is that it adds or subtracts one feature at a time. It results in trapping the search in local minima, because it fails to encode the probing of all the potential combinations.

In [19] Lin and Cuningham proposed a very fast method for input selection introducing the fuzzy curve concept. A fuzzy curve is a non-linear continue curve, which establishes a connection between a specific input and the output, performing a projection of the multidimensional input output space on the (probed input)- output space. The height of the projected output is the measure of importance of the specific input. If the height is sizeable the respective input is considered significant. On the other hand, a substantial (noisy) inputs result in a flat fuzzy curve. The basic advantage of this approach is the linear time complexity of the probing process, with respect to the number of inputs. The serious weakness of the method arises from the fact that the probe of a particular input ignores the impact of rest set. Hence, a specific input could be rejected when it is probed as stand-alone, but it could be significant when it is combined with another one. On the contrary, potential interdependent inputs could be characterized significant when tested independently. In this paper we introduce the fuzzy surface concept as an extension of the fuzzy curve in an entirely different way. We take advantage of the fast model building capabilities derived from the concept of fuzzy curve and a genetic based probing of more than one input simultaneously, to cope with the weakness mentioned above.

The proposed method

Let us consider an *m*-input single output system described by a nonlinear function of the form: $y = f(\underline{x})$, where $\underline{x} = [x_1,..x_j ...,x_m]$ and and *y* denotes the output of the system. Also, let $\wp_q$ denotes the observation input output data set comprising *q* *m*-input/output patterns: $\wp_{q,m,1} = \{(\underline{x}^k, y^k), k = 1,...,q\}$. Let $\Im_m = \{x_1, x_2,...,x_j,...,x_m\}$ be the set of candidate input set representing the input vector (feature space). Also let $\Im_{n,\ell} \mid 1 \le n \le m, and\ \ell = 1,..., \frac{m!}{(m-n)!}$ be a particular subset of $\Im_m$ comprising n of m inputs. The number *n* represents the cardinality of $\Im_{n,\ell}$. The index ℓ denotes a specific selection of order *n*. There are $\binom{n}{m} = \frac{m!}{(m-n)!}$ subsets of order *n*. The total number of subsets of any order derived from the set of *m* candidates' inputs $\Im$, excluding the empty subset, equals $2^m - 1$. Given a specific subset $\Im_{n,l}$ the respective $\wp_{q,n,\ell}$ of the specific *n*-inputs /output data observation data points is derived as a subset of $\wp_{q,m,1}$. For each *n*-input - output datum on $\wp_{q,n,\ell}$ a fuzzy rule with crisp output is created in the following form:

$$R^k : if\ x_1\ is\ A_1^k\ and\ x_2\ is\ A_2^k,...,and\ x_j\ is\ A_j^k,...,and\ x_n\ is\ A_n^k\ then\ y\ is\ y_k \qquad (1)$$

The membership function $\mu_j^k(x_j)$ of the respective fuzzy set A_j^k is given by:

$$\mu_j^k(x_j) = \exp[-(2 \cdot \frac{x_j^k - x_j}{\sigma_j})^2] \qquad (2)$$

Each bell-shaped function is located at x_j^k; the parameter σ_j has a fixed value per input variable, equal to *5-15% of* the x_j variable range. A fuzzy rule base is generated comprising *q* rules, R^k, *k=1,...,q* in the form of equation (1) .

Having determined the product as the fuzzy implication method and using the centroid defuzzyfication technique, the output of the fuzzy model is given by the formula:

$$FS_{n,l}(\underline{x}) = \frac{\sum_{k=1}^{m}\left[\prod_{j=1}^{m}(\mu_j^k(x_j^k))\right]\cdot y^k}{\sum_{k=1}^{m}\mu_j^k(x_j)} \tag{3}$$

Equation (3) provides a continuous and parameter free surface, which approximates the input output data, and behaves as a fuzzy model. The mean absolute percentage error is used to estimate the quality of the approximation:

$$E_{q,n,\ell} = \frac{100}{q}\sum_{k=1}^{q}\frac{\left|FS_{n,l}(\underline{x}^k) - y^k\right|}{\left|y^k\right|}\cdot\% \tag{4}$$

Since the number of rules equals the number of data, the risk of over fitting the data set is major. In order to estimate the validation of the model the data set is subdivided into two subsets. Each subset comprises one half (q/2) of input output patterns. The fuzzy surface is built on the first data subset $\wp_{q/2,n,\ell}$ and it is evaluated on the whole data $\wp_{q,n,\ell}$ according to (4).

It has to be pointed out that the complexity of the aforementioned fuzzy model is preventive for using it as a regular model in order to identify the system. Nevertheless, it is a coarse model that can be easily built, avoiding any time-consuming adaptation processes. Parameters σ_j play a very important role in the proper identification of the system properly. If the values of the parameters are extremely small the fuzzy surface-based model results in over fitting the data increasing $E_{q,n,\ell}$ due to poor generalization. On the other hand, very large value of the parameters σ_j result in inadequate identification performance increasing the $E_{q,n,\ell}$, due to poor identification. Empirical studies have shown that a value calculated by the formula:

$$\sigma_j = \delta\cdot\left(\max(x_j^k) - \min(x_j^k)\right),\ k = 1,\ldots,q \text{ Where } \delta\in[0.05, 0.15] \tag{5}$$

is appropriate option. Moreover, this calculation is adopted in [19]. Therefore, all values of the tunable parameters are calculated by the formula (5) overcoming any iterative time-consuming processes.

For each subset $\Im_{n,\ell}$ a fuzzy surface $FS_{n,l}$ can be built on $\wp_{q/2,n,\ell}$ according to equation (3). For each $FS_{n,l}$ an evaluation measure $R_{n,\ell}$, related both to the modeling performance and to the complexity of $FS_{n,l}$, is defined by the formula:

$$R_{n,l} = w \cdot \frac{100 * E_{q,n,\ell}}{E_{\max}} + (1-w) \cdot \frac{n}{m} \mid w \in (0,1) \qquad (6)$$

where $E_{\max} = \max(E_{q,1,\ell}),\ \left| \ell = 1,...,m \right.$. It is used for the normalization of $E_{q,n,\ell}$ values. The smaller the value of $R_{n,\ell}$ the greater the importance of the respective $FS_{n,l}$. If a specific subset $\Im_{n,\ell}$ comprises non-important inputs, then both terms in (6) increase, raising the respective $R_{n,l}$ value. Inserting significant inputs or deducting negligible ones the respective terms in (6) decrease resulting in lower $R_{n,l}$ values. In the case that two input combinations are equivalent the preferred combination is the one with the lowest order *n*. Therefore, the objective of the proposed input selection method is to track down the subset $\Im_{n,\ell}$ with the minimum $R_{n,\ell}$ value, among all the $2^n - 1$ subsets derived from the set of candidate's inputs $\Im_m$.

The genetic algorithm

In the case that the cardinality *m* of $\Im_m$ is small, the probing of all possible combinations is feasible. On the contrary, for a large set of candidates' inputs the number of combinations is prohibitively high. In general, the minimization of the measure given in (6) can be formulated as a combinatorial unconstrained optimization problem. A simple genetic algorithm with binary encoding, adaptive mutation and crossover rate [18], is applied. In this paper the chromosome of each individual consists of *m* genes of one bit. Each bit encodes the presence '1' or absence '0' of a particular input variable $x_j\ j = 1,...,m$ to the construction of a specific $\Im_{n,\ell} \subseteq \Im_m$. Hence, the phenotype of each individual represents a specific subset $\Im_{n,\ell}$ to be validated. The fitness function, used to assess the quality of each individual, is given by equation (6). In order to enhance the search performance of the genetic algorithm, a specific local search operator, namely Digital Hill climbing Operator (DHCO) [17] is applied to the elite of each generation. This operator selects randomly a relatively small number of bits (i.e. 4 bits) from the elite's chromosome and derives all the potential chromosomes (15 chromosomes) by permuting the selected bits and keeping the rest ones intact. The chromosome with the best fitness value is adopted as the new elite chromosome replacing the initial one.

Experimental results

In order to highlight some functional aspects of the proposed approach two artificial examples are examined initially. Then, the method is applied on two real world applications: the Fisher Iris recognition Benchmark and the Box & Jenkins [4] gas furnace problem.

Example I

The first example concerns the importance evaluation of the inputs for a three-input one output non-linear function [19]:

$$y = [x_1^{1.5} - 1.5\sin(3x_2)]^2 + 5\mathrm{x}_3, \quad x_1, x_2, x_3 \in [0,3] \qquad (7)$$

300 input output data pairs $[x_1, x_2, x_3, y]^k, k = 1,2,...,300$ were produced. The first 150 were used to build the fuzzy surfaces and all the 300 data to calculate the $E_{q,n,\ell}$ in (6). For each pair the input values x_1, x_2, x_3 are randomly generated within the interval $[0,3]$ and the output is calculated according to (7). In (6), the value of the weight w equals $w = 0.8$. Since the number of candidate inputs is small, all the combinations were evaluated without genetic optimization. The evaluation results are illustrated in Table 1, ranked in ascending order with respect to their importance. It is apparent that the combination comprising all the inputs is the most important one. Moreover, a ranking of the importance of each individual input is possible by examining the results for $n = 1$. The order x_2, x_1, x_3 is also recognized in [19]. The time required to evaluate all the combinations was 0.38 sec on a Pentium III 1.2Ghz computer.

n	ℓ	$\Im_{n,l}$	$R_{n,l}$
3	1	x_1, x_2, x_3	9.10
2	1	x_1, x_2	25.04
2	2	x_2, x_3	28.13
1	1	x_2	46.51
2	3	x_1, x_3	47.65
1	2	x_1	65.80
1	3	x_3	80.06

Table 1: Input Evaluation for (7)

Note that the results in Table 1 are not mutually exclusive. Let $\succ$ be a comparative operator regarding the relative importance of two objects. i.e. $a \succ b$ indicates that 'a' is more important than 'b'. The implication:

$$x_r x_t \succ x_s x_t \Leftrightarrow x_r \succ x_s \ \forall r, s, t \quad (8)$$

has to be satisfied [8]. For example: $x_1 x_2 \succ x_1 x_3 \Leftrightarrow x_2 \succ x_3$. This *reliability* constraint is confirmed in Table 1.

Example II

The next example is a modification of the previous one. A fake input is added to the input space, which has no impact to the output of the system. Equation (7) is modified as follows:

$$y = [x_1^{1.5} - 1.5\sin(3x_2)]^2 + 5\mathrm{x}_3, \quad x_1, x_2, x_3, x_4 \in [0,3] \quad (9)$$

The input data points are created extending each input vector $[x_1, x_2, x_3]^k$, created in the previous example with one more component, which represents the x_4 input value. Hence, 300 data points in the form $[x_1, x_2, x_3, x_4, y]^k, k = 1,...300$ are created. The x_4 values are random number in the range [0,3]. The first 150 data is used to craft the fuzzy surfaces, as in Example I. After applying the proposed method, the best combination was x_1, x_2, x_3 with $R_{3,1} = 8.93$. The input variable x_4 is clearly rejected. The combination x_1, x_2, x_3, x_4 that comprises the x_4 input was valuated with $R_{4,1} = 21.21$ which is 225% greater than the best combination. Additionally, the input x_4 was valuated with $R_{1,4} = 85.43$ as the worst option.

Example III

In this example we are dealing with the examination of our method in the case of interdependent inputs. Suppose we have a system governed by the non linear equation:

$$y = \sin(x_1) + \sin(x_2) + 10^{-4} x_3 \textit{ where } x_2 = 3 \cdot \sin(x_1) \qquad (10)$$

The inputs x_1, x_2 are interdependent, while the input x_3 affects slightly the output. 300 input-output data pairs were generated in a manner similar to Example I. The input vector comprises random numbers in [0,3] range and the output for each input vector is calculated by (10). The evaluation of the inputs shows that the best combination was x_1. Both the x_2 and the x_3 inputs were rejected. Note that following to the methodology referred to [19] the input x_2 is detected as an important input because it provides a non-flat fuzzy curve.

The Fisher Iris Benchmark

The second example is a real world application, the well-known Fisher Iris benchmark. This benchmark comprises four candidates inputs that represent measured attributes of a crinum family such as sepal-length, petal length etch. The lilies are classified into three categories represented by an integer number from 1 to 3, according to their attributes. The data set consists of 150 input-output data points. Our method is applied to track down the attributes, which are necessary and sufficient to describe a classification system. The first 75 points were used for building the fuzzy surfaces. The proposed input selection method suggests the combination x_1, x_2, x_3, x_4 as the best one, with $R_{4,1} = 3.10$. The next best in order was x_3, x_4 with $R_{3,1} = 3.18$. All combinations comprising the x_2 input obtain relatively large values and the x_2 valuated as standalone was the worst option. Following, the methodology presented in [19], the input x_2 would be rejected due to a flat fuzzy curve although its combination with the rest inputs affects considerably the output of the system. This a major advantage of the proposed algorithm. Introducing a pseudo input to the set of candidates inputs as in Example II, and reapplying our input selection algorithm the combination x_1, x_2, x_3, x_4 is recognized again as the most important one acquiring a valuation $R_{4,1} = 2.46$. The combination x_3, x_4 is again the next best option with $R_{2,1} = 2.52$, while the combination

x_1, x_2, x_3, x_4, x_5 that comprises the fake input x_5 is worst enough obtaining valuated $R_{5,1} = 13.85$. Moreover, the x_5 input is the last in order option.

The Gas Furnace Problem

The Box & Jenkins [4] gas furnace problem, a well-known real world application, is considered to estimate the validity of the proposed method. This dynamic problem involves a single input *u(t)* that represents the gas flow at time t, and a single output *y(t)* which corresponds to the co_2 concentration in the exhaust gas of the furnace. The intention of many modeling techniques [4], [13], [19] is the prediction of *y(t)*, utilizing past values of both the input *u(t)* and the output *y(t)*. The candidates input set consists of 20 inputs *u(t), u(t-1),….,u(t-10), y(t-1),y(t-2),….y(t-10)* that is $2^{20} - 1 = 1{,}048{,}576$ combinations. The targeted output is *y(t)*. The genetic algorithm mentioned in *section 0* is applied to track down the combination with the minimum $R_{n,l}$ value. The chromosome of the algorithm comprises 20 bits and the fitness function is given by (6). A population of 25 individuals was employed. After 50 generations the best individual achieved a fitness score of $R_{3,1} = 26.83$ suggesting $\underline{x} = [u(t-2), u(t-4), y(t-1)]$ as the most effective input combination. It has to be pointed out that different methods in literature consent regarding the number of inputs to be used, but result in a different subset of the same order $n = 3$. In [13], the proposed input subset was: $\underline{x} = [u(t-3), u(t-4), y(t-1)]$, in [19] the three most important inputs were: u(t-5),u(t-6),y(t-1) and in [3]: $\underline{x} = [u(t-4), u(t-5), y(t-1)]$. In order to assess the validity of each result a feed forward neural network that consists of one hidden layer that comprises 20 neurons, three inputs and a single output is used as a modeling tool for testing the different combinations in a relatively fair manner, overcoming any complex modeling details referred to each one. The Leverbeng-marquat method is applied to train each neural network for 1000 epochs. The mean square error (mse) is used as a performance measure to estimate both the identification and prediction quality of each network. The simulation results are summarized in Table 2.

Input Selection	R	Training Error (mse)	Checking Error (mse)
u(t-2),u(t-4),y(t-1)	26.83	0.032	0.68
u(t-5),u(t-6),y(t-1)	42.77	0.070	1.12
u(t-4),u(t-5),y(t-1)	35.60	0.061	0.67
u(t-3),u(t-4),y(t-1)	28.61	0.031	0.73

Table 2: comparison of the proposed input selection approach

From the experimental results it is clearly shown that the proposed input selection method tracks down the input vector with the greater impact to the output of the system. Additionally, the importance measure $R_{n,\ell}$ is highly correlated with the importance of each combination. The total simulation time was in average 4.33 minutes on a Pentium III computer 1.2Ghz and the total number of tested combinations by the genetic algorithm was: 50 generations * (25 evaluations per generation for the reproduction+16 evaluations per generation for *DHCO* operator)=2048.

Conclusion

A novel, fast and consistent (in terms of equation 8) approach for calculating the importance of each subset of features, that together are assumed to influence the output of a system, from a set of candidates features determined intuitively, is presented in this work. The proposed methodology gets rid of the fast modeling property of the fuzzy curve technique, extended to the fuzzy surface concept. A genetic based search is used to encode the probing of all the potential feature combinations, providing the advantage of detecting potentially interdependent inputs. Extended simulation results confirm the effectiveness of the proposed method.

References:

1. Battiti, R., ‘Using mutual information for selecting features in supervised neural net learning’ IEEE Transactions on Neural Networks 5 (4), (1994), 537–550.

2. Bradley, P.S., Mangasarian, O.L., Street, W.N., ‘Feature selection in mathematical programming’, INFORMS Journal on Computing 10 (2), (1998), 209–217.

3. Barada s. and sing H, ‘ Generating optimal adaptive fuzzy-neural models of dynamical systems with application to control,’ IEEE trans on syst. man and cyb. Part c, vol 28, no 3, pp 371-391, 1998.

4. Box G. E. P. and Jenkins G. M., "Time series analysis, forecasting and control,", San Francisco, CA: Holden Day, 1970.

5. C. C. Lee, “Fuzzy Logic in Control Systems: Fuzzy Logic Controller- Part I,” IEEE Trans. on Syst., Man, and Cybern., vol. 20, No. 2., 1990.

6. C.T. Lin and C.S.G. Lee, “Real-Time Supervised Structure/Parameter Learning for Fuzzy Neural Network,” IEEE Conf. on Fuzzy Systems, pp. 1283-1291, 1992.

7. Fukunaga K., Introduction to Statistical Pattern Recognition, second ed., Academic Press, Boston, 1990.

8. I.J.Leontaritis and S.A. Billings, ‘model selection and validation methods for non-linear systems’, int. j. control, vol 45, no 1, pp. 311-341, 1987.

9. Hopkins, C., Routen, T., Watson, T., ‘Problems with using genetic algorithms for neural network feature selection’, 11th European Conference on Artificial Intelligence. (1994), 221–225.

10. Kittler, J. ‘Mathematical methods of feature selection in pattern recognition’, International Journal of Man–Machine Studies 7 (1975), 609–637.

11. L.-X. Wang and J. M. Mendel, "Back-propagation fuzzy system as non-linear dynamic system identifiers," IEEE Conf. on Fuzzy Systems, pp. 1409-1416, 1992.

12. Malki, H.A., Moghaddamjoo, A., ‘Using the Karhunen–Loeve transformation in the back-propagation training algorithm’ IEEE Transactions on Neural Networks 2 (1), (1991), 162–165.

13. L. Wang and R. Langari, ‘Building Sugeno-type models using fuzzy descretization and orthogonal parameter estimation techniques,’ IEEE Trans. Fuzzy systems, vol. 3, no. 4, pp. 454-458, 1995.

14. M. Sugeno and K. Tanaka, "Successive identification of a fuzzy model and its applications to prediction of a complex system," Fuzzy Sets and Systems, vol. 42, pp. 315-334, 1989.

15. Piramuthu, S. ‘Evaluating feature selection methods for learning in data mining applications’, European Journal of Operational Research 156 (2004), 483–494.

16. Siedlecki, W., Sklansky, J., ‘A note on genetic algorithms for large-scale feature selection’, Pattern Recognition Letters 10 (5), (1989), 335–347.

17. S. E. Papadakis and J. B. Theocharis, "*A GA based Fuzzy Modeling Approach for Generating TSK models*," Fuzzy Sets and Systems, vol 131, no. 2, 121-152, 2002.

18. S. Papadakis and J. Theocharis, “A Novel Genetic Training Algorithm With Application to Fuzzy Neural Networks,” IASTED Int. Conf. On Syst. Modelling Identification and Control pp. 125-128, Feb. 1996.

19. Y. Lin and G. A. Cunningham, “A New Approach to Fuzzy-Neural System Modeling,” IEEE Trans. Fuzzy Systems, vol. 3, pp. 190-198, May 1995.

Authors:

S.E. Papadakis - Technological Educational Institution of Thessaloniki, Greece, E-mail address: *steliospapadakis@mail.gr*

P. Tzionas - Technological Educational Institution of Thessaloniki, Greece, E-mail address: *ptzionas@teithe.gr*

V.G. Kaburlasos - Technological Educational Institution of Thessaloniki, Greece, E-mail address: *vgkabs@teikav.edu.gr*,

J.B. Theocharis - Aristotle University of Thessaloniki, Greece, E-mail address: theochar@vergina.ee.auth.gr