

Robust Agglomerative Clustering Algorithm for Fuzzy Modeling Purposes

Victor H. Grisales^{1,2*}, José J. Soriano¹, Sergio Barato¹, Diana M. Gonzalez¹

¹Laboratorio de Automática, Microelectrónica
e Inteligencia Computacional - LAMIC
Universidad Distrital FJDC
Cra. 8 No. 40-62 P7, Bogotá, Colombia

²Groupe Diagnostic, Supervision et
CONduite qualitatifs - DISCO
Laboratoire d'Analyse et d'Architecture des Systèmes LAAS-CNRS
7 Av. du Colonel Roche, 31077 Toulouse, France

vhgrisales@ieee.org, josoriano@udistrital.edu.co
sbarato,dmgonzalez {@ieee.udistrital.edu.co}

Abstract – This paper addresses Takagi-Sugeno-Kang (TSK) fuzzy model identification. An enhanced algorithm that uses clustering techniques for the approximation of nonlinear systems from data is presented. The algorithm combines the parallel axis version of the Gustafson-Kessel (GK) algorithm with the Fuzzy C-Regression Models (FCRM) in order to maintain the interpretability and improve the global accuracy of the model. A low sensibility to noise and automatic detection of the number of clusters is achieved by using robust statistic and competitive agglomeration techniques similar to the techniques developed in the Robust Competitive Agglomeration (RCA) algorithm. Finally, two numeric examples concerning to static and dynamic nonlinear systems are shown to demonstrate the effectiveness of the proposed algorithm.

I. INTRODUCTION

Currently the use of fuzzy modeling techniques has been increased to approximate nonlinear complex systems. These models have obtained a high performance index where traditional techniques hardly could get it. Among of these fuzzy models, the Takagi-Sugeno-Kang (TSK) model has attracted a great attention [1][2]. This model consists of If-Then rules, where the antecedents are fuzzy sets, while the consequents are functions dependent on the input variables. The TSK model has demonstrated a great capability and flexibility to generate approximations of nonlinear systems from data.

In order to construct TSK models it is necessary to determine the fuzzy sets (membership functions) of the premise part and the parameters of the consequents. Based on the previous knowledge of the system, it could be assumed the type of the membership functions and an initial guess of its parameters. In the consequent, given that functions are usually chosen to be affine, the least squares method can be applied to find the coefficients. Unfortunately, this approach has the drawback of not being able to tune the antecedents, limiting the approximation capability of the model.

There is a difficulty finding the membership functions parameters because it is a nonlinear optimization. The

gradient descent method is typically used. This method has the following drawbacks: (1) The convergence of the method is sensible to the initial parameters; (2) even though the suitable initial parameters were given, it can have a high index of global prediction but lacking of local interpretability. An option to avoid the previous problem is to derive the consequents and antecedents of the model from a partition obtained by means of clustering techniques in the product space of inputs and outputs [3].

With the purpose of searching linear tendencies, three algorithms are traditionally used: the Gustafson-Kessel (GK) algorithm, methods based on Maximum Likelihood Estimation (MLE) algorithm and Fuzzy c-Regression Models (FCRM) [4][5][6]. From these techniques, clusters with rotated axis to the input variables are obtained. This could generate some errors or lack of interpretability when the corresponding TSK models are derived. Another issue is the presence of noise in data or outliers, with a negative influence in the partition generated by the clustering algorithm. Furthermore, in unknown systems it is difficult to determinate beforehand the appropriate number of clusters (rules), which is a requirement for these techniques.

In this paper an algorithm is proposed, in which clusters with parallel axis to the input variables and linear regression submodels are obtained, similar to those used in FCRM. The fact of having non rotated clusters reduces the decomposition errors, without loss of interpretability. In addition, the robust statistics and competitive agglomeration techniques are applied as those used in the Robust Competitive Agglomeration (RCA) and RFRA algorithms [7][8], with the purpose of offering robustness in presence of noise and to obtain the appropriate number of clusters.

The rest of the paper is organized as follows. Section II describes the characteristics of the TSK model; the method used in order to derive the model from data clustering and the problems that appear when traditional techniques are used. Section III presents the proposed robust agglomerative clustering algorithm. Section IV shows the use of the approach in two numerical examples related to both static and dynamic nonlinear systems. Finally, conclusions are given in Section V.

* Supported in part by Programme de Coopération Post-gradué "Systèmes de Commande" France-Colombie.

II. TSK MODEL GENERATION BY MEANS OF FUZZY CLUSTERING

A. TSK Models

The linear TSK models are fuzzy models that consist of rules with the following structure:

$$R_i: \text{If } x_1 \text{ is } A_{i1} \dots \text{ and } x_p \text{ is } A_{ip} \text{ Then } g_i = \mathbf{a}_i^T \mathbf{x} + b_i, i=1, \dots, c \quad (1)$$

Where c is the number of rules, \mathbf{x} is the input vector, A_i and g_i are the multidimensional fuzzy set and the affine function of the i th rule, respectively. In the same way, \mathbf{a}_i and b_i are the parametric vector and the scalar displacement of the i th linear function. The degree of fulfillment for the i th rule is given by:

$$\beta_i(\mathbf{x}) = \mu_{A_{i1}}(x_1) \wedge \mu_{A_{i2}}(x_2) \wedge \dots \wedge \mu_{A_{ip}}(x_p) \quad (2)$$

Where \wedge represents a T-norm [1][3]. The TSK system output is inferred as follows:

$$y = \frac{\sum_{i=1}^c \beta_i(\mathbf{x})(\mathbf{a}_i^T \mathbf{x} + b_i)}{\sum_{i=1}^c \beta_i(\mathbf{x})} \quad (3)$$

The expression (3) shows that the TSK models can play the role of function regressors, they can approximate with certain degree of accuracy any function $y = f(x)$.

B. Existing Algorithms

The parameters obtained in the GK and MLE algorithms are the fuzzy covariance matrix and fuzzy mean, which indicate the direction and the midpoint of each cluster respectively. From the geometric interpretation of the eigenvectors of the covariance matrix and the fuzzy mean, hyperplane equations can be formed, obtaining the consequents parameters corresponding to each cluster. This estimation is equivalent to use Total Least Square (TLS) [3]. The FCRM algorithm seeks to generate the functional prototypes that acts like a local regressor and whose linear combination performs like a global regressor. In order to find the parameters it is possible to choose linear subfunctions so that the algorithm identifies hyperplanar clusters, where the parameters of each submodel are obtained directly from the algorithm. In this case the membership functions play the role of the label of the data. These labels determine with which one of the regression submodels is more related. This algorithm has been put forward by Hathaway and Bezdek [3][6].

The partitions obtained in the algorithms are equivalent to the membership functions of multivariate, which generally are difficult to interpret. With the purpose of obtaining univariate membership functions the multivariate membership functions are projected over each one of the input variables [9]. This projection can present two types of errors: (1) by decomposition, because the obtained clusters have a certain degree of rotation with regard to the input variables. (2) by approach, because the data obtained are projected points, which must be approximated by

parametric functions. The decomposition error can be reduced by using the projection with eigenvectors. However, it deteriorates the transparency and interpretability of the model, since each variable would be expressed as a linear combination of the real variables.

In order to reduce this problem without deteriorating the interpretability, a modification of the parallel axis version of the algorithm MLE has been applied [10]. This algorithm produces clusters of hyperellipsoidal shape, but without rotation with regard to the axis of the input variables. In addition, the parameters of the consequents are estimated through weighted least square method (WLS) in each iteration. The drawback of this algorithm is based on the exponential nature of its distance; it needs to have suitable input parameters to converge in an optimal solution.

Another problem presented is when data experimentally obtained are noisy; this leads to non optimum partitions, when traditional clustering algorithms are used. Furthermore, in these techniques it is necessary to determine previously the number of clusters; this can be a hard task in complex systems without previous knowledge. Having in mind the problems and drawbacks cited above, we propose an algorithm, in which clusters with parallel axis to the input variables and linear regression submodels are obtained. This is accomplished by mixing the parallel axis version of the GK clustering algorithm with the FCRM algorithm; with the purpose of reducing the sensitivity problems at the initialization. Once the algorithm is obtained the robust statistics and competitive agglomeration techniques are applied to have the capability to reject noisy data and appropriate number of clusters detection.

III. PROPOSED ALGORITHM

The original objective functional for the GK algorithm is the following:

$$J(\mathbf{Z}; \mathbf{U}, \mathbf{V}, \mathbf{A}) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A}_{i_s} (\mathbf{z}_k - \mathbf{v}_i) \quad (4)$$

The following restrictions hold

$$\sum_{i=1}^c \mu_{ik} = 1 \quad 1 \leq k \leq N \quad (5)$$

and

$$|\mathbf{A}_{i_s}| = \eta_i \quad \eta_i > 0 \quad \forall i \quad (6)$$

Where c is the number of clusters, N is the number of data points, \mathbf{z}_k is a vector that contains the k th taken data, \mathbf{v}_i is a prototype vector as signed as the i th cluster center, which has the same dimension as \mathbf{z}_k . \mathbf{A}_{i_s} is a symmetrical matrix corresponding to i th cluster, μ_{ik} is the membership function of the k th data to the i th cluster, m influences the fuzziness of the clusters and η_i is the hypervolume of the i th cluster [4]. Being \mathbf{A}_{i_s} symmetrical, the functional given by (4) describes rotated clusters of hyperellipsoidal shape.

Therefore, if clusters with principal axis parallel to the data axis are desired, the matrix \mathbf{A}_{i_d} must be a diagonal matrix,

$$J(\mathbf{Z}; \mathbf{U}, \mathbf{V}, \mathbf{A}) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m (\mathbf{z}_k - \mathbf{v}_i)^T \mathbf{A}_{i_d} (\mathbf{z}_k - \mathbf{v}_i) \quad (7)$$

Subject to (5) and (6). Being \mathbf{A}_{i_d} a diagonal matrix, the functional described by the equation (7) can be rewritten as follows:

$$J(\mathbf{Z}; \mathbf{U}, \mathbf{V}, \mathbf{A}) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \cdot d_{ik}^2 = \left[(\mathbf{x}_k - \mathbf{v}_{ix})^T \mathbf{A}_{ix_d} (\mathbf{x}_k - \mathbf{v}_{ix}) + \sigma_i (y_k - v_{iy})^2 \right] \quad (8)$$

Where $\mathbf{z}_k = [\mathbf{x}_k^T, y_k]^T$, and vector \mathbf{x}_k and y_k are the input and output components of the k th data respectively; similarly, $\mathbf{v}_i = [\mathbf{v}_{ix}^T, v_{iy}]^T$, where vector \mathbf{v}_{ix} and constant v_{iy} contain the input and output components of the center of the i th cluster; \mathbf{A}_{ix_d} is a diagonal matrix, where each element corresponds to each input variable of the matrix \mathbf{A}_{i_d} and σ_i corresponds to the component related with the output variable of \mathbf{A}_{i_d} .

$$\mathbf{A}_{i_d} = \begin{bmatrix} \mathbf{A}_{ix_d} & 0 \\ 0 & \sigma_i \end{bmatrix}$$

Taking the functional given in (8) as a reference, where clusters with parallel axis with regard to the input variables are produced, the term related to the output is modified with the purpose of producing linear submodels as follows:

$$J(\mathbf{Z}; \mathbf{U}, \mathbf{V}, \mathbf{A}, \boldsymbol{\theta}) = \sum_{i=1}^c \sum_{k=1}^N \mu_{ik}^m \left[(\mathbf{x}_k - \mathbf{v}_{ix})^T \mathbf{A}_{ix_d} (\mathbf{x}_k - \mathbf{v}_{ix}) + \sigma_i (y_k - f(\mathbf{x}_k, \boldsymbol{\theta}_i))^2 \right] \quad (9)$$

Where $f(\mathbf{x}_k, \boldsymbol{\theta}_i) = \mathbf{a}_i^T \mathbf{x}_k + b_i$ represents each one of the i th linear submodels. Also, in the same way that in GK algorithm there exists an extra restriction related to the hypervolume of each cluster:

$$|\mathbf{A}_{ix_d}| \sigma_i = \eta_i \quad \eta_i > 0 \quad \forall i \quad (10)$$

This conditions avoids the trivial solution $\mathbf{A}_{ix_d} = 0, \sigma_i = 0$. In (9) it is possible to notice that the term related to the output variable is practically equal to the functional proposed in the FCRM algorithm, except for the σ_i coefficient:

$$J(\mathbf{X}; \mathbf{U}, \boldsymbol{\Omega}) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m [y_k - f_i(\mathbf{x}_k, \boldsymbol{\theta}_i)]^2 \quad (11)$$

Obviously the functional in (9) must be also subject to the restriction (5). In order to minimize (9) subject to (5) and (10), Lagrange multipliers are applied. Combining the developments done in [4] and [6] the following equations are obtained:

$$\mathbf{v}_{ix} = \frac{\sum_{k=1}^N (\mu_{ik}^{l-1})^m \mathbf{x}_k}{\sum_{k=1}^N (\mu_{ik}^{l-1})^m} \quad (12)$$

$$\mathbf{F}_{ix_d}^l = \text{diag} \left[\frac{\sum_{k=1}^N (\mu_{ik}^{l-1})^m (\mathbf{x}_k - \mathbf{v}_{ix}^l) (\mathbf{x}_k - \mathbf{v}_{ix}^l)^T}{\sum_{k=1}^N (\mu_{ik}^{l-1})^m} \right] \quad (13)$$

$$\boldsymbol{\Phi}_i^l = \begin{bmatrix} (\mu_{i1}^{l-1})^m & 0 & \cdots & 0 \\ 0 & (\mu_{i2}^{l-1})^m & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (\mu_{iN}^{l-1})^m \end{bmatrix} \quad (14)$$

$$\mathbf{X}_e = \begin{bmatrix} \mathbf{x}_1^T & 1 \\ \mathbf{x}_2^T & 1 \\ \vdots & \vdots \\ \mathbf{x}_N^T & 1 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad (15)$$

$$\boldsymbol{\theta}_i^l = (\mathbf{X}_e^T \boldsymbol{\Phi}_i^l \mathbf{X}_e)^{-1} \mathbf{X}_e^T \boldsymbol{\Phi}_i^l \mathbf{y} \quad (16)$$

$$f_{iy}^l = \frac{\sum_{k=1}^N (\mu_{ik}^{l-1})^m (y_k - f(\mathbf{x}_k, \boldsymbol{\theta}_i^l))^2}{\sum_{k=1}^N (\mu_{ik}^{l-1})^m} \quad (17)$$

$$\mathbf{A}_{ix_d}^l = \left[\eta_i f_{iy}^l \mid \mathbf{F}_{ix_d}^l \right]^{1/n} (\mathbf{F}_{ix_d}^l)^{-1} \quad (18)$$

$$\sigma_i^l = \left[\eta_i f_{iy}^l \mid \mathbf{F}_{ix_d}^l \right]^{1/n} (f_{iy}^l)^{-1} \quad (19)$$

$$d_{ik}^2 = (\mathbf{x}_k - \mathbf{v}_{ix}^l)^T \mathbf{A}_{ix_d}^l (\mathbf{x}_k - \mathbf{v}_{ix}^l) + \sigma_i^l (y_k - f(\mathbf{x}_k, \boldsymbol{\theta}_i^l))^2 \quad (20)$$

$$\mu_{ik}^l = \frac{1}{\sum_{j=1}^c (d_{ik}^l / d_{jk}^l)^{2/(m-1)}} \quad (21)$$

Where l indicates the iteration number and n indicates the dimension of the input-output product space.

Robust and Competitive Agglomeration Formulation

In real applications, training data sets usually contain noise and outliers which can seriously affect the performance of the clustering algorithm. In addition there exist the problem of determine the appropriate number of clusters. Considering this problems Krishnapuram and

Frigui have used robust statistics and competitive agglomeration techniques to generate the RCA algorithm [7]. Chiang, Su and Chen applied the RCA algorithm on the FCRM algorithm; this technique is called RFRA [8]. On the basis of the previous explanation, the robust and agglomeration competitive formulations are developed for the proposed algorithm

$$J_R(\mathbf{Z}; \mathbf{B}, \mathbf{U}) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \rho_i(d_{ij}^2) - \alpha \sum_{i=1}^c \left[\sum_{k=1}^N w_{ik} \mu_{ik} \right]^2 \quad (22)$$

With d_{ij}^2 described by (20) and subject to the restrictions (5) and (10). In (22) ρ_i is the robust loss function associated to the i th cluster, $w_{ik} = \partial \rho_i(d_{ij}^2) / \partial d_{ij}^2$ is the weight function that represents the typicality of the x_k data with respect to the i th cluster. The second term is the negative of the sum of the square of the clusters cardinalities and it is minimized when cardinality of a cluster is n and the rest of the clusters are empty. The loss function in the first term reduces the noise effects and the weight function discards the noisy data when it calculates the cardinality in the second one. Once α is correctly chosen, the functional J can be used to find compact clusters, while the data is partitioned into a minimum number of clusters. Initially α must be small to allow the formation of small clusters. Next, α must be increased gradually to foment the agglomeration process. When the optimum number of clusters is found, α must be reduced with the purpose of stopping the agglomeration process and allow the algorithm to converge. An appropriate choice of α is explained in [7]. The related expressions are as follows:

$$\alpha(k) = \lambda(k) \frac{\sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \rho_i(d_{ij}^2)}{\sum_{i=1}^c \left[\sum_{k=1}^N w_{ik} \mu_{ik} \right]^2}, \quad (23)$$

$$\lambda(k) = \begin{cases} \lambda_0 e^{-|k_0 - k|/\tau} & \text{if } k > 0 \\ 0 & \text{if } k = 0 \end{cases}$$

In this case λ and α are functions of the number of iterations to fulfill the behavior described previously. To minimize (22) with regard to each of the cluster parameters, μ_{ik} is fixed, obtaining the following expression:

$$\sum_{k=1}^N w_{ik} \mu_{ik}^m \frac{\partial d_{ij}^2}{\partial \beta_i} \quad (24)$$

Where β_i denotes the parameters of the i th cluster. As a result of (24), similar equations to (12)-(17) are obtained, with the ones difference that the term μ_{ik}^m , which is replaced by $\mu_{ik}^m w_{ik}$ and (18)-(21) stay the same.

Based on the developed process in [7], the μ_{ik} value can be expressed as follows:

$$u_{ik} = \frac{1/\rho_i(d_{ik}^2)}{\sum_{l=1}^c 1/\rho_l(d_{lk}^2)} + \frac{\alpha(N_i - \bar{N}_t)}{\rho_i(d_{ik}^2)} = u_{ik}^{RR} + u_{ik}^{Bias} \quad (25)$$

Where $N_i = \sum_{j=1}^N w_{ij} \mu_{ij}$ is called the robust cardinality of the i th cluster and $\bar{N}_t = \sum_{l=1}^c N_l / \rho_l(d_{lk}^2) / \sum_{l=1}^c 1/\rho_l(d_{lk}^2)$ is denoted the weighted average of cardinalities. If the robust cardinality is less than a pre-specified threshold, the corresponding cluster is discarded. Several functions of loss and weight exist; in this case it will be taken the same functions used in [7], which are adaptations of the Tukey's biweight functions, given by:

$$w_i(d^2) = \begin{cases} 1 - \frac{d^2}{2T_i^2} & \text{if } d^2 \in [0, T_i] \\ \frac{[d^2 - (T_i + qS_i)]^2}{2q^2S_i^2} + \frac{5T_i + qS_i}{6} & \text{if } d^2 \in [T_i, T_i + qS_i] \\ 0 & \text{if } d^2 > [T_i + qS_i] \end{cases} \quad (26)$$

$$\rho_i(d^2) = \begin{cases} d^2 - \frac{d^6}{6T_i^2} & \text{if } d^2 \in [0, T_i] \\ \frac{[d^2 - (T_i + qS_i)]^3}{6q^2S_i^2} + \frac{5T_i + qS_i}{6} & \text{if } d^2 \in [T_i, T_i + qS_i] \\ \frac{5T_i + qS_i}{6} + K_i & \text{if } d^2 > [T_i + qS_i] \end{cases}$$

Where $K_i = \max_{1 \leq j \leq c} \left\{ \frac{5T_j + qS_j}{6} \right\} - \frac{5T_i + qS_i}{6}$ for $i = 1, \dots, c$

ensures that any noisy data will have the same weight function in all clusters. In (26) T_i and S_i denote the median (*Med*) and median of absolute deviations (*MAD*) respectively, which are used to normalize the distances. The constant q is a tuning parameter, which must start with a large value and is decreased with time run. This parameter is chosen as a function of the number of iterations and can be calculated by:

$$q_k = \max(q_{\min}, q_{k-1} - \Delta q) \quad q_0 = 12, q_{\min} = 4 \text{ and } \Delta q = 1 \quad (27)$$

IV. APPLICATIONS

In this section, two examples of fuzzy modeling concerning to static and dynamic nonlinear systems are shown to demonstrate the effectiveness of the proposed algorithm.

In the first one, a three-dimensional function $z=f(x,y)$ that corresponds to static nonlinear behavior is modeled, defined by:

$$z = 100(y - x)^2 - (1 - y)^2 + \varepsilon \quad (x, y) \in [-1, 1]^2 \quad (28)$$

Where $\varepsilon \sim N(0, 1)$ is an added noise distributed normally with mean 0 and standard deviation of 1. For this example 441 training points were taken. The algorithm was run initially with 10 clusters and the threshold to discard clusters was established equal to 10. Five clusters were obtained. A comparison between the clusters obtained through the proposed and GK algorithm is depicted in the Fig.1. The original and the obtained model are depicted in the Fig.2. For comparison purposes, the GK and MLE algorithm are implemented by applying the orthogonal projection method in order to get univariate membership functions.

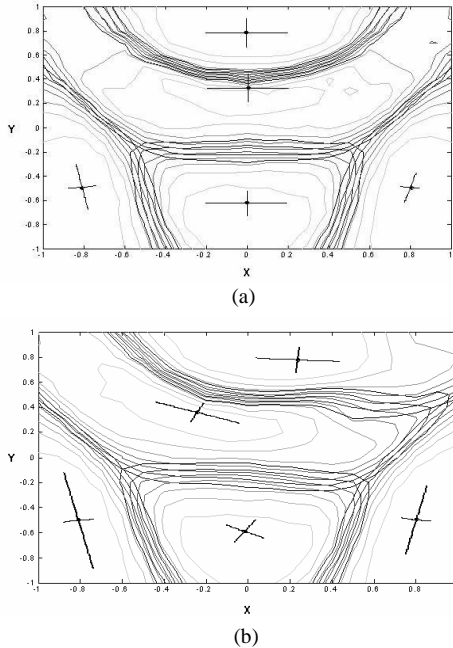


Fig.1. Projection over the input variables of clusters. (a) Generated by the proposed algorithm. (b) Generated by the GK algorithm.

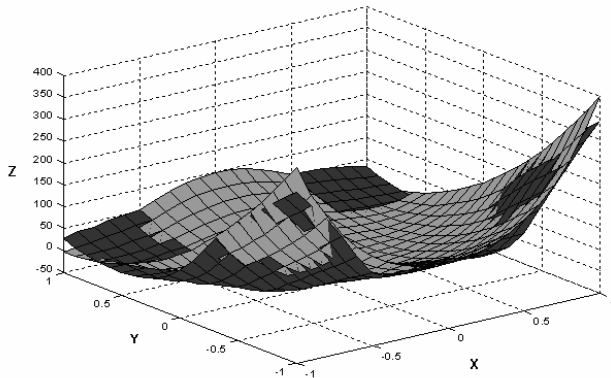


Fig.2. Comparison between the original model (black) and the obtained model through the proposed algorithm (gray).

The performance index $RMSE = 1/N \sqrt{\sum_{j=1}^N (y_j - \hat{y}_j)^2}$ (Root Mean Square Error) was used to evaluate the models. The results are registered in Table I.

TABLE I. PERFORMANCE INDEX COMPARISON

Algorithm	MLE	GK	Proposed
RMSE	23.0247	24.5280	16.9507

In the second example a nonlinear dynamic system is considered, which is described by:

$$y(k+1) = 0.8y(k) + v(k) + \varepsilon(k) \quad \text{where}$$

$$v = \begin{cases} \max(-1.5, u), & u \leq -0.4 \\ 0, & -0.4 \leq u \leq 0.2 \\ \min(u, 0.5), & u \geq 0.2. \end{cases} \quad (29)$$

Where u is the input signal. In order to evaluate the model behavior in the presence of different levels of noise, a Gaussian distributed noise with mean zero and three different values of standard deviation was added, $\varepsilon(k) \sim N(0, \sigma)$, with $\sigma_0=0.05$, $\sigma_1=0.1$, and $\sigma_2=0.3$. Initially a training signal uniformly distributed was applied in $[-0.4, 0.4]$ with 400 sample data. For the approximation process a first order NARX model was selected, so that $y(k+1)$ output is a function of the previous output $y(k)$ and the $u(k)$ input, this means $y(k+1)=f(y(k), u(k))$.

The input and output data were submitted to the clustering process using the proposed algorithm. Initially 10 clusters were considered and a discard limit of 5, finally 4 clusters were obtained. In order to illustrate the algorithm behavior in presence of noise, a case where the noise has a greater standard deviation was considered. The corresponding membership functions for the input and output are shown in Fig.3.

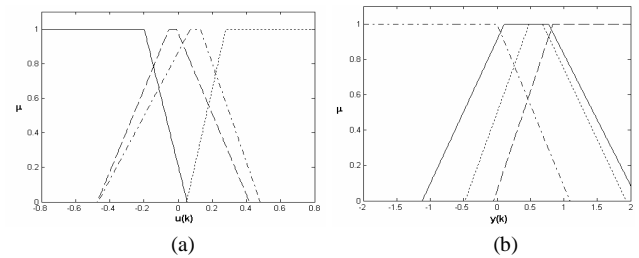


Fig.3. Obtained membership functions: (a) Corresponding to the system input u . (b) Corresponding to the system output y

A comparison between the real model and the approximate model outputs due to the training signal is illustrated in Fig. 4(a). To validate the model a sinusoidal signal $u(k) = 0.35\sin(2\pi k/25)$ was applied. In the Fig. 4(b) the output of the TSK model, in presence of the validation signal is compared with the output of the real system. In both cases, the approximate model using the proposed algorithm gives a close output to the real model.

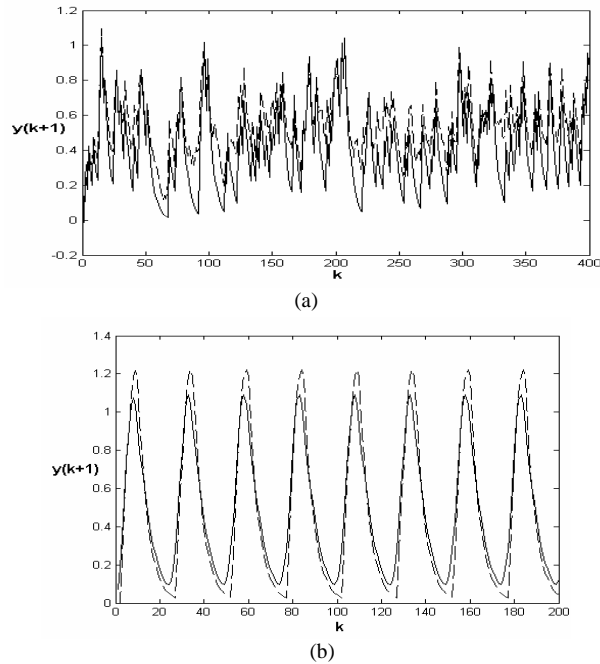


Fig. 4. Outputs due to (a) Training signal, (—) Original model without presence of noise, (---) TSK model from the proposed algorithm and (b) Validation signal, (—) Original model without presence of noise, (---) TSK model output from the GK algorithm.

To evaluation purposes, the GK and MLE algorithms were run over the same training signal and to the obtained models which the validation signal was applied. Over the obtained outputs, the RMSE criteria were applied; the results are shown in Table II for several noise levels, in the training and validation signals respectively.

TABLE II. COMPARISON OF THE RMSE PERFORMANCE INDEX FOR THE TRAINING AND VALIDATION SIGNALS

σ	RMSE Performance Index					
	Training Signal			Validation Signal		
	Proposed Algorithm	GK	MLE	Proposed Algorithm	GK	MLE
0.05	0.0691	0.049	0.144	0.0908	0.042	0.083
0.1	0.0979	0.084	0.141	0.1856	0.338	0.176
0.3	0.1686	1.446	2.328	0.1067	1.491	1.687

V. CONCLUSIONS

The developed algorithm reduces the decomposition error in the membership functions projected, due to the fact that clusters are parallel to the axis, without deterioration of the model interpretability. Unlike of the obtained membership functions with GK or MLE algorithms, in which the projection error is greater due to the generation of rotated clusters. If is desired to reduce this error, the variables from the obtained model must be expressed like a linear combinations of the real input variables, which lead to vanish the interpretability.

Furthermore, submodels that describe well enough the general model are obtained, in comparison with generated models with other algorithms, as it is proved by the RMSE index. In situations where the data extraction from the system can cover a great amount of operation ranges, the proposed algorithm can become in a good modeling option of nonlinear dynamic systems. It was not necessary to indicate the number of clusters required, since the algorithm finds them by itself, when the agglomeration techniques are applied.

In order to choose the threshold, one must consider carefully the limits, if this is too high with regard to the number of clusters, the algorithm will end discarding all clusters. On the other hand, with a very small number the agglomeration process would not be carried out. Finally through the showed examples it was illustrated that the algorithm is able to detect and to reduce the effect of outliers in the training data thanks to the robust statistics applied.

VI. ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees and reviewers for their valuable suggestions.

VII. REFERENCES

- [1] T. Takagi, M. Sugeno, Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE TO Systems, Man and Cybernetics*, Vol.15, No. 1, 1985, pp.116-132.
- [2] M. Sugeno, T. Yasukawa, A Fuzzy-Logic-Based Approach to qualitative modeling, *IEEE Trans. On Fuzzy Systems*, Vol.1, No.1, 1993, pp. 7-31.
- [3] R. Babuska, *Fuzzy Modeling for Control*, Kluwer Academic Publishers, 1998.
- [4] D. Gustafson, W. Kessel, Fuzzy Clustering With a Fuzzy Covariance Matrix, *IEEE CDC*, 1979, pp.761-766.
- [5] J. C. Bezdek, J. C. Dunn, Optimal Fuzzy Partitions: A Heuristic for Estimating the Parameters in a Mixture of Normal Distributions”, *IEEE Trans. On Computers*, 1975, pp. 835-838.
- [6] R. Hataway, J. Bezdek, Switching Regression Models and Fuzzy Clustering, *IEEE Trans. On Fuzzy Systems*, Vol.1, No.3, 2002, pp. 195-204.
- [7] H. Frigui, R. Krishnapuram, A Robust Competitive Clustering Algorithm With Applications in Computer Vision, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.21, No.5, 1999, pp. 450-465.
- [8] Ch. Chuang, Sh. Su, S. Chen, Robust TSK Fuzzy Modeling for Function Approximation With Outliers, *IEEE Trans. On Fuzzy Systems*, Vol.9, No.6, 2001, pp. 810-821.
- [9] E. Kim, M. Park, S. Kim, A Transformed Input-Output Approach to Fuzzy Modeling, *IEEE Trans. On Fuzzy Systems*, Vol.6, No.4, 1998, pp. 596-604.
- [10] J. Abony, R. Babuska, F. Szeifert, Modified Gath-Geva Fuzzy Clustering for Identification of Takagi-Sugeno Fuzzy Models, *IEEE Trans. On Syst, Man and Cybernetics, Part B*, Vol.32, No.5, 2002, pp. 612-621.