AUTHOR QUERY FORM

ELSEVIER	**Journal:** ISA Transactions **Article Number:** 1423	**Please e-mail or fax your responses and any corrections to:** **E-mail: corrections.essd@elsevier.river-valley.com** **Fax: +44 1392 285879**

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list.

For correction or revision of any artwork, please consult http://www.elsevier.com/artworkinstructions.

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Click on the 'Q' link to go to the location in the proof.

Location in article	Query / remark click on the Q link to go Please insert your reply or correction at the corresponding line in the proof
Q1	We have deleted "+" symbol in the below equation. Please check, and correct if necessary.
Q2	We have deleted "cdot" in the below equation. Please check, and correct if necessary.
Q3	We have changed ' $<='$ ' into ' $\leq'$ ' in the below equation. Please check, and correct if necessary.
Q4	Please provide the year for the following reference(s): Refs. [1, 40].

Thank you for your assistance.

ARTICLE IN PRESS

ISA Transactions xx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes

Dejan Dovžan *, Igor Škrjanc

Faculty of Electrical Engineering Tržaška 25, Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 14 October 2009
Received in revised form
9 December 2010
Accepted 11 January 2011
Available online xxxx

Keywords:
Recursive fuzzy *c*-means clustering
Recursive fuzzy identification
Clustering
Online recursive identification

ABSTRACT

In this paper we propose a new approach to on-line Takagi–Sugeno fuzzy model identification. It combines a recursive fuzzy *c*-means algorithm and recursive least squares. First the method is derived and than it is tested and compared on a benchmark problem of the Mackey–Glass time series with other established on-line identification methods. We showed that the developed algorithm gives a comparable degree of accuracy to other algorithms. The proposed algorithm can be used in a number of fields, including adaptive nonlinear control, model predictive control, fault detection, diagnostics and robotics. An example of identification based on a real data of the waste-water treatment process is also presented.

1. Introduction

Takagi–Sugeno models are a powerful practical engineering tool for the modeling and control of complex systems. They expand and generalize the well-known concept of gain scheduling. T–S models utilize the idea of linearization in a fuzzily defined region of the state space. Due to the fuzzy regions (clusters), the nonlinear system is decomposed into a multi-model structure consisting of linear models [1]. This enables the T–S fuzzy model to approximate virtually any nonlinear system within a required accuracy, provided that enough regions are given [2].

To build the T–S model the structure and the parameters of the local models must be identified [3]. Structure identification includes an estimation of the cluster centers (antecedent parameters), which is usually done by fuzzy clustering. Then for each cluster the sub-model's parameters are estimated, which is usually done with a least-squares method [4].

The identification can be made off-line [5–8] or on-line [9–11]. The methods for off-line identification suppose that all the data are available at the start of the training process (identification). The use of this method for on-line identification is limited [9]. For on-line identification the whole model structure must be re-trained using time consuming iterative techniques such as back-propagation [7], genetic algorithms [12,13] or other nonlinear techniques [14].

On-line fuzzy model identification methods mainly use the recursive or weighted recursive least squares for the identification of the sub-model parameters. The key differences among them are the different methods of fuzzy structure identification, i.e., how the clustering and the structure evolution are made.

For on-line identification of nonlinear systems a variety of methods were developed in recent years. The approaches use different mechanisms for the rules update. Some of the methods that are based on fuzzy logic or a combination of fuzzy logic and neural networks are listed below.

The evolving fuzzy neural network method (EFuNN) [15], which was introduced as a part of the ECOS framework [16] combines the neural network and fuzzy logic to build the nonlinear model. The distance between new data sample and cluster center is used to adjust rules. The fuzzy output space adjusts through the output error through Widrow–Hoffs' LMS algorithm. A detailed description can be found in [17]. Similar to the EFuNN is the dynamic evolving neural-fuzzy inference system algorithm (DENFIS) [10]. It inherits and develops the EFuNN's dynamic features. The DENFIS employs the evolving clustering method (ECM) [10], which is a maximum-distance-based clustering method. A scatter partitioning of the input space is used for the purpose of creating the fuzzy rules. The ECM is a one-pass algorithm for a dynamic estimation of the number of clusters in a set of data and for finding their current centers in the input data space. With respect to the predefined maximum distance between the new data sample and the current cluster centers and the cluster center radius the position of the centers is either updated and the radius is increased or a new cluster is introduced. The distance of any new input data to the cluster centers never exceeds the predefined maximum distance. The DENFIS uses triangular membership functions and the first-order T–S-type fuzzy rules

* Corresponding author.
E-mail addresses: dejan.dovzan@fe.uni-lj.si (D. Dovžan), igor.skrjanc@fe.uni-lj.si (I. Škrjanc).

0019-0578/$ – see front matter © 2011 ISA. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.isatra.2011.01.004

Please cite this article in press as: Dovžan D, Škrjanc I. Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes. ISA Transactions (2011), doi:10.1016/j.isatra.2011.01.004

are employed. It uses a recursive, weighted least-squares estimator to estimate the sub-models.

The evolving Takagi–Sugeno model (eTS) [9], which is a further development of the evolving rule-based models (eR) [18] uses the informative potential of a new data sample as a trigger to update the rule base. The eR generates a new rule if there is significant new information present in the collected data. It uses the recursive clustering algorithm [19] based on subtractive clustering [4], which is an improved version of mountain clustering [20], to estimate the clusters. Based on a calculation of the potential between the new data sample and the cluster centers the old clusters are updated or, if the potential exceeds a certain predefined potential threshold, new clusters are introduced. For a sub-model parameters estimation the recursive or weighted recursive least squares are used with the resetting co-variance matrix and the parameters initialization, which is needed because of the rule insertion.

The self-organizing fuzzy neural network (SOFNN) method [21] uses the N-first nearest neighborhood heuristic to define the width of the Gaussian membership functions. The centers of the clusters are obtained with the rival penalized algorithm (RPLC) [22]. The SOFNN structure identification includes adding, pruning and combination of neurons. The new neurons are added when the error criterion or the sample-coverage criterion are not met; the pruning of a neuron occurs when the neuron makes a little contribution to error reduction. The combination of the neuron occurs when two neurons or membership functions are highly similar.

The SAFIS-sequential adaptive fuzzy inference system [23] is similar to the SOFNN, but with different criteria for the rule adding. For a new training sample the adding of the rule is based on the input-data coverage and occurs when a new training data sample is outside the distance threshold to the existing centers of the neurons (i.e., the centers of Gaussian membership functions) and its influence on the output is large enough (the new neuron is added when its contribution to the output is significant). Both the SOFNN and SAFIS use Gaussian membership functions as apposed to the incremental construction learning algorithm (ICLA) [24] that uses triangular membership functions. In the ICLA the structure identification is driven by the error reducing and the structure-evolving mechanism. This ensures the fine partition for the input-data region with high output variation and the coarse partition for the input-data region with the small output variation. This is in contrast to the unformed SOFNN partitioning and the time-decaying SAFIS partitioning. The ICLA should make more appropriate input space partitions.

There are also other on-line methods that deal with the fuzzy structure identification, based on more or less the same principles as the above-described methods, such as the self-constructing neural network (SCFNN) [25], the dynamic fuzzy neural network (D-FNN) [26] and the general dynamic fuzzy neural network (GD-FNN) [27]. They use the error to update the fuzzy rules, like with the SOFNN. The self-constructing neural fuzzy interface network (SONFIN) [28] uses the distance from the cluster centers to update the rules like with the DENFIS; the neural fuzzy control network algorithm (NFCN) [29] uses Kohen's feature maps; the neuro-fuzzy ART-based structure and parameter learning TSK model (Neuro-FAST) [30] uses the adaptive resonance theory (ART) concept; and the generalized adaptive neuro-fuzzy inference system (GANFIS) [31] is using modified mountain clustering [32].

In this article a recursive c-means clustering method is developed and tested against some established on-line clustering methods (DENFIS [10], ETS [9], RAN [33], ESOM [34], EFuNN [15] and Neural gas [35]) on a benchmark problem of the Mackey–Glass time series [36].

The proposed method is similar to Angelov and Filev's on-line clustering [9]. The difference is that their method is based on the subtractive clustering off-line method [4], whereas our method is based on the fuzzy c-means off-line clustering method [37]. The positions of the centers in our method depend on a weighted mean of the data belonging to the i-th cluster, whereas with Angelov's method the centers can only be the data samples with the largest potential. Also, the width of the membership function in our method depends on the fuzzy variance and changes depending on the data pattern, whereas with Angelov's method the width is fixed and predefined. Both methods use the Gaussian membership functions. The local linear sub-models are, in both cases, updated with recursive least squares.

The paper shows that the proposed method gives good results with a small number of clusters which makes it computationally efficient and transparent. It can also be easily compared to other established methods. The method can be used for clustering large numbers of data, where the off-line methods cannot be used because of memory demands. The method can be used in combination with control algorithms that use fuzzy models, to construct the adaptive versions of the algorithms (for example fuzzy predictive functional controllers). It can also be easily modified to construct a recursive version of the Gustafson–Kessel clustering algorithm, which is probably one of the most frequently used algorithms when constructing the fuzzy models for control purposes.

2. Fuzzy c-means clustering

In this section we will first give a brief description of the fuzzy c-means clustering algorithm [37], which is the foundation for our further development of a recursive fuzzy clustering algorithm.

We assume that each observation consists of m samples, which are grouped into an m-dimensional vector $x(k)^T = [x_1(k), \ldots, x_m(k)]$, $x(k) \in \mathbb{R}^m$, where $x_m(k)$ stands for the m-th measurement at the time instant k. A set of n observations is then denoted as $X = \{x(k) \mid k = 1, 2, \ldots, n\}$, $X \in \mathbb{R}^{n \times m}$.

The main objective of the clustering is to partition the data set X into c subsets, which are called clusters. The data matrix X is given as follows:

$$X = \begin{bmatrix} x_1(1) & x_2(1) & \cdots & x_m(1) \\ x_1(2) & x_2(2) & \cdots & x_m(2) \\ \vdots & \vdots & \vdots & \vdots \\ x_1(n) & x_2(n) & \cdots & x_m(n) \end{bmatrix}. \quad (1)$$

The data vector at the time instant k is defined as (the rows of the matrix X) $x(k)^T = [x_1(k), \ldots, x_m(k)]$, $x(k) \in \mathbb{R}^m$. The fuzzy partition of the set X is a family of fuzzy subsets $\{A_i \mid 1 \le i \le c\}$. These fuzzy subsets are defined by their membership functions, which are implicitly defined in the fuzzy partition matrix $U = [\mu_i(k)] \in \mathbb{R}^{c \times n}$. The i-th row of the matrix U contains the values of the membership function of the i-th fuzzy subset A_i of the data matrix X. The partition matrix satisfies the following conditions: the membership degrees are real numbers from the interval $\mu_i(k) \in [0, 1], 1 \le i \le c, 1 \le k \le n$; the total membership of each of the samples in all the clusters equals one $\sum_{i=1}^{c} \mu_i(k) = 1, 1 \le k \le n$; none of the fuzzy clusters is empty nor do any contain all the data $0 < \sum_{k=1}^{n} \mu_i(k) < n, 1 \le i \le c$. This means that the fuzzy partition matrix U belongs to the fuzzy partition set, which is defined as:

$$M = \left\{ U \in \mathbb{R}^{c \times n} \mid \mu_i(k) \in [0, 1], \forall i, k; \sum_{i=1}^{c} \mu_i(k) = 1, \forall k; \right.$$
$$\left. 0 < \sum_{k=1}^{n} \mu_i(k) < n, \forall i \right\}. \quad (2)$$

The c-means algorithm for clustering in n dimensions produces c mean vectors that represent c classes of data. The algorithm

relies on a distortion measure $d(x(k), v_i)$ between the points in data space $x(k)$, $v_i \in \mathbb{R}^m$, where $x(k)$ denotes a certain point in that space, an observation, and v_i stands for the centroid. A variety of different norms can be used to define the distortion measure, such as L_1, L_2 and L_∞, or any other specific to the problem. The algorithm is based on a minimization of the fuzzy c-means objective function, which is introduced as the weighted-criterion function

$$J(X, U, V) = \sum_{i=1}^{c}\sum_{k=1}^{n}(\mu_i(k))^\eta d^2(x(k), v_i) \tag{3}$$

subject to the constraints

$$\sum_{i=1}^{c}\mu_i(k) = 1 \;\forall k, \tag{4}$$

where V is a matrix of cluster centroid vectors v_i, $V = [v_1, \ldots, v_c]^T$, and the overlapping factor or the fuzziness parameter η that influences the fuzziness of the resulting partition is denoted as η; from the hard ($\eta = 1$) to the partition that is completely fuzzy ($\eta \to \infty$). In our approach the standard value $\eta = 2$ is used. In the case of the classical c-means clustering algorithm the distortion measure is defined as L_2 norm.

The problem of finding the fuzzy clusters in the data set X is now solved as a constrained optimization problem using Lagrange multipliers, which consider the minimization of the function in Eq. (3) over the domain X, and taking into account the constraints in Eq. (4).

The objective function is now defined as follows

$$J(X, U, V, \lambda) = \sum_{i=1}^{c}\sum_{k=1}^{n}(\mu_i(k))^\eta (x(k) - v_i)^T(x(k) - v_i) + \sum_{k=1}^{n}\lambda_k\sum_{i=1}^{c}(\mu_i(k) - 1) \tag{5}$$

where λ_k, $k = 1, \ldots, n$ are the Lagrange multipliers.

The minimum of the objective function $J(X, U, V, \lambda)$ is obtained via the Lagrange multipliers method and is given as follows:

$$\mu_i(k) = \left(d_{ik}^2\sum_{j=1}^{c}\left(\frac{1}{d_{jk}^2}\right)^{\frac{1}{\eta-1}}\right)^{-1} \tag{6}$$

where d_{ik} defines the Euclidian distance (L_2-norm) between the observation $x(k)$ and the cluster centroid v_i as follows:

$$d_{ik}^2 = (x(k) - v_i)^T(x(k) - v_i), \quad 1 \le i \le c, \; 1 \le k \le n. \tag{7}$$

The cluster centroid v_i is defined as the weighted mean of the data belonging to the i-th cluster, where the weights are the membership degrees and are given as follows:

$$v_i = \frac{\sum_{k=1}^{n}\mu_i^\eta(k)x(k)}{\sum_{k=1}^{n}\mu_i^\eta(k)}. \tag{8}$$

3. Recursive fuzzy c-means clustering

When the behavior of the process that generates the observed data changes during the time, the clustering should be done recursively to obtain the clusters that describe the current behavior.

3.1. The recursive center calculation

To develop the recursive fuzzy clustering algorithm we will first define the cluster centroid vector $v_i^T = [v_{i1}, \ldots, v_{in}]$ according to the current observation, i.e., the weighted mean of the data according to the current membership degrees. This introduces the notation $v_i(r)$, which means the cluster centroid at the time instant r that is obtained by weighting with the current membership degrees. The cluster centroid in the next observation is denoted as

$$v_i(r+1) = \frac{\sum_{k=1}^{r+1}\mu_i^\eta(k)x(k)}{\sum_{k=1}^{r+1}\mu_i^\eta(k)} = \frac{\sum_{k=1}^{r}\mu_i^\eta(k)x(k) + \mu_i^\eta(r+1)x(r+1)}{\sum_{k=1}^{r}\mu_i^\eta(k) + \mu_i^\eta(r+1)} \tag{9}$$

where $\mu_i(k)$, $k = 1, \ldots, r+1$ denotes the membership degree of the observation vector $x(k)^T = [x_1(k), \ldots, x_m(k)]$, $k = 1, \ldots, r+1$ to the cluster i at the time instant k. Introducing the relation between the old cluster centroid and a new one as follows:

$$v_i(r+1) = v_i(r) + \Delta v_i(r+1) \tag{10}$$

and taking into account Eq. (9) the Eq. (14) is obtained:

$$v_i(r+1) = \frac{\sum_{k=1}^{r}\mu_i^\eta(k)x(k)/\sum_{k=1}^{r}\mu_i^\eta(k)}{\sum_{k=1}^{r+1}\mu_i^\eta(k)/\sum_{k=1}^{r}\mu_i^\eta(k)} + \frac{\mu_i^\eta(r+1)x(r+1)}{\sum_{k=1}^{r}\mu_i^\eta(k) + \mu_i^\eta(r+1)} \tag{11}$$

$$v_i(r+1) = \frac{v_i(k)\sum_{k=1}^{r}\mu_i^\eta(k)}{\sum_{k=1}^{r+1}\mu_i^\eta(k)} + \frac{\mu_i^\eta(r+1)x(r+1)}{\sum_{k=1}^{r}\mu_i^\eta(k) + \mu_i^\eta(r+1)} \tag{12}$$

$$v_i(r+1) = v_i(k) - \frac{v_i(k)\mu_i^\eta(r+1)}{\sum_{k=1}^{r+1}\mu_i^\eta(k)} + \frac{\mu_i^\eta(r+1)x(r+1)}{\sum_{k=1}^{r}\mu_i^\eta(k) + \mu_i^\eta(r+1)} \tag{13}$$

$$\Delta v_i(r+1) = \frac{\mu_i^\eta(r+1)(x(r+1) - v_i(r))}{\sum_{k=1}^{r}\mu_i^\eta(k) + \mu_i^\eta(r+1)}. \tag{14}$$

The cluster centroid increment in Eq. (14) cannot be calculated in the present form, because we cannot calculate the denominator in Eq. (14). The calculation of the membership degrees requires all the past r observations. This is against the recursive approach. An approximate calculation of this term can be made by introducing the exponential weighting of the past membership degrees, which are calculated at each time instant. The weights of the past data are decreasing exponentially.

The term in the denominator is denoted as $s_i(k) \in \mathbb{R}^c$ and calculated as

$$s_i(r+1) = \gamma_v s_i(r) + \mu_i^\eta(r+1) \tag{15}$$

where $s_i(r)$ is defined as follows

$$s_i(r) = \sum_{k=1}^{r}\mu_i^\eta(k). \tag{16}$$

The parameter γ_v, $(0 \le \gamma_v \le 1)$ denotes the forgetting factor of a past observation, i.e., the forgetting factor of the past membership degrees. The $\Delta v_i(r+1)$ can now be written as:

$$\Delta v_i(r+1) = \frac{\mu_i^\eta(r+1)(x(r+1) - v_i(r))}{s_i(r+1)}. \tag{17}$$

Please cite this article in press as: Dovžan D, Škrjanc I. Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes. ISA Transactions (2011), doi:10.1016/j.isatra.2011.01.004

The current membership degree $\mu_i(r+1)$ is next defined as follows:

$$\mu_i(r+1) = \left(d^2_{i,r+1} \sum_{j=1}^{c} \left(\frac{1}{d^2_{j,r+1}} \right)^{\frac{1}{\eta-1}} \right)^{-1} \quad (18)$$

where $d^2_{i,r+1}$ defines the quadratic distance from the cluster centroid as follows:

$$d^2_{i,r+1} = (x(r+1) - v_i(r))^T (x(r+1) - v_i(r)), \quad 1 \le i \le c. \quad (19)$$

3.2. The recursive fuzzy covariance calculation

Around each cluster centroid the distribution of the data can be described using the fuzzy covariance matrix, $F_i(r) \in \mathbb{R}^{m\times m}$, $i = 1, \ldots, c$ and r stands for the number of samples used to calculate the matrix. The fuzzy covariance matrix is defined as the weighted covariance matrix as follows:

$$F_i(r) = \frac{\sum_{k=1}^{r} \mu_i^{\eta}(k)(x(k) - v_i^r)(x(k) - v_i^r)^T}{\sum_{k=1}^{r} \mu_i^{\eta}(k)} \quad (20)$$

where v_i^r stands for the centroid vector of the i-th cluster calculated for the set of r samples. The fuzzy covariance matrix at the next sample $(r+1)$ can be expressed as follows

$$F_i(r+1) = \frac{\sum_{k=1}^{r} \mu_i^{\eta}(k)(x(k) - v_i^{r+1})(x(k) - v_i^{r+1})^T}{\sum_{k=1}^{r} \mu_i^{\eta}(k) + \mu_i^{\eta}(r+1)} + \frac{\mu_i^{\eta}(r+1)(x(r+1) - v_i^{r+1})(x(r+1) - v_i^{r+1})^T}{\sum_{k=1}^{r} \mu_i^{\eta}(k) + \mu_i^{\eta}(r+1)} \quad (21)$$

where v_i^{r+1} stands for the centroid vector of the i-th cluster calculated for the set of $r+1$ samples. Taking into account Eq. (20), introducing it into Eq. (21), and using Eqs. (15) and (16), the following approximate recursive expression for the fuzzy clustering matrix is obtained

$$F_i(r+1) = \gamma_c \frac{s_i(r)}{s_i(r+1)} F_i(r) + \frac{\mu_i^{\eta}(r+1)}{s_i(r+1)} \cdot (x(r+1) - v_i(r+1))(x(r+1) - v_i(r+1))^T. \quad (22)$$

The initial value for the matrix F_i can be set as $F_i(0) \approx, I \in \mathbb{R}^{m\times m}$.

3.3. Applying the recursive least squares

The centers of the fuzzy clusters and their distribution are used to define the new membership functions' distribution and using the recursive least squares method the fuzzy model is obtained. Using the projection of the cluster onto the independent variables, the input membership functions are obtained. Here we are assuming that the first $m-1$ measured variables represent the input variables and the last m-th variable in the data matrix X represents the output. In our case the clusters are approximated by the Gaussian membership functions, with center v_i and the variance $\sigma^2_{i,j} = \eta_m f_{i,j}$, where using η_m the overlapping between the membership functions is defined and $f_{i,j}$ stand for the diagonal elements of the fuzzy covariance matrix. The membership function of the i-th cluster and the j-th component of $x(k)$ is therefore defined as

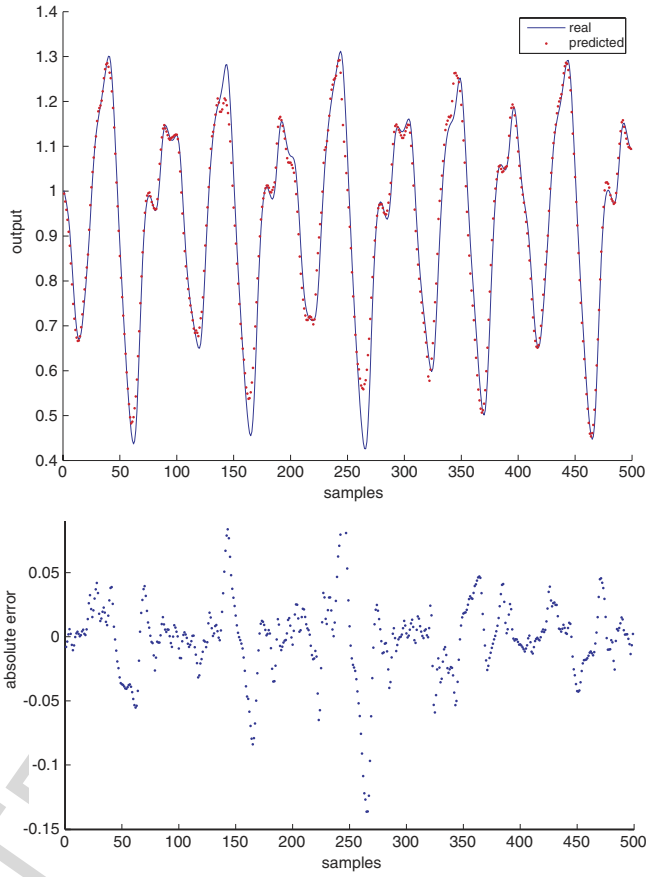

Fig. 1. 85-step-ahead prediction of the M–G time-series. The dots represent the model output with 100 clusters, the solid line is the real data. In the lower graph the absolute error between model and process output is represented.

$$\mu_i(x_j(k)) = \mathrm{e}^{-\frac{(x_j(k)-v_{ij}(k))(x_j(k)-v_{ij}(k))^T}{2\sigma^2_{i,j}(k)}}, \quad i = 1, \ldots, c, \; j = 1, \ldots, m-1. \quad (23)$$

The $m-1$ input variables define the hyperspace, and the subspaces in this hyperspace are defined as the Cartesian product of the subspaces. These imply the definition of the membership degree in each subspace as the product of the membership degrees as follows:

$$\beta_i(k) = \prod_{j=1}^{m-1} \mu_i(x_j(k)). \quad (24)$$

The fuzzy recursive least squares algorithm [38,10] is then obtained as follows:

$$\psi_i^T(k+1) = \beta_i(k)[1, x_1(k), x_2(k), \ldots, x_{m-1}(k)]$$
$$y_i(k) = \beta_i(k) x_m(k)$$
$$P_i(k+1) = \frac{1}{\lambda_r}\left(P(k) - \frac{P_i(k)\psi_i(k+1)\psi_i^T(k+1)P_i(k)}{\lambda_r + \psi_i^T(k+1)P_i(k)\psi_i(k+1)} \right)$$
$$\theta_i(k+1) = \theta_i(k) + P_i(k)\psi_i(k+1)(y_i(k) - \psi_i^T(k+1)\theta_i(k)) \quad (25)$$

where λ_r stands for the exponential forgetting factor, which should be set between 0.95 and 1 [39] to deal with time-varying processes, P_i stands for the covariance matrix, which is set to $P_i(0) = 10^5 I$, $I \in \mathbb{R}^{m\times m}$, and θ_i represents the parameters of the i-th local model and is written as follows:

$$\theta_i^T = [\theta_{i,0}, \theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,m-1}]. \quad (26)$$

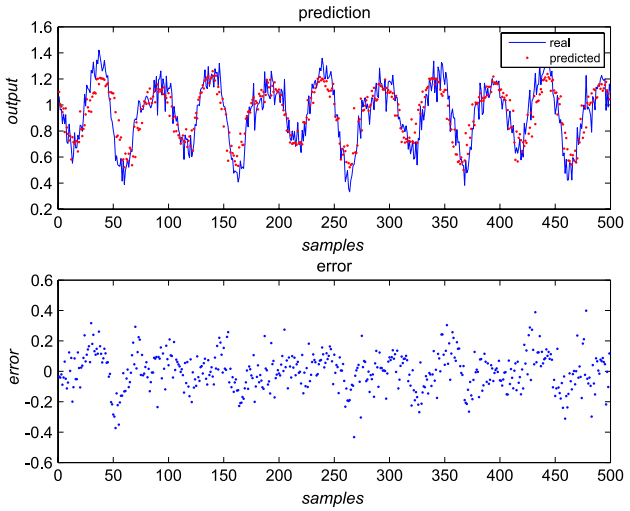

Fig. 2. 85-step-ahead prediction of the noisy M–G time-series. The upper graph represents the real output and the model output. The dots represent the model output with 100 clusters, the solid line is the real data. The lower graph represents the error between the real and predicted output.

Each local model contributes to the output of the model with the corresponding membership value. The whole set of fuzzy model parameters can be written in the matrix as follows:

$$\Theta = [\theta_1, \theta_2, \ldots, \theta_c]. \quad (27)$$

3.4. The recursive algorithm

The recursive algorithm can now be described in the following steps:

1. Step 0: Initialization:
   - define the number of clusters (c), the fuzziness (η, η_m) and the forgetting factors (λ_r, γ_v, γ_c),
   - determine the initial value of P_i for $i = 1, \ldots, c$,
   - determine the initial F_i for $i = 1, \ldots, c$
   - determine the initial centers: $v_i = x(k)$, for $i = k = 1, \ldots, c$ and membership degrees $\mu_i(x(k)) = 1$ for $i = k$, $\mu_i(x(k)) = 0$ for $i \neq k$, $i, k = 1, \ldots, c$,
   - calculate the initial s_i for $i = 1, \ldots, c$ from Eq. (16),
2. Step 1: calculate the current membership degree from Eqs. (18) and (19),
3. Step 2: calculate the $s_i(r+1)$ from Eq. (15) and Δv_i from Eq. (17),
4. Step 3: calculate the new centers from Eq. (10),
5. Step 4: calculate the new fuzzy variance matrix $F_i(r+1)$ from equation Eq. (22),
6. Step 5: calculate the membership functions μ_i from Eq. (23) and the membership degrees β_i from Eq. (24),
7. Step 6: apply the recursive least squares under equations Eq. (25) and return to Step 1.

Step 0 is executed only once in the procedure. The first three tasks under Step 0 are done by the user (definition of forgetting factors (rule of thumb Eq. (33)), number of clusters, initial covariance matrix ($P_i(0) \approx 10^5 I$), initial fuzzy covariance matrix ($F_i(0) \approx I$) and setting the fuzziness ($\eta_m \approx 0.25$ to 1) and overlapping factors

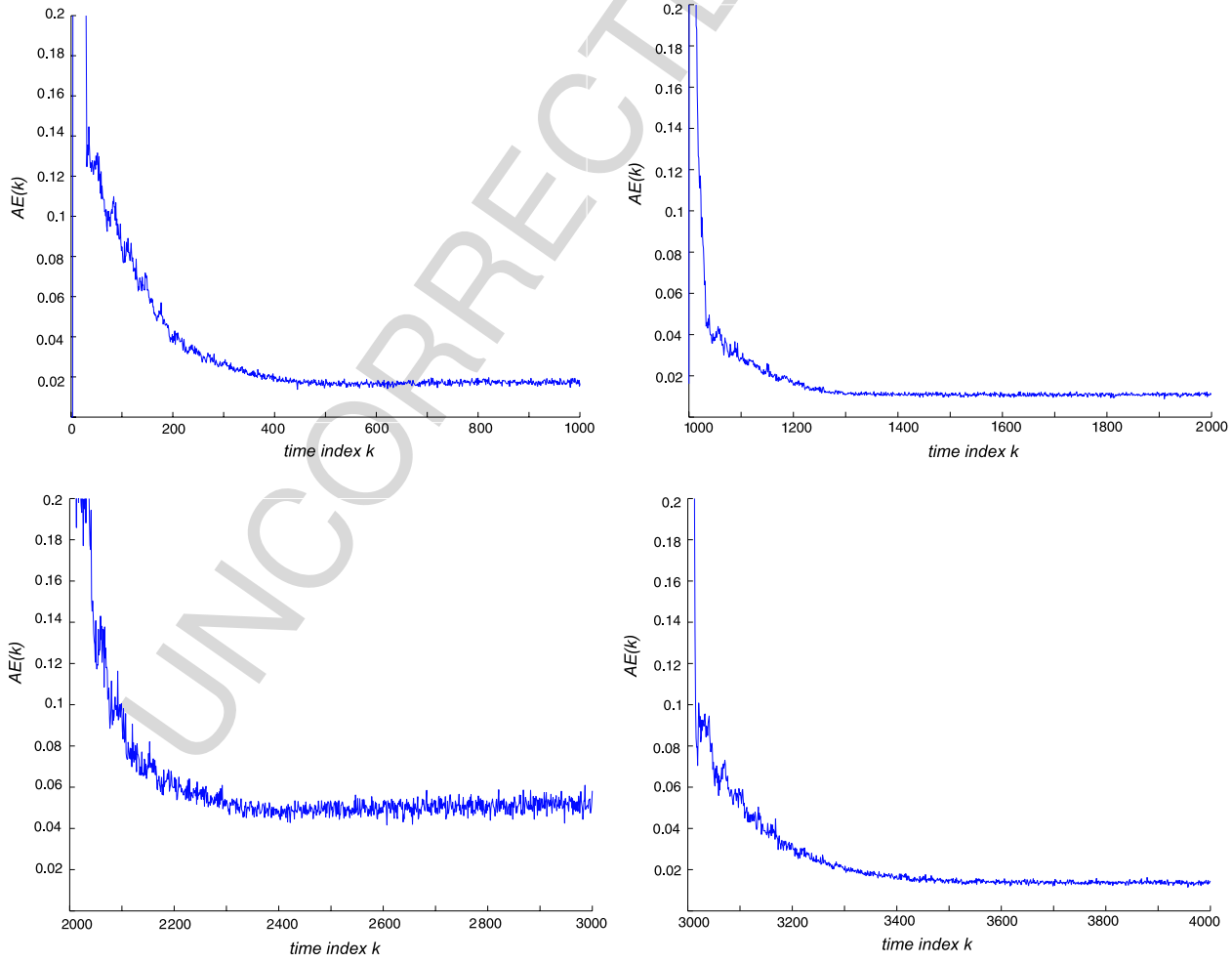

Fig. 3. Plot of instantaneous AE for time-varying process.

Please cite this article in press as: Dovžan D, Škrjanc I. Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes. ISA Transactions (2011), doi:10.1016/j.isatra.2011.01.004

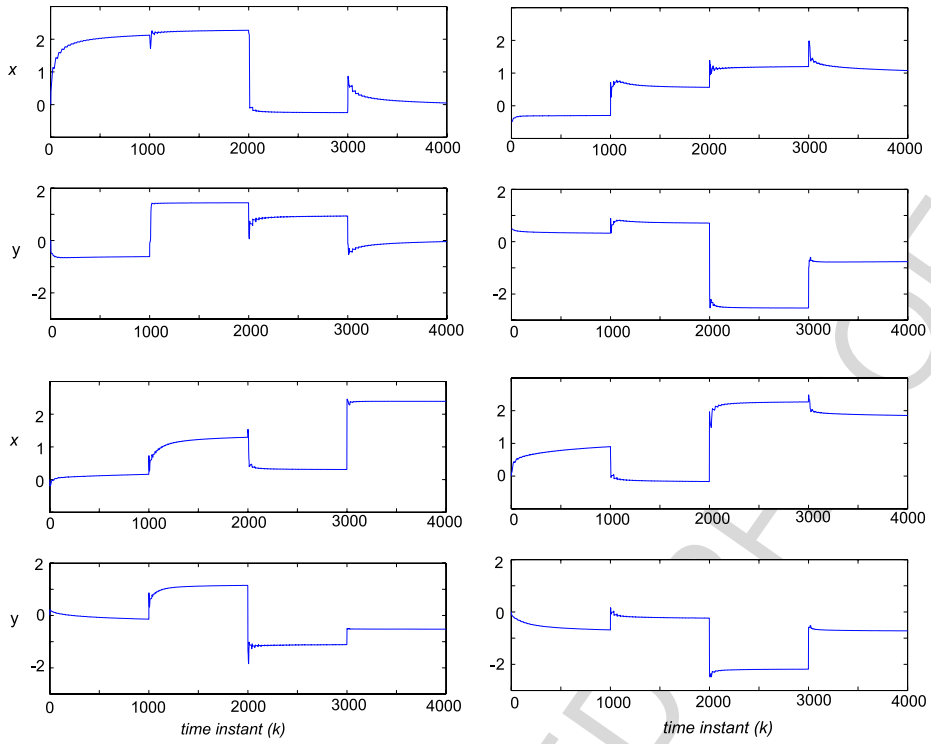

Fig. 4. Plot of cluster path. In each box the path of one cluster is represented. The upper graph in the box represents the path in the input space and the lower graph in the output space.

($\mu \approx 2$)). Everything else is initialized from the algorithm itself. The steps from Step 1 to Step 6 are executed for the time instances $k \geq c + 1$ for every new sample.

The forgetting factors affect the speed of the adaptation and the smoothness of estimates. For time varying processes, the factors must be lower than one to ensure the forgetting. By lowering the forgetting factors the fluctuations of estimates will increase and the method will be more sensitive to outliers.

The fuzziness factors define the smoothness of the nonlinearity approximation. The higher they are the more smooth nonlinearity we get, but to high values cause a bigger identification error. The values that give a good result are: $\eta_m =$ from 0.25 to 1 and $\eta = 2$.

The number of clusters must be set in advance. The number depends on the data and the desired approximation quality. This can be in some cases the disadvantage of the method. If you are doing classification and clustering of the data with unknown input and output range it is better to use methods with evolving structure like eTS. If you are identifying the model for prediction or model based control of the process the input and output data range is usually known and therefore we can use the fixed number of clusters.

4. Simulation examples

The proposed method was tested on a standard benchmark problem of the Mackey–Glass time-series [36] prediction and on the example from Kumar's paper [40].

Mackey–Glass time series

The chaotic time series is generated from the M–G differential delay equation defined by the following equation:

$$x(t) = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t). \quad (28)$$

The aim is to use past values of x to predict some future value of x. We assume $x(0) = 1.2$, $\tau = 17$ and the value of the signal is predicted 85 steps ahead, based on the values of the signal at the current moment, 6, 12 and 18 steps back.

Output: $[x(t + 85)]$ (29)

Input: $[x(t - 18)x(t - 12)x(t - 6)x(t)]$. (30)

For the validation of the model we used the non-dimensional error index (NDEI), defined as the ratio of the root mean square error to the standard deviation of the target data. In this simulation, 3000 data points $t \in [201, 3200]$ were created for the training, and $500t \in [5001, 5500]$ were created for the testing, the same as in [10]. The fuzzy model was built on the first 3000 data samples and then this model was used to predict the output for the 500 testing samples. The data was not normalized for the rFCM method. The results from other methods are also taken from [10] and are shown in Table 1. Fig. 1 shows the comparison between the real data and the fuzzy prediction model with 100 clusters.

We can see that our method gives a comparable degree of accuracy to other methods that use a reasonable number of rules. From Table 1 we can see that the proposed method gives better

Table 1
Comparison of recursive c-means method and other fuzzy identification methods on a 85 steps ahead prediction of the M–G time-series.

Methods	Rules (nodes, clusters)	NDEI
DENFIS	58	0.276
eTS	113	0.0954
RAN	113	0.375
ESOM	114	0.32
EFuNN	193	0.401
DENFIS	883	0.033
ESOM	1000	0.044
Neural gas	1000	0.062
EFuNN	1125	0.094
rec. FCM	3	0.4849
rec. FCM	10	0.4562
rec. FCM	58	0.3085
rec. FCM	100	0.1250

Please cite this article in press as: Dovžan D, Škrjanc I. Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes. ISA Transactions (2011), doi:10.1016/j.isatra.2011.01.004

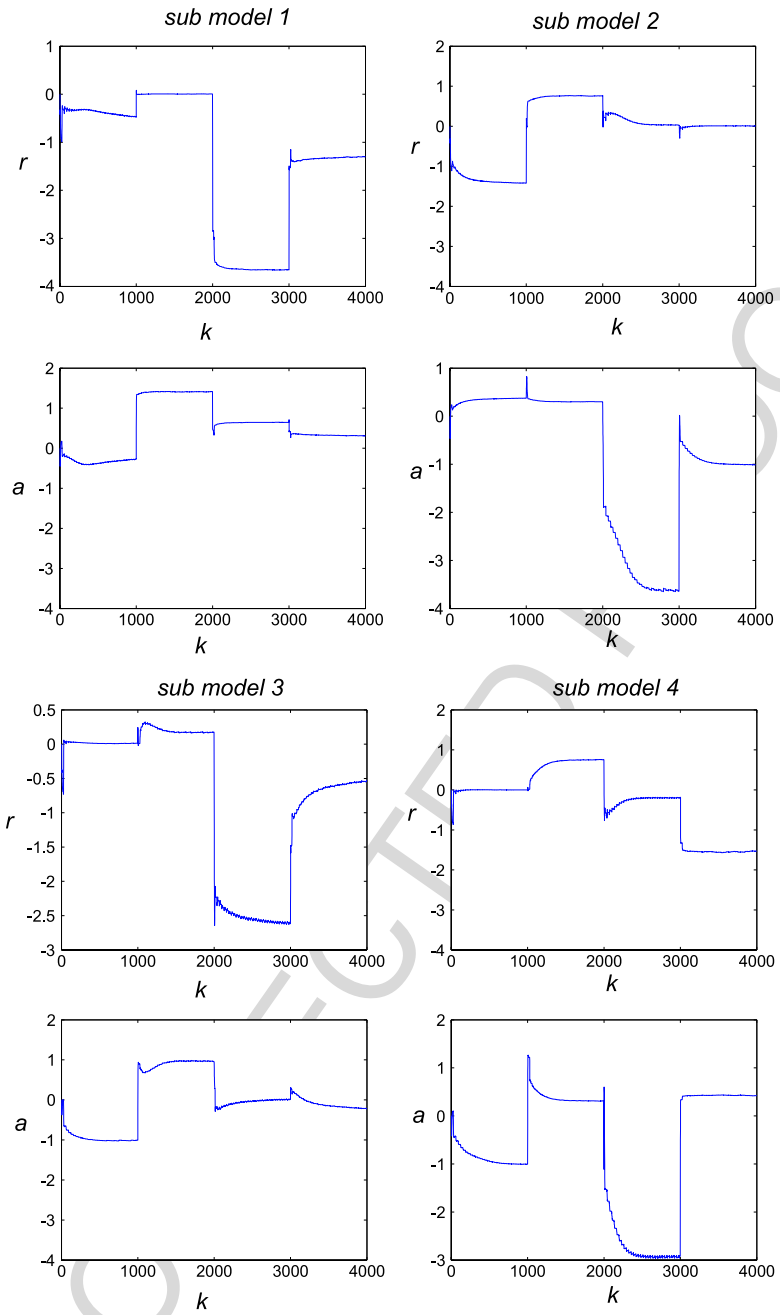

Fig. 5. Plot of the sub model parameters.

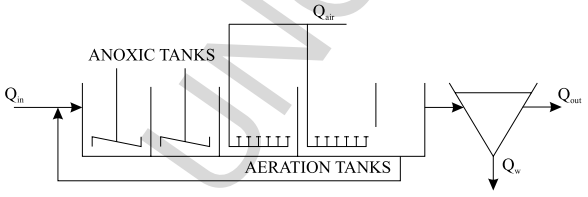

Fig. 6. Schematic representation of simulation benchmark.

results than methods like RAN, ESOM, and EFuNN (for around 100 clusters or rules).

In order to test the robustness of the recursive FCM we added Gaussian noise with a variance of 0.01 to the process. The obtained model gave a NDEI of 0.4809 for 100 clusters. The output of the model is shown in Fig. 2.

Identification of time-varying nonlinearity

This example was taken from Kumar's paper [40]. The process is described with the following equation:

$$y = f(x, p) = \frac{-10x}{2p + x^2} + p^2 \tanh(x), \tag{31}$$

where p is the time-varying parameter and $x(t) = -0.5 + |3 \sin 10t|$ is the input. The process is simulated from $t = 0$ to $t = 40$, with sample time $T_s = 0.01$. Parameter p varies with time as:

$$p(t) = \begin{cases} 1.5, & 0 \le t < 10 \\ 2, & 10 \le t < 20 \\ 1, & 20 \le t < 30 \\ 1.5, & 30 \le t < 40. \end{cases} \tag{32}$$

The test was done with added zero mean Gaussian noise with variance 0.01. To test the performance of the algorithm the

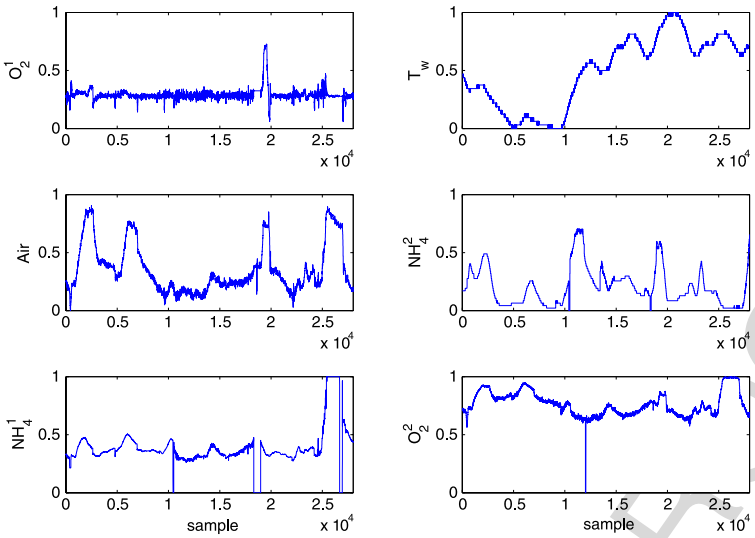

Fig. 7. Normalized training set.

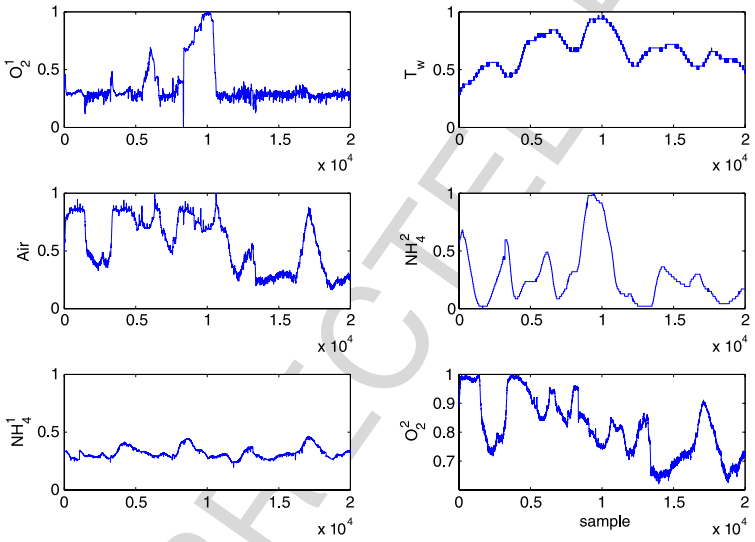

Fig. 8. Normalized validation set.

instantaneous absolute estimation error $AE(k)$ was calculated at each time instant k as $AE(k) = 1/300 \sum_{j=1}^{300} |f(x^j, p(T_s k)) - y_{fm}|$, where x^j are uniformly distributed points from interval $[-0.5, 2.5]$ and y_{fm} is the fuzzy model output. The estimation performance over considered time span ($t = 0$ to $t = 40$) was assessed by defining the energy of estimation-error signal $AE(k)$ as $\sum_{k=0}^{4000} |AE(k)^2|$. For a detailed explanation please see [40].

The values of the algorithm parameters were chosen as: $\eta = 2$, $\eta_m = 0.25$, the forgetting factor $\lambda_r = 0.998$, $\gamma_v = 0.998$ and $\gamma_c = 0.998$, number of clusters $c = 4$. To get better results the resetting [39] of least-square covariance matrix, s_k, and the fuzzy covariance matrix was introduced. The resetting is done when the output of the process and model output differ by more than 1. The forgetting factors were chosen based on the rule of thumb given in [39]:

$$\lambda = 1 - 2/N, \tag{33}$$

where λ is the forgetting factor and N are the data samples, that affect the estimates.

The overall performance $\left(\sum_{k=0}^{4000} |AE(k)^2|\right)$ of the proposed recursive algorithm was a bit better then the ones reported in [40]. The proposed method gave the overall performance of 20.45, while the best value in [40] was 225.49. The absolute estimation error is shown on Fig. 3. The paths of the cluster centers are shown on Fig. 4. Fig. 5 shows the parameters of the sub models, where the output of the i-th sub model is calculated as $y = ax + r$.

The same test was made when the process parameter p was uncertain. The random chosen number from uniform distribution interval $[-0.02\ 0.02]$ was added to the p at each time instant k. The results were almost the same as with no parameter uncertainty. The overall performance of 25.48 was a bit higher.

Both the tests were also made for fixed membership functions. The first 1000 samples were used to identify the model using the presented on-line identification method. Then the parameters of the membership degrees were fixed and the tuning of the sub-models was done only using the recursive fuzzy least squares. For the first test we got the overall performance of 25.63 and for the second test (uncertain parameter p) the overall performance was 28.59.

For comparison also eTS and SAFIS were tested on this example. While the proposed method gave about the same AE value each simulation run, the results of the SAFIS and eTS varied substantially. The best overall AE given by these two methods using the

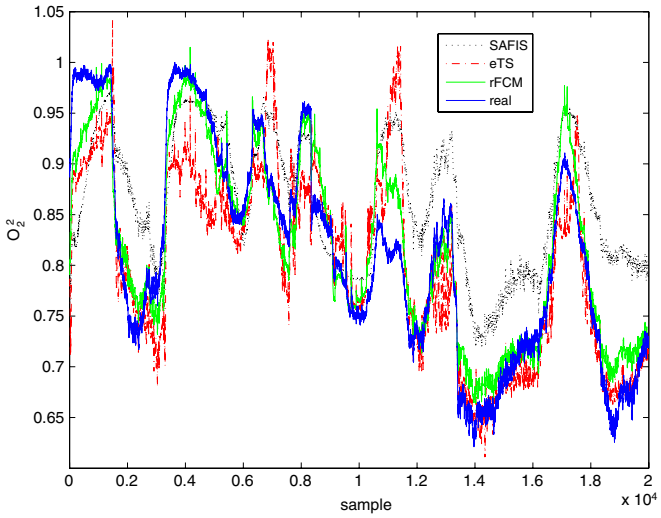

Fig. 9. Normalized output (validation).

proposed settings was 183.43 for SAFIS (11 clusters) and 1123.2 for eTS (5 clusters). The results of these two methods can probably be improved with the optimization of the parameters and probably some covariance resetting.

Identification of model for predicting the oxygen in the waste-water treatment process

In this example the SAFIS, eTS and the proposed method are used to identify a static prediction model of O_2 in a waste-water treatment process.

Waste-water treatment plants are large nonlinear systems subject to large perturbations in flow and load, together with uncertainties concerning the composition of the incoming waste-water [41]. The process consists of five sequentially connected reactors along with a 10-layer secondary settling tank. The plant layout, model equations and control strategy are described in detail on the web page (http://www.ensic.u-nancy.fr/costwwtp). Schematic representation is shown in Fig. 6. The data from which the models were identified was collected from a waste-water treatment plant. The set of measurements are: ammonia in the last aerobic tank (NH_4^2), oxygen in the first aerobic reactor (O_2^1), ammonia in the first aerobic tank (NH_4^1), waste-water temperature (T_w), total airflow rate (*Air*) and oxygen in the last aerobic reactor (O_2^2). The last measurement is the output of our model, the rest of them are the inputs. Measurements were divided into two sets. The training set (Fig. 7) and the validation set (Fig. 8). The measurements in the training set have a lot of outliers. For the identification of the model the measurements were normalized as it was noticed that eTS performs better with normalized values. First the training set was used to identify the model on-line with the methods. The trained model was then used to estimate the output of the process on the validation set. Fig. 9 represents the results of the validation. The eTS method evolves to 85 clusters and the SAFIS evolves to 26 clusters. The parameters for the eTS were set as proposed in [19]. For SAFIS the parameters were tuned as: $\kappa = 2$, $\varepsilon_{\max} = 0.001$, $\varepsilon_{\min} = 1e-4$, $e_g = 5e-6$, $e_p = 5e-10$ and $\gamma = 1$. At this point it is worth mentioning that the authors find the parameter tuning of the SAFIS very time consuming and difficult. The settings for the rFCM (the proposed method) were chosen as: $c = 15$, $\eta = 2$ and $\eta_m = 0.25$. The forgetting factors were set to one.

The maximal error for the SAFIS was 0.1735, for the eTS 0.2111 and for rFCM 0.1007. The sum squared error for the SAFIS was $6.7e-3$, for the eTS $3.18e-3$ and for the rFCM $1.03e-3$. The histograms of the errors are shown in Fig. 10. The autocorrelations of the errors and errors are shown in Fig. 11.

Although the eTS gave better results with M–G prediction, the result of this experiment shows that the proposed method can better cope with outliers. This is very important for on-line identification, where preprocessing of the data is limited. The model obtained with the rFCM is a bit better than the model obtained by SAFIS and eTS.

5. Conclusion

In this paper the recursive version of the fuzzy *c*-means clustering algorithm was derived. The algorithm can be used for on-line clustering and identification of the Takagi–Sugeno fuzzy model. The advantage of T–S fuzzy model approach over the neuro-fuzzy approach (for example SAFIS), is that at each time step you can extract the linear model that is valid at a certain time instant. This model can be then used for example for setting the linear controller parameters.

The developed method is simple to program, easy to tune and computationally effective as it does not require re-training of the

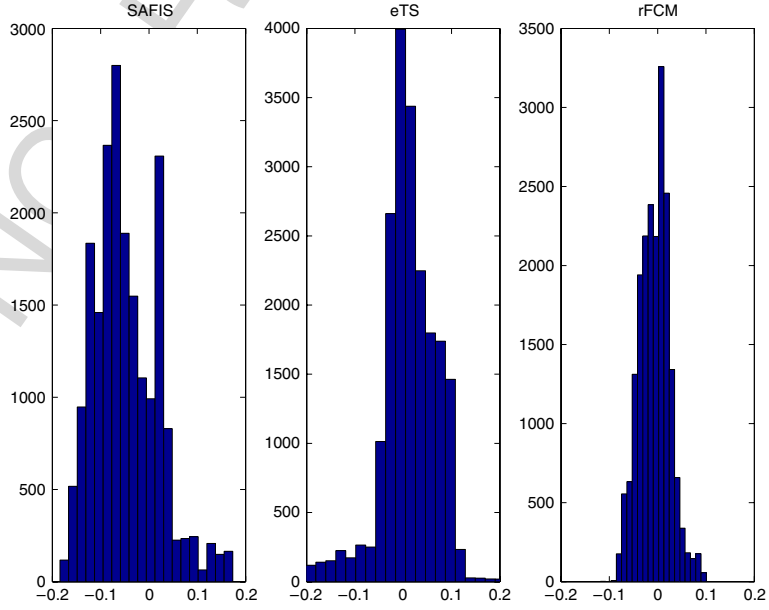

Fig. 10. The error histogram.

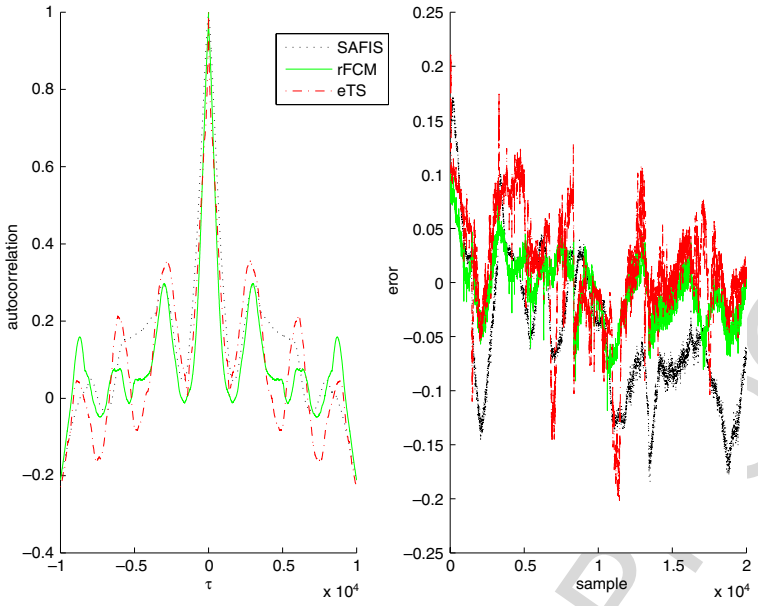

Fig. 11. The autocorrelation and error graph.

whole model. In addition, the model is updated in only a fraction of a second. For the M–G time-series example the model with 100 clusters was updated in 0.015 s using a Pentium 4 processor with a 2.4 GHz CPU and 1 GB of RAM, where the algorithm was written in Matlab code. The degree of accuracy can be easily compared with other methods.

The method requires more memory than eTS or SAFIS because the fuzzy covariance matrix must be stored. On the other hand the memory demands are constant because of the fixed number of clusters.

By developing the recursive fuzzy *c*-means clustering and combining it with recursive least squares a fast and robust algorithm for identifying the fuzzy model from numerical data was obtained. The algorithm is robust with respect to the initial parameters and with respect to noisy data. The latter is achieved by averaging, which takes place in both cluster estimation and in the least-squares estimation procedures.

Acknowledgement

We would like to thank Dr. Angelov for providing us with the source code for the eTS method.

References

[1] Johanson TA, Murray-Smith R. Operating regime approach to nonlinear modeling and control. In: R. Murray-Smith, T.A. Johanson, Multiple model approaches to modeling and control. Hants (UK). Taylor Francis. p. 3–72.
[2] Hwang CL, Chang LJ. Fuzzy neural-based control for nonlinear time-varying delay systems. IEEE Trans Syst Man Cybern B 2007;37(6):1471–85.
[3] Takagi T, Sugeno M. Fuzzy identification of systems and its applications. IEEE Trans Syst Man Cybern 1985;SMC-15:116–32.
[4] Chiu SL. Fuzzy model identification based on cluster estimation. J Intell Fuzzy Syst 1994;2:267–78.
[5] Kukolj D, Levi E. Identification of complex systems based on neural and Takagi–Sugeno fuzzy model. IEEE Trans Syst Man Cybern B 2004;34(1):272–82.
[6] Kim E, Park Minkee, Ji S, Park Migon. A new approach to fuzzy modeling. IEEE Trans Fuzzy Syst 1997;5(3):328–37.
[7] Shing J, Jang R. ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans Syst Man Cybern 1993;23(3):665–85.
[8] Kasabov N. Learning fuzzy rules and approximate reasoning in fuzzy neural networks and hybrid systems. Fuzzy Sets and Systems 1996;82(2):665–85.
[9] Angelov PP, Filev DP. An approach to online identification of Takagi–Sugeno fuzzy models. IEEE Trans Syst Man Cybern B 2004;34(1):484–97.
[10] Kasabov NK, Song Q. DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. IEEE Trans Fuzzy Syst 2002;10(2):144–54.
[11] Leng G, Prasad G, McGinnty TM. An on-line algorithm for creating self-organizing fuzzy neural networks. Neural Netw 2004;17(10):1477–93.
[12] Chiang CK, Chung HY, Lin JJ. A self-learning fuzzy logic controller using genetic algorithms with reinforcements. IEEE Trans Fuzzy Syst 1997;5(3):460–7.
[13] Hoffmann F, Pfister G. Learning of a fuzzy control rule base using messy genetic algorithms. Genetic algorithms and soft computing, vol. 8. 1996. p. 279–305.
[14] Berenji HR. A reinforcement learning-based architecture for fuzzy logic control. In: Special issue on fuzzy logic and neural networks for pattern recognition and control. vol. 6. no. 2. February 1992. p. 267–292.
[15] Kasabov N. Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. IEEE Trans Syst Man Cybern B 2001;31(6):902–18.
[16] Kasabov N. ECOS: a framework for evolving connectionist systems and the ECO learning paradigm. In: Proc. ICONIP 1998. 1998. p. 1222–35.
[17] Kasabov N. Evolving fuzzy neural networks-algorithms, applications and biological motivation. In: Methodologies for the conceptation, design and application of soft computing. Singapore. 1998. p. 271–4.
[18] Angelov PP. Evolving rule-based models: a tool for design of flexible adaptive systems. Heidelberg: Springer-Verlag; 2002.
[19] Angelov P. An approach for fuzzy rule-base adaptation using on-line clustering. Integr Methods Hybrid Syst 2004;35(3):275–89.
[20] Yager RR, Filev DP. Learning of fuzzy rules by mountain clustering. In: Proc. SPIE conf. applicat. Fuzzy logic technol. 1993. p. 246–54.
[21] Qiao J, Wang H. A self-organizing fuzzy neural network and its applications to function approximation and forecast modeling. Neurocomputing 2008;71(4–6):564–9.
[22] Xu L, Krzyzak A, Oja E. Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. IEEE Trans Neural Netw 1993;4:636–49.
[23] Rong Hai-Jun, Sundararajan N, Huang Guang-Bin, Saratchandran P. Sequential adaptive fuzzy inference system (SAFIS) for nonlinear system identification and prediction. Fuzzy Sets and Systems 2006;157(9):1260–75.
[24] Wang D, Zeng Xiao-Jun, Keane JA. An incremental construction learning algorithm for identification of T–S fuzzy systems. In: Proc. FUZZ 2008. IEEE international conference on fuzzy systems 2008. 2008. p. 1660–6.
[25] Lin FJ, Lin CH, Shen PoH. Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive. IEEE Trans Fuzzy Syst 2001;9(5):751–9.
[26] Wu S, Er MJ. Dynamic fuzzy neural networks—a novel approach to function approximation. IEEE Trans Syst Man Cybern B 2000;30(2):358–64.
[27] Wu S, Er MJ, Gao Y. A fast approach for automatic generation of fuzzy rules by generalized dynamicc fuzzy neural networks. IEEE Trans Fuzzy Syst 2001;9(4):578–94.
[28] Juang CF, Lin CT. An on-line self-constructing neural fuzzy inference network and its applications. IEEE Trans Fuzzy Syst 1998;6(1):12–32.
[29] Lin CT. A neural fuzzy control system with structure and parameter learning. Fuzzy Sets and Systems 1995;70:183–212.
[30] Tzafestas SG, Zikidis KC. NeuroFAST: on-line neuro-fuzzy ART-based structure and parameter learning TSK model. IEEE Trans Syst Man Cybern B 2001;31(5):797–802.
[31] Azeem MF, Hanmandlu M, Ahmad N. Structure identification of generalized adaptive neuro-fuzzy inference systems. IEEE Trans Fuzzy Syst 2003;11(5):666–81.

[32] Azeem MF, Manmandlu M, Ahmad N. Modified mountain clustering in dynamic fuzzy modeling. In: 2nd int. conf. inform. technol. 1999. p. 63–8.
[33] Patt J. A resource allocating network for function interpolation. Neural Comput 1991;3(2):213–25.
[34] Deng D, Kasabov N. Evolving self-organizing maps for online learning, data analysis and modeling. In: Proc. IJCNN 2000 neural networks neural computing: new challenges perspectives new millennium. vol. 6. 2000. p. 3–8.
[35] Fritzke B. A growing neural gas network learns topologies. Adv Neural Inf Process Syst 1995;7.
[36] Mackey MC, Glass L. Oscillations and chaos in physiological control systems. Science 1977;197:287–9.
[37] Bezdek JC. Pattern recognition with fuzzy objective function algorithms. New York: Plenum Press; 1981.
[38] Goodwin GC, Sin KS. Adaptive filtering prediction and control. Upper Saddle River (New York): Prentice-Hall; 1984.
[39] Åström KJ, Wittenmark B. Adaptive control. New York: Addison-Wesley; 1995.
[40] Kumar M, Stoll R, Stoll N. Deterministic approach to robust adaptive learning of fuzzy models. IEEE Trans Syst Man Cybern B. 36 (4) 767–780.
[41] Škrjanc I, Teslic L. Monitoring of waste-water treatment plant using Takagi–Sugeno fuzzy model. In: Electrotechnical conference. 2008. MELECON 2008. The 14th IEEE Mediterranean. 2008. p. 67–70. doi:10.1109/MELCON.2008.4618412.

Please cite this article in press as: Dovžan D, Škrjanc I. Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes. ISA Transactions (2011), doi:10.1016/j.isatra.2011.01.004