Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematical Modelling 32 (2008) 1159–1178
APPLIED MATHEMATICAL MODELLING
www.elsevier.com/locate/apm

Rule generation for hierarchical collaborative fuzzy system

Paulo Salgado *

CITAB – Centro de Investigação e Tecnologias Agro-Ambientais e Biológicas, Universidade de Trás-os-Montes e Alto Douro, Dep. Engenharias, 5001-801 Vila Real, Portugal

Received 1 October 2005; received in revised form 1 February 2007; accepted 16 March 2007
Available online 6 April 2007

Abstract

A new method of rule generation for the hierarchical collaborative fuzzy system, HCFS, is proposed. This HCFS is structured like various parallel fuzzy subsystems and it overcomes the dimensionality problem and the lack of interpretability of most of the traditional fuzzy systems, when dealing with complex real-world problems. An association process of different fuzzy systems is presented in this work, through the use of a relevance concept of a fuzzy system. The result of this aggregation is a collaborative structure where all sub-models have the ability to gradually improve the overall accuracy of approximation by adding their own contributions. For this structure we propose a new algorithm to be used in the procedures of the three learning phases: the structure building, the parametric identification and the division of the learning data among the various levels of the hierarchical structure. This new fuzzy modelling technique automatically generates and tunes the sets of fuzzy rules in the hierarchical collaborative structure (HCS). The effectiveness of the proposed HCFS model in handling high-dimensional and complex problems is demonstrated through various numerical simulations.

Keywords: Hierarchical fuzzy systems; Fuzzy modelling; Fuzzy clustering

1. Introduction

Fuzzy modelling is one of the currently used techniques for modelling non-linear, uncertain and complex systems. An important characteristic of fuzzy models, FM, is the partitioning of the space of system variables into fuzzy regions using fuzzy sets [1]. In each region, the characteristics of the system can be simply described using a rule. Typically, a fuzzy model consists of a rule base with rules associated to particular regions, where the information available is transparent and easily readable. This characteristic of fuzzy systems has been largely exploited in pattern recognition [2], control [3], image processing [4] and other fields.

However, as the complexity of the described systems increases, the number of rules of a standard fuzzy system also increases exponentially with the number of variables involved [5–9] or with the complexity of the relationships in the samples patterns [10]. The aforementioned exponential growth of the total number of fuzzy

* Tel.: +351 259 350 359; fax: +351 259 350 480.
E-mail address: psal@utad.pt

0307-904X/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.apm.2007.03.007

rules implies that a fuzzy system is always a trade-off between accuracy and dimensionality. The interpretability of the resulting rules by the human operator, one of most important features on the fuzzy system description, is also reduced.

An alternative to solve the curse of dimensionality is the hierarchical fuzzy modelling [11–15]. By arranging the inputs in hierarchical ways and allowing the consequent part of a rule to be an antecedent to another rule, the number of fuzzy rules would be a linear or nearly-linear function of the number of inputs. In this case, the input space is decomposed into subspaces of lower dimension, and each input variable is considered only at a certain level of the hierarchy. Another approach defines the hierarchy as a prioritisation of rules [16]. With this kind of hierarchy, the higher levels contain more specific rules and the lower levels contain the generic rules, which are applied only when no applicable specific rule is available. However, given a set of high-dimensional training data, for these hierarchical modelling it is not easy to determine an appropriate, if not optimal, network structure. Very few methods have yet been proposed to automatically generate rules for such hierarchical systems, most of them by the use of genetic algorithms (GA) [17,18]. As it is well known, GA is a very time-consuming searching process and, sometimes, it is computationally prohibitive when a very high-dimensional problem is being considered.

The division of the input space is one of the most important aspects of fuzzy modelling. This division is a coarse setting of model complexity, thus determining the performance of the model fine-tuning. For an appropriate division of input space, several methods have been reported [19–22]. Some of these methods have registered the division by merging similar membership functions or inserting new ones. Other authors [23–25] have proposed clustering algorithms to obtain fuzzy rules from the given input–output data.

These strategies have been applied on the conventional flat fuzzy system, where all rules were on the same level (rule base) and used by a common inference engine, which was identified from single data training. This structure of fuzzy system and these strategies are therefore inadequate in many real applications, which have different relationships with distinct complexities to be modelled. This problem suggests the use of a hierarchical structure of fuzzy rule bases to make a convenient organisation of fuzzy information in the fuzzy subsystems of the hierarchical structure.

The aim of this paper is to propose an efficient methodology that automatically generates fuzzy rules from real data, organized in a Hierarchical Collaborative Structure (HCS) of fuzzy systems, and to make the optimisation of its fuzzy sets membership parameters by combining various algorithms. The HCFS model $f(x)$ is a hierarchical structure containing a set of n fuzzy subsystems, where each fuzzy subsystem $f_1(x), f_2(x), \ldots, f_n(x)$ contributes with its particular description to the response of the overall system. Each subsystem may contain information related with particular aspects of the input–output region or merely collaborate to the performance of $f(x)$, and whose contribution is measured by its relevance. This structure was shown very useful to split the inside greenhouse air temperature and humidity flat fuzzy models into fuzzy sub-models [26].

In this work the HCFS model is obtained directly from the data set. In general, this approach involves two major steps: structure identification and parameter identification. Moreover, a third step is developed to make the partitioning of the input–output space through various hierarchical sub-models, each part being used in the training process of the corresponding level. We propose a new learning strategy for these tasks, combining the unsupervised and supervised learning algorithms to create and adapt the fuzzy rules, by the ability to make automatically a partition of the data set that will be used by each subsystem in the learning process. In the corresponding procedure, the model parameters, initially found by forward RLS algorithm, are recurrently adapted by the backward RLS algorithm. The final set of rules (of each sublevel) is appropriately determined by removing the useless fuzzy rules.

The paper is organized as follows. In Section 2, the flat fuzzy system is described and HCFS structure is presented. The concepts of relevance of a set of rules and the relevance of the fuzzy system are defined. Section 3 describes how a HCFS is evaluated in the modelling process. Also, a new hierarchical collaborative fuzzy system algorithm, HCFSA, is proposed, which automatically generates rules to the hierarchical fuzzy system. For this we use a set of methods: Cluster Rule Generation; Recursive Least-Square; Data Rejection Algorithm and Backward Fuzzy Rule Selection algorithm. Next, the HCFSA is tested on a benchmark of non-linear model identification and on a computer image processing problem. Finally, Section 5 concludes and gives some suggestions for future work.

2. The relevance of the fuzzy system

2.1. The fuzzy system

A generic fuzzy model is defined as a collection of fuzzy rules R_l of the form of (1):

$$R_l : \text{If } x_1 \text{ is } A_{l1} \text{ and } \cdots \text{ and } x_n \text{ is } A_{ln} \text{ then } y = z_l(\vec{x}), \tag{1}$$

where $\vec{x} = (x_1, \ldots, x_n)^{\mathrm{T}} \in U$ and $y \in V$ are linguistic variables, A_{li} are fuzzy sets of the universes of discourse $U_i \in \boldsymbol{R}$ with membership function $\mu_{li}(x_i)$, and $z_i(\vec{x})$ is a function of the input variables. Typically, z can take one of the following three forms: fuzzy set (Mamdani type fuzzy systems), singleton (Takagi–Sugeno) or polynomial function (Takagi–Sugeno–Kang, TSK) type fuzzy systems. Takagi–Sugeno fuzzy systems with centre average defuzzification, product-inference rule and singleton fuzzification are in the following form:

$$z(\vec{x}_k; \theta) = \sum_{l=1}^{M} p_l(\vec{x}_k) \cdot \theta_l = \boldsymbol{p}^{\mathrm{T}}(\vec{x}_k) \cdot \theta, \tag{2}$$

where $\boldsymbol{p}(\vec{x}) = [p_1(\vec{x}), \ldots, p_l(\vec{x}), \ldots, p_M(\vec{x})]^{\mathrm{T}}$ are the vector of fuzzy basis functions (FBF) of fuzzy system. The FBF's are given by

$$p_l(\vec{x}) = \mu_l(\vec{x}) \Big/ \sum_{l=1}^{M} \mu_l(\vec{x}), \tag{3}$$

where M represent the number of rules, $\mu_l(\vec{x}) = \prod_{i=1}^{n} \mu_{li}(x_i)$ is the firing strength of rule l and $\theta = [\theta_1, \ldots, \theta_l, \ldots, \theta_M]^{\mathrm{T}}$ is the vector of θ's parameters, where the θ_l is the point at which the output fuzzy set l achieves its maximum value (for TS systems they are the consequent constants). The defuzzified output y of the fuzzy model is calculated as a *weighed average* [27,28] of the outputs of all fuzzy rules.

Our approach is to develop a fuzzy model for an unknown function $f : R^n \to R$ simply from the data set.

2.2. The relevance of the fuzzy system

Fuzzy logic systems, FLS, are based on a set of rules that map regions in an input space, U, to regions in an output space, V, describing a region in a product space $S = U \times V$. The fuzzy rules are fuzzy relations in the product space S. These relationships are realized by a set of rules $\Im$, which create a power set of fuzzy rules $\widetilde{P}(\Im)$. In this context, the contributions of the different rules in the behaviour of the fuzzy system will be unequal, i.e., with different relevance. Moreover, in a hierarchical fuzzy system, for different parts of the input space the fuzzy subsystems will have a variety of performances that we need to measure. The adoption of fuzzy logic theory and the focus on the degree of relevance as an extended fuzzy measure seems to be an appropriate approach to automating relevance perception of the rules and fuzzy systems. The relevance is a measure of the relative importance of the rules that describe the region S. The relevance is a special fuzzy measure that involves the relativity of a support region, which we see as a fuzzy measure only if the support of rules agrees with region S.

Definition 1. The relevance of the rule $R \in \widetilde{P}(\Im)$ on a region S can be characterized by a real, positive value. The normalized relevance function maps the power set of fuzzy rules $\widetilde{P}(\Im)$ on the real interval $[0,1]$, i.e.: $\Re_S(R) \in [0, 1]$.

A higher relevance on region S corresponds to more important fuzzy rules on describing the relationship in the S regions and vice-versa. In Definition 2 we propose a relevance function.

Definition 2. The relevance of rule l of the fuzzy system (2) on point $(\vec{x}_k, y_k) \in S$ is defined as

$$\Re_S^l(\vec{x}_k, y) = \begin{cases} p_l(\vec{x}_k); & \text{if } H(\vec{x}_k) \geqslant \gamma \\ p_l(\vec{x}_k)(H(\vec{x}_k)/\gamma)^{\delta}; & \text{otherwise,} \end{cases} \tag{4}$$

where $H(\vec{x}_k) = \sum_{l=1}^{M} \mu_l(\vec{x}_k)$, γ is a threshold value (less than or equal to one), and $\delta > 0$ is a real constant.

The δ parameter controls the fuzziness of the relevance of the fuzzy rule and of the fuzzy system on the input region. A small value of δ corresponds to an enlargement of influence of the fuzzy rule and that of the fuzzy system. On the other hand, for a large value of δ the relevance falls drastically, similar to a crisp set. This relevance function is associated with the firing of fuzzy rules through the fuzzy basis functions (FBF) $p_l(\vec{x}_k)$. The relevance function has two parts, which correspond to: (1) high firing values, i.e. $H(\vec{x}_k) \geqslant \gamma$, with its relevance approaching the unitary value, and (2) low relevance values ($H(\vec{x}_k) < \gamma$).

The relevance of fuzzy system (2) will be obtained by the aggregation of the relevance of the individual rules. The aggregation operation must obey a set of axioms [29]. For the fuzzy system whose relevance of fuzzy rule is measured according to Definition 2, its relevance on point $(\vec{x}_k, y_k) \in S$ is expressed in Definition 3.

Definition 3. The relevance of fuzzy system (2) in $(\vec{x}_k, y_k) \in S$ is defined as

$$\Re_S(\vec{x}_k, y) = \begin{cases} 1; & \text{if } H(\vec{x}_k) \geqslant \gamma \\ (H(\vec{x}_k)/\gamma)^{\delta}; & \text{otherwise} \end{cases} \tag{5}$$

This relevance function is defined for the well-known region S, where the fuzzy system establishes its input–output relationships. Therefore, the relevance of one rule or set of rules simultaneously characterizes the level of relevance of the rule in its region and the degree of region coverage by the fuzzy system. This new proposal enables the aggregation of different fuzzy systems, with a different set of rules and different or shared covered regions.

Fig. 1 represents a hierarchical collaborative fuzzy system, HCFS, with n sub-models. Each ith sub-model has two outputs: the first is the output response of this fuzzy subsystem and the other output is the relevance of the fuzzy subsystem, which represents its contribution to the final system description.

Therefore, the output of the aggregation of one model i with the previous sub-models is the weighed sum of the current contribution with the aggregated previous sub-models (6)

$$Y_i(\vec{x}) = y_i(\vec{x}) \cdot R_i(\vec{x}) + Y_{i-1}(\vec{x}) \cdot \Re_{i-1}(\vec{x}), \tag{6}$$

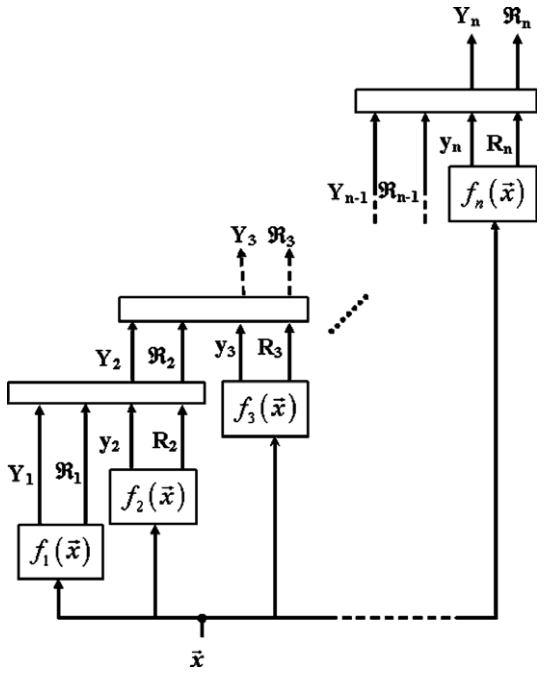

Fig. 1. The hierarchical collaborative fuzzy system, HCFS.

where $R_i(\vec{x})$ represents the relevance function of the ith fuzzy subsystem covering the point $\boldsymbol{x}_i$ of the universe of discourse. The relevance $\Re_i(\vec{x})$ reveals the effective contribution (or belief of contribution) of the corresponding fuzzy system up to the ith level and according to the axiomatic defined in [29]. For HCFS the relevance aggregation is here defined as:

$$\Re_i(\vec{x}) = T(R_i(\vec{x}); \Re_{i-1}(\vec{x})), \tag{7}$$

where T is a t-conorm operation, as for example the *max* operation. The t-conorm operation is commutative, which implies the structure to be a collaborative one.

3. Generating rules for the hierarchical collaborative fuzzy system

The HCFS fuzzy model is a hierarchical fuzzy model where each sub-model is a flat FLS (here assumed the type of Eq. (2)) with an extra output: *the relevance of fuzzy subsystem* (see Fig. 1). Each fuzzy sub-model identification method follows the three generic steps: structure identification, data selection and parameter estimation. Unsupervised learning can be used to obtain the initial fuzzy rule base from sample data while supervised learning is effective in the FLS parameters fine-tuning. In the proposed method the determination of the number of fuzzy rules and the positioning of fuzzy sets in the domain of input space is done by a Cluster Rule Generation algorithm, CRG. Model parameters are generally associated with rule conclusion, whose estimation is performed by the recurrent (forward or backward) least squares optimization algorithm, RLS [30] (supervised method). The CRG identification method consists of establishing a volume of cluster influence *V in each zone.* This influence is applied on regions of the space where the multivariable cluster membership functions have high values (typically above 0.5). Volume V determines the granularity of the adaptive fuzzy model. The shape and position of clusters are adjusted by considering the statistical property of points that belong to the cluster. This method will be adapted for learning process on HCFS structure.

In the hierarchical structure HCFS, each sub-model has selective and limited abilities of learning context, defined here by modelling clusters with a predefined volume in each level. In the modelling task, each sub-model competes with the others in adequately modelling the training data set.

The process starts with the first subsystem in the HCFS structure trying to model the initial training set. For that it uses the CRG method to create clusters (rules) and to adequately position their centers; subsequently it tries to optimize the consequent parameters through the RLS method, most often without much success. The next task will be to exclude from the training set those points that mostly hinder the model's performance. This removal of unfit data is accompanied by the adaptation of both the model's structure and parameters. The excluded points make the training data set for the next higher hierarchical fuzzy system, where the previously described learning methods are now applied. At the end of the entire process we have a fuzzy system organized as one hierarchical structure.

So, the distribution of training data by different levels of the hierarchical structure is the main task of the proposed algorithm, together with the selection of the minimum number of rules that are sufficient for its description. These processes involve the use of recursive techniques for removing data patterns from the training data set, based on the criterion of minimizing the square error. The Data Rejection Algorithm, DRA, implements this strategy. After this training process, we use a Backward Selection Algorithm to remove the useless fuzzy rules.

In the following sections these techniques are detailed.

3.1. Fuzzy rules extracted from partitioned numerical data

As mentioned earlier, a self-constructing rule generator obtains initial fuzzy rules during the structure identification phase. Initially, we propose a novel approach for the flat FM system, the Cluster Rule Generation algorithm, CRG. Our goal is to cluster the given data set into several groups such that similar points are grouped into the same cluster while dissimilar points are in different clusters. Later, we adapt this algorithm in order to be used in the HCFS structure.

A FM system for a given set of input–output data is obtained in an iterative process consisting of two steps. First, the data is partitioned into a set of fuzzy clusters based on the conjugation of two similarity tests (input

and output) and, if the result is positive, by weighing its membership degree to the cluster. When new training data is considered, the existing clusters with some degree of similarity are adjusted. In this case, membership functions associated with each cluster are defined according to statistical means and variances of data points included in the cluster. Furthermore, when the degree of similarity is not enough a new clustering can be created, without the necessity of restarting the identification process.

For a system with n inputs and one output, we define a fuzzy cluster j as a pair $(m_j(\vec{x}), \bar{y}_j)$, where $m_j(\vec{x})$ is in agreement with Definition 2, i.e.

$$m_j(\vec{x}) = \frac{\mu_j(\vec{x}_k)}{\sum_{i=1}^{M}\mu_i(\vec{x}_k)}\,\mathfrak{R}_S(\vec{x}_k, y), \tag{8}$$

with

$$\mu_j(\vec{x}) = \prod_{i=1}^{n}\mu_{ji}(x_i) = \prod_{i=1}^{n}\exp(-(x_i - \bar{x}_{ji})^2/\sigma_{ji}^2) \tag{9}$$

and $\mathfrak{R}_S$ as defined in (5), where $\vec{x} = (x_1, \ldots, x_n)^{\mathrm{T}}$, $\bar{x}_j = (\bar{x}_{j1}, \ldots, \bar{x}_{jn})^{\mathrm{T}}$, and $\vec{\sigma}_j = (\sigma_{j1}, \ldots, \sigma_{jn})^{\mathrm{T}}$ denote the input vector, the mean vector, and the standard deviation vector, respectively, and $\bar{y}_j$ denotes the height of cluster j. The Gaussian [31] multivariate membership function (9) defines the domain of influence of the cluster.

Let M be the number of existing fuzzy clusters and B_j be the *size* of cluster j. Initially M and B's have null value. This algorithm uses clusters with the same volume V.

For an input–output instance P_k, $(\vec{x}_k, y_k)$, where $\vec{x}_k = (x_{1,k}, \ldots, x_{n,k})^{\mathrm{T}}$, we perform the input–output similarity test on cluster j, by conjugation (10):

$$m_j(\vec{x}_k) > \rho \quad \text{and } |y_k - \bar{y}_j| < \tau \cdot d, \tag{10}$$

where ρ and τ, $0 \leqslant \rho, \tau \leqslant 1$, are predefined thresholds and d is the output range of variable y, i.e., $d = y_{\max} - y_{\min}$ being $y_{\max}$ and $y_{\min}$ the maximum output and the minimum output, respectively, of the given data set.

When the instance P_k has passed both the input- and output-similarities tests with respect to cluster j, this cluster should be modified to include instance P_k as its member, i.e.:

$$\bar{\boldsymbol{x}}_j^{(k+1)} = (B_j^{(k)}\bar{\boldsymbol{x}}_j^{(k)} + m_j(\vec{x}_k)\cdot\vec{x}_k)/(B_j^{(k)} + m_j(\vec{x}_k)), \tag{11}$$

$$\bar{y}_j^{(k+1)} = (B_j^{(k)}\bar{y}_j^{(k)} + m_j(\vec{x}_k)\cdot y_k)/(B_j^{(k)} + m_j(\vec{x}_k)), \tag{12}$$

$$s_{ji}^{(k+1)} = \frac{B_j^{(k)}}{B_j^{(k)} + m_j(\vec{x}_k)}\left[s_{ji}^{(k)} + \frac{(x_{ki} - \bar{x}_{ji}^{(k)})^2}{B_j^{(k)} + m_j(\vec{x}_k)}\right] \quad \text{for } i = 1, \ldots, n, \tag{13}$$

$$B_j^{(k+1)} = B_j^{(k)} + m_j(\vec{x}_k). \tag{14}$$

This process is repeated for all data points ($k = 1, 2, \ldots, n_p$) and for all clusters $j = 1, \ldots, M$. Note that M is not changed in this case and s_{ji} is the recursive equation of the statistically variances of the data points included in the cluster. Eq. (11) is the mean value of cluster j.

On the other hand, if no cluster has satisfied the similarity criterions, i.e. for $\forall j$ we have

$$m_J(\vec{x}_k) = \max_j(m_j(\vec{x}_k)) < \rho \quad \text{or } |y_k - \bar{y}_J| > \tau \cdot d, \tag{15}$$

then a new cluster with volume V is created. For this case, we assume that instance P_k is not close enough to any existing cluster and a new fuzzy cluster $N = M + 1$ is placed in the space region centred in point P_k, i.e.

$$\bar{\boldsymbol{x}}_N = \vec{x}_k, \quad \bar{y}_N = y_k, \quad \vec{\sigma}_N = \vec{\sigma}_0, \quad B_N = 1, \tag{16}$$

where $\vec{\sigma}_0 = (\sigma_0, \ldots, \sigma_0)^{\mathrm{T}}$ is a user-defined constant vector. Note that the new cluster N contains only one member with maximum degree, $m_N(\vec{x}_k) = 1$.

The deviation vector $\vec{\sigma}_j$ is collinear with vector $\vec{s}_j$, i.e.:

$$\sigma_{ji} = s_{ji}\sqrt[n]{\frac{V}{\prod_{r=1}^{n} s_{jr}}} i = 1,\ldots,n. \tag{17}$$

The result of the previous method is a set of ellipsoidal clustering, with the main axis parallel to the axis coordinates of the input space and with constant volume. The process is iterated ($k = 1,\ldots,n_p$) until all input–output instances have been processed. At the end, the clusters with low B value will be removed:

$$\text{Clusters removed} : B_j < \eta, \quad j = 1,\ldots,M, \tag{18}$$

where η is a user-defined constant that represents the minimum support of cluster. The removal occurs for clusters that do not cover enough "territory" of the input space. This process prevents the growth of the number of clusters, by removing overlapped clusters and clusters positioned away from the region to be modelled.

The aforementioned algorithm will be executed for a predefined number of iterations, n_{iter}. At the beginning of each iteration, v, parameters B are weakened by multiplying them by a scalar parameter value α, with $0 \leqslant \alpha \leqslant 1$, so that the clusters are more easily adapted to the new learning process:

$$B_j^{(v+1)} = \alpha B_j^{(v)}, \quad j = 1,\ldots,M. \tag{19}$$

Note that, at each iteration, when the training data is considered once again, the existing clusters can be adjusted and news clusters can be created.

The strong similarity between the cluster representation and the antecedent set of fuzzy rule (1), both with membership Gaussian function, as well as between the consequent parameters θ's and $\bar{y}_j$, suggest us the construction of fuzzy rules based on clusters. At the end we have a set of M initial fuzzy rules for the given input–output data set. In example 1, the CRG algorithm is automatically used to identify a grey cross image.

Example 1. A cross image represented in Fig. 2a is used to illustrate the previous approach. The grey-level, 50×50 pixels image originates a set of R^3 points, (x_1, x_2, I), where I is the intensity in the $\boldsymbol{x} = (x_1, x_2)$ spatial coordinates. In the context of image processing, the fuzzy system is a static function map $I_x = f(x_1, x_2)$ that makes the correspondence between the pixel coordinates (x_1, x_2) and the value of its colour intensity, I_x. Fig. 2b shows the image generated by the flat fuzzy system identified by the proposed CRG algorithm. Fig. 2c shows the image partition by fuzzy memberships, represented by the ellipse level line $\mu(\vec{x}) = \exp(-1)$ (ellipsoid lines) and the correspondent centre of rules (asterisk point). We can see that the cluster have horizontal and vertical orientation and all clusters have the same volume. The cross is described by five ellipsoidal fuzzy rules, while the remaining 25 rules describe the image background.

The CRG algorithm will also be used to the structural identification of each hierarchical level of the HCFS fuzzy model, with little adaptations of the previous algorithm that will be related afterwards, by defining supplementary criterions to create and removing clusters.

The identification process begins with the first model and with the entire available data set. The CRG algorithm is used to find the best cluster from that data, by adapting the number, the centre and the shape

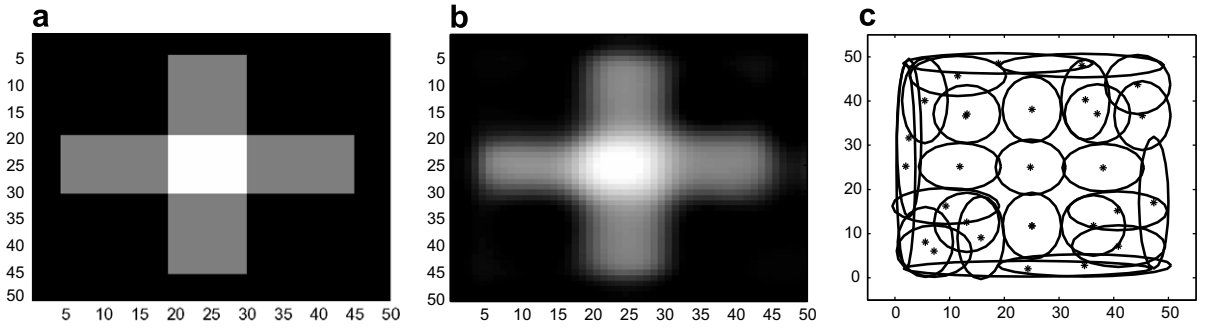

Fig. 2. (a) Grey cross image (50×50 pixels); (b) image representation by a fuzzy system, with 30 rules; (c) fuzzy set repartition (the ellipsoids corresponds to the exp(−1) level and the asterisks to the centres).

of clusters. However, in the HCFS fuzzy model, a selection of data training points could be excluded from the training set in order to improve the quality criterion of the sub-model. This removal can be due to two processes: exclusion of one single point from the training data set, or a set of points is excluded due to the removal of its cluster (fuzzy rule); each algorithm is proposed in Sections 3.2 and 3.3, respectively. This rejected data set from the identification process of one hierarchical level is reused as the training data set of the next hierarchical model. For a sufficient number of hierarchical levels, all training data is used in the identification process.

Let $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_k, \ldots\}$ be the set of all training data, X_i^* be a subset of the initial training data X, $X_i^* \subset X$, used in the identification of level i, and K_i be the index set defined by

$$K_i = \{k | \vec{x}_k \in X_i^* \subset X\}. \tag{20}$$

This subset of training data will be used new criterions to create and remove clusters. The generation of a new cluster is now restricted to the following condition:

$$m_J(\vec{x}_k) = \max_j(m_j(\vec{x}_k)) < \rho \quad \text{or } |y_k - \bar{y}_J| < \tau \cdot d \quad \text{for } k \in K_i \tag{21}$$

This supplementary condition confines the new clusters centres in the space region defined by X_i^*. However, the adaptation process continues to be done by Eqs. (11)–(14) and it is performed for all point in X.

For the cluster removal process a new measure is also necessary. To implement this strategy an additional variable C is defined. Associated with each clusters, C_j contains the sum of membership functions of all points of X_i^* in cluster j (note that B_j is the sum of the membership functions (or values) of all points of X in the cluster), i.e.

$$C_j = C_j + m_j(\vec{x}_k) \quad \text{for } k \in K_i. \tag{22}$$

This new variable is used in the cluster removal test, by replacing the B variable in test Eq. (18) (note that, $C_j \leqslant, B_j$). This condition assures that cluster covers an important region with points of X_i^*.

In the iterative process of the CRG algorithm, this variable is weakened at the beginning of a new iteration, in a similar way to what happened to variable B (Eq. (19)):

$$C_j^{(v+1)} = \alpha C_j^{(v)}, \quad j = 1, \ldots, M. \tag{23}$$

So, cluster that do not satisfactorily cover the X_i^* data set are certainly removed through test (21). At the end of the CRG algorithm, clusters are created that are strongly linked to region of data set X_i^*.

3.2. The recursive least squares training of fuzzy systems

In Section 3.1, a rough fuzzy model for the ith level is automatically generated from the given training data by using the proposed algorithm. From (2), it is clear that the performance of the constructed fuzzy model that approximates the behaviour of the given data set will be influenced by consequent parameters. Consequently, in the parameter identification process, a parameter learning procedure is used such that the procedure can further use the given training data to fine-tune the parameters of the constructed fuzzy model [30]. Additionally, if it can supply information about data points that where rejected from the data training set then the model performance can be improved. For that to occur, an efficient method for adjusting parameters must be done.

If we constrain the f_i's to be linear functions of the θ's parameters, then the problem becomes a least square error (LSE) problem. For our RLS adaptive fuzzy system we have the following problem: each ordered sequence of points $(\vec{x}_k, y_i(\vec{x}_k))$, $k = 0, 1, 2, \ldots, N$ determines an adaptive fuzzy model of Eq. (2) such that $\overline{E}_i = \frac{1}{N}\sum_{k=0}^{N}(d_k - y_i(\vec{x}_k)R_i(\vec{x}_k))^2$ is minimized, where d_k is the desired output for the system when the input is $\vec{x}_k$ and R_i is the relevance of the ith level. In the following, index "i" is omitted for notational convenience.

Consider the *design matrix* $P \in I^{N \times M}$ that stores the description of all training points, that is

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p}_1^{\mathrm{T}}(\vec{x}_1) \\ \vdots \\ \boldsymbol{p}_N^{\mathrm{T}}(\vec{x}_N) \end{bmatrix} = \begin{bmatrix} p_1(\vec{x}_1) & \cdots & p_M(\vec{x}_1) \\ \vdots & \ddots & \vdots \\ p_1(\vec{x}_N) & \cdots & p_M(\vec{x}_N) \end{bmatrix}, \tag{24}$$

where M is the number of rules and N the number of training data points. Matrix $\boldsymbol{P}$ is often called *regressor matrix* and each column correspond to a FBF function.

Using (2), the training data can be rewritten as $\boldsymbol{y} = \boldsymbol{P}\theta + e$, where $\boldsymbol{y} = [y_1 \ \cdots \ y_N]^{\mathrm{T}}$ regroups all data outputs and $\boldsymbol{e} = [e_1 \ \cdots \ e_N]^{\mathrm{T}}$ stores the corresponding residuals representing modelling errors, additive noise and further uncertainties. Parameter vector θ can be determined by minimizing the Euclidean norm of the residual vector that characterizes the LSE problem, with a solution expressed by

$$\hat{\theta} = \boldsymbol{S}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{y}, \tag{25}$$

where $\boldsymbol{S} = (\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P})^{-1}$ is the inverse of *variance matrix*. Therefore, the error vector between the predictions of the fuzzy system from the training set inputs and the actual outputs observed in the training set is $\boldsymbol{e} = \boldsymbol{Q}\boldsymbol{y}$, where $\boldsymbol{Q} = \boldsymbol{I}_M - \boldsymbol{P}\boldsymbol{S}\boldsymbol{P}^{\mathrm{T}}$ is the projection matrix.

If a single example is added (removed) from the training set, then the design matrix acquires (loses) a new row and it is the projection matrix what expands (contracts) while the regressor matrix maintains the same size. Generally, the number of training data N is large and greater than the number of rule output parameters M, and the use of the forward RLS algorithm proved to be computationally more adequate. So, vector θ can be updated using the RLS algorithm by using the following recursive equations [32]:

$$\theta_r = \theta_{r-1} + \boldsymbol{S}_r\boldsymbol{p}_r(\boldsymbol{d}_r - \boldsymbol{p}_r^{\mathrm{T}}\theta_r \cdot R(\vec{x}_r)), \tag{26}$$

$$\boldsymbol{S}_r = \boldsymbol{S}_{r-1} - \frac{\boldsymbol{S}_{r-1}\boldsymbol{p}_r\boldsymbol{p}_r^{\mathrm{T}}\boldsymbol{S}_{r-1}}{1 + \boldsymbol{p}_r^{\mathrm{T}}\boldsymbol{S}_{r-1}\boldsymbol{p}_r}, \tag{27}$$

where $r = 1, 2, \ldots$ is the iteration number, that is, vector θ is adapted recursively for each new sample of data r. At the beginning, $\boldsymbol{S}_0 = \zeta \cdot \boldsymbol{I}$ where ζ is a parameter with high value, $\boldsymbol{I}$ is the identity matrix and $\theta_0 = 0$, a null vector. At the rth iteration Eqs. (26) and (27) minimize the sum of square errors of the ordered data points with index from 1 to r. So, after the learning process with r points, the system has a sum of square error given by

$$\overline{E}_{J,J} = \sum_{j=0}^{r}(d_j - \boldsymbol{p}_j^{\mathrm{T}}\theta_r R(\vec{x}_j))^2 \tag{28}$$

and mean square error given by

$$\overline{E}_{J,J} = \frac{1}{r}E_{J,J} \tag{29}$$

Let $E_{I,J}$ be the sum of square errors of the system in set $I = \{(\vec{x}_j, y_j)|j = 1, \ldots, s\}$ with s data points, which has been trained with set $J = \{(\vec{x}_j, y_j)|j = 1, \ldots, r\}$ with r data points.

On the other hand, the exclusion on point k from the training set is dealt by the backward RLS algorithm with the adaptation of matrix $\boldsymbol{S}$ and vector θ given by

$$\theta_{r-1} = \theta_r - \boldsymbol{S}_r\boldsymbol{p}_r(\boldsymbol{d}_k - \boldsymbol{p}_k^{\mathrm{T}}\theta_r R(\vec{x}_k)), \tag{30}$$

$$\boldsymbol{S}_{r-1} = \boldsymbol{S}_r + \frac{\boldsymbol{S}_r\boldsymbol{p}_k\boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{S}_r}{1 - \boldsymbol{p}_k^{\mathrm{T}}\boldsymbol{S}_r\boldsymbol{p}_k}. \tag{31}$$

The removal of the kth point also affects the sum of square errors, whose value will be recalculated through Eq. (28). This new calculus involves determining the error in all $r - 1$ remaining points and the computation of the corresponding matrix $\boldsymbol{S}_{r-1}$ and vector θ_{r-1}. A more efficient way to express the square error is through the incremental method of the relation given by

$$E_{I,I} = E_{J,J} - \beta_k, \tag{32}$$

where $I = J \setminus \{(\vec{x}_k, y_k)\}$ and β_k represent the reduction of the sum of square errors by exclusion of the kth point. It is easy to prove that β_k results from the product of the square error of fuzzy system at the kth point with the scalar $(1 + \xi_k)$, that is

$$\beta_k = (1 + \xi_k)(d_k - \boldsymbol{p}_k^{\mathrm{T}}\theta_r R(\vec{x}_k))^2, \tag{33}$$

where ξ_k is a positive scalar, with magnitude less than one, where

$$\xi_k = \boldsymbol{p}_k^{\mathrm{T}} \boldsymbol{S}_{r-1} \boldsymbol{p}_k. \tag{34}$$

So, the exclusion of point k from the data training set leads to the reduction of the sum of square errors by a β_k value, which is the square error of the model at the kth point weighed by the $(1 + \xi_k)$ coefficient. However, Eq. (34) shows to be computationally hard because it requires a previous determination of the $\boldsymbol{S}_{r-1}$ matrix. This problem is solved if the $\boldsymbol{S}_{r-1}$ matrix is replaced by an expression of $\boldsymbol{S}_r$ matrix. So, with some algebraic manipulations, ξ_k can be expressed by

$$\xi_k = \frac{\boldsymbol{p}_k^{\mathrm{T}} \boldsymbol{S}_r \boldsymbol{p}_k}{1 - \boldsymbol{p}_k^{\mathrm{T}} \boldsymbol{S}_r \boldsymbol{p}_k}, \tag{35}$$

where $\boldsymbol{S}_r$ is a previously obtained matrix.

Note that the sum of square errors E is always reduced when a data point is excluded from the training data set. The choice of kth point to be removed is based on finding the greatest decrease in the sum of squared errors E (Eq. (32)). A selected criterion is monitored to decide when to stop the removing process. In this work, the Data Rejection Algorithm, DRA, is stopped according to the following definition:

Definition 4. The Data Rejection Algorithm stops when the mean square error $\overline{E}$ is reduced to a value lower than the predefined maximum error, χ, i.e.,

$$\overline{E}_{J,J} < \chi, \tag{36}$$

or when the mean square error $\overline{E}$ starts to grow. This last situation corresponds to a β_k value for which the following inequality is true:

$$\beta_k < \frac{1}{k} E_{J,J} = \overline{E}_{J,J}. \tag{37}$$

The rejection of these points or regions of the training data set are due to the intrinsic fault of capacity of the actual structure in describing its contained information. Generally, the conventional learning methods force the model to fit the description of all the training data, which usually yields a bad model for all data. So, here we propose an algorithm that chooses the data points to exclude from the training data. Naturally, the rejected points are now placed in the training data set of the next level of the hierarchical fuzzy system to participate into the identification process of its system.

The Data Rejection Algorithm, DRA, is summarized as follows:

Step 1: After the forward RLS algorithm, where we applied all training data $J = \{(\vec{x}_j, y_j) | j = 1, \ldots, r\}$ and $r = N$, the parameter vector θ_N and the variance matrix S_N are obtained. The mean square error of the system is $\overline{E}_{J,J}$, given by Eq. (29).

Step 2: If $\overline{E}_{J,J} < \chi$ or $\overline{E}_{J-1,J-1} > \overline{E}_{J,J}$ then stop the data rejection process.

Step 3: Find point k at which value β_k is higher, i.e. $k = \mathrm{argmax}(\beta_j, j = 1, \ldots, r)$, where β_j is given by Eq. (33). This point k will be removed from the data training set J.

Step 4: Update the parameter vector and its variance matrix due to the exclusion of point k from training data by using Eqs. (30) and (31).

Step 5: Reduce the set $J \leftarrow J \setminus \{(\vec{x}_k, y_k)\}$, $r \leftarrow r - 1$ and update the error $E_{J,J} \leftarrow E_{J,J} - \beta_k$. Go to Step 2.

In conclusion, the DRA algorithm rejects from the data set all points that, though their removal, reduce the modelling error. The result is a fuzzy sub-model perfectly adequate for data set points (or regions) that are modelled by the current hierarchical level.

3.3. Adding and removing a fuzzy rule

After training with the previous algorithms, the final system represents a set of patterns that contains important information with regard to the HCFS fuzzy sub-model. The θ parameters reflect a least square

means of these patterns. However, the number of rules will be shown as excessive to represent the identified patterns and some of them can be removed without a significant increase of square error. This situation requires a new algorithm to remove the useless fuzzy rules. Although in the present work there is no need for the task of adding fuzzy rules, its study is important for a better understanding of the process of removing fuzzy rules. So, next we propose two recursive techniques for adding and removing fuzzy rules.

The philosophy of the Forward (Backward) Fuzzy Rule Selection Algorithm, FFRS (BFRS) algorithm, is to sequentially add (remove) from the fuzzy system, one at a time, those rules that cause the largest decrease (smallest increase) in the residual or in the prediction error function. This decrease (increase) in residual can be efficiently calculated as described in the following.

The forward selection algorithm increases the fuzzy system by adding a new FBF that has great contribution to the fuzzy system performance as measured by the error function. Adding a new fuzzy rule to the fuzzy system which already has m rules results in a $(m+1)$th regressor model. Applying it to a trained set with p patterns has the effect of adding an extra column to the *regressor matrix*. If the old *regressor matrix* is $\boldsymbol{P}_m$ and the new FBF function is $q_{m+1}(\vec{x}) = [p_{m+1}(\vec{x}_1), \ldots, p_{m+1}(\vec{x}_p)]^{\mathrm{T}}$, then the new design matrix is

$$\boldsymbol{P}_{m+1} = [\,\boldsymbol{P}_m \quad \boldsymbol{q}_{m+1}\,]. \tag{38}$$

The new variance matrix is $\boldsymbol{S}_{m+1}$, where

$$\boldsymbol{S}_{m+1}^{-1} = \boldsymbol{P}_{m+1}^{\mathrm{T}}\boldsymbol{P}_{m+1} = \begin{bmatrix} \boldsymbol{S}_m^{-1} & \boldsymbol{P}_m^{\mathrm{T}}\boldsymbol{q}_{m+1} \\ \boldsymbol{q}_{m+1}^{\mathrm{T}}\boldsymbol{P}_m & \boldsymbol{q}_{m+1}^{\mathrm{T}}\boldsymbol{q}_{m+1} \end{bmatrix}. \tag{39}$$

Applying the formula for the inverse of a partitioned matrix yields

$$\boldsymbol{S}_{m+1} = \begin{bmatrix} \boldsymbol{S}_m & \boldsymbol{0} \\ \boldsymbol{0}^{\mathrm{T}} & 0 \end{bmatrix} + \frac{1}{\boldsymbol{q}_{m+1}^{\mathrm{T}}\boldsymbol{Q}_m\boldsymbol{q}_{m+1}} \begin{bmatrix} \boldsymbol{S}_m\boldsymbol{P}_m^{\mathrm{T}}\boldsymbol{q}_{m+1} \\ -1 \end{bmatrix} \begin{bmatrix} \boldsymbol{S}_m\boldsymbol{P}_m^{\mathrm{T}}\boldsymbol{q}_{m+1} \\ -1 \end{bmatrix}^{\mathrm{T}}, \tag{40}$$

where $\boldsymbol{Q}_m = \boldsymbol{I} - \boldsymbol{P}_m\boldsymbol{S}_m\boldsymbol{P}_m^{\mathrm{T}}$ is the projection matrix for the original fuzzy system with m fuzzy rules. With this result we can calculate the new projection matrix $\boldsymbol{Q}_{m+1}$, which will contain the contribution of $(m+1)$th fuzzy rule

$$\boldsymbol{Q}_{m+1} = \boldsymbol{Q}_m - \frac{\boldsymbol{Q}_m\boldsymbol{q}_{m+1}\boldsymbol{q}_{m+1}^{\mathrm{T}}\boldsymbol{Q}_m}{\boldsymbol{q}_{m+1}^{T}\boldsymbol{Q}_m\boldsymbol{q}_{m+1}}. \tag{41}$$

The sum-squared-error, can be conveniently written in terms of $\boldsymbol{Q}_m$, for the mth rule fuzzy system, and $\boldsymbol{y}$ by $E_m = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{Q}_m\boldsymbol{y}$. After the addition of the fuzzy rule, the equivalent cost function is increased (negatively) by

$$\Delta E_m = E_{m+1} - E_m = -\boldsymbol{y}^{\mathrm{T}}\frac{\boldsymbol{Q}_m\boldsymbol{q}_{m+1}\boldsymbol{q}_{m+1}^{\mathrm{T}}\boldsymbol{Q}_m}{\boldsymbol{q}_{m+1}^{\mathrm{T}}\boldsymbol{Q}_m\boldsymbol{q}_{m+1}}\boldsymbol{y}. \tag{42}$$

Eq. (42) shows that rule $m+1$ has an individual contribution to the sum-squared-error. The forward selection algorithm therefore aims to select a new rule (new FBF function) for which $|\Delta E_m|$ is maximum.

The process of removing fuzzy rules is similar to the explained for the addition of fuzzy rules. We are interested in knowing the new regression matrix $\boldsymbol{S}_{m-1}$ and the new projection matrix $\boldsymbol{Q}_{m-1}$, after the jth fuzzy rule has been removed from the fuzzy system with m fuzzy rules and for which we know the old regression matrix $\boldsymbol{S}_m$ and the old projection matrix $\boldsymbol{Q}_m$. The present algorithm has a main difference from the previous algorithm: any column can be removed from the P_m while in the adding rule algorithm the new column is always inserted to the right of P_m, just after the mth column.

The projection matrix is invariant to permutation of the columns of the regression matrix, and for this reason we put it at the end column position. From Eq. (41), but now with $\boldsymbol{Q}_m$ in place of $\boldsymbol{Q}_{m+1}$, $\boldsymbol{Q}_{m-1}$ in place of $\boldsymbol{Q}_m$, $\boldsymbol{q}_j$ in place of $\boldsymbol{q}_{m+1}$, the result is

$$\boldsymbol{Q}_{m-1} = \boldsymbol{Q}_m - \frac{\boldsymbol{Q}_m\boldsymbol{q}_j\boldsymbol{q}_j^{\mathrm{T}}\boldsymbol{Q}_m}{\boldsymbol{q}_j^{\mathrm{T}}\boldsymbol{Q}_m\boldsymbol{q}_j}. \tag{43}$$

The removal of one fuzzy rule from a fuzzy system results always in an increase of the cost function, given by

$$\Delta E_{m-1} = E_{m-1} - E_m = -\boldsymbol{y}^{\mathrm{T}} \frac{\boldsymbol{Q}_m \boldsymbol{q}_j \boldsymbol{q}_j^{\mathrm{T}} \boldsymbol{Q}_m}{\boldsymbol{q}_j^{\mathrm{T}} \boldsymbol{Q}_m \boldsymbol{q}_j} \boldsymbol{y}. \tag{44}$$

The proposed Backward Fuzzy Rule Selection algorithm has as its main objective to choose the fuzzy rules to be removed from the fuzzy system, one in each iterative cycle, that lead to a smaller increment in the error, $|\Delta E_m|$.

In this reduction rules process, the new regression matrix $\boldsymbol{S}_{m-1}$, obtained by removal of the jth fuzzy rule, has a smaller dimension than the knowledge of the old regression matrix, $\boldsymbol{S}_m$, with less one row and less one column. Without loss of generality, in the first place we assume that the jth rule to be removed corresponds to the last column of $\boldsymbol{P}_m$. In this case, the relation between $\boldsymbol{S}_{m-1}$ and $\boldsymbol{S}_m$ can be obtained through Eq. (40), by replacing index m with $m-1$ and index $m+1$ with m, and by rewriting it as in the following:

$$\boldsymbol{S}_m = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{d} \\ \boldsymbol{d}^{\mathrm{T}} & e \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_{m-1} & \boldsymbol{0} \\ \boldsymbol{0}^{\mathrm{T}} & 0 \end{bmatrix} + \frac{1}{a} \begin{bmatrix} \boldsymbol{S}_{m-1} \boldsymbol{b} \boldsymbol{b}^{\mathrm{T}} \boldsymbol{S}_{m-1} & -\boldsymbol{S}_{m-1} \boldsymbol{b} \\ -\boldsymbol{b}^{\mathrm{T}} \boldsymbol{S}_{m-1} & 1 \end{bmatrix}, \tag{45}$$

where $\boldsymbol{a} = \boldsymbol{q}_m^{\mathrm{T}} \boldsymbol{Q}_{m-1} \boldsymbol{q}_m$ and $\boldsymbol{b} = \boldsymbol{P}_{m-1}^{\mathrm{T}} \boldsymbol{q}_m$. The relationship between the sub-matrix of both sides of Eq. (45) results in an equation system, whose solution is

$$\boldsymbol{S}_{m-1} = \boldsymbol{C} - \frac{1}{e} \boldsymbol{d}^{\mathrm{T}} \boldsymbol{d}. \tag{46}$$

The removal of jth rule (which is not the last rule) from the fuzzy system is equivalent to the removal of a column j in the middle of matrix $\boldsymbol{P}_m$. Despite that, matrix $\boldsymbol{S}_{m-1}$ can be calculated in a way similar to that previously related in Eq. (45): matrix $\boldsymbol{C}$ is obtained from $\boldsymbol{S}_m$ by removing the jth row and the jth column; and vector $\boldsymbol{d}$ is equal to the previously removed column, but without the jth element.

The final element of the fuzzy rules selection is a method to stop the selection procedure, therefore to seek a compromise between complexity and accuracy. Several metrics have been introduced to measure this compromise at a model with linear parameters [32], since the output of the fuzzy system is linear with respect to the θ parameters, these metrics can be applied here. In this work, a particular metric is used: the Akaike final prediction error (FPE, criterion [33]).

$$F = \frac{1 + v(m/N)}{1 - v(m/N)} \frac{E_m}{N}, \tag{47}$$

where m is the number of rules (equal to the number of parameters) and v is a weighing factor, equal to 2 in this research. This equation penalization makes a compromise between accuracy (E_m/N term) and complexity of models, by taking into account the number of m fuzzy rules.

The above algorithm provides a method for optimally add/remove fuzzy rules from a fuzzy system, with less decrease/increase in the final prediction error, FPE. The Backward Fuzzy Rule Selection (BFRS) algorithm is summarized as follows:

Backward Fuzzy Rule Selection (BFRS) algorithm:

Step 1: Let the fuzzy system with $m \leftarrow M$ rules and the initial values E_m and the FPE, respectively of the cost function (square error) and of the final prediction error.

Step 2: Compute the increase on the cost function ΔE_j, if rule j is removed by using Eq. (44). Repeat this computation for all fuzzy rules of the fuzzy system, $j = 1, \ldots, m$.

Step 3: Set $k = \mathrm{argmin}(\Delta E_j)$ and compute the FPE for the smallest decrease in the FPE function, FPE_k.

Step 4: If $\mathrm{FPE}_k < \mathrm{FPE}$ then remove the kth rule from the fuzzy system and its support regions goes to the upper hierarchical model. Calculate $\boldsymbol{S}$ by (46) and set $m \leftarrow m - 1$. Go to Step 2.

Step 5: Stop the removing rules process and compute the optimal weights in the final fuzzy system from Eq. (25).

3.4. The HCFS identification algorithm

The identification of the hierarchical collaborative Fuzzy System is made by Hierarchical Collaborative Fuzzy System algorithm, BFRS. It uses the CRG, the RLS, the DRA and the BFRS algorithms as presented in the following:

Hierarchical Collaborative Fuzzy System Algorithm, HCFSA

- Let P be the set of input–output training data of the HCFS model.
- For the 1st level, the index set is $K = K_1 = \{1, 2, \ldots, N\}$, where N is the total number of training data.
- Let P_i be the set of points to identify in the ith level of the HCFS. P_i is a subset of P: $P_i = \{(\vec{x}_k, d_k)|k \in K_i\}$, where $K_i \subset K$. For the first level we have $d_1(\vec{x}_k) = y_k$ with $k \in K_1$ and $P_1 = P$.

Step 1: For the ith level, the data to be identified is the pairwise: $(\vec{x}_k, d_i(\vec{x}_k))$, where $k \in K_i$ and $d_i(\vec{x}_k) = y_k - Y_{i-1}(\vec{x}_k)\Re_{i-1}(\vec{x}_k)$. $Y_{i-1}(\vec{x}_k)$ is the fuzzy output of the HCFSA just at the $(i-1)$th level and $\Re_{i-1}(\vec{x}_k)$ is corresponding relevance value.

Step 2: Use the CRG algorithm, with appropriate volume V_i, to generate the appropriate FBF and the corresponding fuzzy rule that covers the P_i data set.

Step 3: Apply the RLS algorithm to estimate the θ parameters by using Eqs. (26) and (27) with data points that belong to P_i.

Step 4: While stop criterion of Definition 4 is not met, execute recursively Steps 5–6 (i.e., the DRA algorithm), else go to Step 7.

Step 5: Reject from P_i the input–output pair of data $(\vec{x}_j, y_j)$ that minimize the most the sum of the square error (Eq. (32)). This task corresponds to choosing the data point with maximum β_j value, i.e. $j = \{j|\beta_j > \beta_k, \forall k \in K_i\}$. The data point j is moved to the next level data set P_{i+1}: $K_i \leftarrow K_i \setminus \{j\}$ and $K_{i+1} \leftarrow K_{i+1} \cup \{j\}$.

Step 6: Readjust the θ parameters by using Eqs. (30) and (31).

Step 7: Use the BFRS algorithm to remove the useless relevant rules. All points connected to the removed rules are placed in the training data of the next level.

Step 8: The response of the HCFS model at the ith level is done by

$$Y_i(\vec{x}_k) = Y_{i-1}(\vec{x}_k) \cdot \Re_{i-1}(\vec{x}_k) + y_i(\vec{x}_k) \cdot R_i(\vec{x}), \tag{48}$$

with relevance value

$$\Re_i(\vec{x}) = \max(R_i(\vec{x}_k); \Re_{i-1}(\vec{x}_k)) \quad \forall \vec{x}_k \in P \tag{49}$$

Jump to the next level, $i = i + 1$. If $i > n$, then *exit*.

4. Application of HCFSA

Two examples are used to illustrate the HCFSA algorithm: the modelling of a non-linear system and the identification of a cross image. In the first one, the output of the recurrent system has been corrupted by a high value of noise, while the second has abrupt changes in image details, which makes these examples hard tests for the HCFSA performance. The output of the hierarchical collaborative fuzzy system is a result of the aggregation of its sub-models according to Eqs. (48) and (49), with $\Re_0(\vec{x}) = 0$ and $Y_0(\vec{x}) = 0$. Each sub-model employed is a multi-input single-output fuzzy logic system of the form of Eq. (2) with relevance given by Definition 3. At the end of the aggregation of various sub-models, the output of the HCFSA is $Y(\vec{x}) = Y_n(\vec{x})$ with relevance $\Re(\vec{x}) = \Re_n(\vec{x})$. The membership function used was the multivariable Gaussian membership (9).

The following data is valid for two examples: The CRG algorithm is used in the structural identification of the HCFSA sub-models. The number of iteration was 50 with $\alpha = 0.8$, in all sublevels. For each level i, a constant volume V_i of the Gaussian membership functions is specified. The choice of the volume value has a strong impact on the type of information that this level describes as well as on the number of rules that are generated. Usually, the identification process begins with high volume value, in an attempt to describe large regions of the input–output space, and is decreased from one sub-model to the next.

After the structural identification made by CRG algorithms the parameters of the fuzzy models are identified by the forward RLS algorithm. After that, a removal of data points is made in order to exclude "undesirable" data points from the training data set and its inclusion in the next level training data. Simultaneously, the model parameters are backwards adapted. This task is made by the DRA algorithm. For both examples we used parameter $\chi = 0{,}001$ and relevance parameters $\gamma = 0.7$ and $\delta = 2$. Finally, the useless rules are removed.

Note that, after the choice of the algorithms parameters, generation of the fuzzy rules for each sub-model, input–output data partition through various hierarchical sub-levels, and tuning of its parameters is completely automatic.

Example 2. Consider an illustrative benchmark non-linear system [34,35]:

$$y(k) = \frac{2.5y(k-1)y(k-2)}{1+y^2(k-1)+y^2(k-2)} + 0.3\cos\left(\frac{y(k-1)+y(k-2)}{2}\right) + 1.2u(k-1) + e(k),$$

where $e(k)$ is a random number of a strong Gaussian noise sequence given by $e \sim N(0,0.316^2)$ and the system input $u(k) = (1/2)(\sin(\pi \cdot k/20) + \sin(\pi \cdot k/50))$, given the initial conditions $y(0) = y(-1) = 0$. Note that input variable has a range value of 2 units while the white noise has a standard deviation of 0.316.

A data sequence of 1000 samples was generated and plotted in Fig. 3a. Fig. 3b represents a noise-free response of the system, $\tilde{y}(k),\ k = 1, 2, \ldots$. The first 500 data samples $\{u(k), y(k)\}$ of corrupted noise data, from $k = 1$ to $k = 500$ were used as an estimation set and the other 500 samples were used to validate the obtained model. The regression sub-models employed are multi-input single-output fuzzy logic systems, with the input vector given by $\vec{x}(k) = [y(k-1)y(k-2)u(k-1)]^{\mathrm{T}}$. All training data was used in the first phase of the identification of the first sub-model of the HCFS. For a volume $V_1 = 0.5$ and $\eta = 10$ we obtained a partition of input space with 13 fuzzy rules. In the second phase (the parameter identification), only 35.7% of data points were used for the identification of the consequent parameters of the fuzzy rules of the first sub-model. The

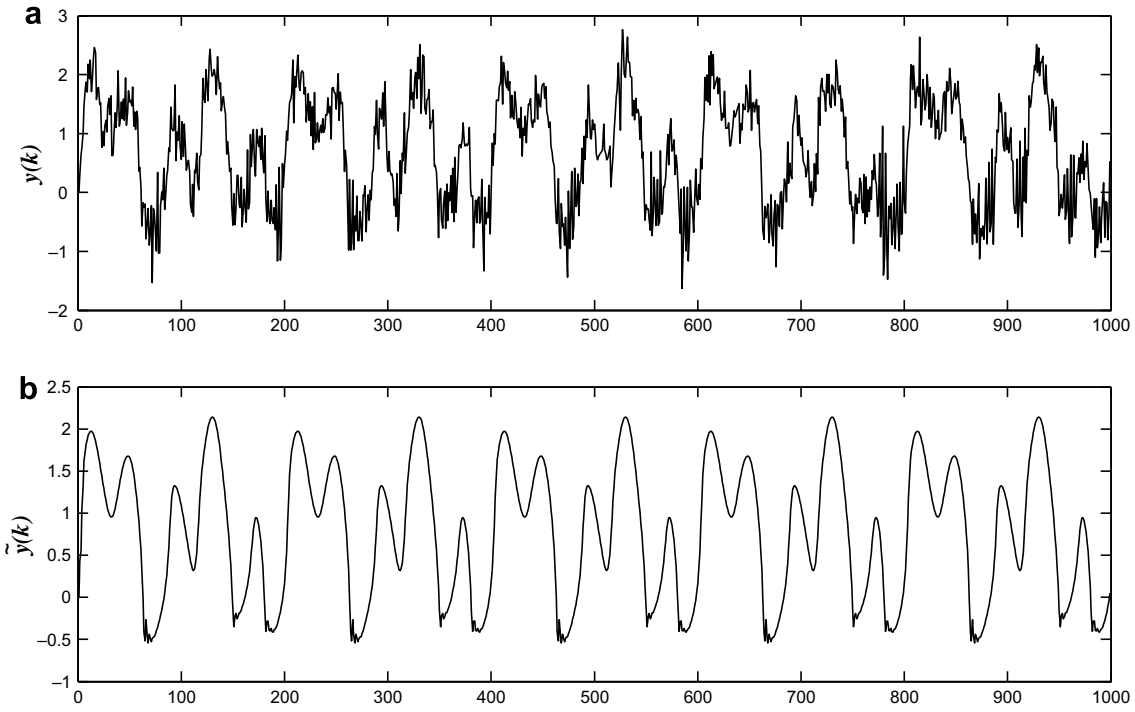

Fig. 3. The output $y(k)$'s in Example 2. (a) Noisy response, $y(k)$; (b) a noise-free response, $\tilde{y}(k)$.

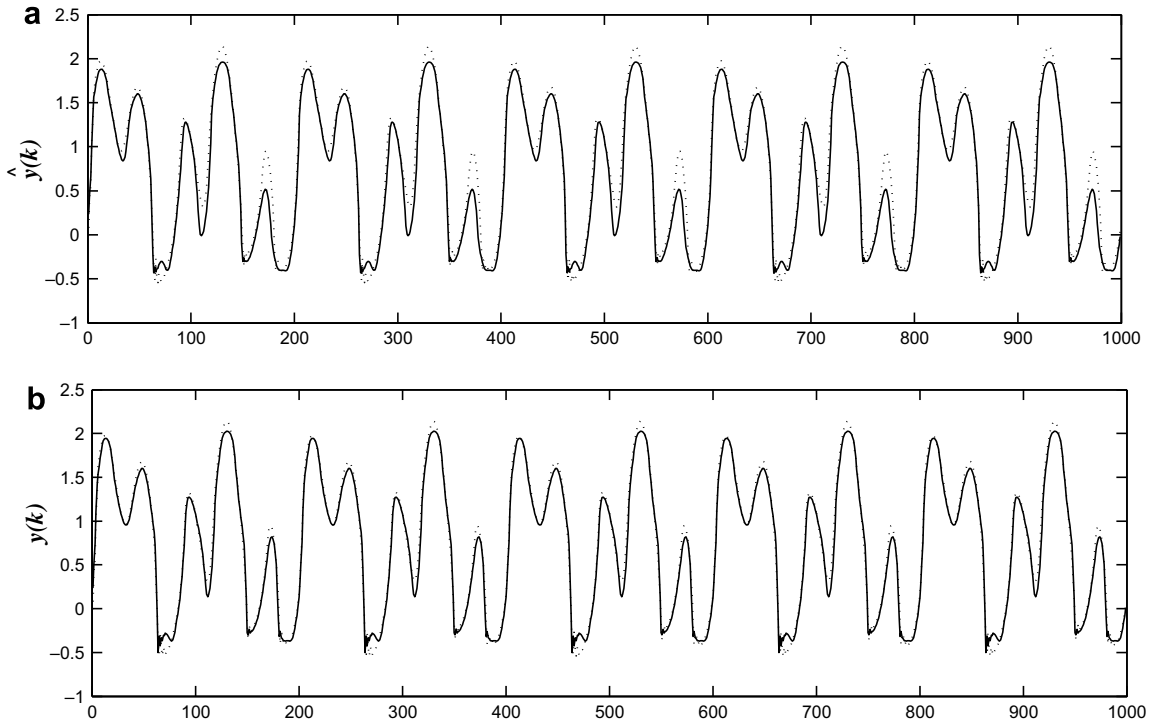

Fig. 4. Model predictions over the training data set ($k = 1, \ldots, 500$) and over the validation data set ($k = 501, \ldots 1000$) (dotted line: noise-free response, solid line: model prediction): (a) Model prediction by first sub-model of HCFS; (b) model prediction by HCFS.

remaining data points are moved to the training data set to be used in identification of the second level of the hierarchical structure. For the second level, we used a volume V_2 that is equal to half of V_1 and with $\eta = 5$. In this manner, we obtained seven fuzzy rules.

Given the same initial condition of the real model, the HCFS model identified by the HCFSA algorithm was used to iteratively generate the model output $\hat{y}(k)$, with the input $\vec{x}(k) = [\hat{y}(k-1)\hat{y}(k-2)u(k-1)]^{\mathrm{T}}$, where the past system output terms $y(k-1)$ and $y(k-2)$, were replaced by model predictions $\hat{y}(k-1)$ and $\hat{y}(k-2)$, and system input $u(k-1)$ is assumed known. This free running fuzzy model tests the stability of the obtained model. The iterative model outputs so generated are plotted in Fig. 4a and b for the response of the first sub-model and for the HCFS, respectively. The mean square error between the fuzzy output model $\hat{y}(k)$ and the noisy-free observations $\tilde{y}(k)$, $k = 1, 2, \ldots$, over both the training and testing sets was 7.902×10^{-3}. Note that the error of the response of the first sub-model when running alone was 2.782×10^{-2}.

A second experience was realized by increasing the CRG parameter η value to 22, using the identification of the first level. This change removes clusters that have a low index of data covering or cluster over-positioning. The number of rules of the fist level is decreased to 5 rules and the second is increased to 10. Fig. 5 shows the response of the first sub-model working alone and of the combined sub-models. The simulation mean square error was 4.635×10^{-2} for the first model and 1.308×10^{-2} for the entire HCFS model. The number of points used in this identification process was equal to the previous experience.

From these results we conclude that the HCFS algorithm is able to automatically model the dynamical process into a two-level hierarchical HCFS, where the fundamental information is modelled by the first level with fewer rules, while accuracy (the "last touch") is given by the second level.

In previous studies [35], this system was used to test other fuzzy model strategies. In all cases the noise variance was much lower and the number of rules is significantly greater. The robustness of the algorithm was also demonstrated with respect to the high level of noise, where it was capable of predicting the system output free of noise.

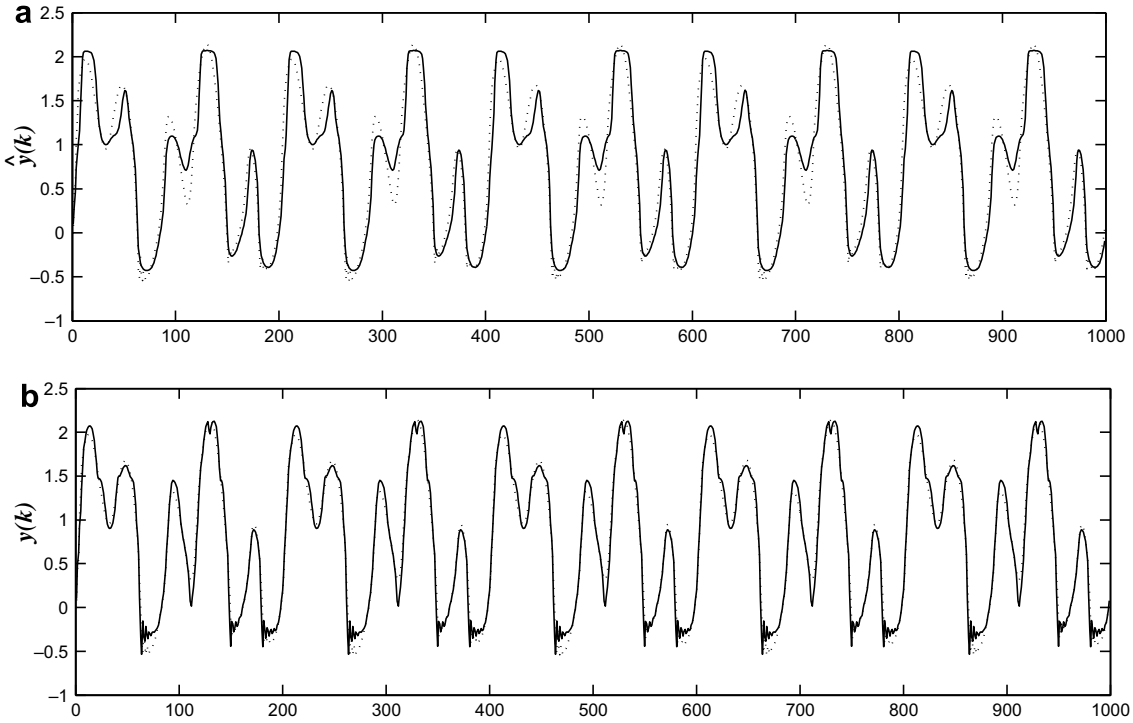

Fig. 5. Model predictions over the training data set ($k = 1, \ldots, 500$) and over the validation data set ($k = 501, \ldots 1000$) for $\eta = 22$ (dotted line: noise-free response, solid line: model prediction): (a) Model prediction by first sub-model of HCFS; (b) model prediction by HCFS.

Example 3. The Abington Cross image (Fig. 2) is a critical image for recognition by a flat FLS (2), as proved in Example 1. The main reasons are the variety of cues such as texture, intensity, edge, which are not adequate to be partitioned by smooth membership functions. The HCFS model and the above algorithm were used to model the cross image into five levels. The CRG algorithm is used interactively with levels with the following parameters: $\tau = 0.02$, $\rho = e^{-2}$, $\sigma = 0.02$, η parameters: 100 (high value), 30, 2.5,1.6, 1, and with cluster volume: 200 (high value), 40, 5 and 0.5, from the first to the last level. The first value of volume allows us to cover all the image range with only one rule, while the last volume is adequate to clustering a reduced number of pixels. The intermediate volumes allows the generation of clusters of different sizes. The number of rules generated for each sub-model was: 1, 6, 22, 9 and 106, respectively from level 1 to level 5. The percentage of rejected

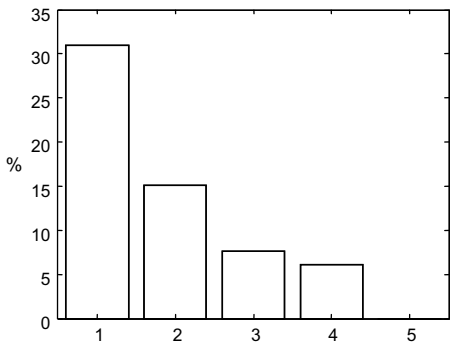

Fig. 6. Percentage of rejected points by the learning algorithm in each level.

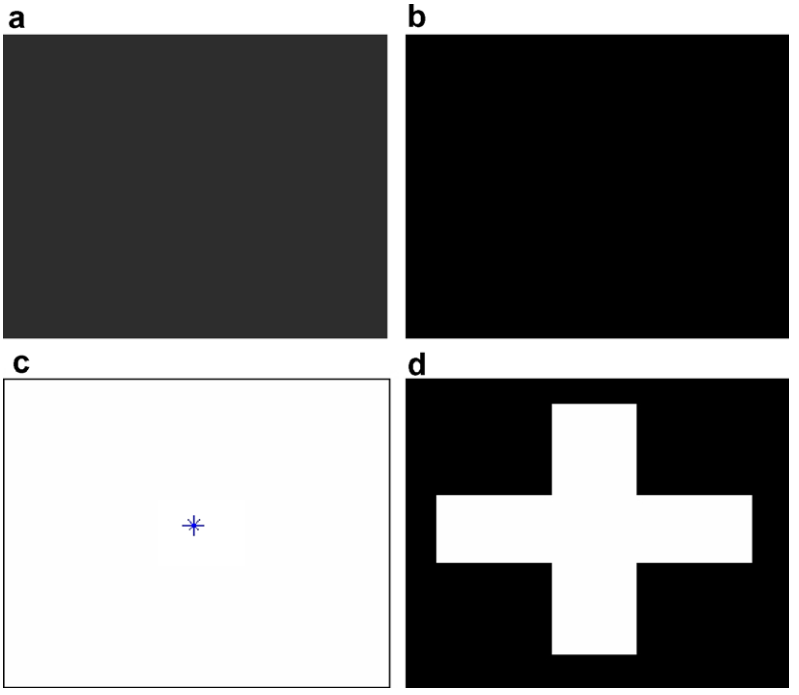

Fig. 7. Results of the 1st sub-model. (a) Image description after Step 2; (b) final image of the level, (c) centre of rule (asterisk point); (d) rejected points (white region).

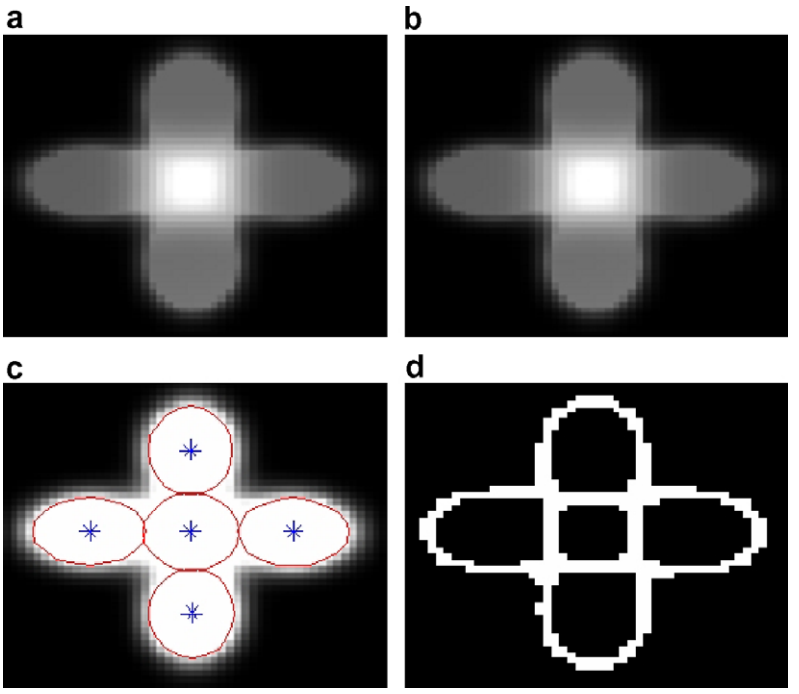

Fig. 8. Results of the 2nd sub-model. (a) Image description after Step 2, (b) final image of hcfs, just second level, (c) relevance of fuzzy sub-model (white = 1) and fuzzy rule placement (ellipses lines) and its centre (asterisk point), (d) rejected points (white region).

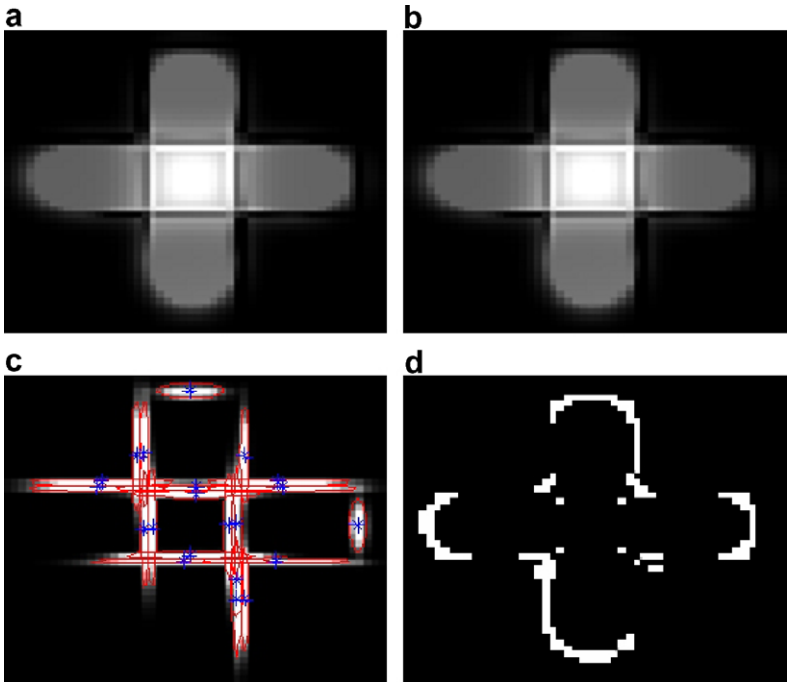

Fig. 9. Results of the 3rd sub-model. (a) Image description after Step 2, (b) final image of HCFS, just third level, (c) relevance of fuzzy sub-model (white = 1) and fuzzy rule placement (ellipses lines) and its centre (asterisk point), (d) rejected points (white region).

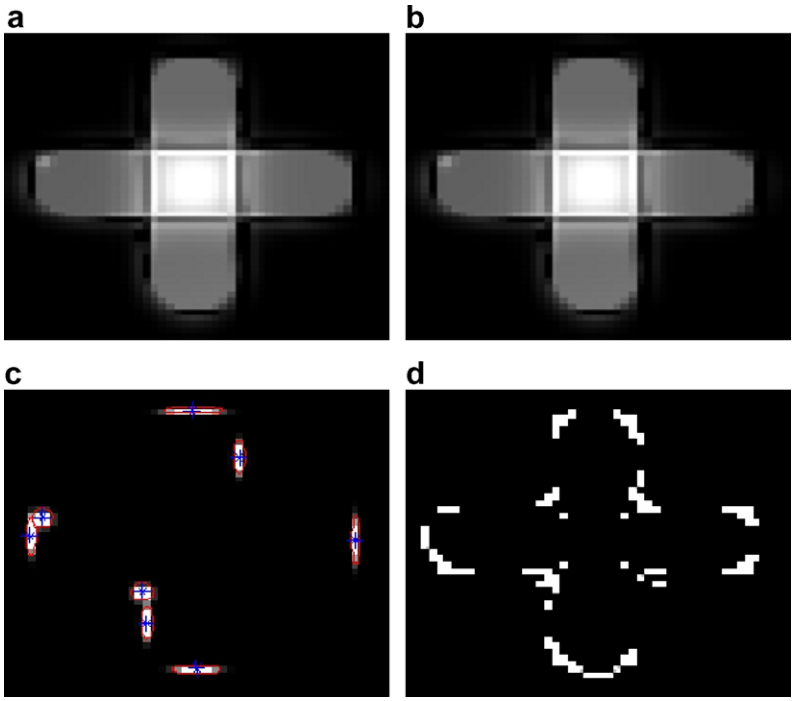

Fig. 10. Results of the 4th sub-model. (a) Image description after Step 2, (b) final image of HCFS, just 4th level, (c) relevance of fuzzy sub-model (white = 1) and fuzzy rule placement (ellipses lines) and its centre (asterisk point), (d) rejected points (white region).

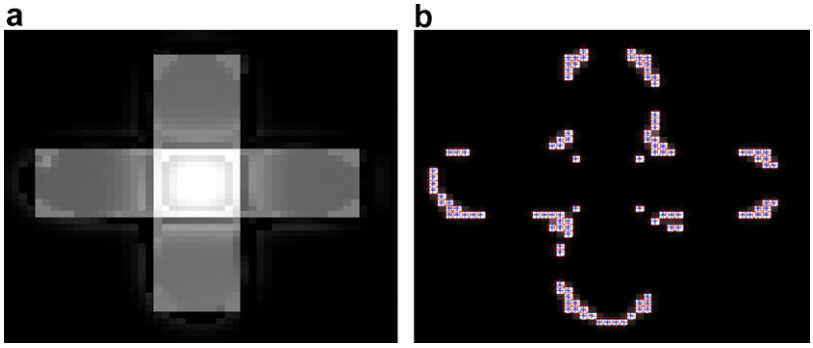

Fig. 11. (a) Cumulative image generated by the hierarchical collaborative fuzzy system, HCFS, from levels 1–5; (b) relevance of fuzzy 5th sub-model (white = 1) and fuzzy rule placement (ellipses lines) and its centre (asterisk point).

points by the algorithm in each level is displayed in Fig. 6. So the first level of the HCFS model identified, with only one rule, about 70% of image pixels, which correspond to the image background; the second level, with only 6 rules, absorbed more than 15% of the pixels, and so on. 85% of image pixels were identified with only seven rules.

Fig. 7 shows the results of the first level. The upper-left hand image is generated by the sub-model after the RLS learning phase (Steps 2 and 3 of the HCFS algorithm), while the opposite right is after removing the undesirable pixels, whose pixels positions are represented in white in Fig. 7d. The position of fuzzy rule is represented in Fig. 7c by the asterisk point. As expected, all cross points are transferred to the next level and the current level plays a part in the description of the background image.

The results of the HCFS algorithm in the second level of the HCFS are displayed in Fig. 8, where the upper left and the upper right images are the image description just of level 2 of the HCFS model, respectively before and after the use of BFRS algorithm. Fig. 8c shows the position and shape of fuzzy rules and its relevance on the image while Fig. 8d shows the rejected data points. Note that the nine rules occupy the cross regions, two overlapped in each arm of the cross and one in the cross centre. Figs. 9 and 10 display the results to the 3rd and 4th levels, respectively. These last sub-models improved the border cross representation by the hierarchical fuzzy model. Finally, the last sub-model, with small volume ellipsoids, refines small regions of the image. Fig. 11a shows the image generated by the complete HCFS model while Fig. 11b displays the rules of the last level. This example is illustrative of HCFS's capability to describe complex images with a small number of rules, where its sub-models have distinct modelling tasks within the hierarchical structure.

5. Conclusion and future work

In this paper we present the hierarchical collaborative fuzzy systems, HCFS, and propose a new learning methodology for it. The presented approach is an efficient method for automatic generation of fuzzy rules for HCFS. The present hierarchical structure of the fuzzy system and its associated algorithms spread out the utilization of the fuzzy technical modelling to a complex modelling system, preserving its main advantages, such as readability and transparency of the fuzzy systems.

Several aspects of the presented algorithm can be easily modified to include other types of relevance functions and aggregations. These modifications have a great impact on the HCFS sub-models. In addition, different criteria can be used in the Data Rejection Algorithm, DRA, and in the Backward Fuzzy Rule Selection (BFRS) algorithm. These alterations could produce a different partition of information into HCFS sublevels.

Acknowledgment

This work was supported by Fundação para a Ciência e Tecnologia (FCT) under Grant POSI/SRI/41975/2001.

References

[1] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[2] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[3] L.X. Wang, A Course in Fuzzy Systems and Control, Prentice-Hall, Englewood Cliffs, NJ, 1997.
[4] S.-Y. Kung, J. Taur, S.-H. Lin, Synergistic modeling and application of hierarchical fuzzy neural networks, Proc. IEEE 87 (9) (1999) 1550–1574.
[5] J.S. Jang, Input selection for ANFIS learning, in: Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE'96), New Orleans, LA, 1996, pp. 1493–1499.
[6] T. Nakashima, T. Morisawa, H. Ishibuchi, Input selection in fuzzy rule-based classification systems, in: Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE'97), Barcelona, Spain, 1997, pp. 1457–1462.
[7] G.V.S. Raju, J. Zhou, R.A. Kisner, Hierarchical fuzzy control, Int. J. Control 54 (5) (1991) 1201–1216.
[8] H. Ying, G. Chen, Necessary conditions for some typical fuzzy systems as universal approximators, Automatica 33 (7) (1997) 1333–1338.
[9] M. Brown, K.M. Bossley, D.J. Mills, C.J. Harris, High dimensional neurofuzzy systems: Overcoming the curse of dimensionality, in: Proc. Int. Joint Conf. on Fuzzy Systems and the 2nd Int. Fuzzy Engineering Symp. Yokohama, Japan, March 1995, pp. 2139–2146.
[10] Min-You Chen, D.A. Linkens, Rule-base self-generation and simplification for data-driven fuzzy models, Fuzzy Sets Syst. 142 (2004) 243–265.
[11] Li-Xin Wang, Analysis and design of hierarchical fuzzy systems, IEEE Trans. Fuzzy Syst. 7 (5) (1999) 617–624.
[12] L.X. Wang, Universal approximation by hierarchical fuzzy systems, Fuzzy Sets Syst. 93 (1998) 223–230.
[13] G.V.S. Raju, J. Zhou, Adaptive hierarchical fuzzy controller, IEEE Trans. Syst. Man Cybern. 23 (1993) 973–980.
[14] H.P. Chen, T.M. Parng, A new approach of multi-stage fuzzy logic inference, Fuzzy Sets Syst. 78 (1996) 51–72.
[15] Fu-Lai Chung, Ji-Cheng Duan, On multistage fuzzy neural network modeling, IEEE Trans. Fuzzy Syst. 8 (2) (2000) 125–142.
[16] R.R. Yager, On the construction of hierarchical fuzzy systems models, IEEE Trans. Syst. Man Cybern. – Part C 28 (1) (1998) 55–66.
[17] Tachibana, Kanta, Furuhashi, A structure identification method of submodels for hierarchical fuzzy modeling using the multiple objective genetic algorithm, Int. J. Intell. Syst. 17 (2002) 495–513.
[18] M.R. Delgado, F.V. Zuben, F. Gomide, Hierarchical genetic fuzzy systems, Inform. Sci. 136 (2001) 29–52.
[19] Ruck Thawonmas, Function approximation based on fuzzy rules extracted from partitioned numerical data, IEEE Trans. Syst. Man Cybern. 29 (4) (1999) 525–534.
[20] Y. Lin, G.A. Cunningham III, S.V. Coggeshall, Using fuzzy partitions to create fuzzy systems from input–output data and set the initial weights in a fuzzy neural network, IEEE Trans. Fuzzy Syst. 5 (1997) 614–621.
[21] C.T. Chao, Y.J. Chen, C.C. Teng, Simplification of fuzzy-neural systems using similarity analysis, IEEE Trans. Syst. Man. Cybern. 26 (2) (1996) 344–354.
[22] Magne Setnes, Robert Babuška, Uzay Kaymak, Hans R. van Nauta Lemke, Similarity measures in fuzzy rule base simplification, IEEE Trans. Syst. Man Cybern. 28 (3) (1998) 376–386.
[23] Thomas A. Runkler, James C. Bezdek, Alternating cluster estimation: A new tool for clustering and function approximation, IEEE Trans. Fuzzy Syst. 7 (4) (1999) 377–393.
[24] C.-C. Wong, C.-C. Chen, A hybrid clustering and gradient descent approach for fuzzy modelling, IEEE Trans. Man. Cybern. B 29 (1999) 686–693.
[25] Janos Abonyi, Robert Babuška, Ferenc Szeifert, Modified Gath–Geva Fuzzy Clustering for Identification of Takagi–Sugeno Fuzzy Models, IEEE Trans. Syst. Man Cybern. – Part B 32 (5) (2002) 612–621.
[26] P. Salgado, J. Boaventura, Greenhouse climate hierarchical fuzzy modelling, Control Eng. Pract. 13(5) 613–628.
[27] W. Pedcrycz, M. Reformat, Rule-based modelling of nonlinear relationships, IEEE Trans. Fuzzy Syst. 5 (2) (1997) 256–269.
[28] E. Roventa, T. Spircu, Averaging procedures in defuzzification processes, Fuzzy Sets Syst. 136 (2003) 375–385.
[29] P. Salgado, Clustering and hierarchization of fuzzy systems, Soft Computing Journal, Springer-Verlag, 2006.
[30] J.S. Jang, ANFIS: Adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (1993) 665–684.
[31] N. Gershenfeld, B. Schoner, E. Metois, Cluster-weighted modelling for time-series analysis, Nature 397 (1999) 329–332.
[32] L. Ljung, System Identification: Theory for the User, Prentice-Hall, Englewood Cliffs, NJ, 1987.
[33] H. Akaike, Fitting autoregressive models for prediction, Ann. Inst. Statist. Math. 21 (1969) 425–439.
[34] C.J. Harris, X. Hong, Neurofuzzy mixture of experts network parallel learning and model construction algorithms, IEE Proc. Control Theory Appl. 148 (6) (2001) 456–465.
[35] X. Hong, C.J. Harris, Nonlinear model structure design and construction using orthogonal least squares and D-optimality design, IEEE Trans. Neural Networks 13 (5) (2002) 1245–1250.