

# A New Fuzzy Set Merging Technique Using Inclusion-Based Fuzzy Clustering

Samia Nefti, *Member, IEEE*, Mourad Oussalah, *Member, IEEE*, and Uzay Kaymak, *Member, IEEE*

**Abstract**—This paper proposes a new method of merging parameterized fuzzy sets based on clustering in the parameters space, taking into account the degree of inclusion of each fuzzy set in the cluster prototypes. The merger method is applied to fuzzy rule base simplification by automatically replacing the fuzzy sets corresponding to a given cluster with that pertaining to cluster prototype. The feasibility and the performance of the proposed method are studied using an application in mobile robot navigation. The results indicate that the proposed merging and rule base simplification approach leads to good navigation performance in the application considered and to fuzzy models that are interpretable by experts. In this paper, we concentrate mainly on fuzzy systems with Gaussian membership functions, but the general approach can also be applied to other parameterized fuzzy sets.

**Index Terms**—Fuzzy clustering, fuzzy modeling, fuzzy sets, inclusion, merging.

## I. INTRODUCTION

CLUSTERING has been acknowledged as a powerful tool to analyze data and to build data-driven models. It is similar to the ability of humans to divide objects into groups or clusters sharing common characteristics. Children for instance can quickly label objects referring to buildings, vehicles, people, animals, etc. Since the introduction of the fuzzy set theory, its application to classification has produced very appealing outcomes, both from the theoretical perspective as testified by the exponential growth of literature in fuzzy clustering, and from the practical viewpoint as suggested by the emerging successful applications. In fuzzy modeling, clustering is equivalent to finding the fuzzy sets sharing the same features or characteristics. In data-driven approaches, the membership functions that represent the linguistic terms in the model are derived from these fuzzy sets (clusters). Usually, one observes a large degree of overlap among the fuzzy sets obtained. In this sense, the fuzzy sets are said to be similar according to some concept or measure of similarity. In order to increase the

interpretability of fuzzy models, these similar fuzzy sets are often combined or merged [1], [2].

### A. Merging Based on Similarity

In the field of merging membership functions in data-driven modeling, the following three issues of paramount importance have emerged: 1) how to quantify the similarity between the fuzzy sets, 2) how to determine which sets to merge, and 3) how to merge the selected fuzzy sets.

Regarding the first issue, the similarity between fuzzy sets is typically quantified by using a similarity measure. The latter is evaluated between pairs of fuzzy sets, and it is assumed that the pairs corresponding to high values of similarity measure represent compatible concepts. Various similarity measures have been proposed for fuzzy sets [3], [4]. In general, one distinguishes the following two types of similarity measures [1]: 1) geometric similarity measures and 2) set-theoretic similarity measures. Geometric similarity measures consider similarity as proximity of fuzzy sets, and are best suited for measuring similarity among nonoverlapping fuzzy sets [5]. Set-theoretic similarity measures are based on the application of set-theoretic operators such as the union and the intersection. They are considered to be more suitable for quantifying similarity between overlapping fuzzy sets [1]. Since automated methods for fuzzy system construction often result in highly overlapping fuzzy sets, set-theoretic fuzzy measures are usually used for simplifying fuzzy rule bases. Setnes *et al.* proposed to use the Jaccard index to quantify similarity [1]. However, measures related to inclusion can also be applied [6].

Regarding the second issue (determining which sets are to be merged), one can distinguish two main approaches. One approach is to determine groups of sufficiently similar fuzzy sets and to merge them. Each group consists of two or more fuzzy sets to be merged. The relation induced by the similarity measure from pairwise similarity assessment of fuzzy sets is, in general, symmetric, but unfortunately, not transitive. Therefore, the induced relation is not a similarity relation in the algebraic sense, and hence, determining the groups of similar fuzzy sets need not have a unique solution. To deal with such a problem, the transitive closure of fuzzy relations can be used [7], but the results are not satisfactory all the time. Consequently, some authors have proposed to apply an iterative solution to the problem [1]. In this approach, similarity between all pairs of fuzzy sets, which are candidates for merging, is considered. Then, one successively merges the two most similar fuzzy sets  $p$  and  $q$  (by replacing them with a new fuzzy set  $p^*$ ). This process continues until no more fuzzy sets can be merged, because the maximum similarity is below a user-defined threshold. Since only

Manuscript received October 17, 2005; revised December 21, 2006.

S. Nefti is with the Department of Computer Science, School of Sciences, University of Salford, Salford M5 4WT, U.K. (e-mail: S.Nefti-Meziani@salford.ac.uk).

M. Oussalah is with the University of Birmingham, Electronics, Electrical and Computing Department, Edgbaston, Birmingham B15 2TT, U.K. (e-mail: M.Oussalah@bham.ac.uk).

U. Kaymak is with the Econometric Institute, Erasmus School of Economics, Erasmus University Rotterdam, 3000 DR, Rotterdam, The Netherlands (e-mail: u.kaymak@ieee.org).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TFUZZ.2007.902011

two fuzzy sets are merged at a time, the problem of transitivity is circumvented. In both approaches, the user must specify a merging threshold, which indicates the similarity above which the fuzzy sets should be merged.

The third issue regarding the actual merging method is solved heuristically. The fuzzy set  $p^*$ , which replaces the merged sets  $p$  and  $q$  is determined from some kind of combination of information from  $p$  and  $q$ . Setnes *et al.* took the support of  $p \cup q$  as the support of the new fuzzy set  $p^*$  [1]. This guarantees the coverage of the whole premise space when  $p^*$  replaces  $p$  and  $q$  in the premise of the rule base. The core of  $p^*$  is then computed by averaging the cores of  $p$  and  $q$ . The problem of determining the fuzzy set that will replace the merged ones is similar to the prototype-merging problem in pattern recognition literature. For example, in view of Chang's algorithm [8],  $p^*$  can be computed as a weighted average between  $p$  and  $q$ . Alternatively, in the modified Chang algorithm (MCA) presented by Bezdek *et al.* [9], a simple arithmetic mean of  $p$  and  $q$  can be used to compute  $p^*$  regardless of the weight values.

### B. Merging in the Parameters Space

The way the above three issues are addressed in similarity-based merging has made it an effective method. However, there are also some drawbacks associated with these methods as follows.

- 1) Usually, user intervention is required to guide the merging process by detecting the fuzzy sets that are sufficiently similar in the data space. Additional parameters are introduced, such as the merging threshold. The optimal values of these parameters for a given problem may be tedious to determine. Adaptive threshold approaches have been proposed in [6], but as mentioned by the authors, the results are often context dependent, which restricts their generalization.
- 2) If the fuzzy sets are merged sequentially, two most similar fuzzy sets at a time, the resulting solution may be far away from the optimal one, both from the viewpoint of the size of the resulting fuzzy rule base and its representation power [10]. Furthermore, the pairwise comparisons and sequential merging increase the computational burden of the method and the results become sensitive to the order in which the fuzzy sets are merged.
- 3) The methods for determining the fuzzy set that replaces the merged sets in data space are mostly based on heuristics. The methods that originate from the pattern recognition community may result in gaps in the final rule base, which means that after the merging phase the fuzzy system may not be able to derive an output for all possible inputs. The method proposed by Setnes *et al.* [1] guarantees the coverage of the rule premise space, but it is unknown how the merge results influence the final system performance if the merging was done in another way.

An alternative method to merging fuzzy sets in the data space is merging them in the parameters space. When one inspects the fuzzy sets obtained by automated techniques, it is often easy to recognize visually groups of fuzzy sets that are similar. This suggests that a clustering approach could be used for merging

those fuzzy sets. Since fuzzy models often use a parameterized representation of fuzzy sets, similar fuzzy sets will also have similar parameter values. Hence, clustering in the parameters space can reveal the underlying similarity structure between the fuzzy sets.

### C. Paper Outline

In this paper, we consider merging parameterized fuzzy sets in the parameters space and its application to data driven modeling. We propose a new technique for merging fuzzy sets by evaluating the inclusion degrees of each datum in the underlying prototype in the parameters space. For this purpose, we propose a novel fuzzy clustering algorithm, where the objective function implicitly accounts for overall inclusion degrees [11] of each datum in the underlying prototype. A distance minimization criterion is also incorporated into the objective function in order to prevent the trivial solution of a very large prototype. The prototypes obtained from the novel clustering algorithm are used for replacing the similar fuzzy sets.

The outline of the paper is as follows. In Section II, we propose a new merging algorithm based on clustering in the parameters space. This algorithm accounts for the inclusion degree of each datum in the prototype class. In this paper, we consider only fuzzy systems with Gaussian membership functions, but the results could be generalized to other types of fuzzy sets, too. The application of the proposed clustering algorithm in rule base simplification is discussed in Section III. In Section IV, we discuss a mobile robot navigation problem as an illustrative application example. The results show that our algorithm for rule base simplification is in agreement with expert intuitive reasoning regarding the simplification of fuzzy systems. We also compare our results with another algorithm for merging fuzzy sets in the data space. Finally, conclusions are given in Section V.

## II. CLUSTERING IN THE PARAMETERS SPACE OF FUZZY SETS

Two fuzzy sets  $p$  and  $q$  are similar if their membership functions have approximately the same values over all elements of the domain on which they are defined. This is also the basis for using set-theoretic similarity measures, which are evaluated on discrete domains. On nondiscrete (continuous) domains, the membership functions are usually evaluated on discrete samples, which are obtained according to some discretization process. In many fuzzy systems, however, the shape of the membership functions is not completely free. Instead, parametric membership functions are used, so that the general mathematical description of the membership functions is fixed. The use of parametric membership functions in parameters space simplifies their storage in computers, and they can be evaluated for any element of the continuous domain [12]. For the parametric membership functions, the similarity between fuzzy sets can also be evaluated by considering the parameters of the sets: Similar fuzzy sets will have approximately the same values of their parameters. For example, two triangular fuzzy sets  $T(x; a_1, b_1, c_1)$  and  $T(x; a_2, b_2, c_2)$  [13], [14] that are similar would have similar values for their parameters  $a_i$ ,  $b_i$ , and  $c_i$  for  $i = 1, 2$  in the parameters space.

Gaussian membership functions are often used in fuzzy systems. Especially when the fuzzy systems are obtained by automated methods, Gaussian membership functions are useful because of their differentiability, so that supervised learning algorithms can be used, such as various gradient descent learners [15] and neurofuzzy systems [13]. In the rest of this paper, we concentrate on fuzzy systems with Gaussian membership functions, but the general approach can also be applied to fuzzy systems with other types of membership functions. A Gaussian membership function  $\mu_i$  can be defined by three parameters, i.e., the mean  $m_i$ , the standard deviation  $\sigma_i$ , and the height  $h_i$ . We assume that all membership functions are normalized, so that their heights equal 1. In that case, each membership function is characterized by the two parameters  $\mu_i$  and  $\sigma_i$ , and it can be represented as a point in parameters space  $\mathbb{R}^2$ . Because similar membership functions would have similar parameters in the parameters space, a fuzzy clustering algorithm applied to data in this space could be used to detect groups of similar membership functions. We now turn our attention to fuzzy clustering.

Fuzzy clustering partitions a data set into a number of overlapping groups. One of the first and most commonly used fuzzy clustering algorithms is the FCM (FCM) algorithm [16]. Since the introduction of FCM, many derivative algorithms have been proposed and different applications have been investigated. Typically, FCM allows the determination of the membership value  $u_{ij}$  of each datum  $\mathbf{x}_i (i = 1, \dots, n)$  to cluster  $j$  based on the distance from datum  $\mathbf{x}_i$  to cluster prototype  $\mathbf{v}_j (j = 1, \dots, c)$ . The number  $c$  of classes is supposed to be known beforehand by the user. The obtained partition is optimal in the sense of minimizing the objective function

$$J = \sum_{i=1}^n \sum_{j=1}^c u_{ij}^\alpha d_{ij}^2 \quad (1)$$

subject to the constraint

$$\sum_{j=1}^c u_{ij} = 1. \quad (2)$$

The symbol  $d_{ij}$  indicates the distance from the cluster prototype  $\mathbf{v}_j$  to the datum  $\mathbf{x}_i$ . The parameter  $\alpha > 1$  is the fuzziness parameter, which controls the spread of the membership functions induced by  $u_{ij}$ . The partition matrix  $\mathbf{U}$ , whose elements consist of the computed membership values  $u_{ij}$ , indicates the partitioning of the data set into different clusters  $j$ , or, equivalently, the membership grade of each datum  $\mathbf{x}_i$  to the class  $j$  whose prototype is  $\mathbf{v}_j$ .

For any semidefinite and positive matrix  $\mathbf{A}$ , the distance  $d_{ij}$  can be written as

$$d_{ij}^2 = (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{A} (\mathbf{x}_i - \mathbf{v}_j). \quad (3)$$

Thus, any choice of a matrix  $\mathbf{A}$  induces a specific kind of distance interpretation, and consequently, generates its own meaning of cluster shape. For instance, if  $\mathbf{A}$  is the identity matrix,  $d_{ij}$  corresponds to a Euclidean distance and it induces roughly spherical clusters. Gustafson and Kessel [17] have focused on the case where the matrix  $\mathbf{A}$  is different for each cluster  $j$ .  $\mathbf{A}_j$  is obtained from the covariance matrix of cluster

$j$ , while the determinant of each  $\mathbf{A}_j$ , which stands for the volume of the cluster, is kept constant. This enables detection of ellipsoidal clusters. Bezdek *et al.* [18], [19] have investigated the case where one of the eigenvectors of the matrix (that corresponds to the largest eigenvalue) is maximized. This allows the detection of linear clusters such as lines or hyperplanes. Dave [20] proposed a special formulation of the objective function that yields a better description of circular shape. Later, he also proposed a method to deal with random noise in data sets [21]. Krishnapuram and Keller [22] put forward another formulation of  $J$  where the membership values are not normalized according to (2). Instead, the algorithm is implicitly constrained by the formulation of the objective function  $J$ .

The preceding indicates clearly that neither the formulation of the matrix  $\mathbf{A}$  nor that of the objective function  $J$  is completely fixed. Therefore, some flexibility is allowed, depending on the application. In rule base simplification, it is important that the membership function that replaces a group of similar membership functions represents a general concept that includes the concepts it replaces. In this way, the coverage of the variable space by the fuzzy rules does not deteriorate. Further, large overlap between the rules improves the interpolation property of the fuzzy model. Indeed, the inclusion characteristic of the replacing membership functions has been observed in previous studies on rule base simplification as in [1]. Although the similarity between fuzzy sets is quantified by a symmetric set-theoretic measure such as the Jaccard index, there are explicit mechanisms in the merging algorithm that biases the replacing function towards the inclusion of the membership functions it replaces. Setnes *et al.*, for example, achieve this by taking the union of the supports of the membership functions, as explained before [1].

For the merging purpose, one could cluster the Gaussian membership functions, which are represented by their mean and standard deviation, in  $\mathbb{R}^2$ , by using a fuzzy clustering algorithm such as the FCM algorithm. The prototype of a cluster then represents another membership function, which could be used to replace the membership functions in that cluster. However, the distance structure in the product space of fuzzy set parameters need not correspond to an inclusion structure between fuzzy sets. Since the replacing function must be biased towards the inclusion of membership functions it replaces, one must constrain the fuzzy clustering algorithm additionally in order to detect the inclusion of fuzzy sets in the cluster prototype. For this purpose, we introduce an inclusion concept hidden in the distance structure, which accounts for inclusion in a global sense. In the following, we discuss an inclusion index for Gaussian fuzzy sets defined in the parameters space and its incorporation in fuzzy clustering.

#### A. Construction of Inclusion Index

The goal here is to model the amount of inclusion between two membership functions. Let  $m$  and  $\sigma$  be the parameters corresponding to the mean and standard deviation, respectively, of a Gaussian membership function  $G$ . Let us denote by  $G_i$  the Gaussian membership function characterized by  $(m_i, \sigma_i)$ . Let  $Id(G_1, G_2)$  stand for the degree of inclusion of the Gaussian  $G_1$  in the Gaussian  $G_2$ . Globally, we require that the more the

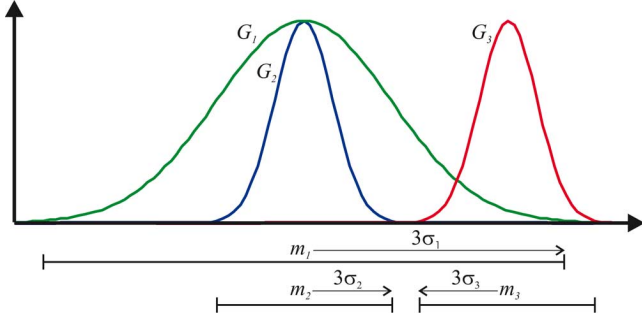


Fig. 1. Interval representation for Gaussian membership functions.

former fuzzy set is a subset of the latter, the larger the value of  $Id(G_1, G_2)$ .

For any Gaussian membership function  $G$ , it is known that almost 98% of the membership function is concentrated within the interval  $[m - 3\sigma, m + 3\sigma]$ . Each membership function can, therefore, be represented as an interval centred at  $m_i$  and with length  $6\sigma_i$ , as shown in Fig. 1. For the sake of clarity, we will denote with  $\sigma'$  (respectively,  $\sigma'_i$ ) the values of  $3\sigma$  (respectively,  $3\sigma_i$ ). Inspired from the fuzzy set-theoretical inclusion, the inclusion grade of  $G_1$  into  $G_2$  can be represented as the ratio  $(G_1 \cap G_2)/(G_1)$ . From this perspective, the inclusion grade equals zero whenever fuzzy sets  $G_1$  and  $G_2$  are disjoint ( $G_1 \cap G_2 = \emptyset$ ) and reaches the maximum value, which is one, whenever  $G_1$  is a subset of  $G_2$  ( $G_1 \subset G_2$ ). Strictly speaking, there are various formulations of fuzzy set inclusion indices in the literature of fuzzy set theory. However, the aforementioned ratio is among the best candidates that preserve the logical relations held by the standard set inclusion index [23], [24].

The use of the aforementioned interval interpretation of the Gaussians allows us to construct a model of the previous ratio. An estimation of  $Id(G_1, G_2)$  can be obtained as

$$Id(G_1, G_2) = \frac{L(M_1 \cap M_2)}{2\sigma'_1} \quad (4)$$

where  $M_1$  represents the interval  $[m_1 - \sigma'_1, m_1 + \sigma'_1]$ ,  $M_2$  represents the interval  $[m_2 - \sigma'_2, m_2 + \sigma'_2]$ , and  $L: \mathbb{I} \rightarrow \mathbb{R}$  is a mapping from the set of intervals on real numbers ( $\mathbb{I}$ ) to the set of real numbers, which assigns for each interval  $[a, b]$  its length  $b - a$ . More specifically, (4) can be rewritten as

$$Id_1(G_1, G_2) = \begin{cases} \frac{I_1}{2\sigma'_1}, & \text{if } 0 \leq I_1 \leq 2\sigma'_1 \\ 0, & \text{if } I_1 < 0 \\ 1, & \text{if } I_1 > 2\sigma'_1 \end{cases} \quad (5)$$

where  $I_1$  denotes  $\sigma'_1 + \sigma'_2 - |m_1 - m_2|$ .

Alternatively, one may use squared values of parameters in order to avoid the use of the absolute value in (5). Then, a counterpart of (5) becomes

$$Id_2(G_1, G_2) = \begin{cases} \frac{I_2}{4(\sigma'_1)^2}, & \text{if } 0 \leq I_2 \leq 4(\sigma'_1)^2 \\ 0, & \text{if } I_2 < 0 \\ 1, & \text{if } I_2 > 4(\sigma'_1)^2 \end{cases} \quad (6)$$

with  $I_2 = (\sigma'_1 + \sigma'_2)^2 - (m_1 - m_2)^2$ . Equation (6) can be rewritten using step functions  $H$  as

$$Id_2(G_1, G_2) = \frac{I_2}{4(\sigma'_1)^2} H(I_2) + \frac{4(\sigma'_1)^2 - I_2}{4(\sigma'_1)^2} H(I_2 - 4(\sigma'_1)^2) \quad (7)$$

which holds almost everywhere. Note that the standard step functions  $H$  are defined as

$$H(x - a) = \begin{cases} 0, & \text{if } x < a \\ 1/2, & \text{if } x = a \\ 1, & \text{if } x > a \end{cases} \quad (8)$$

From this perspective,  $Id_2(G_1, G_2)$ , as defined in (7), corresponds to a straightforward interpretation of the previous fuzzy set-theoretical inclusion grade, and therefore, it preserves all the intuitive features and properties of the fuzzy set-theoretical inclusion grade. As soon as  $|m_i - m_j| > 3(\sigma_i + \sigma_j)$ , which means that the two membership functions are far away from each other,  $Id(G_i, G_j)$  vanishes. If the fuzzy sets have the same mean, i.e.,  $|m_i - m_j| = 0$  and  $\sigma_i \leq \sigma_j$ , then  $Id(G_i, G_j)$  equals one. Otherwise, the inclusion grade captures the overlap between the two fuzzy sets with respect to that of  $G_i$ .

The index  $Id_2(G_1, G_2)$  attains its maximum value when the first fuzzy set is fully included in the second. For cluster merging purposes, however, it may be useful to assess not only whether a fuzzy set is included in the other one, but also the extent to which the including distribution is large compared to the included one. In this course, one may consider omitting the upper bound of the index. The index then becomes

$$Id_3(G_1, G_2) = \begin{cases} \frac{I_2}{4(\sigma'_1)^2}, & \text{if } I_2 \geq 0 \\ 0, & \text{if } I_2 < 0 \end{cases} \quad (9)$$

or, equivalently, by using step functions

$$Id_3(G_1, G_2) = \frac{I_2}{4(\sigma'_1)^2} H(I_2). \quad (10)$$

Loosely speaking, the latter captures not only the inclusion of  $G_1$  in  $G_2$  but also the extent to which the spread of  $G_1$  is smaller than that of  $G_2$ . It holds that as soon as a physical inclusion  $G_1 \subset G_2$  occurs, then  $Id_3(G_1, G_2) \geq 1$ . Otherwise, the index still captures the degree of overlapping between the two distributions with respect to the distribution  $G_1$ . Consequently,  $Id_3$  offers a better performance when the relative size of distributions is desired as part of the evaluation index, which, in turn, endows the inclusion index with discrimination power. In particular, as exhibited in Fig. 2, the inclusion index  $Id_3$  allows us to discriminate between the different scenarios of inclusion induced by the different pairs of membership functions. In this course, the formulation in (10) leads to the following ordering [subscript 3 is omitted from  $Id_3$  for clarity purposes, i.e.,  $Id_a$  stands for evaluation of index  $Id_3$  in the case of pair of fuzzy sets pertaining to Fig. 2(a)]:

$$Id_a(G_2, G_1) \geq Id_b(G_2, G_1) \geq Id_c(G_2, G_1) \geq Id_d(G_2, G_1) = Id_e(G_2, G_1) = 0 \quad (11)$$

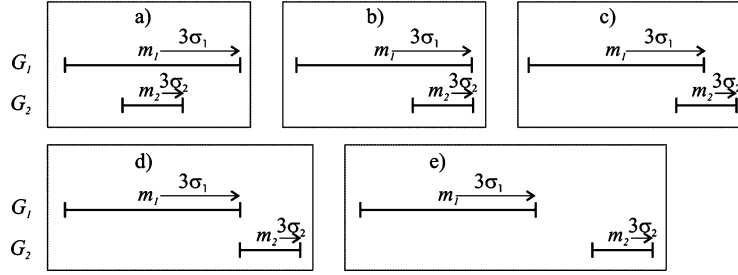


Fig. 2. Illustration of inclusion for several pairs of fuzzy sets.

which is in agreement with the intuition regarding the given examples.

### B. Inclusion-Based Fuzzy Clustering Algorithm

We now introduce a fuzzy clustering algorithm that incorporates the inclusion index. The main idea is to obtain class prototypes such that all elements of that class are roughly included in the Gaussian class prototype in the sense of evaluation given by index  $Id_3$ . Ideally for each class, one would look for a Gaussian prototype that maximizes the overall inclusion degrees over the set of Gaussians belonging to that class, or equivalently, that maximizes the total sum of inclusion degrees of each Gaussian into that prototype, while each inclusion index is weighted by the corresponding value of the membership value  $u_{ij}^\alpha$ . Consequently, the problem boils down to maximizing the quantity  $\sum_{j=1}^c \sum_{i=1}^n Id_3(G_i, G_{v_j}) U_{ij}^\alpha$ , or equivalently, minimizing the objective function

$$J = - \sum_{j=1}^c \sum_{i=1}^n Id_3(G_i, G_{v_j}) U_{ij}^\alpha \quad (12)$$

subject to (2). Here,  $G_i$  corresponds to the  $i$ th Gaussian of initial datum with mean  $m_i$  and standard deviation  $\sigma_i$ , and  $G_{v_j}$  corresponds to the Gaussian prototype of the  $j$ th class whose mean and standard deviations are  $m_{v_j}$  and  $\sigma_{v_j}$ , respectively.

Unfortunately, the optimization problem (12) and (2) without further constraints would lead to the extreme solution where the spread (standard deviation) of all the prototypes tend to their maximal values. To circumvent this effect, a term can be added to (12) to minimize the distance from each prototype to all elements (Gaussians) of the same class as the aforementioned prototype. A possible formulation of such reasoning consists of a linear combination of objective functions (1) and (12). This leads to the following optimization problem:

$$\begin{aligned} \text{Minimize } J &= - \sum_{j=1}^c \sum_{i=1}^n Id_3(G_i, G_{v_j}) u_{ij}^\alpha \\ &+ w \sum_{j=1}^c \sum_{i=1}^n (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{A} (\mathbf{x}_i - \mathbf{v}_j) u_{ij}^\alpha, \end{aligned} \quad w > 0 \quad (13)$$

subject to (2). In (13), each Gaussian  $G_i$  is interpreted as a vector  $\mathbf{x}_i$  with coordinates  $(m_i, 3\sigma_i)$ . Similarly, the  $j$ th prototype can

be represented as a vector with coordinates  $(m_{v_j}, 3\sigma_{v_j})$ . The weight  $w$  is a positive number, which acts as a regularization term and controls the balance between the inclusions and distances while quantifying the relative importance of both terms with respect to each other. Indeed, (13) indicates a balance between maximizing inclusion indices of each element in the prototype  $G_{v_j}$ , and minimizing the distance from the prototype to these elements. Besides, as neither the distances  $d_{ij}$  nor the inclusion evaluations  $Id_3$  are normalized entities and due to the presence of negative parts in (13),  $w$  plays a role of an adjustment factor. Note that the value of  $w$  that ensures a rational behavior for the aforementioned optimization is not unique as will be pointed out later. On the one hand, choosing  $w$  relatively large makes the distance minimization requirement a predominant part in the objective function (13). This, in turn, brings the aforementioned optimization closer to the standard FCM algorithm. On the other hand, taking  $w$  relatively small makes the inclusion evaluation a predominant part in the objective function (13). Thereby, (13) tends towards the optimization (12), which, in turn, induces prototypes with a maximum spread as already mentioned. Some possible suggestions for  $w$  are provided later on.

Let  $\mathbf{B}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$  and  $\mathbf{B}_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ . Then,  $Id_3$  can be rewritten as

$$Id_3(G_i, G_{v_j}) = \frac{1}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} S_{ij} H(S_{ij}) \quad (14)$$

with

$$S_{ij} = (\mathbf{x}_i + \mathbf{v}_j)^T \mathbf{B}_2 (\mathbf{x}_i + \mathbf{v}_j) - (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{B}_1 (\mathbf{x}_i - \mathbf{v}_j). \quad (15)$$

Therefore, a counterpart of the objective function in (13) is

$$\begin{aligned} J &= - \sum_{j=1}^c \sum_{i=1}^n \frac{1}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} S_{ij} H(S_{ij}) u_{ij}^\alpha \\ &+ w \sum_{j=1}^c \sum_{i=1}^n (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{A} (\mathbf{x}_i - \mathbf{v}_j) u_{ij}^\alpha, \quad w > 0. \end{aligned} \quad (16)$$

We can now calculate the update equations that lead to the optimal solution by combining (16) and (2) using augmented Lagrangian methodology. The optimal solutions are reported in Proposition 1 whose proof is given in the Appendix.

- Step 1 Fix the number of clusters  $c$ , the parameter of fuzziness  $\alpha$ , the value of  $w$  and initialise the matrix  $\mathbf{U}$  by using the fuzzy c-means algorithm.  
 Step 2 Determine  $\mathbf{v}_j$  from (18).  
 Step 3 Determine the new matrix  $\mathbf{U}$  using previous evaluation of  $S_{ij}$  and (17).  
 Step 4 Test if matrix  $\mathbf{U}$  is stable; if stable, stop; else return to Step 2.

Fig. 3. Inclusion-based fuzzy clustering algorithm.

*Proposition 1:* Minimizing the objective function (13) subject to constraint (2) leads to

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left[ \frac{-\frac{1}{4}(\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} S_{ik} H(S_{ik}) + w d_{ik}^2}{-\frac{1}{4}(\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} S_{ij} H(S_{ij}) + w d_{ij}^2} \right]^{\frac{1}{\alpha-1}}} \quad (17)$$

with  $d_{ij}^2 = (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{A}(\mathbf{x}_i - \mathbf{v}_j)$ . Further

$$\mathbf{E} \mathbf{v}_j = \mathbf{F} \quad (18)$$

where

$$\mathbf{E} = \sum_{i=1}^n \left[ \frac{1}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} (\mathbf{B}_1 - \mathbf{B}_2) H(S_{ij}) u_{ij}^\alpha + 2w \mathbf{A} \mathbf{x}_i u_{ij}^\alpha \right] \quad (19)$$

and

$$\mathbf{F} = \sum_{i=1}^n \left[ \frac{1}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} (\mathbf{B}_1 + \mathbf{B}_2) \mathbf{x}_i H(S_{ij}) u_{ij}^\alpha + 2w \mathbf{A} \mathbf{x}_i u_{ij}^\alpha \right]. \quad (20)$$

In our case, the matrix  $\mathbf{A}$  in (13)–(20) coincides with identity matrix, which makes the distance metric equivalent to the Euclidean distance.

Note that one needs to ensure that the solution  $\mathbf{v}_j$  does not make  $S_{ij}$  vanish as step function is not differentiable for  $S_{ij} = 0$ . This requirement is usually satisfied, since numeric optimization provides only an approximate result. In order to simplify the resolution of (17), where the quantity  $H(S_{ij})$  is also a function of  $\mathbf{v}_j$  to be used in computing  $\mathbf{v}_j$ , we evaluate  $S_{ij}$ , thereby  $H(S_{ij})$ , by using the previous estimate of  $\mathbf{v}_j$  (from the previous step of the iterative algorithm). The matrix solution is then  $\mathbf{v}_j = \mathbf{E}^{-1} \mathbf{F}$ . This approximation is well justified from at least three arguments. First,  $H(S_{ij})$  is a binary variable, and so any small error in  $\mathbf{v}_j$ , if relatively close to the true  $\mathbf{v}_j^*$ , has no influence at all on the estimation of  $H(S_{ij})$ . Second, as the number of iterations increases, the estimations of  $\mathbf{v}_j$  tend to be very close to each other, which offers appealing justification to the first argument. Third, this reasoning is very similar in spirit to iterative approaches applied for solving linear/nonlinear systems in optimization problems [25].

A possible initialization of  $\mathbf{v}_j$  consists of using the FCM algorithm. In that case, the proposed inclusion-based fuzzy clustering algorithm can be summarized as in Fig. 3.

- Step 1 Fix the number of clusters  $c$ , the parameter of fuzziness  $\alpha$ , the value of  $w$  and initialise the matrix  $\mathbf{U}$  and the prototypes  $\mathbf{v}_j$  by using the fuzzy c-means algorithm.  
 Step 2 Evaluate quantities  $S_{ij}$  and  $H(S_{ij})$  by using the estimates  $\mathbf{v}_j$ .  
 Step 3 Determine the new matrix  $\mathbf{U}$  using (17).  
 Step 4 Determine prototypes  $\mathbf{v}_j$  using (18).  
 Step 5 Test if matrix  $\mathbf{U}$  is stable; if stable, stop; else return to Step 2.

Fig. 4. Approximate inclusion-based fuzzy clustering algorithm.

Alternatively, (19) and (20) can be computed using an approximated model, where  $H(S_{ij})$  is evaluated at the previous step. Then, the previous algorithm is modified as in Fig. 4.

### C. Choice of the Weighting Parameter $w$

In this section, we consider possible approaches to selecting appropriate values for the parameter  $w$ , since it is an important parameter for the clustering results. At least two different strategies can be suggested for selecting an appropriate value of  $w$ .

1) *Constraint-Based Methodology:* In this case, the choice of  $w$  is constrained by the ranges assigned to the entities involved in the optimization problem, like the matrix  $\mathbf{U}$  or the data matrix  $\mathbf{X}$ . Especially, when one considers (17), an inconvenient choice of  $w$  may lead to membership values that are slightly greater than one or that are slightly negative. Even though it is customary in optimization to round the value off to the closer boundary range (either 0 or 1 for  $u_{ij}$ ) [26], this situation can be avoided by assigning a specific range to values of  $w$ . Proposition 2 now holds.

*Proposition 2:* If the condition  $w \geq (\max_i \sigma_i^2)/(4 \min_i \sigma_i^2)$  holds, then it is guaranteed that  $0 \leq u_{ij} \leq 1$ .

The proof of the proposition can be found in the Appendix. Note that the aforementioned condition is only sufficient and not necessary, as the aim is not to find the optimal bounds of  $w$ , which are not obvious. In other words, the above lower bound on  $w$  is not the most conservative one.

In the case where both components of data  $\mathbf{x}_i$  lie within the unit interval, it might be desirable that the prototype vectors also have their components within the unit interval. In this respect, Proposition 3 holds.

*Proposition 3:* When the data  $\mathbf{x}_i$  lie within the unit square and if the condition  $w \geq 0.5$  holds, then both components of the prototype vector  $\mathbf{v}_j (j = 1, \dots, c)$  lie within the unit interval.

The proof of Proposition 3 can also be found in the Appendix. It can be proven from (18) that regardless the value of  $w$ , if the mean component of data  $\mathbf{x}_i$  lies within the unit interval, then the mean component of the prototype vectors  $\mathbf{v}_j$  also lies within the unit interval. However, the fact that the spread (standard deviation) components also lie within the unit interval does not necessarily entail a spread component of the prototype, since the inclusion constraint would intuitively force the prototype's spread to be maximized. Consequently, the requirement of both prototype components lying within the unit interval is restrictive, but can be justified in some applications or automated systems where all parameters are bounded from the above.

Note, again, that similar to Proposition 2, the condition pointed out in Proposition 3 is only a sufficient condition.

Besides, the condition in Proposition 2 often entails that of Proposition 3 as soon as there is large discrepancy among the values of  $\sigma_i$ .

Given the previous condition, a rational choice for  $w$  is to use the lower bound of the inequality (i.e.,  $w = (\max_i \sigma_i^2)/(4 \min_i \sigma_i^2)$  in case of Proposition 2 or  $w = 0.5$  in case of Proposition 3). This choice can be justified as the most pessimistic evaluation in the absence of further evidence, which ensures coherence in the sense of the fulfillment of the desired constraints. Such reasoning is quite common in possibilistic reasoning schemes, where the lower bound provides an indication about the necessity-based evaluation [27].

2) *Empirical Methodology*: In this case, the choice of  $w$  relies either on expert knowledge or on intensive simulation and/or experimental tests. Inspired by robust statistics theory and in the light of suggestions regarding robust fuzzy clustering [26], a possible solution is to let  $w$  vary at each iteration of the algorithm and be calculated as (21), shown at the bottom of the page, where the superscripts  $k$  and  $k-1$  stand for the  $k$ th and  $(k-1)$ th iteration, and  $\eta$  is a proportionality factor. Equation (21) indicates a choice of  $w$  that is proportional to the ratio of overall inclusion evaluations and the overall distance evaluations, while the proportionality term  $\eta$  allows us to control the convergence. It should be for instance small at the beginning to encourage the formation of small clusters and then should increase gradually to encourage agglomeration. A combination of empirical and constraint-based approaches can also be obtained by constraining the choice of  $\eta$  such that either the  $\mathbf{U}$ 's constraints or  $\mathbf{V}$ 's constraints are held. However, it was found in practice that such a formulation sometimes leads to instability due to the variability of  $\eta$  at each iteration.

Another empirical formulation has been found to work well in practice provided both components of the objects in the data set lie within the unit interval. In this case

$$w = \max \left( \frac{1}{(1 - \min_i \sigma_i)(1 + \min_i \sigma_i)}, \frac{1}{\max_i \sigma_i^2} \right). \quad (22)$$

As for most empirical formulations, the motivation for (22) is mainly experimental, despite being related to the constraint-based approach when turned into inequality and accounting for  $\mathbf{V}$ 's constraint under certain conditions. This choice of  $w$  has worked well in our experimental data set.

### III. RULE BASE SIMPLIFICATION WITH INCLUSION-BASED FUZZY CLUSTERING

In this section, we consider how rule base simplification can be achieved by using the inclusion-based fuzzy clustering algorithm from Section II. Rule base simplification is concerned with complexity reduction, transparency, and interpretability of fuzzy models. While complexity reduction is concerned with controlling the number of rules in a fuzzy system, transparency and interpretability are concerned with constraining the fuzzy

membership functions such that semantic interpretations can be attached to the fuzzy concepts that the fuzzy sets represent. In recent years, fuzzy rule base simplification and data-driven generation of interpretable fuzzy models have received much attention in the literature (see, e.g., [2], [28], and [29]). In [30], interpretability is maintained by constraining the location of the membership functions during the learning phase. Another approach based on merging similar fuzzy sets after initial rule base construction is investigated in [1], [4], [31], and [32]. Evolutionary approaches to the generation of interpretable fuzzy models have been studied, among others, in [33] and [34]. Recently, an online approach to obtaining interpretable fuzzy models has also been proposed [35].

When fuzzy rules are obtained by automated techniques, there is usually much redundancy in the resulting rule base regarding the distribution of similar fuzzy sets that represent compatible concepts [1]. In many cases, groups or clusters of similar fuzzy sets can be recognized when the fuzzy sets in these rule bases are inspected visually (see, e.g., Fig. 6). Based on this observation, we suggest a clustering-based approach to rule base simplification. The algorithm takes into account the inclusion of the fuzzy sets in the cluster prototypes.

#### A. Generation of Simplified Models

Given the formulation of the inclusion-based fuzzy clustering algorithm from Section II-B, we can now sketch the fuzzy modeling and simplification approach based on the proposed merging algorithm. The modeling consists of three phases. In Phase I, the initial fuzzy model is generated by using a suitable automated fuzzy modeling technique (which is not described in this paper). Some possibilities are fuzzy clustering (e.g., [31] and [36]), neurofuzzy and gradient-descent learning systems (e.g., [37] and [38]), or evolutionary adaptation (e.g., [39]–[41]). Phase I results in a set of fuzzy rules with associated antecedent membership functions and consequent parameters (or consequent membership functions in case of Mamdani-type systems). In Phase II, the membership functions are considered in the parameters space. Each Gaussian membership function is represented by its mean and standard deviations. The merging algorithm is then applied to determine the prototypes that will replace the initial fuzzy sets. In general, the fuzzy sets are replaced by the prototype for which they have the highest membership. This completes Phase II. In Phase III, the new rule base is screened further. The replacement operation may lead to rules with the same antecedents. These rules are merged, and possibly the consequent parameters are reestimated. Furthermore, the rule base may further be adapted in order to optimize the rule premises and the consequents. The proposed modeling and simplification procedure is summarized in Fig. 5.

#### B. Related Work and Discussion

The proposed algorithm requires as prerequisite the specification of the number of classes used by the merging algorithm.

$$w^{(k)} = \eta \frac{\sum_{j=1}^c \sum_{i=1}^n \frac{1}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} S_{ij}^{k-1} H(S_{ij}^{k-1}) (w_{ij}^{k-1})^\alpha}{\sum_{j=1}^c \sum_{i=1}^n (\mathbf{x}_i - \mathbf{v}_j^{k-1})^T \mathbf{A} (\mathbf{x}_i - \mathbf{v}_j^{k-1}) (w_{ij}^{k-1})^\alpha} \quad (21)$$

**Phase I:**

Determine a fuzzy system from data: learn fuzzy rules, membership functions and consequent parameters. If not available already, convert membership functions into Gaussians parameterised by mean and standard deviation.

**Phase II:**

For each feature do:

- Normalize Gaussian means and three times the standard deviation to the unit interval.
- Represent Gaussian membership functions as points in the parameters space.
- Determine groups of compatible membership functions by using inclusion-based fuzzy clustering.
- Replace each membership function in the rulebase by the prototype to which it has highest membership.

**Phase III:**

Screen rulebase:

- Remove redundant rules, *i.e.* rules whose antecedents already exist in the rulebase.
- Re-estimate rule consequents and, if needed, optimise the final rulebase.

Fig. 5. Fuzzy modeling and rule base simplification using inclusion-based clustering.

The number of classes could be determined by one of the many cluster validity criteria proposed in the literature; see, for instance, [16] and [42]. The inclusion-based clustering algorithm can easily be accommodated to account for an unknown number of clusters and use some cluster validity index. For example, the authors of [43] have proposed a new validity index, which was applied to the FCM algorithm. Such methods can be extended to our inclusion-based method straightforwardly. Another relevant issue is how to perform rule base simplification, given that each rule possesses both a premise and a consequent part. After the simplification, there may be rules that conflict with each other as they may have different consequents for the same premise. A conflict resolution mechanism is needed to resolve such cases [1]. The handling of such situations in our paper is explained in Section IV. Finally, regarding the accuracy of models obtained from rule base simplification, there is always a tradeoff between the accuracy of the model and its transparency, which is related to the minimization of the number of fuzzy rules (see, for instance, [44]). This tradeoff is not investigated explicitly in this paper.

#### IV. APPLICATION EXAMPLE

In this section, we describe the application of our proposed algorithm to a mobile robotics problem. The three phases from Fig. 5 are described in detail. Further, the performance of the algorithm is compared to the rule base simplification approach proposed in [1].

##### A. Mobile Robot Navigation

The example we consider in this section is a real-time mobile robot navigation task. Information from two sensors is considered for an obstacle avoidance task. The standard fuzzy system proposed by Wang and Mendel [30] has been used as a controller for the behavior of the robot in front of obstacles without any prior knowledge about interactions of the robot with the environment. As discussed in Section III, the design methodology for the fuzzy system is divided into three phases. Phase I refers to the configuration phase, where the fuzzy rules, the member-

ship functions, the strength of the rules, and other parameters are generated online by using a backpropagation through-time (BPTT) algorithm. Phase II concerns the complexity reduction of the rule base generated from Phase I. The interpretability of the rule base is improved by using the proposed inclusion-based fuzzy clustering approach for the simplification of the rule base. This phase can be done offline, since no input from the real-time system is required. Phase III deals with the adaptation of the consequent parts of the rules, where conflicting rules are resolved. This phase is performed online. We now discuss these three phases in more detail.

1) *Configuration of the Fuzzy System:* In Phase I, a fuzzy system is generated by using online data. The robot used in this experiment is equipped with eight ultrasonic sensors for obstacle detection. The inputs for the obstacle avoidance controller are the ultrasonic measurements provided by robot sensors. Only two measurements are considered to be important for this task, namely,  $d_1$  and  $d_2$ , corresponding, respectively, to the lateral and the frontal sensors with respect to a robot configuration. The output corresponds to the orientation  $\theta$  of the controller, with respect to the robot axis. Note that the velocity of the robot, when moving, is assumed to be constant.

We use information in the form of relative distance values. Note that the sensors provide evaluative information rather than instructive information. By using this information, one can define an objective function in order to evaluate the performance of the controller. Using the relative distance values, we define the following objective function:

$$J = \frac{1}{2} \sum_{k=1}^N \sum_{n=1}^2 \|S_{n,k} - S_{n,k}^d\|. \quad (23)$$

In (23), the symbol  $S_{n,k}$  stands for the distance from sensor  $n$  to the obstacle or object identified by that sensor at sample time  $k$ . Similarly, the symbol  $S_{n,k}^d$  stands for the minimum or the desired distance from sensor  $n$  to that obstacle at sample time  $k$ . Wang and Mendel's standard fuzzy system [30], which is based on Takagi–Sugeno rules, is used as the architecture for the fuzzy controller. The system is initialized randomly with 25 rules. Hence, 25 fuzzy sets were used for each input. The optimal parameters are obtained by minimizing the objective function (23) after the controller produces a sequence of control actions. A BPTT algorithm [45], which is not described further in this paper, is used for the training. The algorithm determines the optimal parameters for the Gaussian membership functions of the fuzzy system and the rule strengths.

2) *Interpretability and Complexity Reduction:* As shown in Fig. 6(a) and (b), the rule base obtained after the learning stage in Phase I contains highly overlapping fuzzy sets. Inclusion-based fuzzy clustering algorithm is applied to reduce the complexity of the rule base and to improve its interpretability. The membership matrix  $\mathbf{U}$  and the matrix  $\mathbf{V}$  of cluster prototypes are obtained from the clustering algorithm, which converged to a stable solution in less than 15 iterations in most cases. Based on knowledge about the process dynamics, four clusters have been generated for  $d_1$  and  $d_2$ . The prototypes determined by inclusion-based fuzzy clustering are depicted in Fig. 6(c) and (d) (in thick lines).



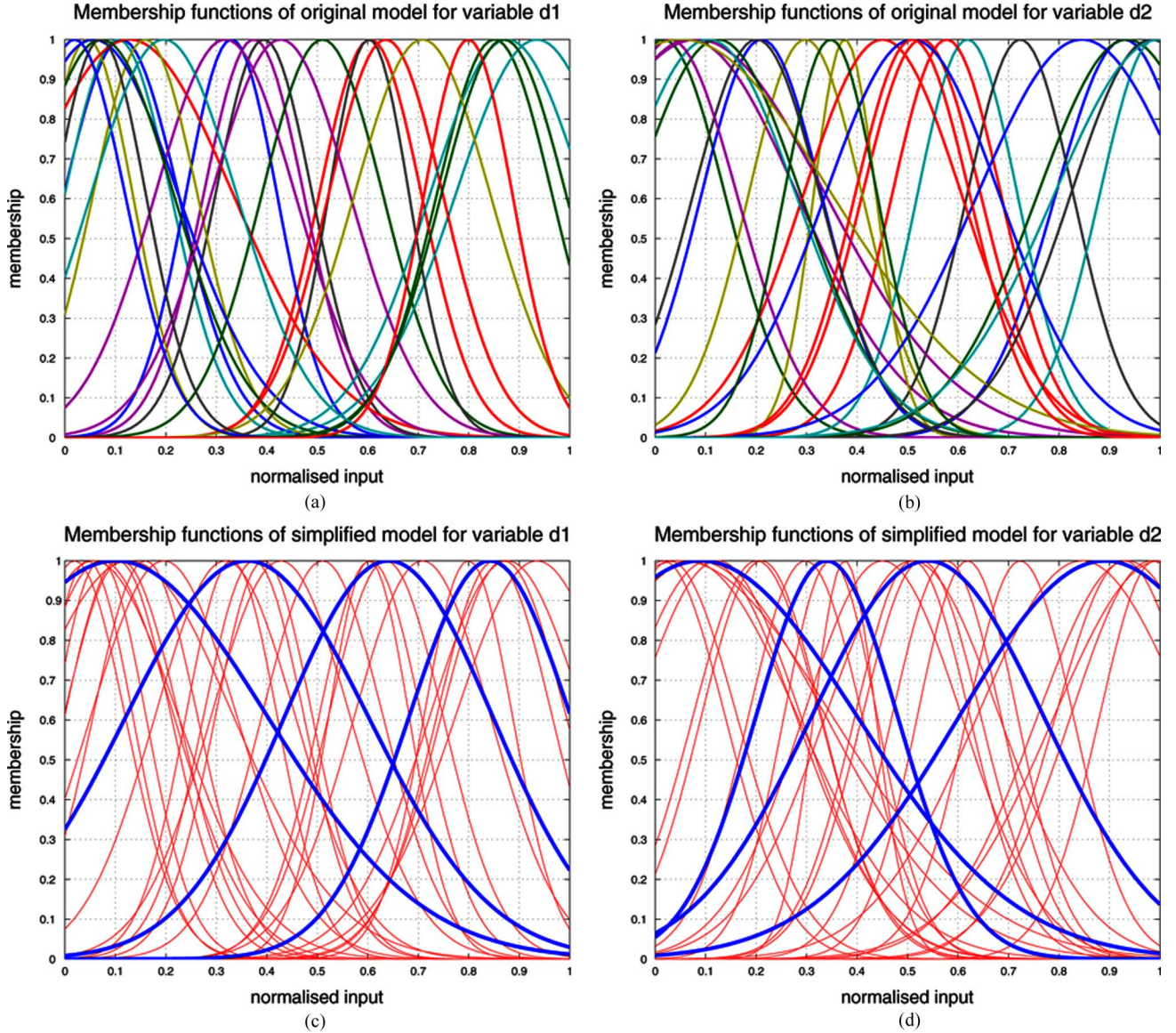


Fig. 6. Fuzzy sets for the sensor (a)  $d_1$  and (b)  $d_2$  after learning in Phase I. Fuzzy sets for the sensor (c)  $d_1$  and (d)  $d_2$  in the simplified rule base.

The parameters of these cluster prototypes are as follows:

$$\begin{aligned}
 d_1 : & \quad C_{1,1}(0.10, 0.30) \quad C_{1,2}(0.36, 0.24) \\
 & \quad C_{1,3}(0.64, 0.21) \quad C_{1,4}(0.84, 0.16) \\
 d_2 : & \quad C_{2,1}(0.09, 0.31) \quad C_{2,2}(0.34, 0.14) \\
 & \quad C_{2,3}(0.54, 0.23) \quad C_{2,4}(0.89, 0.29).
 \end{aligned}$$

Now, let us assign the following labels to these prototypes:

$$\begin{aligned}
 d_1 : & \quad C_{1,1} : TP \quad C_{1,2} : P \quad C_{1,3} : M \quad C_{1,4} : L \\
 d_2 : & \quad C_{2,1} : TP \quad C_{2,2} : P \quad C_{2,3} : M \quad C_{2,4} : L.
 \end{aligned}$$

Each membership function in the rule base from Phase I is replaced automatically with the closest prototype by using the matrix  $\mathbf{U}_1$  for the sensor  $d_1$  and the matrix  $\mathbf{U}_2$  for  $d_2$ . The new partition in the reduced rule base is shown in Fig. 6(c) and (d).

To study how the rules have been modified, let us consider the following sample of the rules generated after the learning in Phase I.

- Rule 1: IF  $d_1$  is near (0.1117, 0.1144) and  $d_2$  is near (0.9290, 0.1364), THEN  $\theta$  is 0.01.
- Rule 2: IF  $d_1$  is near (0.8511, 0.1150) and  $d_2$  is near (0.0158, 0.1235), THEN  $\theta$  is  $-0.68$ .
- Rule 3: IF  $d_1$  is near (0.6051, 0.1021) and  $d_2$  is near (0.5277, 0.1181), THEN  $\theta$  is  $-0.30$ .
- Rule 4: IF  $d_1$  is near (0.1050, 0.1061) and  $d_2$  is near (0.6188, 0.0998), THEN  $\theta$  is 0.25.
- Rule 5: IF  $d_1$  is near (0.3742, 0.1000) and  $d_2$  is near (0.0679, 0.2152), THEN  $\theta$  is 0.60.

Tables I and II show the partition matrices  $\mathbf{U}_1$  and  $\mathbf{U}_2$  that are obtained from the clustering step. By using the partition matrix, one can determine automatically which prototype will replace a particular membership function, by checking the prototypes with the highest membership values. For example, the membership function  $F_{1,1}$  defined by the parameters (0.1117, 0.1144)

TABLE I  
PARTITION MATRIX  $U_1$  OF  $d_1$

	$\mu_{C_{1,1}}(F_{1,j})$	$\mu_{C_{1,2}}(F_{1,j})$	$\mu_{C_{1,3}}(F_{1,j})$	$\mu_{C_{1,4}}(F_{1,j})$
$F_{1,1}$	0.8179	0.1382	0.0290	0.0149
$F_{1,2}$	0.7792	0.1531	0.0435	0.0242
$F_{1,3}$	0.7202	0.1945	0.0549	0.0303
$F_{1,4}$	0.6556	0.2232	0.0765	0.0447
$F_{1,5}$	0.6300	0.2429	0.0807	0.0464

TABLE II  
PARTITION MATRIX  $U_2$  OF  $d_2$

	$\mu_{C_{2,1}}(F_{2,j})$	$\mu_{C_{2,2}}(F_{2,j})$	$\mu_{C_{2,3}}(F_{2,j})$	$\mu_{C_{2,4}}(F_{2,j})$
$F_{2,1}$	0.0271	0.0573	0.1242	0.7915
$F_{2,2}$	0.6353	0.2429	0.0895	0.0323
$F_{2,3}$	0.0373	0.2383	0.6715	0.0529
$F_{2,4}$	0.0454	0.1854	0.6351	0.1341
$F_{2,5}$	0.8509	0.1012	0.0361	0.0118

for  $d_1$  has the largest membership to the prototype  $C_{1,1}$  (membership of 0.8179) and the membership function  $F_{2,1}$  defined by the parameters (0.9290, 0.1364) of  $d_2$  has the largest membership to the prototype  $C_{2,4}$  (membership of 0.7915). Hence,  $F_{2,1}$  is replaced by  $C_{2,4}$  during rule base simplification.

After applying the proposed algorithm, the premise parts of rules 1, 15, 21, and 25 coincide as well as those pertaining to rules 2 and 18; rules 4, 10, and 22; rules 5, 12, and 19; rules 6 and 13, and the rules 14 and 17. Therefore, the underlying rules can be merged, reducing the number from 25 to 14 rules. After reestimating the parameters of the merged rules, the sample of rules becomes as follows.

- Rule 1, 15, 21, 25: IF  $d_1$  is near  $C_{1,1}$  and  $d_2$  is near  $C_{2,4}$ , THEN  $\theta$  is 0.01.
- Rule 2, 18: IF  $d_1$  is near  $C_{1,4}$  and  $d_2$  is near  $C_{2,1}$ , THEN  $\theta$  is  $-0.6$ .
- Rule 3: IF  $d_1$  is near  $C_{1,3}$  and  $d_2$  is near  $C_{2,3}$ , THEN  $\theta$  is  $-0.30$ .
- Rule 4, 10, 22: IF  $d_1$  is near  $C_{1,1}$  and  $d_2$  is near  $C_{2,3}$ , THEN  $\theta$  is 0.2.
- Rule 5, 12, 19: IF  $d_1$  is near  $C_{1,2}$  and  $d_2$  is near  $C_{2,1}$ , THEN  $\theta$  is 0.61.

Alternatively, we can also write these rules using the previous linguistic labels. For instance, rules 12 and 14 can be rewritten as follows.

- Rule 12: IF  $d_1$  is  $TP$  and  $d_2$  is  $TP$ , THEN  $\theta$  is 0.7
- Rule 14: IF  $d_1$  is  $L$  and  $d_2$  is  $P$ , THEN  $\theta$  is  $-0.5$ .

3) *Resolving Conflicts and Adapting Consequent Parts:* After replacing the fuzzy sets with prototypical fuzzy sets generated from Phase II, the performance of the fuzzy system, in general, deteriorates since the consequent parts of the rules are not optimal. Therefore, one must redetermine the rule consequents. In this phase, the adaptation of the simplified rule base and the handling of conflict situations are performed. We use a real-time recurrent learning (RTRL) algorithm [46] for the adaptation in our mobile navigation application, although

other adaptation methods can also be used in this phase. In our approach, the adaptation of the consequent parameters occurs in the least squares sense, based on the squared error between the output of each rule and the output  $r$  of its simplified version, i.e., it is based on  $e = (r - r')^2$ . Furthermore, we apply a heuristic for rule replacement in the case of conflicting rules. After reducing the rule base of the mobile robot controller by using inclusion-based fuzzy clustering, the online adaptation algorithm RTRL is applied to optimize the performance of the mobile robot. Simultaneously, a procedure for dealing with conflict situations has been performed to guarantee successful navigation. As a result of this optimization, 11 rules have been deleted from the rule base. For example, when we consider the Rules 1 and 2 that were discussed before, the change of the consequent parameters as a result of the RTRL algorithm is as follows.

#### Phase I

- Rule 1: IF  $d_1$  is near (0.1117, 0.1144) and  $d_2$  is near (0.9290, 0.1364), THEN  $\theta$  is 0.01.
- Rule 2: IF  $d_1$  is near (0.8511, 0.1150) and  $d_2$  is near (0.0158, 0.1235), THEN  $\theta$  is  $-0.68$ .

#### Phase II

- Rule 1: IF  $d_1$  is near (0.1007, 0.3020) and  $d_2$  is near (0.8911, 0.2900), THEN  $\theta$  is 0.17.
- Rule 2: IF  $d_1$  is near (0.8421, 0.1607) and  $d_2$  is near (0.0904, 0.3113), THEN  $\theta$  is  $-0.68$ .

#### Phase III

- Rule 1: IF  $d_1$  is near (0.1007, 0.3020) and  $d_2$  is near (0.8911, 0.2900), THEN  $\theta$  is 0.17.
- Rule 2: IF  $d_1$  is near (0.8421, 0.1607) and  $d_2$  is near (0.0904, 0.3113), THEN  $\theta$  is  $-0.43$ .

To illustrate the performance of the robot as a result of our proposed training and simplification method, we have conducted a typical experiment. The robot has been instructed to follow a path with obstacles, and its response has been tested after the three phases discussed previously. Table III lists the number of fuzzy sets and the number of rules in the rule base. It also shows the value of the performance index obtained as the sum of squared errors. After the learning in Phase I, the fuzzy system has 25 rules and 50 fuzzy sets in total. The performance index is 0.0012, which indicates a very satisfactory result. After the rule base simplification in Phase II, there are still 25 rules in the rule base, but the total number of fuzzy sets is reduced to eight. There are now conflicting rules as they share the same premise but have different consequents. Because of the conflicting rules and the nonoptimal values of the consequents, the performance index increases to 0.0092 indicating a loss of performance. After the conflict resolution and real-time consequent adaptation in Phase III, the number of rules is reduced to 14. The number of fuzzy sets in the rule base is now eight and the performance index is 0.0057. This corresponds to a small decrease in performance compared to the initial learned rule base, while the fuzzy system is simplified significantly. Fig. 7 shows the paths followed by the robot after each of the three phases.

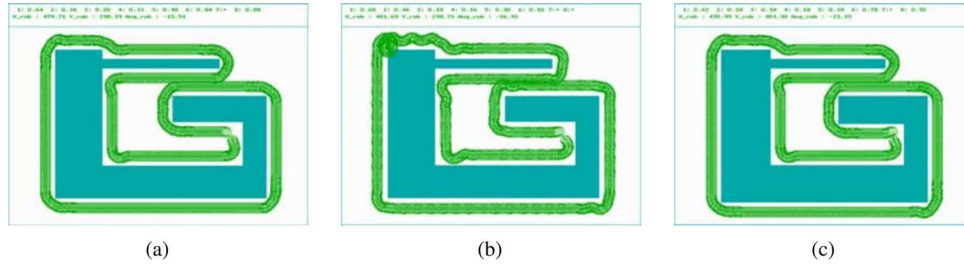
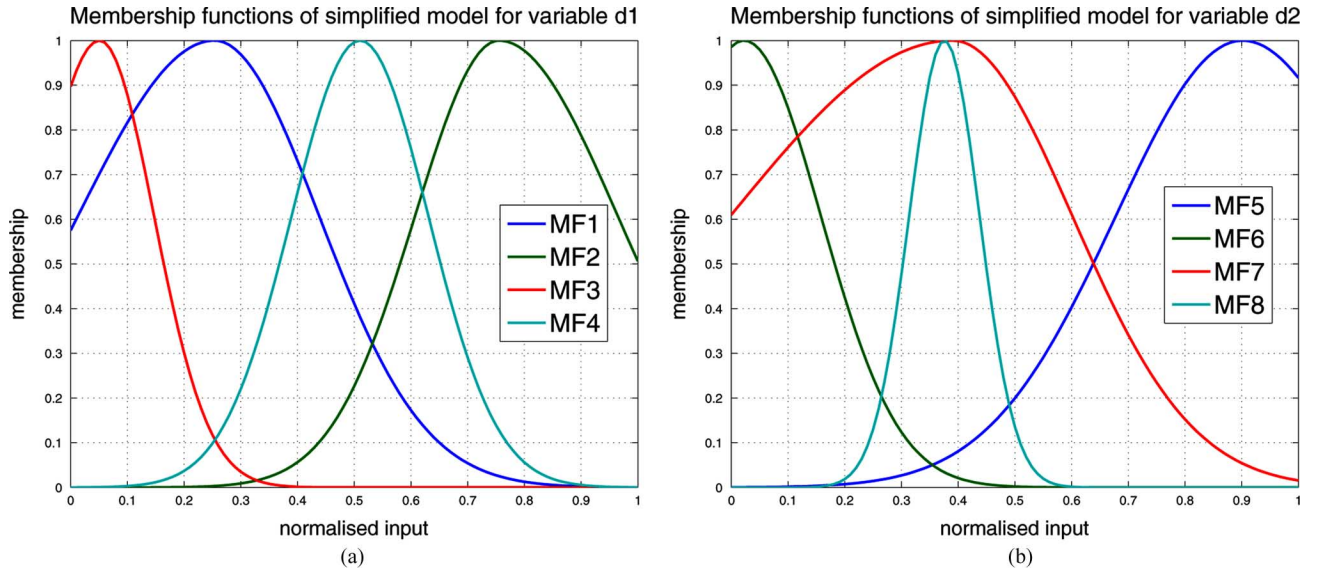


Fig. 7. Paths followed by the mobile robot after (a) Phase I, (b) Phase II, and (c) Phase III.

TABLE III  
PERFORMANCE OF THE FUZZY CONTROLLER DURING THE THREE PHASES OF MODEL CONSTRUCTION

	Number of membership func.		Number	Performance
	$d_1$	$d_2$	of rules	index
Phase I	25	25	25	0.0012
Phase II	4	4	25	0.0092
Phase III	4	4	14	0.0057

Fig. 8. Simplified fuzzy sets for the sensor (a)  $d_1$  and (b)  $d_2$  after using Setnes' approach.

### B. Comparison With Setnes' Approach

Setnes' algorithm [1] is an alternative approach to merging similar fuzzy sets and simplifying the rule base. In this section, we compare our approach with Setnes' algorithm and analyze three important issues in fuzzy modeling: accuracy, complexity, and interpretability.

1) *Accuracy*: Consider again the same initial rule base that has been generated from Phase I. After applying Setnes' method, the membership functions of the simplified model for  $d_1$  and  $d_2$  are shown in Fig. 8. Part of the initial rule base looks like as in the following after applying Setnes' algorithm and reestimating the consequence parameters.

- Rule 1: IF  $d_1$  is MF1 and  $d_2$  is MF5, THEN  $\theta$  is 0.015.  
 Rule 2: IF  $d_1$  is MF2 and  $d_2$  is MF6, THEN  $\theta$  is  $-0.680$ .  
 Rule 3: IF  $d_1$  is MF2 and  $d_2$  is MF7, THEN  $\theta$  is  $-0.386$ .

Rule 4: IF  $d_1$  is MF1 and  $d_2$  is MF7, THEN  $\theta$  is 0.525.

Rule 5: IF  $d_1$  is MF3 and  $d_2$  is MF7, THEN  $\theta$  is 0.395.

Table IV shows the comparison of the performance of the controller in terms of average squared error before and after simplification of the rule base using both our and Setnes' method in three different experiments.

Table V compares both approaches in terms of the output of the fuzzy system in two different experiments. We run the first experiment with four fuzzy sets for each of the input variables, and we run the second experiment with two fuzzy sets per variable. The threshold parameters of Setnes' algorithm were selected such that the resulting number of fuzzy sets after the merger was the same as the number of fuzzy sets from the approach proposed in this paper. The results show that inclusion-based fuzzy clustering performs better than Setnes' method when performance is quantified in terms of the squared error (see, also, Fig. 9).



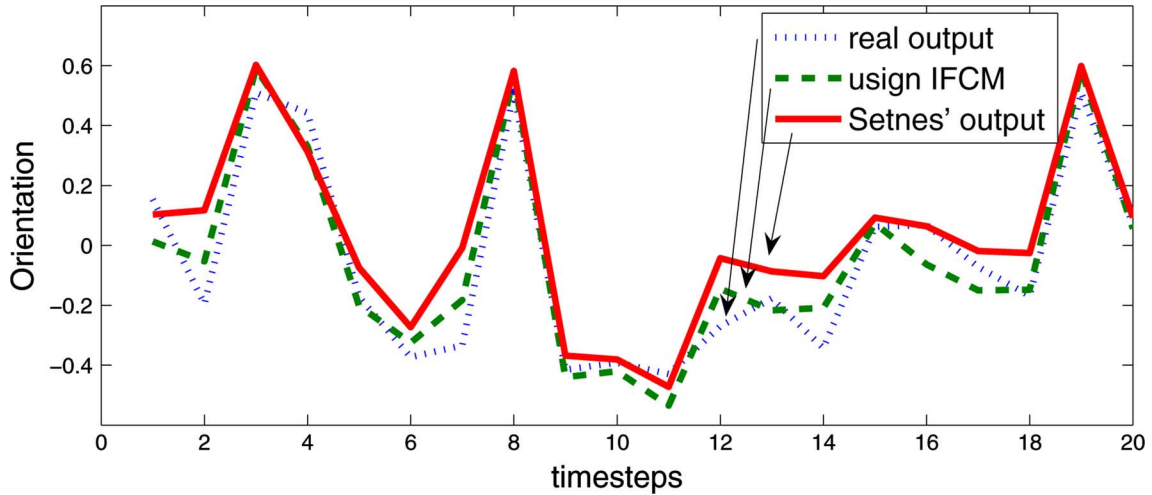


Fig. 9. Real and the predicted outputs using both approaches in experiment 1.

TABLE IV  
PERFORMANCE OF THE FUZZY CONTROLLER BEFORE  
AND AFTER SIMPLIFICATION

Method	Experiment	Nbr of rules	Nbr of MbF $d_1$	Nbr of MbF $d_2$	Perf. Index
Inclusion Fuzzy Clustering	Before simplification	25	25	25	0.0012
	1	14	4	4	0.0092
	2	4	2	2	0.0360
	3	16	4	5	0.0083
Setnes	Before simplification	25	25	25	0.0012
	1	10	4	4	0.0145
	2	4	2	2	0.0450
	3	10	4	5	0.0110

2) *Complexity*: As shown in Table IV, our proposed method reduces the number of rules from 25 to 14 in the first experiment, and to four in the second experiment. In comparison, Setnes' method reduces the number of rules from 25 to ten in the first experiment, and to four in the second experiment. We observe that Setnes' method tends to produce smaller number of rules for a given number of fuzzy sets. However, the accuracy of the model is worse than the accuracy of the model obtained with our approach. Note also that the use of Setnes' approach requires determining all pairwise similarity values and the user needs to set a threshold  $\lambda$  above which two membership functions are considered to be similar. The computation of all pairwise similarity values has some computational burden, and the results are sensitive to the choice of the similarity threshold value  $\lambda$ . Additionally, the intransitive nature of the threshold-based similarity merging may undermine its performance and lead to a counter-intuitive result. The reader should also note that Setnes' method does not allow explicit control of the number of membership functions. Hence, the tuning of the thresholds can be very tedious if a desired model complexity is required. In general, the optimal values of the similarity threshold  $\lambda$  and the set threshold  $\gamma$  have to be determined experimentally in Setnes' approach. The complexity of a simplified model can be controlled more explicitly in our proposed approach by specifying the number of clusters in the inclusion-based clustering step, which is an advantage over Setnes' approach.

TABLE V  
OUTPUT OF THE CONTROLLER WITH FOUR AND TWO  
FUZZY SETS PER VARIABLE

Number of fuzzy sets	Desired output	Output using proposed method	Output using Setnes' method
4	-0.1952	-0.0530	0.1173
	0.4425	0.3301	0.3131
	-0.3341	-0.1826	-0.0085
	-0.2701	-0.1452	-0.0429
	-0.3430	-0.2086	-0.1029
2	-0.1000	-0.1385	-0.3928
	0.2600	0.2337	0.3594
	-0.3000	-0.0349	-0.0367
	-0.1000	-0.1319	-0.3990
	-0.1500	0.1963	0.0720

3) *Interpretability*: In order to compare the interpretability of the models obtained from the two rule base simplification approaches, we have studied the fuzzy sets obtained, respectively, from the Setnes' approach and the proposed approach for the variable  $d_2$ . These fuzzy sets are shown in Figs. 10 and 11 for two sets of experiments.

In general, the membership functions obtained from Setnes' approach are nonsymmetric because of the sequential nature of the merging, even though the similarity measure used is symmetric (Jaccard index). In the inclusion-based clustering approach, the membership functions in the simplified model have the same shape (in this case, symmetric Gaussians) as the fuzzy sets in the complex model, since the simplified fuzzy sets are parameterized in the same way as the initial fuzzy sets. Note also that some of the classes obtained using Setnes' method can be included in another one. Considering the case of Fig. 10(a), for example, MF8 is a subset of MF7. Similarly, MF9 is a subset of MF7 in Fig. 11(a). The presence of subsets in the model decreases its interpretability. It is not possible in our proposed approach to obtain fuzzy sets that are included in another. Table VI summarizes the comparison between the proposed rule base simplification approach and Setnes' approach in terms of complexity, accuracy and interpretability of the simplified models.

## V. DISCUSSION AND CONCLUSION

In this paper, we have proposed a new merging method based on clustering of fuzzy sets in the parameters space of mem-

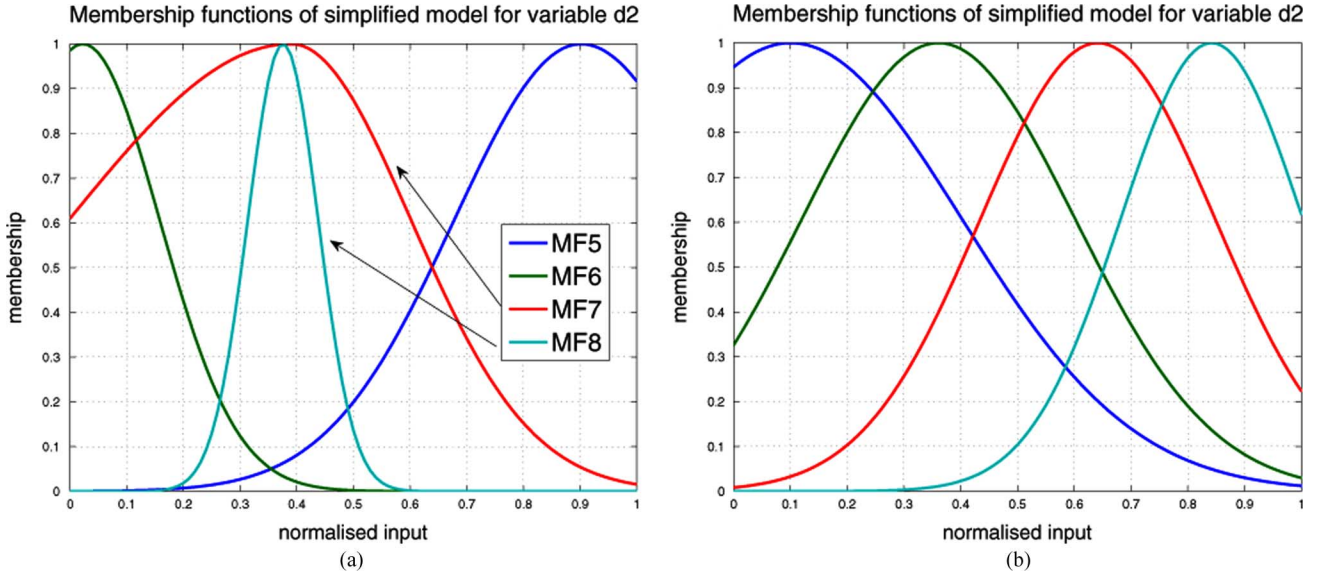


Fig. 10. Fuzzy sets obtained for variable  $d_2$  using (a) Setnes' method and (b) the proposed method for experiment 1.

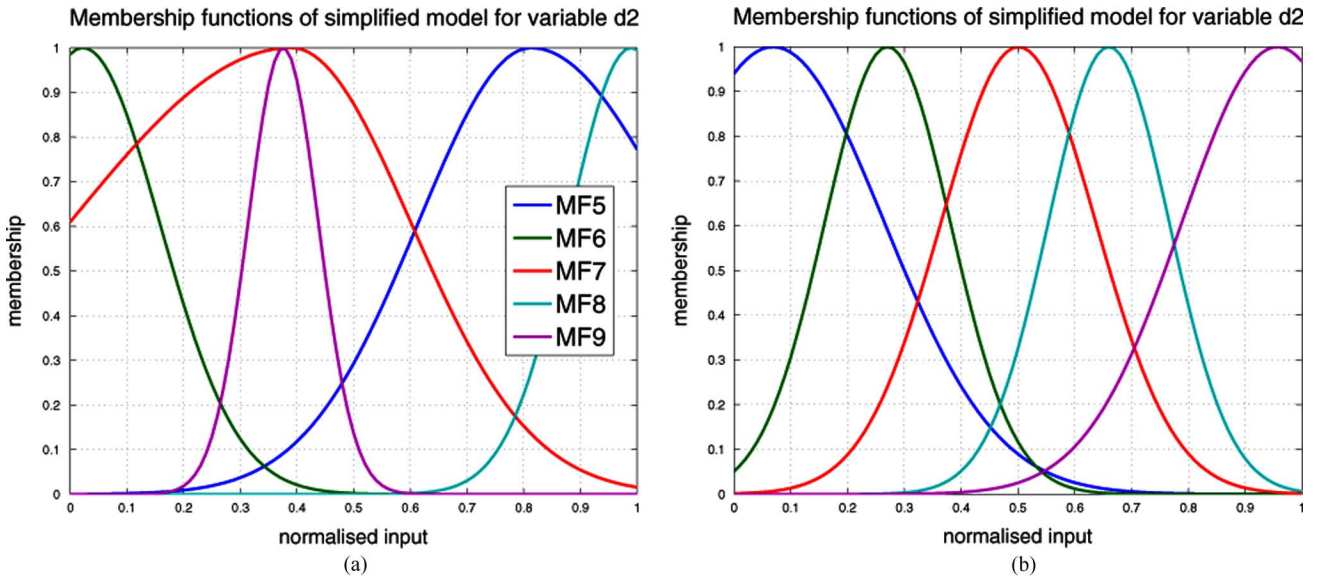


Fig. 11. Fuzzy sets obtained for variable  $d_2$  using (a) Setnes' method and (b) the proposed method for experiment 3.

TABLE VI  
COMPARISON BETWEEN TWO RULE BASE SIMPLIFICATION METHODS

method	Complexity		Accuracy	Interpre- tability
	Pre-proc. need	Number of rules and membership func.		
Setnes'	High	Small	Medium	Fair
Proposed	Low	Medium	High	Good

bership functions, taking into account the degree of inclusion between the cluster prototypes and the fuzzy sets. The originality of our approach resides in using fuzzy clustering for determining fuzzy sets that are compatible with one another in feature space. We have applied our inclusion-based fuzzy clustering algorithm for simplifying fuzzy rule bases that are constructed in a data-driven manner. The advantage of the proposed algorithm is that it allows explicit control over the desired complexity of

the simplified model and avoids the problem of sensitivity to the order of merging, which is present in similarity-based simplification approaches that are proposed earlier in the literature.

In the proposed approach, the fuzzy sets are replaced by the most compatible prototypical fuzzy set that is determined from the inclusion-based clustering algorithm. After reducing the number of fuzzy sets, rules with the same antecedents can be merged for conflict resolution, and the performance of the simplified system can be improved by reestimating the rule consequents. We applied the proposed algorithm successfully to mobile robot navigation. The results indicate that our strategy of merging can generate a compact and interpretable fuzzy model, which is in agreement with expert reasoning. These results are also supported by our performance comparison of the proposed method with another rule base simplification approach based on similarity-driven merging of fuzzy sets.

A number of extensions of our work can be considered. In future work, we will study the behavior of the inclusion-based clustering algorithm for other standard modeling tasks, such as classification. In this context, the extension of inclusion-based clustering for dealing with noise may be of interest. Another interesting problem is determining the optimal number of fuzzy sets in the simplified model by using cluster validity and/or cluster merging techniques, without using other means, such as the expert knowledge about the process.

## APPENDIX

### A. Proof of Proposition 1

Minimizing

$$J = - \sum_{j=1}^c \sum_{i=1}^n Id_3(G_i, G_{v_j}) u_{ij}^\alpha + w \sum_{j=1}^c \sum_{i=1}^n (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{A}(\mathbf{x}_i - \mathbf{v}_j) u_{ij}^\alpha$$

subject to

$$\sum_{j=1}^c u_{ij} = 1$$

leads to the following augmented Lagrangian:

$$\begin{aligned} J(\mathbf{U}, \mathbf{V}, \beta) = & - \sum_{j=1}^c \sum_{i=1}^n Id_3(G_i, G_{v_j}) u_{ij}^\alpha \\ & + w \sum_{j=1}^c \sum_{i=1}^n (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{A}(\mathbf{x}_i - \mathbf{v}_j) u_{ij}^\alpha \\ & + \sum_{i=1}^n \beta_i \left( \sum_{j=1}^c u_{ij} - 1 \right). \end{aligned} \quad (24)$$

Substituting the  $Id_3$  expression (14) into (24) leads to

$$\begin{aligned} J(\mathbf{U}, \mathbf{V}, \beta) = & - \frac{1}{4} \sum_{j=1}^c \sum_{i=1}^n (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} S_{ij} H(S_{ij}) u_{ij}^\alpha \\ & + w \sum_{j=1}^c \sum_{i=1}^n (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{A}(\mathbf{x}_i - \mathbf{v}_j) u_{ij}^\alpha \\ & + \sum_{i=1}^n \beta_i \left( \sum_{j=1}^c u_{ij} - 1 \right) \end{aligned} \quad (25)$$

with  $S_{ij}$  defined as in (15). The necessary conditions for optimality are found by setting the derivatives of  $J$  with respect to its parameters to zero. Therefore, provided  $S_{ij} \neq 0$ , which ensures the derivability of the step function  $H(S_{ij})$

$$\begin{aligned} \frac{\partial J}{\partial u_{ij}} = & - \frac{\alpha}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} S_{ij} H(S_{ij}) u_{ij}^{\alpha-1} \\ & + w \alpha (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{A}(\mathbf{x}_i - \mathbf{v}_j) u_{ij}^{\alpha-1} \\ & + \beta_i = 0 \end{aligned} \quad (26)$$

$$\frac{\partial J}{\partial \beta_i} = \sum_{j=1}^c u_{ij} = 0 \quad (27)$$

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{v}_j} = & -2w \sum_{i=1}^n \mathbf{A}(\mathbf{x}_i - \mathbf{v}_j) u_{ij}^\alpha \\ & - \frac{1}{4} \sum_{i=1}^n (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} (\mathbf{B}_2(\mathbf{x}_i \\ & + \mathbf{v}_j) + \mathbf{B}_1(\mathbf{x}_i - \mathbf{v}_j)) H(S_{ij}) u_{ij}^\alpha = 0. \end{aligned} \quad (28)$$

Equation (28) is obtained by noting that the derivative of the step function  $H(x)$  with respect to variable  $x$  is the Dirac function  $\delta(x)$ , which is zero valued for all nonzero values of  $x$ . Consequently, the result (28) is obtained.

Equation (26) entails

$$u_{ij} = \left[ \frac{\beta_i}{\frac{\alpha}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} S_{ij} H(S_{ij}) + w \alpha d_{ij}^2} \right]^{\frac{1}{\alpha-1}} \quad (29)$$

with  $d_{ij}^2 = (\mathbf{x}_i - \mathbf{v}_j)^T \mathbf{A}(\mathbf{x}_i - \mathbf{v}_j)$ . Now, using (27), (29) is equivalent to

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left[ \frac{-\frac{1}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} S_{ik} H(S_{ik}) + w \alpha d_{ik}^2}{-\frac{1}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} S_{ij} H(S_{ij}) + w \alpha d_{ij}^2} \right]^{\frac{1}{\alpha-1}}}. \quad (30)$$

In order to determine the prototype vector  $\mathbf{v}_j$ , (28) can be rewritten as

$$\begin{aligned} & \left[ \sum_{i=1}^n \frac{1}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} (\mathbf{B}_1 - \mathbf{B}_2) H(S_{ij}) + 2w \mathbf{A} u_{ij}^\alpha \right] \mathbf{v}_j \\ & = \sum_{i=1}^n \left[ \frac{1}{4} (\mathbf{x}_i^T \mathbf{B}_2 \mathbf{x}_i)^{-1} (\mathbf{B}_1 + \mathbf{B}_2) \mathbf{x}_i H(S_{ij}) u_{ij}^\alpha + 2w \mathbf{A} \mathbf{x}_i u_{ij}^\alpha \right]. \end{aligned}$$

Therefore

$$\mathbf{E} \mathbf{v}_j = \mathbf{F} \quad (31)$$

where  $\mathbf{E}$  and  $\mathbf{F}$  are defined by (19) and (20), respectively.

### B. Proof of Proposition 2

Let  $\mathbf{x}_i = \begin{pmatrix} m_i \\ \sigma_i \end{pmatrix}$  and  $\mathbf{v}_j = \begin{pmatrix} m_{v_j} \\ \sigma_{v_j} \end{pmatrix}$ . Then, we can write the membership value  $u_{ij}$  as

$$u_{ij} = \frac{1}{\left[ \frac{4w d_{ik}^2 - ((\sigma_i + \sigma_{v_k})^2 - (m_i - m_{v_k})^2) H_{ik}}{4w d_{ij}^2 - ((\sigma_i + \sigma_{v_j})^2 - (m_i - m_{v_j})^2) H_{ij}} \right]^{\frac{1}{\alpha-1}}} \quad (32)$$

where  $d_{ij}^2 = (\sigma_i - \sigma_{v_j})^2 + (m_i - m_{v_j})^2$  and

$$H_{ik} = H((\sigma_i + \sigma_{v_k})^2 - (m_i - m_{v_k})^2). \quad (33)$$

Note that if each term of the sum used in the denominator of the right-hand side of (32) is positively valued, then the overall sum, and thereby,  $u_{ij}$  is also positively valued. In other words,  $w$  is chosen such that

$$4w d_{ik}^2 - ((\sigma_i + \sigma_{v_k})^2 - (m_i - m_{v_k})^2) H_{ik} \geq 0. \quad (34)$$

For example

$$w \geq \max_{i,k} \left\{ \frac{((\sigma_i + \sigma_{v_k})^2 - (m_i - m_{v_k})^2)H_{ik}}{4((\sigma_i - \sigma_{v_k})^2 + (m_i - m_{v_k})^2)} \right\}. \quad (35)$$

Since  $H_{ik}$  is binary valued

$$w \geq \max_{i,k} \left\{ \frac{((\sigma_i + \sigma_{v_k})^2 - (m_i - m_{v_k})^2)}{4((\sigma_i - \sigma_{v_k})^2 + (m_i - m_{v_k})^2)} \right\}. \quad (36)$$

Given that all entities  $m_i, \sigma_i, m_{v_k}$ , and  $\sigma_{v_k}$  lie within the unit interval, the maximum of the right-hand side of (36) is reached when  $(m_i - m_{v_k})$  vanishes while the expression in the numerator is maximized and that of the denominator is minimized, which holds if  $\sigma_{v_k}$  vanishes as well. Consequently, (36) entails

$$w \geq \frac{\max_i \sigma_i^2}{4 \min_i \sigma_i^2}. \quad (37)$$

Note that the lower bound mentioned in the inequality (37)) is far from a conservative bound because neither  $(m_i - m_{v_k})$  nor  $\sigma_{v_k}$  vanishes in practice, and also, the switch from (35) to (36) is not optimal as the evaluations  $H_{ik}$  are always set to their maximal values.

### C. Proof of Proposition 3

Using a similar notation as in the previous section, (19) and (20) are equivalent to

$$\mathbf{E} = \begin{pmatrix} \sum_{i=1}^n \frac{H_{ij} + 2w}{4\sigma_i^2} u_{ij}^\alpha & 0 \\ 0 & \sum_{i=1}^n \frac{2w - H_{ij}}{4\sigma_i^2} u_{ij}^\alpha \end{pmatrix} \quad (38)$$

and

$$\mathbf{F} = \begin{pmatrix} \sum_{i=1}^n \frac{m_i H_{ij} + 2w m_i}{4\sigma_i^2} u_{ij}^\alpha \\ \sum_{i=1}^n \frac{H_{ij} + 2w}{4\sigma_i} u_{ij}^\alpha \end{pmatrix}. \quad (39)$$

Consequently, solving for  $\mathbf{v}_j$ , the matrix equation  $\mathbf{E}\mathbf{v}_j = \mathbf{F}$  leads to

$$\mathbf{v}_j = \begin{pmatrix} \frac{\sum_{i=1}^n \frac{m_i H_{ij} + 2w m_i}{4\sigma_i^2} u_{ij}^\alpha}{\sum_{i=1}^n \frac{H_{ij} + 2w}{4\sigma_i^2} u_{ij}^\alpha} \\ \frac{\sum_{i=1}^n \frac{H_{ij} + 2w}{4\sigma_i} u_{ij}^\alpha}{\sum_{i=1}^n \frac{2w - H_{ij}}{4\sigma_i} u_{ij}^\alpha} \end{pmatrix}. \quad (40)$$

Regarding the first component of  $\mathbf{v}_j$ , note that

$$\sum_{i=1}^n \frac{m_i H_{ij} + 2w m_i}{4\sigma_i^2} u_{ij}^\alpha \leq \sum_{i=1}^n \frac{H_{ij} + 2w}{4\sigma_i^2} u_{ij}^\alpha \quad (41)$$

always holds for all  $i = 1, \dots, n$  and  $j = 1, \dots, c$  regardless the value of  $w$  since  $m_i \leq 1$ . Consequently, the mean component of the prototype vector always lies within the unit interval as soon as the mean components of the data set lie within the unit interval, regardless of  $w$  values. This result is somehow similar to

FCM clustering where the prototypes are bounded by the lower and the upper bound of the data set.

For the second component of  $\mathbf{v}_j$ , we have

$$\sum_{i=1}^n \frac{H_{ij} + 2w}{4\sigma_i} u_{ij}^\alpha \leq \sum_{i=1}^n \frac{2w - H_{ij}}{4\sigma_i^2} u_{ij}^\alpha \quad (42)$$

for all  $i = 1, \dots, n$  and  $j = 1, \dots, c$ , which is equivalent to

$$w \left( \sum_{i=1}^n u_{ij}^\alpha \left( \frac{1}{\sigma_i^2} - \frac{1}{\sigma_i} \right) \right) \geq \sum_{i=1}^n \frac{H_{ij} u_{ij}^\alpha}{2} \left( \frac{1}{\sigma_i^2} - \frac{1}{\sigma_i} \right). \quad (43)$$

Hence

$$w \geq \frac{\sum_{i=1}^n H_{ij} u_{ij}^\alpha \left( \frac{1}{\sigma_i^2} - \frac{1}{\sigma_i} \right)}{\left( 2 \sum_{i=1}^n u_{ij}^\alpha \left( \frac{1}{\sigma_i^2} - \frac{1}{\sigma_i} \right) \right)}. \quad (44)$$

The maximum of the right-hand side of (44) is reached when  $H_{ij} = 1$ , which forces the inequality to  $w \geq (1/2)$ . Since the condition (42) ensures that the spread component of  $\mathbf{v}_j$  lies within unit interval, the latter condition is equivalent to (44). Similar to (36), the lower bound (44) is not the most conservative since the hypothesis of  $H_{ij} = 1$  for all  $i, j$  is not very realistic. Consequently, (44) is only a sufficient condition but not necessary.

### ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable comments that helped improve the content, presentation, and quality of this paper.

### REFERENCES

- [1] M. Setnes, R. Babuška, U. Kaymak, and H. R. van Nauta Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 376–386, Jun. 1998.
- [2] S. Guillaume, "Designing fuzzy inference systems from data: An interpretability-oriented review," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 3, pp. 426–443, Jun. 2001.
- [3] V. V. Cross, "An analysis of fuzzy set aggregators and compatibility measures," Ph.D. dissertation, Dept. Comp. Sci. Eng., Wright State Univ., OH, 1993.
- [4] M. Setnes, "Fuzzy rule-base simplification using similarity measures," M.Sc. thesis, Control Lab., Faculty Electr. Eng., Delft Univ. Technol., Delft, The Netherlands, Jul. 1995.
- [5] R. Zwick, E. Carlstein, and D. V. Budescu, "Measures of similarity among fuzzy concepts: A comparative analysis," *Int. J. Approx. Reason.*, vol. 1, pp. 221–242, 1987.
- [6] U. Kaymak and M. Setnes, "Fuzzy clustering with volume prototypes and adaptive cluster merging," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 6, pp. 705–712, Dec. 2002.
- [7] U. Kaymak and R. Babuška, "Compatible cluster merging for fuzzy modelling," in *Proc. 4th IEEE Int. Conf. Fuzzy Syst.*, Yokohama, Japan, Mar. 1995, vol. 2, pp. 897–904.
- [8] C.-L. Chang, "Finding prototypes for nearest neighbor classifiers," *IEEE Trans. Comput.*, vol. C-23, no. 11, pp. 1179–1184, Nov. 1974.
- [9] J. C. Bezdek, T. R. Reichherzer, G. S. Lim, and Y. Attikiouzel, "Multiple-prototype classifier design," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 28, no. 1, pp. 67–79, Feb. 1998.
- [10] R. A. Mollineda, F. J. Ferri, and E. Vidal, "An efficient prototype merging strategy for the condensed 1-NN rule through class-conditional hierarchical clustering," *Pattern Recognit.*, vol. 35, no. 12, pp. 2771–2782, Dec. 2002.

- [11] S. Nefti, M. Oussalah, K. Djouani, and J. Pontnau, "Clustering and adaptive robot navigation in an unknown environment," in *Proc. IEEE Int. Conf. Intell. Syst. (INES)*, Vienna, Austria, Sep. 1998, pp. 1998–2005.
- [12] S. Helmer, "Evaluating different approaches for indexing fuzzy sets," *Fuzzy Sets Syst.*, vol. 140, no. 1, pp. 167–182, 2003.
- [13] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Upper Saddle River, NJ: Prentice-Hall, 1997.
- [14] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, NJ: Prentice-Hall, 1995.
- [15] Y. Shi, M. Mizumoto, N. Yubazaki, and M. Otani, "A learning algorithm for tuning fuzzy rules based on the gradient descent method," in *Proc. 5th IEEE Int. Conf. Fuzzy Syst.*, New Orleans, LA, Sep. 1996, pp. 55–61.
- [16] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
- [17] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE Conf. Decision Control*, San Diego, CA, 1979, pp. 761–766.
- [18] J. C. Bezdek, C. Coray, R. Gunderson, and J. Watson, "Detection and characterization of cluster substructure, I. linear structure: Fuzzy c-lines," *SIAM J. Appl. Math.*, vol. 40, no. 2, pp. 339–357, 1981.
- [19] J. C. Bezdek, C. Coray, R. Gunderson, and J. Watson, "Detection and characterization of cluster substructure, II. fuzzy c-varieties and convex combinations thereof," *SIAM J. Appl. Math.*, vol. 40, no. 2, pp. 358–372, 1981.
- [20] R. N. Dave, "Fuzzy shell-clustering and applications to circle detection in digital images," *Int. J. General Syst.*, vol. 16, no. 4, pp. 343–355, 1990.
- [21] R. N. Dave, "Characterization and detection of noise in clustering," *Pattern Recognit. Lett.*, vol. 12, no. 11, pp. 657–664, Nov. 1991.
- [22] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, May 1993.
- [23] D. Dubois and H. M. Prade, *Fuzzy sets and systems: Theory and applications*, ser. Mathematics in Science and Engineering. New York: Academic, 1980, vol. 144.
- [24] D. Dubois and H. M. Prade, Eds., *Fundamentals of Fuzzy Sets*, ser. The Handbooks of Fuzzy Sets Series. Dordrecht, The Netherlands: Kluwer, 2000, vol. 7.
- [25] P. E. Gill, W. Murray, and M. Wright, *Practical Optimization*. New York: Academic, 1981.
- [26] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 450–465, May 1999.
- [27] D. Dubois and H. Prade, *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. New York: Plenum, 1988.
- [28] Y. C. Jin, "Fuzzy modeling of high-dimensional systems: Complexity reduction and interpretability improvement," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 2, pp. 212–221, Apr. 2000.
- [29] R. Mikut, J. Jäkel, and L. Gröll, "Interpretability issues in data-based learning, of fuzzy systems," *Fuzzy Sets Syst.*, vol. 150, no. 2, pp. 179–197, 2005.
- [30] L. X. Wang, "Design and analysis of fuzzy identifiers of nonlinear dynamic-systems," *IEEE Trans. Autom. Control*, vol. 40, no. 1, pp. 11–23, Jan. 1995.
- [31] M. Setnes, "Supervised fuzzy clustering for rule extraction," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Seoul, South Korea, Aug. 1999, vol. 3, pp. 1270–1274.
- [32] M. Setnes, R. Babuška, and H. B. Verbruggen, "Rule-based modeling: Precision and transparency," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 28, no. 1, pp. 165–169, Feb. 1998.
- [33] H. Roubos and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 516–524, Aug. 2001.
- [34] H. L. Wang, S. Kwong, Y. C. Jin, W. Wei, and K. F. Man, "Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction," *Fuzzy Sets Syst.*, vol. 149, no. 1, pp. 149–186, 2005.
- [35] J. V. Ramos and A. Dourado, "On-line interpretability by fuzzy rule base simplification and reduction," in *Proc. Eur. Symp. Intell. Technol., Hybrid Syst., Implement. Smart Adapt. Syst. (EUNITE)*, Aachen, Germany, Jun. 2004, pp. 430–422.
- [36] P. D'Urso, "Fuzzy clustering for data time arrays with inlier and outlier time trajectories," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 5, pp. 583–604, Oct. 2005.
- [37] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, Nov./Dec. 1992.
- [38] J. Zhang, "Modeling and optimal control of batch processes using recurrent neuro-fuzzy networks," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 417–427, Aug. 2005.
- [39] F. Hofman and G. Pfister, F. Herrera and J. L. Verdegay, Eds., "Learning of a fuzzy control rule base using messy genetic algorithm," in *Genetic Algorithms and Soft Computing*, ser. Studies in Fuzziness and Soft Computing. Berlin, Germany: Physica-Verlag, 1996, pp. 279–305, no. 8.
- [40] C. H. Chou, "Genetic algorithm-based optimal fuzzy controller design in the linguistic space," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 3, pp. 372–385, Jun. 2006.
- [41] J. Casillas, O. Cordon, M. J. del Jesus, and F. Herrera, "Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 1, pp. 13–29, Feb. 2005.
- [42] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 301–315, Jun. 1998.
- [43] M. Y. Chen and D. A. Linkens, "Rule-base self-generation and simplification for data-driven fuzzy models," *Fuzzy Sets Syst.*, vol. 142, no. 2, pp. 243–265, 2004.
- [44] H. Ishibuchi, T. Murata, and S. I. B. Türk, "Single-objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems," *Fuzzy Sets Syst.*, vol. 89, no. 2, pp. 135–150, 1997.
- [45] S. Nefti, "Approche neuro-floue pour la modelisation et la commande des systèmes nonlinéaires multi-variables: Application à la robotique," Ph.D. dissertation, Dept. Comp. Sci., Univ. Paris XII, Paris, France, Jul. 1999.
- [46] S. Nefti, M. Oussalah, K. Djouani, and J. Pontnau, "Intelligent adaptive mobile robot navigation," *J. Intell. Robot. Syst.*, vol. 30, no. 4, pp. 311–329, Apr. 2001.



**Samia Nefti** (M'04) received the M.S. degree in electrical engineering, D.E.A. degree in automatic control, and the Ph.D. degree in robotics and artificial intelligence from the University of Paris XII, Paris, France, in 1992, 1994, and 1998, respectively.

In November 1999, she joined Liverpool University, Liverpool, U.K., as a Senior Research Fellow working on European research project Occupational Therapy Internet School (OTIS). Afterwards, she has been involved in the U.K. Engineering and Physical Sciences Research Council (EPSRC) on Railway project with Manchester Metropolitan University, Manchester, U.K., where her work was concerned mainly with model-based predictive control, modeling, and decision making. Currently, she is a Lecturer at Computer Science Department, University of Salford, Salford, U.K. Her current research is focused on fuzzy and neural-fuzzy clustering, neurofuzzy modeling, and cognitive behavior modeling in the area of robotics.

Dr. Nefti is a full Member of the Informatics Research Institute (IRIS), a Chartered Member of the British Computer Society, and the IEEE Computer Society. She is a member of the international program committees of several conferences and is an active member of the European Network for Advancement of Artificial Cognition Systems.



**Mourad Oussalah** (M'97) received the M.S. degree in electronics from the Polytechnic National Institute of Algiers, Algiers, Algeria, in 1992, the D.E.A. degree in robotics and automation from the University of Paris XII, Paris, France, in 1994, and the Ph.D. degree in robotics and data fusion from the University of Evry, Val Essonne, France, in 1998.

He worked as an Assistant Professor from 1996 to 1998 at Evry University. In October 1998, he joined Katholieke Universiteit Leuven, Leuven, Belgium, as a Research Fellow in the project "Active Sensing for Intelligent Machines" (1999–2004), where he applied new artificial intelligence



tools to robotics tasks. In January 2001, he joined Centre for Software Reliability as Senior Research Fellow to work on software dependability within the Interdisciplinary Research Collaboration in Dependability project. Since 2003, he has been lecturing at the University of Birmingham, Birmingham, U.K. His topics of interests include data fusion, reliability analysis, uncertainty handling in the framework of probability, possibility and evidence theories, random sets, and robotics, in which he published more than 100 technical papers.



**Uzay Kaymak** (S'94–M'98) received the M.Sc. degree in electrical engineering, the Degree of Chartered Designer in information technology, and the Ph.D. degree in control engineering from the Delft University of Technology, Delft, The Netherlands, in 1992, 1995, and 1998, respectively.

From 1997 to 2000, he was a Reservoir Engineer with Shell International Exploration and Production. He is currently an Associate Professor of Economics and Computer Science at the Econometric Institute, Erasmus University, Rotterdam, The Netherlands.

His current research interests include fuzzy decision making, multiagent modeling, and adaptive networks in logistics.

Dr. Kaymak is a member of the Erasmus Research Institute on Management and the Dutch School for Information and Knowledge Systems. He serves on various technical committees of the IEEE Computational Intelligence Society and the IEEE Systems, Man and Cybernetics Society. He is also an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS and is a member of the editorial board of *Fuzzy Sets and Systems*.