

# Similarity Measures in Fuzzy Rule Base Simplification

Magne Setnes, Robert Babuška, Uzay Kaymak, and Hans R. van Nauta Lemke

**Abstract**—In fuzzy rule-based models acquired from numerical data, redundancy may be present in the form of similar fuzzy sets that represent compatible concepts. This results in an unnecessarily complex and less transparent linguistic description of the system. By using a measure of similarity, a rule base simplification method is proposed that reduces the number of fuzzy sets in the model. Similar fuzzy sets are merged to create a common fuzzy set to replace them in the rule base. If the redundancy in the model is high, merging similar fuzzy sets might result in equal rules that also can be merged, thereby reducing the number of rules as well. The simplified rule base is computationally more efficient and linguistically more tractable. The approach has been successfully applied to fuzzy models of real world systems.

**Index Terms**—Fuzzy modeling, fuzzy sets, rule base reduction, rule base simplification, similarity measure.

## I. INTRODUCTION

FUZZY modeling is one of the techniques currently being used for modeling of nonlinear, uncertain, and complex systems. An important characteristic of fuzzy models 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. A fuzzy model typically consists of a rule base with a rule for each particular region. Fuzzy transitions between these rules allow for the modeling of complex nonlinear systems with a good global accuracy. One of the aspects that distinguish fuzzy modeling from other black-box approaches like neural nets is that fuzzy models are *transparent* to interpretation and analysis (to a certain degree).

However, the transparency of a fuzzy model is not achieved automatically. A system can be described with a few rules using distinct and interpretable fuzzy sets but also with a large number of highly overlapping fuzzy sets that hardly allow for any interpretation. When a fuzzy model is developed using expert knowledge, usually the model designer takes care that the model remains interpretable. On the other hand, some degree of redundancy and thus unnecessary complexity cannot be avoided when automated techniques are applied to acquire fuzzy models from data.

Manuscript received April 14, 1996; revised March 2, 1997. This work was supported in part by the Research Council of Norway.

The authors are with the Department of Electrical Engineering, Control Laboratory, Delft University of Technology, 2600 GA Delft, The Netherlands (e-mail: m.setnes@et.tudelft.nl).

Publisher Item Identifier S 1083-4419(98)02624-7.

This article introduces a new approach for simplifying fuzzy rule-based models by reducing redundant information that is found in the form of *similar* fuzzy sets in the rule base. Similar fuzzy sets are highly overlapping fuzzy sets that describe almost the same region in the domain of some model variable. In such cases, the model uses more fuzzy sets than necessary since these fuzzy sets represent more or less the same concept. We propose to use a similarity measure for identifying similar fuzzy sets, and to replace these by a common fuzzy set representative for the original ones. This common fuzzy set is created by merging the similar fuzzy sets. If the redundancy in the model is high, merging similar fuzzy sets might result in equal rules that can be subsequently deleted from the rule base, resulting in reduction of the number of rules.

Description of a system using natural language is an advantage of fuzzy modeling. A simplified rule base makes it easier to assign qualitatively meaningful linguistic terms to the fuzzy sets, and it reduces the number of terms needed. It becomes easier for experts to validate the model and the users can understand better and more quickly the operation of the system. A model with fewer fuzzy sets and fewer rules is also better suited for the design and implementation of a nonlinear (model-based) controller, or for simulation purposes, and it has lower computational demands.

Several methods have been proposed for optimizing the size of the rule base obtained with automated modeling techniques (such as compatible cluster merging [2], fuzzy binary box tree [3], or membership function fusion and annihilation [4]). Similarity driven rule base simplification differs from other reduction methods in that its main objective is to reduce the number of fuzzy sets used in the model. It does not necessarily alter the number of rules. Reduction of the number of rules might follow from rule base simplification if the rules become equal as a result of the merging process. If no rules are combined, simplification is still achieved by reducing the number of fuzzy sets.

In the following we briefly discuss fuzzy modeling, redundancy and similarity before proposing an algorithm for rule base simplification. This paper is organized as follows. Section II reviews fuzzy modeling and shows why similar fuzzy sets appear in the model. In Section III, the concept of similarity is discussed and a suitable similarity measure for fuzzy rule base simplification is chosen. Section IV discusses merging of fuzzy sets and rules, and presents an algorithm for similarity based simplification. In Section V, an application of the algorithm to a real-world problem is described. Finally,

conclusions and comments on further research are given in Section VI.

## II. FUZZY MODELING

Fuzzy modeling has recently been applied with success to a variety of problems, especially in control engineering [5]–[8]. Fuzzy modeling is suited for modeling based on data as well as for modeling based on knowledge acquisition. In the latter case, the model is built using human expert knowledge. This knowledge is captured in IF-THEN rules with fuzzy predicates that establish relations between the relevant system variables. The fuzzy predicates are associated with linguistic labels, and the model is in fact a qualitative description of the system using rules like

IF *temperature* is moderate and *volume* is small  
 THEN *pressure* is low

where the meanings of the linguistic terms *moderate*, *small*, and *low* are defined by fuzzy sets in the domain of the system variables *temperature*, *volume* and *pressure*, respectively. Such models are often called *linguistic models*. However, knowledge acquisition is not a trivial task. Experts are not always available, and when they are, their knowledge is not always consistent, systematic and complete, but often incomplete and episodic [9]. Automated modeling using systems measurements gives a more versatile approach in the sense that it is independent of domain experts, and it facilitates adaptation and self-tuning based on information becoming available during operation. Moreover, data and knowledge can be easily combined when building fuzzy models.

Different types of fuzzy models exist. The Mamdani model [10], which uses linguistic rules with a fuzzy premise part and a fuzzy consequent part, is often used in knowledge acquisition. Another structure that has been used extensively in literature is the Takagi–Sugeno (TS) [5] model. In the examples in Section V we use this approach as it is well suited for automatic approaches to fuzzy modeling [5], [7]. It is explained in more detail below.

### A. The TS Fuzzy Model

The TS fuzzy model uses rules that differ from the Mamdani type in that their consequents are mathematical functions instead of fuzzy sets. The TS model is based on the idea describing the system with a set of local input-output relations that have the following structure:

$$R_i : w_i(\text{IF } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and } \dots \text{ and } x_n \text{ is } A_{in} \text{ THEN } y_i = f_i(\cdot)) \quad (1)$$

where  $R_i$  is the  $i$ th rule in the rule base,  $x_1, \dots, x_n$  are the premise variables,  $y_i$  is the rule output,  $A_{i1}, \dots, A_{in}$  are fuzzy sets defined for the respective premise variables, and  $w_i$  is the rule weight. Typically,  $w_i = 1, \forall_i$ , but it can be adjusted during model reduction. Usually,  $f_i(\cdot)$  is a linear function of the premise variables

$$f_i(x_1, x_2, \dots, x_n) = p_{i0} + p_{i1}x_1 + p_{i2}x_2 + \dots + p_{in}x_n. \quad (2)$$

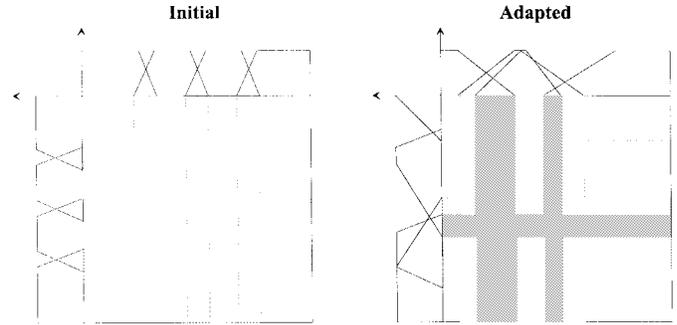


Fig. 1. Fuzzy partition of two premise variables. Adapting the fuzzy sets which define an initial fuzzy partition for the premise space can result in similar fuzzy sets.

Each rule defines a hyperplane in the premise-consequent product space, and the model approximates the real system’s hypersurface with these hyperplanes. The overall output of the model (the inferred value  $\hat{y}$ ) is calculated by taking the weighted average of the rule consequences

$$\hat{y} = \frac{\sum_{i=1}^N w_i \beta_i y_i}{\sum_{i=1}^N w_i \beta_i} \quad (3)$$

where  $N$  is the number of rules,  $\beta_i$  is the degree of activation of the  $i$ th rule’s premise, and  $y_i$  is the contribution of that rule. Given the inputs  $x_1, x_2, \dots, x_n$ , the degree of activation is calculated as

$$\beta_i = \prod_{j=1}^n \mu_{ij}(x_j), \quad i = 1, 2, \dots, N \quad (4)$$

where  $\mu_{ij}(x_j)$  is the membership function of the fuzzy set  $A_{ij}$  for input variable  $x_j$  in the premise of the  $i$ th rule, see (1). Because of the linear structure of the rule consequents, well-known parameter estimation techniques such as least squares can be used to estimate the consequent parameters [7], [8].

### B. Automated Modeling

Two common approaches for obtaining fuzzy models from systems measurements are parameter adaptation and fuzzy clustering. In the first case, one tunes an initial partition of the premise space, while in the second case a partition suitable for a given number of rules is sought automatically. Both techniques can be combined too.

1) *Parameter Adaptation*: In this approach an initial partition of the input space is usually given by a number of equidistant symmetrical fuzzy sets defined for all the premise variables of the system. This partition can be seen as a uniform grid in the premise space. The parameters of the membership functions are adapted using learning algorithms such as backpropagation. During adaptation, the fuzzy sets can drift closer to each other and may end up in overlapping positions [4]. The resulting rule base then contains redundancy in terms of highly similar fuzzy sets, as illustrated in Fig. 1. A drawback with this approach is that the number of rules increases exponentially with the number of inputs, and that an initially transparent model may become unreadable after adaptation.

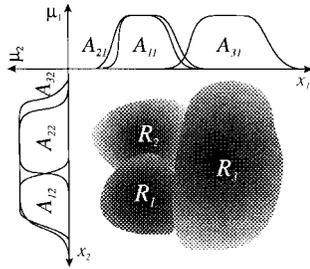


Fig. 2. Schematic diagram of three fuzzy clusters in a two-dimensional premise space. The fuzzy clusters represent rules  $R_1$ ,  $R_2$ , and  $R_3$ . The fuzzy sets in the premise of each rule are found by projecting the clusters onto the premise variables  $x_1$  and  $x_2$ .

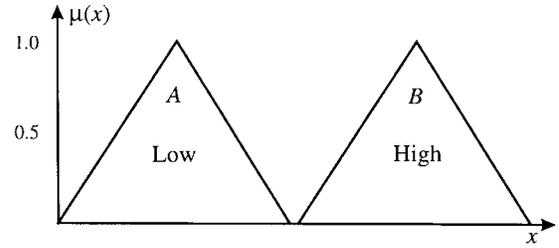
2) *Fuzzy Clustering*: Fuzzy clustering algorithms are unsupervised algorithms that partition data points into a given number of clusters with fuzzy boundaries. By applying fuzzy clustering to data sampled from a system, a fuzzy model of the system can be obtained. Each cluster represents a fuzzy relation, and corresponds to a rule in the rule base. The fuzzy sets in the premise part of the rules are usually identified by projecting the clusters onto the corresponding axis of the data space [7], [8]. This projection usually results in similar fuzzy sets, as illustrated in Fig. 2. In Mamdani models, the fuzzy consequents of the rules are determined by projection too. In TS models, the consequent parameters are derived from the cluster covariance matrix [11] or estimated using a parameter estimation technique. Different approaches to clustering can be found, such as clustering in the output space and inducing clusters in the input space [12] or clustering in the product space of input and output variables [7], [8]. Two well known fuzzy clustering algorithms are the fuzzy c-means [13] and the Gustafson–Kessel (GK) algorithm [11]. The latter is especially suitable for the identification of TS fuzzy models and has been successfully applied to modeling of dynamic systems [7], [8]. The GK algorithm is suitable for approximating  $(n - 1)$ -dimensional hypersurfaces in a  $n$ -dimensional space. It can be used to identify a systems model by clustering data from system (input-output) measurements. However, before applying clustering, the number of clusters must be specified explicitly. Correct specification of the number of clusters is important. A large number results in an unnecessarily complicated rule base, while a small number may result in a poor model. Methods for finding the optimal number of clusters (rules) have been suggested, see, e.g., [2], [12], [14].

### C. Redundancy

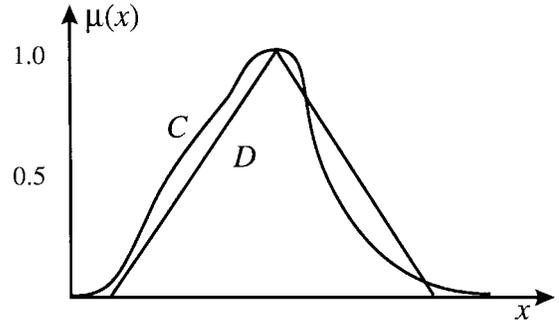
Fuzzy models, especially if acquired from data, may contain redundant information in the form of similarity between fuzzy sets. Three unwanted effects that can be recognized are

- 1) similarity between fuzzy sets in the model;
- 2) similarity of a fuzzy set  $A$  to the universal set  $U$  :  $\mu_U(x) = 1, \forall x \in X$ ;
- 3) similarity of a fuzzy set  $A$  to a singleton set.

As similar fuzzy sets represent compatible concepts in the rule base, a model with many similar fuzzy sets becomes redundant, unnecessarily complex and computationally de-



(a)



(b)

Fig. 3. (a) Distinct fuzzy sets with no degree of equality and (b) overlapping fuzzy sets with a high degree of equality.

manding. Linguistic interpretation of such a model is difficult as it is not trivial to assign qualitatively meaningful labels to highly similar fuzzy sets.

Some of the fuzzy sets extracted from data may be similar to the universal set  $U$  (e.g.,  $A_{32}$  in Fig. 2). Such fuzzy sets are irrelevant. The opposite effect is similarity to a singleton set. During adaptation, membership functions may get narrow, resulting in fuzzy sets almost like singletons (spikes) [4]. If a rule has one or more such fuzzy sets in its premise, it will practically never fire, and thus the rule does not contribute to the output of the model. However, it should be noted that such rules may represent exceptions from the overall model behavior.

### III. SIMILARITY

The concept of similarity is interpreted in different ways depending on the context. The interpretation of similarity in everyday language is “having characteristics in common” or “not different in shape, but in size or position.” This interpretation of similarity differs from the one we use. We define similarity between fuzzy sets as *the degree to which the fuzzy sets are equal*. This definition is related to the concepts represented by the fuzzy sets. Consider the fuzzy sets  $A$  and  $B$  in Fig. 3(a). They have exactly the same shape, but represent clearly distinct concepts, e.g., a low and a high value of  $x$ , respectively. They have zero degree of equality and are considered dissimilar. On the other hand, the two fuzzy sets  $C$  and  $D$  in Fig. 3(b), even though they differ in shape, can be said to have a high degree of equality. They represent compatible concepts and are considered similar.

#### A. Similarity as Degree of Equality

In our approach, fuzzy sets are considered similar if they are defined by overlapping membership functions that assign

approximately the same values of membership to the elements in their universe of discourse. Their similarity is the degree to which they can be considered as equal. The classical definition of equality is crisp. Let  $\mu_A$  and  $\mu_B$  be the membership functions of fuzzy sets  $A$  and  $B$ , respectively. Then  $A$  and  $B$  are *equal*, if  $\mu_A(x) = \mu_B(x)$ ,  $\forall x \in X$ . If we apply this concept of equality to the fuzzy sets in Fig. 3 we get  $A \neq B$  and  $C \neq D$  since their membership functions are different. However,  $C$  and  $D$  can be said to have a high degree of equality and hence they are similar. In the same way that fuzzy sets allow for gradual transition between full membership and total nonmembership, a similarity measure captures a gradual transition between equality and nonequality

$$s = S(A, B) = \text{degree}(A = B), \quad s \in [0, 1] \quad (5)$$

where  $S$  is a similarity measure. The similarity measure is a function assigning a similarity value  $s$  to the pair of fuzzy sets  $(A, B)$  that indicates the degree to which  $A$  and  $B$  are equal or how similar they are.

### B. Similarity Measures for Rule Base Simplification

A similarity measure for fuzzy sets detects highly similar fuzzy sets representing more or less compatible concepts in a fuzzy rule base. Such fuzzy sets should be assigned a high similarity value, whereas more distinct sets should be assigned a lower similarity value. For a correct comparison of similarity values, the similarity measure should not be influenced by the scaling of the domain on which the fuzzy sets are defined. This avoids the need for normalization of the domains.

In the following, let  $A$  and  $B$  be (two) fuzzy subsets of  $X$  with membership functions  $\mu_A(x)$  and  $\mu_B(x)$ , respectively. A similarity measure will be considered as a candidate for an automated rule base simplification scheme if it satisfies the following four criteria:

- 1) Nonoverlapping fuzzy sets should be considered totally nonequal,  $s = 0$

$$S(A, B) = 0 \Leftrightarrow \mu_A(x)\mu_B(x) = 0, \quad \forall x \in X. \quad (6)$$

- 2) Overlapping fuzzy sets should have a similarity value  $s > 0$

$$S(A, B) > 0 \Leftrightarrow \exists x \in X, \quad \mu_A(x)\mu_B(x) \neq 0. \quad (7)$$

- 3) Only equal fuzzy sets should have a similarity value  $s = 1$

$$S(A, B) = 1 \Leftrightarrow \mu_A(x) = \mu_B(x), \quad \forall x \in X. \quad (8)$$

- 4) Similarity between two fuzzy sets should not be influenced by scaling or shifting the domain on which they are defined

$$\begin{aligned} S(A', B') &= S(A, B), \quad \mu_{A'}(l + kx) = \mu_A(x), \\ \mu_{B'}(l + kx) &= \mu_B(x), \quad k, l \in \mathbb{R}, \quad k > 0. \end{aligned} \quad (9)$$

Criterion 1 assures that dissimilar (nonoverlapping) fuzzy sets are excluded from the set of similar fuzzy sets. Various degrees of dissimilarity between distinct fuzzy sets are related to the distance between them, and can be quantified by a *distance measure*. According to criterion 2, overlapping fuzzy sets should be assigned a nonzero degree of similarity and should not be regarded as totally nonequal. Criterion 3 assures that equality is a special case of similarity, in the same way that crisp sets can be considered as a special case of fuzzy sets. Criterion 4 is required for a fair comparison of similarities in the rule base as a similarity measure that satisfies this criterion is not influenced by the numerical values of the domain variables.

Different similarity measures have been proposed for fuzzy sets, and a study of many such measures can be found in [15], [16]. In general, they can be divided into two main groups:

- 1) geometric similarity measures;
- 2) set-theoretic similarity measures.

The theoretical analysis of similarity has been dominated by geometric models. These models represent fuzzy sets as points in a metric space and the similarity between the sets is regarded as an inverse of their distance in this metric space. Denoting the distance between  $A$  and  $B$  as  $D(A, B)$ , the similarity of  $A$  and  $B$  can be written as

$$s = S(A, B) = \frac{1}{1 + D(A, B)}, \quad s \in (0, 1]. \quad (10)$$

Examples of geometric similarity measures are the generalizations of the Hausdorff distance to fuzzy sets [17], [18]. Another example is similarity transformed from the well-known Minkowski class of distance functions

$$D_r(A, B) = \left( \sum_{j=1}^m |\mu_A(x_j) - \mu_B(x_j)|^r \right)^{\frac{1}{r}}, \quad r \geq 1 \quad (11)$$

assuming that the fuzzy sets  $A$  and  $B$  are defined on a discrete universe of discourse  $X = \{x_j \mid j = 1, 2, \dots, m\}$ . For continuous universes, the sum is replaced by integration.

It has been argued in [18] that geometric similarity measures are best suited for measuring similarity (or dissimilarity) among distinct fuzzy sets, while the set-theoretic measures are the most suitable for capturing similarity among overlapping fuzzy sets. The geometric similarity measures represent similarity as proximity of fuzzy sets, and not as a measure of equality. The interpretation of similarity as ‘‘approximate equality’’ can better be represented by a set-theoretic similarity measures. Such measures are based on set-theoretic operations like union and intersection. They also have the advantage above geometrical measures that they are not influenced by scaling and ordering of the domain.

A set-theoretic measure often encountered in the literature is the so-called consistency-index which is the maximum membership degree of the intersection of two fuzzy sets

$$S_C(A, B) = \sup_{x \in X} \mu_{A \cap B} = \max_{x \in X} [\mu_A(x) \wedge \mu_B(x)] \quad (12)$$

where  $\wedge$  is the minimum operator. Some authors use this measure for rule base reduction purposes [3], [4]. However, this

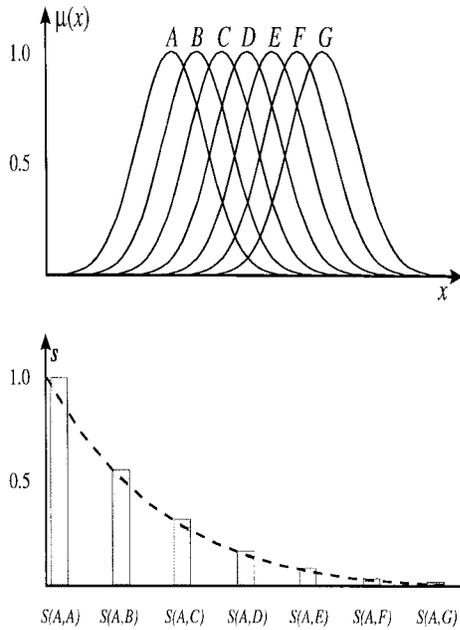


Fig. 4. Fuzzy sets  $A, B, \dots, G$  and the similarity computed for  $S(A, A)$ ,  $S(A, B)$ ,  $\dots$ ,  $S(A, G)$ .

measure does not fulfill criterion 3 as it focuses attention on only one value of the variable  $x$ , rather than performing some sort of averaging or integration. We will use the following similarity measure, based on the set-theoretic operations of intersection and union, to determine the similarity between fuzzy sets [19]

$$S(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (13)$$

where  $|\cdot|$  denotes the cardinality of a set, and the  $\cap$  and  $\cup$  operators represent the intersection and union respectively [20]. Rewriting this expression in terms of the membership functions gives

$$S(A, B) = \frac{\sum_{j=1}^m [\mu_A(x_j) \wedge \mu_B(x_j)]}{\sum_{j=1}^m [\mu_A(x_j) \vee \mu_B(x_j)]} \quad (14)$$

in a discrete universe  $X = \{x_j \mid j = 1, 2, \dots, m\}$ .  $\wedge$  and  $\vee$  are the minimum and maximum operators, respectively. In computer implementation, continuous domains need to be discretized. This similarity measure complies with the four criteria above, and reflects the idea of a gradual transition from equal to completely nonequal fuzzy sets (with  $S(A, B) = 0$ ). An example of the behavior of this similarity measure for fuzzy sets with a varying degree of overlap is shown in Fig. 4.

#### IV. RULE BASE SIMPLIFICATION

As discussed in Section II-B, automated approaches to fuzzy modeling often introduce redundancy in terms of several similar fuzzy sets that describe almost the same region in the domain of some model variable. The similarity measure (13) can be used to quantify the similarity between fuzzy sets in the rule base. Two or more such similar fuzzy sets can be merged to create a new fuzzy set representative of the merged sets. By substituting this new fuzzy set for the ones merged in the rule

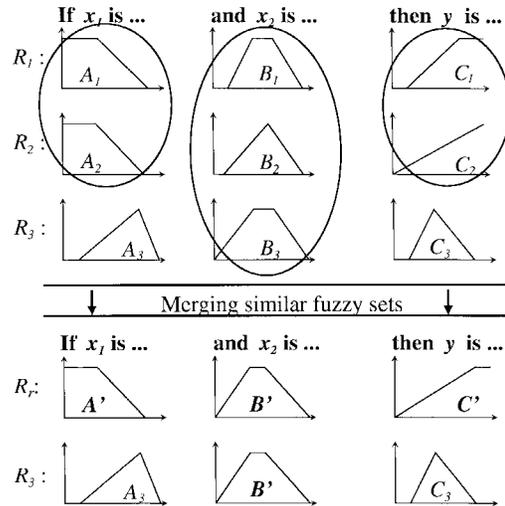


Fig. 5. Example of a Mamdani-type model. Similar fuzzy sets are merged. Rules  $R_1$  and  $R_2$  become equal and can be represented by one rule  $R_r$ .

base, the number of fuzzy sets needed to constitute the model decreases. In this way *rule base simplification* is achieved. This simplification may result in equal rules. In the rule base, only one of the equal rules is needed, and the others can be deleted. In this way *rule base reduction* is achieved. Hence, in our approach, there is a difference between rule base simplification and rule base reduction. The former is the primary objective, and the latter may follow indirectly if the redundancy is high. Fig. 5 illustrates the idea of merging similar fuzzy sets, which results in both rule base simplification and reduction. This idea is exploited in this section for developing the rule base simplification and reduction algorithm that is described in Section IV-D.

#### A. Removing Fuzzy Sets

The rule base may contain irrelevant fuzzy sets. If a fuzzy set in the premise of a rule has a membership function  $\mu(x) \approx 1, \forall x \in X$ , it is similar to the universal set  $U$  and can be removed. The similarity of a fuzzy set  $A$  to the universal set is quantified by  $S(A, U)$ . An example of a fuzzy set quite similar to the universal set is illustrated in Fig. 2. Here  $A_{32}$  can be removed, and only  $A_{31}$  is necessary in the premise of rule  $R_3$  to distinguish the associated region in the premise product space.

A rule whose premise only consists of fuzzy sets very similar to the universal set can be removed. Its contribution is then only biasing the result, and the same output can be achieved by adjusting the consequents of the other rules. The opposite may also occur. During adaptation, the support of one or more fuzzy sets may become so narrow that they become almost like spikes (singletons). Singleton fuzzy sets have extremely low similarity to the universal set (i.e.,  $S(A, U) \approx 0$ ). In some cases, rules with such fuzzy sets in their premise can be removed from the rule base, but in general care must be taken as the rules may represent exceptions. Interaction from the user is typically needed in such cases. Since our aim is to develop an automated simplification method, we do not

consider removing singletons from the rule base. See [4] for an example of a possible approach.

### B. Merging Fuzzy Sets

In general, when two fuzzy sets  $A$  and  $B$  are considered to be very similar, there are three possibilities for simplifying the rule base

- 1) replace  $A$  by  $B$ ;
- 2) replace  $B$  by  $A$ ;
- 3) replace both  $A$  and  $B$  by a new fuzzy set  $C$ .

Two important aspects of the simplified model are its accuracy and coverage of the premise space (uncovered regions should not occur in the premise space as a result of rule base simplification). Assuming that the model's accuracy is measured by some function  $J$  (e.g., sum of squared errors), the effect of replacing  $A$  and  $B$  by  $C$  should be as small as possible with respect to  $J$ . Finding the fuzzy set  $C$  best suited to replace  $A$  and  $B$  becomes a question of evaluating  $J$ . In general, one can say that if the model is more sensitive to changes in  $A$  than to changes in  $B$ ,  $A$  should replace  $B$ , or the common fuzzy set  $C$  should resemble  $A$  more than  $B$ . In particular applications, additional aspects like model granularity (number of linguistic terms per variable), interpretability or physical relevance may be important.

For the discussion about merging fuzzy sets, we define a fuzzy set  $A$  using a *parametric* membership function  $\mu_A(x; a_1, a_2, a_3, a_4)$ ,  $a_1 \leq a_2 \leq a_3 \leq a_4$

$$\mu_A(x; a_1, a_2, a_3, a_4) = \begin{cases} 0, & x \leq a_1, \text{ or } x \geq a_4 \\ 1, & a_2 \leq x \leq a_3 \\ \alpha, & \alpha \in (0, 1), \text{ otherwise.} \end{cases} \quad (15)$$

One way to merge the fuzzy sets is to take the support of  $A \cup B$  as the support of the new fuzzy set  $C$ . This guarantees preservation of the coverage of the whole premise space when  $C$  replaces  $A$  and  $B$  in the premise of the rule base. The kernel of  $C$  is given by aggregating the parameters describing the kernels of  $A$  and  $B$ . Thus merging  $A$  and  $B$ , defined by  $\mu_A(x; a_1, a_2, a_3, a_4)$  and  $\mu_B(x; b_1, b_2, b_3, b_4)$ , respectively, (15) gives a fuzzy set  $C$  defined by  $\mu_C(x; c_1, c_2, c_3, c_4)$  where

$$c_1 = \min(a_1, b_1) \quad (16)$$

$$c_2 = \lambda_2 a_2 + (1 - \lambda_2) b_2 \quad (17)$$

$$c_3 = \lambda_3 a_3 + (1 - \lambda_3) b_3 \quad (18)$$

$$c_4 = \max(a_4, b_4). \quad (19)$$

The parameters  $\lambda_2, \lambda_3 \in [0, 1]$  determines which of the fuzzy sets  $A$  or  $B$  has the most influence on the kernel of  $C$ . In the rest of the article we use  $\lambda_2 = \lambda_3 = 0.5$ . This averaging of the kernels gives a tradeoff between contributions of the rules in which the fuzzy sets occur. Fig. 6 illustrates this method for merging two fuzzy sets  $A$  and  $B$  to create  $C$ .

### C. Merging Rules

In a Mamdani fuzzy model, if  $k$  rules become equal as a result of rule base simplification,  $k-1$  of them can be removed, resulting in rule base reduction, as shown in Fig. 5. If only the premises of the rules become equal, and the consequents

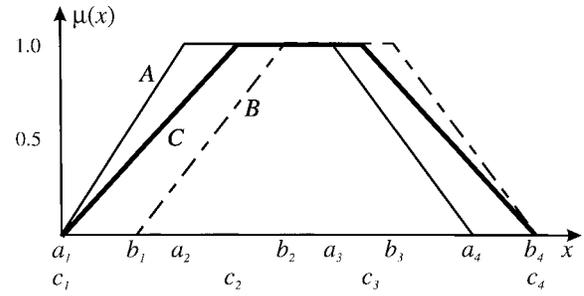


Fig. 6. Creating fuzzy set  $C$  to replace  $A$  and  $B$  in the rule base.

do not, this can indicate a contradiction in the rule base that might need further attention [21]. In the rest of this article we concentrate on TS models since they are well suited for (semi) automated generation of rules from data.

In TS models, the consequents of the rules are not fuzzy, and the similarity is considered only in the premise part of the rules. When the *premise parts* of  $k$  rules get equal, we remove  $k-1$  of these rules. However, the consequent parameters of the rule remaining in the rule base, called  $R_r$ , must be reestimated. This can be done by weighting  $R_r$  with  $k$  ( $w_r = k$ ) and averaging the consequents of all the  $k$  rules with equal premise parts. Thus, the  $k$  rules with equal premise parts,  $R_1, R_2, \dots, R_k$ , are represented by a single rule  $R_r$  with weight  $w_r = k$  and consequent parameters

$$P_r = \frac{1}{k} \sum_{i=1}^k P_i \quad (20)$$

where  $P_i$  is a vector of the consequent parameters  $p_{i0}, p_{i1}, \dots, p_{in}$  in the  $i$ th of the rules with equal premise parts. The model output must now be calculated according to

$$\hat{y} = \frac{\sum_{i=1, i \neq r}^{N-(k-1)} w_i \beta_i y_i + w_r \beta_r y_r}{\sum_{i=1, i \neq r}^{N-(k-1)} w_i \beta_i + w_r \beta_r}. \quad (21)$$

Another approach is to reestimate the consequent parameters in the reduced rule base using training data. This requires more computation, but usually gives a numerically more accurate result than weighting and averaging. Independently of the method used, we refer to this as *merging rules*.

### D. Rule Base Simplification Algorithm

This section describes an algorithm that is developed for rule base simplification in TS models. The same concept can be used also for Mamdani models. Simplification is achieved by removing fuzzy sets similar to the universal set and by merging similar fuzzy sets. Based on the result, rule base reduction is obtained by merging rules with equal premise parts. The approach uses the similarity measure (13) for determining the similarity between the fuzzy sets in the rule base, and requires two thresholds:  $\gamma$  for removing fuzzy sets similar to the universal set and  $\lambda$  for merging fuzzy sets that are similar to one another.

The algorithm starts by iteratively merging similar fuzzy sets. In each iteration, the similarity between all pairs of fuzzy sets for each variable is considered, and the pair of fuzzy sets

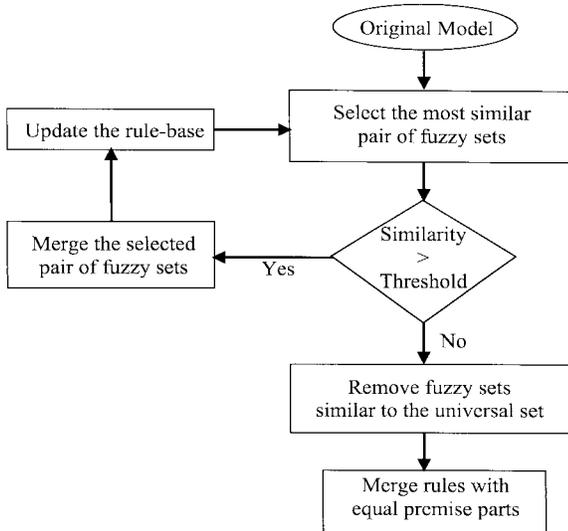


Fig. 7. The simplification algorithm.

having the highest similarity  $s > \lambda$  is merged to create a new fuzzy set. Then the rule base is updated by substituting this new fuzzy set for the fuzzy sets merged to create it. The algorithm then again evaluates the similarities in the updated rule base. This continues until there are no more fuzzy sets for which  $s \geq \lambda$ . Thereafter, the fuzzy sets that have similarity  $s > \gamma$  to the universal set  $U$  are removed. Finally, the rule base is checked for rules with equal premise parts. Such rules are merged as discussed in Section IV-C. Fig. 7 depicts a flowchart of the algorithm that is summarized as follows:

Given a fuzzy rule base  $\mathbf{R} = \{R_i\}_{i=1}^N$ , where  $R_i$  is given by (1), select thresholds  $\gamma, \lambda \in (0, 1)$ .

**Repeat:**

*Step 1: Select two most similar fuzzy sets in  $\mathbf{R}$ .* Calculate  $s_{ikj} = S(A_{ij}, A_{kj})$ ,  $j = 1, \dots, n$ ,  $i = 1, \dots, N$ ,  $k = 1, \dots, N$ . Select  $A_{lq}$  and  $A_{mq}$ ,  $s_{tmq} = \max_{i \neq k} \{s_{ikj}\}$ .

*Step 2: Merge similar fuzzy sets and update  $\mathbf{R}$ .* If  $s_{tmq} \geq \lambda$  merge  $A_{lq}$  and  $A_{mq}$  to create a new fuzzy set  $A$ . Set  $A_{lq} = A$  and  $A_{mq} = A$ .

**Until:** no more fuzzy sets have similarity  $s_{ikj} \geq \lambda$ ,  $i \neq k$ .

*Step 3: Remove fuzzy sets similar to the universal set.* For each fuzzy set  $A_{ij}$  calculate  $S(A_{ij}, U)$ ,  $\mu_U(x_j) = 1$ ,  $\forall x_j$ . If  $S(A_{ij}, U) \geq \gamma$  remove  $A_{ij}$  from the premise of  $R_i$ .

*Step 4: Merge rules with equal premise part.*

The threshold  $\gamma$  for removing the fuzzy sets that are similar to the universal set should generally be higher than the threshold  $\lambda$  for merging. In our applications,  $\gamma = 0.8$  gives good results. The choice of a suitable threshold  $\lambda$  depends on the application. The lower the threshold  $\lambda$ , the less fuzzy sets are used in the resulting model. In general, one can expect the numerical accuracy of the model to decrease as  $\lambda$  decreases. However, this need not always be the case. If the model is highly redundant or overdetermined, the numerical accuracy may improve as a result of simplification.

By using different thresholds, different versions of the model can be obtained. For instance, for explaining the working of a system (operator training, expert evaluation), a comprehensible linguistic description is important. In such cases,

it is reasonable to trade some numeric accuracy for extra transparency and readability. This implies the use of a lower threshold  $\lambda$  than when aiming at applications for prediction or simulation.

The algorithm only merges one pair of fuzzy sets ( $A_{lq}$ ,  $A_{mq}$ ) per iteration. Merging in different premise variables is independent of each other, making it possible to merge more fuzzy sets in one variable than in another. It might happen that the fuzzy sets to be merged were created themselves by merging in a previous iteration. For the fuzzy sets in the original rule base to have an equal influence on the final result, we make use of the fact that if a fuzzy set  $D$  is created by merging, it has multiple occurrences in the rule base. For instance, if  $D$  is created by merging ( $A, B$ ), and later  $D$  is merged with  $C$ , two occurrences of  $D$  and one of  $C$  are merged to create  $E$ . Using the parametric description in (15), the parameters of the membership function  $\mu_E(x; e_1, e_2, e_3, e_4)$  become:  $e_1 = \min(c_1, d_1)$ ,  $e_2 = (c_2 + c_2 + d_2)/3$ ,  $e_3 = (c_3 + c_3 + d_3)/3$ , and  $e_4 = \max(c_4, d_4)$ . This corresponds to merging the three original membership functions  $A, B$ , and  $C$ .

After rule base simplification, rules with equal premise parts are merged. In the following, we reestimate the consequent parameters of the resulting rule base using the same training data from which the original rule base was identified.

## V. EXAMPLES

In this section the proposed approach is applied to two fuzzy models. The first one is an approximation of a static nonlinear function using simulation data. The second one is a static model of the time response of a real multiple-input-single-output (MISO) nonlinear process. Both models are fuzzy rule-based models of the TS type and have been identified from training data using the identification method described in [7], [8] which is based on the GK clustering algorithm (see also Section II-B2). In this algorithm, three parameters have to be preselected: The number of clusters  $K$ , the fuzziness parameter  $m$  and the stopping criterion  $e$ . In the literature,  $m$  is usually chosen to be 2, and this value was also used for obtaining the models in this section. The stopping criterion for the GK-algorithm was  $e = 0.01$ .

### A. Sinusoidal Function

A sinusoidal function is approximated using automated fuzzy modeling. The function is generated according to the formula

$$y = \sin(0.0015x^2) \frac{x^{2.9}}{10000}, \quad x = 0, 1, 2, \dots, 100. \quad (22)$$

Using  $K = 11$ , a fuzzy rule base with rules of the TS type is obtained

$$R_i: \text{ IF } x \text{ is } A_i \text{ THEN } y_i = p_{i0} + p_{i1}x, \quad i = 1, 2, \dots, 11. \quad (23)$$

The two consequent parameters in each rule ( $p_{i0}, p_{i1}$ ) are estimated using least-mean-squares (LMS). The obtained fuzzy model is shown in Fig. 8, where both the local linear models defined by the rule consequents and the fuzzy sets in the

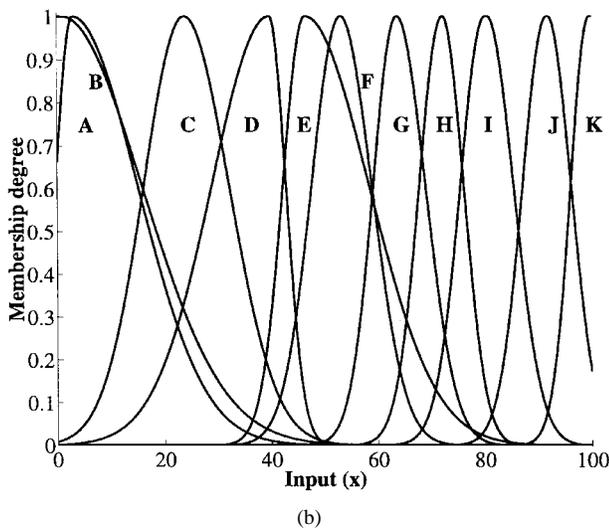
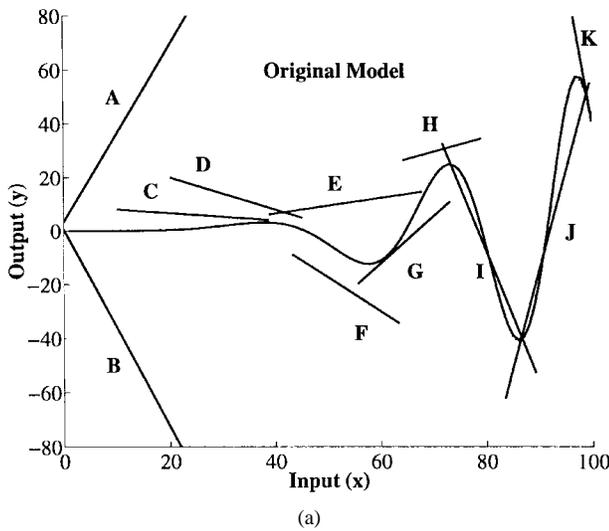


Fig. 8. Original model: (a) Function to be approximated (solid line), model output (dotted line, barely visible as the match is very good), and the local linear models. (b) The identified membership functions of the original model.

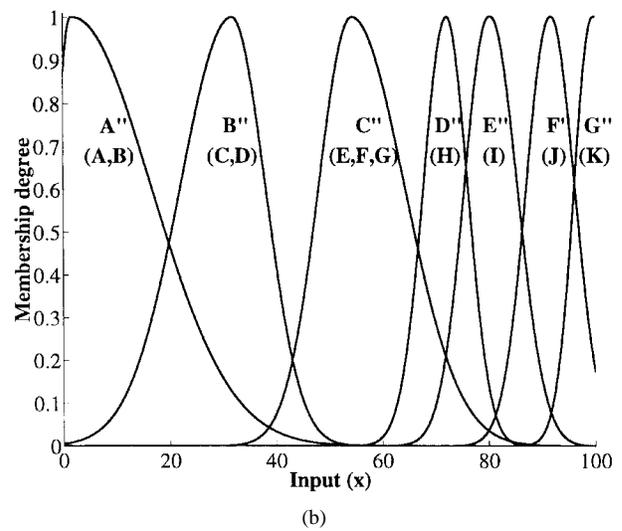
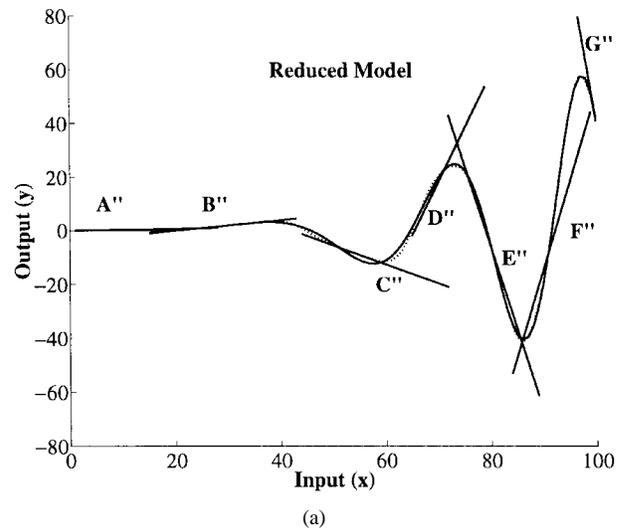


Fig. 9. Reduced model: (a) Function to be approximated (solid line), model output (dotted line) and the local linear models. (b) Membership functions of the reduced model. The letters in parentheses indicate the corresponding fuzzy sets in the original model.

premise of the rule base are plotted. A contradiction in the rule base can be recognized for low values of  $x$  where two rules with completely different consequents [ $A$  and  $B$  in Fig. 8(a)] are fired with approximately the same strength in the same region. The identified membership functions show that the fuzzy sets in the premise of these two contradictory rules are very similar [ $A$  and  $B$  in Fig. 8(b)]. The combined result is a flat model output in this region that corresponds to the behavior of the function to be approximated. We apply the proposed algorithm to this model. It identifies and merges these two highly similar membership functions. The premise parts of the two rules in question become equal, and the two rules corresponding to these two contradictory local models are merged. The result is one rule with a consequent part representing a correct local model of the function in this region. This is illustrated in Fig. 9(a) where we have applied the simplification algorithm with threshold  $\lambda = 0.25$ . Both the number of membership functions and the number of rules are reduced from 11 to 7. Lowering the threshold will influence the performance of the model severely as we will end up

with less local models (rules) than required for a reasonable approximation of the function. Table I shows which fuzzy sets are merged when using different thresholds together with the mean-squared-error (MSE) of the reduced model [the letters  $A, B, \dots, K$  refer to the fuzzy sets in Fig. 8(b)]. The threshold  $\gamma$  was set to 0.8. However, it has no influence (unless set very low) as there are no fuzzy sets in this model with high similarity to the universal set  $U$ .

### B. Enzymatic Soil Removal

In this section we consider fuzzy modeling of enzymatic soil removal. Enzymes are agents that break down soil chemically. The rate of the soil removal is affected by such factors as alkalinity of the solution and temperature. The fuzzy model should predict the percentage soil that is removed by the enzymes, given the elapsed time for the reaction, the temperature of the solution and the alkalinity. It is assumed that the enzyme concentration is large enough, so that it has no influence on the rate of the reaction. More information about

TABLE I  
FUZZY SETS MERGED FOR DIFFERENT VALUES OF  $\lambda$  AND EVALUATION OF THE REDUCED MODELS

Threshold $\lambda$	Merged	No. Rules	MSE
Original	n.a.	11	0.1045
0.80	(A, B)	10	0.1045
0.60	(A, B)(E, F)	9	0.1093
0.30	(A, B)(C, D)(E, F)	8	0.1303
0.25	(A, B)(C, D)(E, F, G)	7	0.6264
0.20	(A, B, C, D)(E, F, G)(H, I)(J, K)	5	21.975

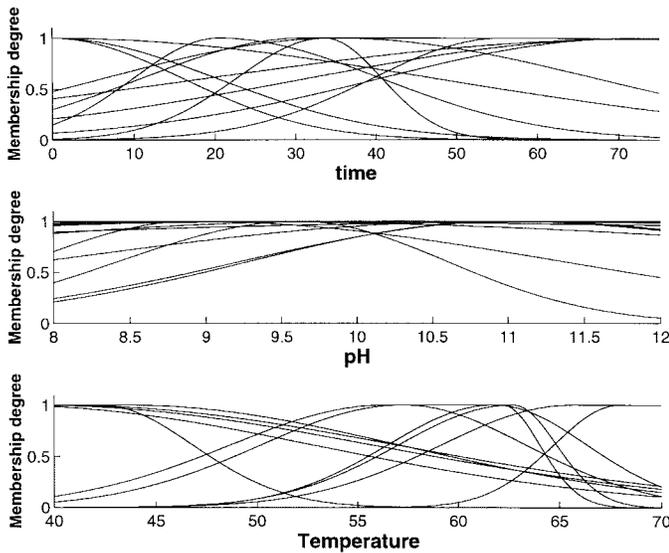


Fig. 10. The membership functions of the original model.

fuzzy modeling of enzymatic soil removal can be found in [22].

1) *Original Model:* The process is identified from measurements of the time responses under particular temperature and alkalinity conditions. Based on these measurements, a TS fuzzy model has been identified by fuzzy clustering in the product space of the three input variables time ( $t$ ), alkalinity ( $pH$ ) and temperature ( $T$ ), and one output variable, the remaining soil ( $y$ ). The rule base obtained from clustering with  $K = 11$  can be written

$$R_i: \text{ IF } t \text{ is } A_{i1} \text{ and } pH \text{ is } A_{i2} \text{ and } T \text{ is } A_{i3} \\ \text{ THEN } y_i, \quad i = 1, 2, \dots, 11. \quad (24)$$

The output (consequent) of each rule is a linear combination of the process variables

$$y_i = p_{i0} + p_{i1} \cdot t + p_{i2} \cdot pH + p_{i3} \cdot T \quad (25)$$

where the parameters  $p_{i0}, p_{i1}, \dots, p_{i3}$  have been estimated with LMS. The model obtained from clustering consists of 11 rules and has the total of 33 fuzzy sets [22]. The fuzzy sets in the premise of this original model are shown in Fig. 10.

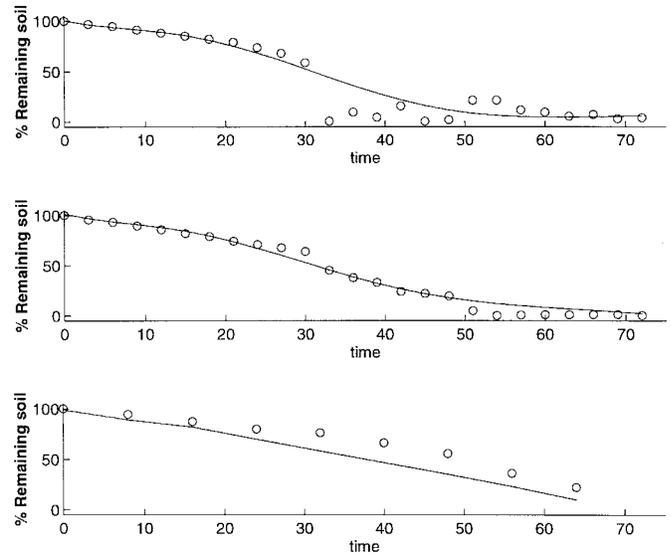


Fig. 11. The original model's response to the evaluation data for three different experimental conditions.

TABLE II  
RESULTS THAT ARE OBTAINED USING DIFFERENT VALUES FOR  $\lambda$  AND  $\gamma = 0.8$

Threshold $\lambda$	Number of fuzzy sets per variable				rules	MSE
	Time	pH	Temp	total		
Original model	11	5	11	33	11	0.0108
0.90	10	2	9	21	11	0.0106
0.80	8	2	6	16	11	0.0108
0.70	4	2	6	12	10	0.0119
0.60	2	2	5	9	9	0.0116
0.40	2	1	3	6	7	0.0120

2) *Model Validation:* The original model is validated using a validation data set. The original model's response to this data is shown in Fig. 11. The original model's response to the evaluation data has been found satisfactory, the MSE is 0.0108. However, the model contains a lot of overlapping fuzzy sets. Fig. 10 shows that there is a high redundancy in terms of similarity among the fuzzy sets in the model. Assigning meaningful labels to all these fuzzy sets for linguistic description and expert evaluation is almost impossible.

3) *Results:* The proposed rule base simplification algorithm is applied several times with different values of the threshold  $\lambda$  for merging. The threshold  $\gamma$  for removing fuzzy sets similar to the universal set is held constant at 0.8. Both rule base simplification and reduction occurs, and different versions of the rule base are obtained, varying in both the number of fuzzy sets (granularity) and the number of rules. MSE is calculated for the different models from their response to the evaluation data, and the obtained results are reported in Table II. The table shows that the original model can be both simplified and reduced quite substantially without sacrificing the accuracy too much. Also we see that lowering the threshold does not always decrease the numerical accuracy (as mentioned in Section IV-

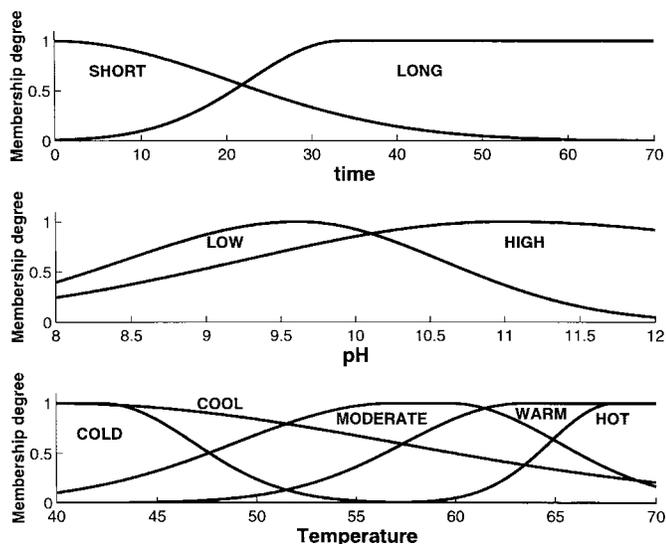


Fig. 12. Membership functions obtained with  $\lambda = 0.6$  and  $\gamma = 0.8$ .

D). From this, we conclude that the original model is highly redundant and overdetermined.

In Fig. 12, the fuzzy sets of the model obtained by using  $\lambda = 0.6$  are plotted. For this case, obtaining a qualitative model by assigning linguistic labels to the fuzzy sets is much easier than for the original model. As shown in the figure, the variable time is partitioned into two regions, “short” and “long;” the variable alkalinity ( $pH$ ) is partitioned into “low” and “high”; and for the variable temperature one obtains the five regions “cold”, “cool”, “moderate”, “warm”, and “hot.” The response of this model to the evaluation data is shown in Fig. 13. There is little difference from the response of the original model (compare with Fig. 11). See Table II for the MSE values.

From Table II we see that for a threshold  $\lambda < 0.6$  all the fuzzy sets defined for the premise variable alkalinity are merged. If we decrease the threshold  $\gamma$ , the process variable alkalinity will be removed from the premise of the rule base. However, it will still remain in the rule consequents, having a linear influence on the process. Expert knowledge confirm that alkalinity is the least influential premise parameter for this process, and that the temperature is the most important quantity that influences the enzyme activity in this application. This is reflected in the higher granularity in the partitioning of this variable.

## VI. CONCLUSION

Fuzzy models, especially if acquired from data, often contain a large number of fuzzy sets. A closer study of the fuzzy sets in the rule base may reveal that many of them are highly similar, representing more or less the same concept. This redundancy hampers linguistic interpretation of the models, and also makes them computationally more demanding. In this article, a similarity driven method for fuzzy rule base simplification has been presented. An algorithm for its implementation has been proposed and applied to two examples: a simulation of a strongly nonlinear function and a real process.

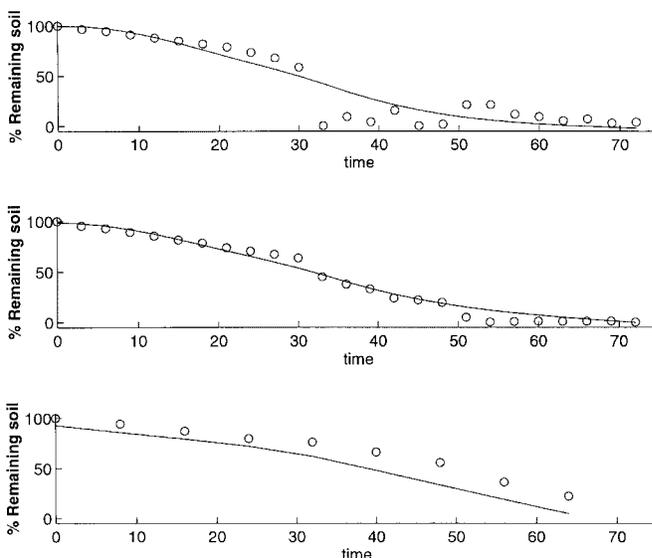


Fig. 13. Response of model obtained with  $\lambda = 0.6$  and  $\gamma = 0.8$ .

We define similarity as a degree of equality. Similar fuzzy sets have highly overlapping membership functions, and we use a set-theoretic similarity measure based on fuzzy intersection and union to quantify the similarity among the fuzzy sets in the rule base. The proposed algorithm simplifies the rule base by merging fuzzy sets having a similarity higher than a given threshold. Merging creates a common fuzzy set that replaces the occurrence of the merged ones in the rule base. This reduces the number of fuzzy sets that are used in the model, and thus *simplifies* the rule base. If the redundancy of the model is high, rules with equal premise might result from this simplification. The algorithm *reduces* the rule base by merging such rules. The similarity measure also identifies fuzzy sets that can be removed from the premise of the rule base because of similarity to the universal set or to a singleton set.

The proposed algorithm is iterative with only one pair of fuzzy sets being merged per iteration. Merging in different variables takes place independently of one another. The number of rules is not necessarily reduced unlike in other methods that reduce the number of clusters during model identification. The proposed algorithm is also computationally less demanding than methods of this type.

Similarity driven rule base simplification makes it possible to easily create different versions of the fuzzy rule base with varying granularity which can be used for different purposes like control, simulation or a comprehensible linguistic description of the system.

The proposed method has been successfully applied to real problems such as modeling of a fermentor tank pressure dynamics [16] and an air-conditioning system [23]. In this article an application to the fuzzy modeling of enzymatic soil removal has been shown.

## ACKNOWLEDGMENT

The authors would like to thank the Unilever Research Laboratories, Vlaardingen, The Netherlands, for providing the measurements for the experimental part of this research.

## REFERENCES

- [1] L. A. Zadeh, "Fuzzy sets," *Inf. Contr.*, vol. 8, pp. 338–353, 1965.
- [2] U. Kaymak and R. Babuška, "Compatible cluster merging for fuzzy modeling," in *Proc. FUZZ-IEEE/IFES'95*, Yokohama, Japan, Mar. 1995, pp. 897–904.
- [3] C.-T. Sun, "Rule-base structure identification in an adaptive-network-based fuzzy inference system," *IEEE Trans. Fuzzy Syst.*, vol. 2, no. 1, pp. 64–73, 1994.
- [4] B. G. Song, R. J. Marks II, S. Oh, P. Arabshahi, T. P. Caudell, and J. J. Choi, "Adaptive membership function fusion and annihilation in fuzzy if-then rules," in *Proc. FUZZ-IEEE/IFES'93*, 1993, pp. 961–967.
- [5] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 116–132, 1985.
- [6] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*. New York: Wiley, 1994.
- [7] R. Babuška and H. B. Verbruggen, "Fuzzy set methods for local modeling and identification," in *Multiple Model Approaches to Nonlinear Modeling and Control*, R. Murray-Smith and T. A. Johansen, Eds. London, U.K.: Taylor and Francis, 1996.
- [8] R. Babuška, "Fuzzy modeling and identification," Ph.D. dissertation, Dept. Elect. Eng., Contr. Lab., Delft Univ. Technol., Delft, The Netherlands, Dec. 1996.
- [9] W. Remmele and B. Ueberreiter, "Systematic knowledge acquisition in expert systems," *Siemens Rev., R&D Special*, pp. 9–14, Spring 1991.
- [10] E. H. Mamdani, "Application of fuzzy algorithms for control of a simple dynamic plant," *Proc. Inst. Electr. Eng.*, no. 121, pp. 1585–1588, 1974.
- [11] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE CDC*, San Diego, CA, 1979, pp. 761–766.
- [12] M. Sugeno and T. Yashukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7–31, 1993.
- [13] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function*. New York: Plenum, 1981.
- [14] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 773–781, July 1989.
- [15] V. V. Cross, "An analysis of fuzzy set aggregators and compatibility measures," Ph.D. dissertation, Wright State Univ. Dayton, OH, 1993.
- [16] M. Setnes, "Fuzzy rule-base simplification using similarity measures," M.Sc. thesis, Dept. Elect. Eng., Contr. Lab., Delft Univ. Technol., July 1995.
- [17] A. L. Ralescu and D. A. Ralescu, "Probability and fuzziness," *Inf. Sci.*, vol. 34, pp. 85–92, 1984.
- [18] R. Zwick, E. Carlstein, and D. V. Budescu, "Measures of similarity among fuzzy concepts: A comparative analysis," *Int. J. Approx. Reas.*, vol. 1, pp. 221–242, 1987.
- [19] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. New York: Academic, 1980.
- [20] G. J. Klir and B. Youan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [21] D. Dubois and H. Prade, "On the validation of fuzzy knowledge bases," in *Fuzzy Reasoning in Information, Decision and Control Systems*, S. G. Tzafestas and A. N. Venetsanopoulos, Eds. Amsterdam, The Netherlands: Kluwer, 1994, ch. 2, pp. 31–49.
- [22] U. Kaymak, "Application of fuzzy methodologies to a washing process," Chartered Designer thesis, Dept. Elect. Eng., Contr. Lab., Delft Univ. Technol., Delft, The Netherlands, Dec. 1994.
- [23] M. Setnes, R. Babuška, and H. B. Verbruggen, "Complexity reduction in fuzzy modeling," in *Proc. 2nd MATHMOD*, Vienna, Austria, Feb. 1997.



**Magne Setnes** was born in 1970 in Bergen, Norway. He received the engineering degree (ing.) in robotics from the Kongsberg College of Engineering, Kongsberg, Norway, in 1992 and the M.Sc. degree (ir.) in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1995. He is currently pursuing the Ph.D. degree at the Control Laboratory, Delft University of Technology, and working in the field of computational intelligence.

His research interests include fuzzy sets, fuzzy logic, and neuro-fuzzy systems for modeling, control, and decision making.



**Robert Babuška** was born in 1967 in Prague, Czechoslovakia. He received the M.Sc. degree in control engineering from the Czech Technical University, Prague, in 1990, and the Ph.D. degree from the Delft University of Technology, Delft, The Netherlands, in 1997.

He is currently an Associate Professor with the Control Laboratory, Electrical Engineering Department, Delft University of Technology. His main research interests include fuzzy set theory, fuzzy systems modeling, identification, and control.



**Uzay Kaymak** was born in Konya, Turkey in 1968. He received the M.Sc. degree (ir.) in electrical engineering from Delft University of Technology, Delft, The Netherlands, in 1992, where he also received the Degree of Chartered Designer in Information Technology in 1995. He is currently pursuing the Ph.D. degree at the Control Laboratory, Delft University of Technology.

His main research interests include fuzzy set theory, fuzzy decision making, and fuzzy control.

Mr. Kaymak is a student member of the Royal Institution of Engineers, The Netherlands.



**Hans R. van Nauta Lemke** was born in 1924. He received the M.Sc. degree (ir.) in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1950.

After ten years of industrial experience with the Van der Heem Company (now part of Philips), he became a Professor at the Faculty of Electrical Engineering, Delft University of Technology, in 1959. He was dean of the Faculty of Electrical Engineering from 1963 to 1965 and 1986 to 1989, Rector Magnificus from 1970 to 1973, and Director of TNO-Defence Research from 1978 to 1981. He retired in 1989, but is still active in the field of fuzzy mathematics, fuzzy control, and decision making.

Mr. van Nauta Lemke is an honorary member of Fuzzy Logic Netherlands and of the Royal Institution of Engineers, The Netherlands.