Compact fuzzy models through complexity reduction and evolutionary optimization

Hans Roubos, hans@ieee.org Magne Setnes, magne@ieee.org
Delft University of Technology, Faculty of Information Technology and Systems,
Control Engineering Laboratory, P.O. Box 5031, 2600 GA Delft, the Netherlands

Abstract— **Genetic Algorithms (GAs) and other evolutionary optimization methods to design fuzzy rules from data for systems modeling and classification have received much attention in recent literature. We show that different tools for modeling and complexity reduction can be favorably combined in a scheme with GA-based parameter optimization. Fuzzy clustering, rule reduction, rule base simplification and constrained genetic optimization are integrated in a data-driven modeling scheme with low human intervention. Attractive models with respect to compactness, transparency and accuracy, are the result of this symbiosis.**

I. Introduction

We focus on learning fuzzy rules from data with low human intervention. Many tools to initialize, tune and manipulate fuzzy models have been developed. We show that different tools can be favorably combined to obtain compact fuzzy rule-based models of low complexity with still good approximation accuracy. A modeling scheme is presented that combine four previously studied tools for rule-based modeling: fuzzy clustering [1], rule reduction by orthogonal techniques [2], similarity driven simplification [3], and evolutionary optimization [4].

Fuzzy clustering [1] is used to obtain an initial rule-based model from sampled data. Since rules obtained and tuned by data-driven techniques often contain redundancy in terms of similar (overlapping) fuzzy sets, similarity driven rule base simplification is applied to [3] detect and merges compatible fuzzy sets in the model and to remove "don't-care" terms. To reduce the number of rules, we apply a simple QR decomposition based rule reduction technique proposed in [2]. Finally, since these methods are based on the separate identification and manipulation of the models premise and consequent parts, a constrained real-coded GA is applied to simultaneously fine-tune (optimize) all parameters in the resulting rule-base. In [4], we showed that such a GA was able to strongly improve the models performance by small alterations to the rules.

By combining these tools in an iterative loop we propose a powerful fuzzy modeling scheme. The algorithm starts with an initial model, obtained here by means of fuzzy clustering in the product space of measured in- and outputs. Successively, rule reduction, rule base simplification and GA-based optimization are applied in an iterative manner. The GA performs a multi-criterion search for model accuracy while trying to exploit the possible redundancy in the model. In the next iteration, this redundancy will be used by the rule reduction and rule base simplification tools to reduce and simplify the rule base. The result is a compact fuzzy rule base of low complexity with high accuracy. When the iterations terminate, a final GA-based optimization is performed to increase accuracy and transparency, as opposed to the GA in the iterative loop which tries to exploit redundancy.

The next section discusses data-driven fuzzy modeling. Then, in Section III the rule reduction and rule base simplification tools are described. Section IV presents the GA-based optimization strategy, and in Section V the resulting, total modeling scheme is given. In Section VI, the method is demonstrated on a nonlinear dynamic systems model known from the literature, and the results are compared to other methods published. Section VII concludes the paper.

II. Data-driven modeling

A. The TS fuzzy model

Rule-based models of the Takagi-Sugeno (TS) type [7] are especially suitable for the approximation of dynamic systems. The rule consequents are often taken to be linear functions of the inputs:

$$R_i: \textbf{If } x_1 \text{ is } A_{i1} \textbf{ and } \ldots x_n \text{ is } A_{in} \textbf{ then } \hat{y}_i = \zeta_{i1}x_1 + \ldots, \zeta_{in}x_n + \zeta_{i(n+1)},\ i = 1, \ldots, M. \quad (1)$$

Here $x = [x_1, x_2, \ldots, x_n]^T$ is the input vector, $\hat{y}_i$ is the output of the ith rule, and $A_{i1}, \ldots, A_{in}$ are fuzzy sets defined in the antecedent space by membership functions $\mu_{A_{ij}}(x_j)$:$\mathbb{R} \rightarrow [0, 1]$. ζ_{ij} are the consequent parameters and M is the number of rules. The total output of the model is computed by aggregating the individual contributions of the rules:

$$\hat{y} = \sum_{i=1}^{M} p_i(x)\hat{y}_i\,, \quad (2)$$

where $p_i(x)$ is the normalized firing strength of the ith rule:

$$p_i(x) = \frac{\prod_{j=1}^{n} A_{ij}(x_j)}{\sum_{i=1}^{M}\prod_{j=1}^{n} A_{ij}(x_j)},\ i = 1, 2, \ldots, M. \quad (3)$$

In the following, we will apply the frequently used triangular membership functions μ_{ij} to describe the fuzzy sets A_{ij} in the rule antecedents

$$\mu(x; a, b, c) = \max\left(0, \min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right)\right). \quad (4)$$

B. Identification from data

Given N input-output data pairs $\{x_k, y_k\}$, the typical identification of the TS model is done in two steps: first the fuzzy rule antecedents are determined, and then least squares parameter estimation is applied to determine the consequents [1], [8]. In the examples in this paper, the antecedents of the initial fuzzy rule

0-7803-5877-5/00/$10.00 © 2000 IEEE

bases are obtained from fuzzy c-means clustering in the product space of the sampled input-output data. Following the approach in [1], [9], each cluster represent a certain region in the systems input-output state-space, and corresponds to a rule in the rule base. The fuzzy sets in the rule antecedent is obtained by projecting the cluster onto the domain of the various inputs.

C. Transparency and accuracy

The initial rule base constructed by fuzzy clustering typically fulfills many criteria for transparency and good semantic properties [10]:

- *Moderate number of rules*: fuzzy clustering helps ensure a comprehensive sized rule base with rules that describe important regions in the data.
- *Normality*: by fitting parameterized functions to the projected clusters, normal and comprehensive membership functions are obtained that can be taken to represent linguistic terms.
- *Coverage*: the deliberate overlap of the clusters (rules) and their position in populated regions of the input-output data space ensure that the model is able to derive an output for all occurring inputs.

However, the approximation capability of the initial rule base remains suboptimal. The projection of the clusters onto the input variables, and their approximation by parametric functions like, e.g., triangular fuzzy sets, introduce a structural error since the resulting premise partition differs from the cluster partition matrix. Also, the separated identification of the rule antecedents and the rule consequents prohibits interactions between them during modeling. To improve the approximation capability of the initial model, a GA based optimization method is applied.

The transparency and compactness of the initial rule base are often also subject to improvement. The distinguishability of the rules and the terms (fuzzy sets) resulting from the projection depends, among others, on the difficult determination of the correct number of clusters (rules) in the data an their position in the product space. To reduce the rule base complexity, we apply two methods for rule reduction and rule base simplification, respectively, as explained in the next section.

III. Rule base reduction and simplification

The complexity of a rule base is determined by the number of rules, and the number of different fuzzy sets used in the rule antecedents. Both can be reduced by the techniques described in this section.

A. Rule reduction with P-QR

Various rule reduction strategies based on rank-revealing techniques like the singular value decomposition (SVD) have been proposed for fuzzy models [6]. Such techniques transform the rule selection problem into the problem of picking the most influential columns of the firing matrix $\mathrm{P} = [p_1, p_2, \ldots, p_M] \in \mathbb{R}^{N \times M}$, which contains the firing strength of all the M rules for the N inputs $\boldsymbol{x}_k$. The elements in the columns $p_i = [p_{1i}, p_{2i}, \ldots, p_{Ni}]^T$ are the normalized firing strengths calculated as in (3).

In [2] it is shown that a simple pivoted QR (P-QR) decomposition of the models firing matrix P can be used to obtain an importance ordering of the rules in the rule base without the need for estimating the SVD of P. Moreover, unlike comparable methods such as the SVD-QR [11], the ordering produced by the P-QR does not depend on an estimate of the effective rank of P.

The QR decomposition of P is given by $\mathrm{P}\Pi = \mathrm{QR}$, where $\Pi \in \mathbb{R}^{M \times M}$ is a permutation matrix, $\mathrm{Q} \in \mathbb{R}^{N \times M}$ has orthonormal columns and $\mathrm{R} \in \mathbb{R}^{M \times M}$ is upper triangular. If P has full rank, then R is nonsingular (invertible). When P is (near) rank deficient, it is desirable to select the permutation matrix Π such that the rank deficiency is exhibited in R, having a small lower right block $\mathrm{R}_{kk} \in \mathbb{R}^{k \times k}$ [12]:

$$\mathrm{R} = \begin{bmatrix} \mathrm{R}_{11} & \mathrm{R}_{12} \\ 0 & \mathrm{R}_{kk} \end{bmatrix}$$

It can be shown that for the $M - k + 1$'th singular value of P, we have $\sigma_{M-k+1}(\mathrm{P}) \leq \|\mathrm{R}_{kk}\|$. Therefore, if $\|\mathrm{R}_{kk}\|$ is small, then P has at least k small singular values.

The QR decomposition is uniquely determined by the permutation matrix Π which can be computed by a column pivoting strategy [13]. The pivoting algorithm favors columns of R with a high norm, related (through orthogonalization) to the norm of the columns of P. For a fuzzy rule base the norms of the columns of P correspond to the firing strength and/or firing frequency of the rules. Thus, P-QR picks first the most active and least redundant of the remaining rules.

B. Similarity driven rule base simplification

The similarity driven rule base simplification method was proposed in [3]. A similarity measure is used to quantify the redundancy among the fuzzy sets in the rule base. Similar fuzzy sets, representing compatible concepts, are merged in order to obtain a generalized concept represented by a new fuzzy set that replaces the similar ones in the rule base. This reduces the number of different fuzzy sets (linguistic terms) used in the model. The similarity measure is also used to detect "don't care" terms, i.e., fuzzy sets in which all elements of a domain have a membership close to 1. Similarity driven simplification differ from rule reduction in that it is driven by the similarity among fuzzy sets defined on the domain of the same antecedent variable, and not in the product space of the inputs. Thus, the models term set can be reduced without necessarily any rules being removed.

We apply a similarity measure based on the set-theoretic operations of intersection and union:

$$S(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

where $|.|$ denotes the cardinality of a set, and the $\cap$ and $\cup$ operators represent the intersection and union respectively. For discrete domains $X = \{x_j | j = 1, 2, \ldots, m\}$, this can be written as:

$$S(A, B) = \frac{\sum_{j=1}^{m} [\mu_A(x_j) \wedge \mu_B(x_j)]}{\sum_{j=1}^{m} [\mu_A(x_j) \vee \mu_B(x_j)]} \tag{6}$$

where $\wedge$ and $\vee$ are the minimum and maximum operators, respectively. S is a symmetric measure in [0,1]. If $S(i, j) = 1$, then the two membership functions A_i and A_j are equal and $S(i, j)$ becomes 0 when the membership functions are non overlapping.

IV. Real Coded Genetic Algorithm

A real-coded GA [14] is used for the simultaneous optimization of the parameters of the antecedent membership functions and the rule consequents. The main aspects of the proposed GA are discussed below and the implementation is summarized in Section IV-E.

A. Fuzzy model representation

Chromosomes are used to describe the solutions. With a population size L, we encode the parameters of each fuzzy model (solution) in a chromosome $s_l, l = 1, \ldots, L$, as a sequence of elements describing the fuzzy sets in the rule antecedents followed by the parameters of the rule consequents. A TS model with M fuzzy rules is encoded as

$$s_l = (\mathrm{ant}_1, \ldots, \mathrm{ant}_M, \zeta_1, \ldots, \zeta_M) \ , \tag{7}$$

where ζ_i contains the consequent parameters ζ_{iq} of rule R_i, and $\mathrm{ant}_i = (a_{i1}, b_{i1}, c_{i1}, \ldots, a_{in}, b_{in}, c_{in},)$ contains the parameters of the antecedent fuzzy sets $A_{ij}, j = 1, \ldots, n$, according to (4). In the initial population $\mathrm{S}^0 = \{s_1^0, \ldots, s_L^0\}$, s_1^0 is the initial model, and $s_2^0, \ldots, s_L^0$ are created by random variation with a uniform distribution around s_1^0.

B. Selection function

The *roulette wheel* selection method [14] is used to select n_C chromosomes for operation. The chance on the roulette-wheel is adaptive and is given as $P_l / \sum_{l'} P_{l'}$, where

$$P_l = \left(\frac{1}{J_l}\right)^2 \ , \quad l, l' \in \{1, \ldots, L\}, \tag{8}$$

and J_l is the performance of the model encoded in chromosome s_l. The inverse of the selection function (P_l^{-1}) is used to select chromosomes for deletion. The best chromosome is always preserved in the population (*Elitist* selection).

C. Genetic operators

Two classical operators, *simple arithmetic crossover* and *uniform mutation*, and four special real-coded operators are used in the GA. In the following, $r \in [0, 1]$ is a random number (uniform distribution), $t = 0, 1, \ldots, T$ is the generation number, s_v and s_w are chromosomes selected for operation, $k \in \{1, 2, \ldots N\}$ is the position of an element in the chromosome, and v_k^{min} and v_k^{max} are the lower and upper bounds, respectively, on the parameter encoded by element k:

1. *Uniform mutation*; a random selected element v_k, $k \in \{1, 2, \ldots, N\}$ is replaced by v_k' which is a random number in the range $[v_k^{min}, v_k^{max}]$. The resulting chromosome is $s_v^{t+1} = (v_1, \ldots, v_k', \ldots, v_m)$.
2. *Multiple uniform mutation*; uniform mutation of n randomly selected elements, where n is also selected at random from $\{1, \ldots, N\}$.
3. *Gaussian mutation*; all elements of a chromosome are mutated such that $s_v^{t+1} = (v_1', \ldots, v_k', \ldots, v_m')$ where $v_k' = v_k + f_k$, $k = 1, 2, \ldots, N$. Here f_k is a random number drawn from a *Gaussian* distribution with zero mean and an adaptive variance $\sigma_k = (\frac{T-t}{T})(\frac{v_k^{max} - v_k^{min}}{3})$. tuning performed by this operator becomes finer and increases.
4. *Simple arithmetic crossover*; s_v^t and s_w^t are crossed over at the kth position. The resulting offsprings are: $s_v^{t+1} = (v_1, \ldots, v_k, w_{k+1}, \ldots, w_N)$ and $s_w^{t+1} = (w_1, \ldots, w_k, v_{k+1}, \ldots, v_N)$, where k is selected at random from $\{2, \ldots, N-1\}$.
5. *Whole arithmetic crossover*; a linear combination of s_v^t and s_w^t resulting in $s_v^{t+1} = r(s_v^t) + (1-r)s_w^t$ and $s_w^{t+1} = r(s_w^t) + (1-r)s_v^t$.
6. *Heuristic crossover*; s_v^t and s_w^t are combined such that $s_v^{t+1} = s_v^t + r(s_w^t - s_v^t)$ and $s_w^{t+1} = s_w^t + r(s_v^t - s_w^t)$.

D. Constraints

The optimization performed by the GA is subjected to two types of constraints: *partition* and *search space*. The partition constraint prohibits gaps in the partitions of the input (antecedent) variables. The coding of a fuzzy set must comply with (4), i.e., $a \le b \le c$. To avoid gaps in the partition, pairs of neighboring fuzzy sets are constrained by $a_R \le c_L$, where L and R denote left and right set, respectively.

The GA search space is constrained by two user defined bound-parameters, α_1 and α_2, that applies to the antecedent and the consequent parameters of the rules, respectively. The first bound, α_1, is intended to maintain the distinguishability of the models term set (the fuzzy sets) by allowing the parameters describing the fuzzy sets A_{ij} to vary only within a bound of $\pm\alpha_1|\mathcal{X}_j|$ around their initial values, where $|\mathcal{X}_j|$ is the length (range) of the domain on which the fuzzy sets A_{ij} are defined. The second bound, α_2, is intended to maintain the local-model interpretation of the rules by allowing the qth consequent parameter of the ith rule, ζ_{iq}, to vary within a bound of $\pm\alpha_2(\max_i(\zeta_{iq}) - \min_i(\zeta_{iq}))$ around its initial value.

The search space constraints are coded in the two vectors, $v^{max} = [v_1^{max}, \ldots, v_N^{max}]$ and $v^{min} = [v_1^{min}, \ldots, v_N^{min}]$, giving the upper and lower bounds on each of the N elements in a chromosome. During generation of the initial partition, and in the case of a uniform mutation, elements are generated at random within these bounds.

E. Genetic algorithm

Given the pattern matrix Z and a fuzzy rule base, select the number of generations T, the population size L, the number of operations n_C and the constraints α_1 and α_2. Let S^t be the current population of solutions $s_l^t, l = 1, \ldots, L$, and let J^t be the vector of corresponding values of the evaluation function:

1. Create initial chromosome s_1^0 from the fuzzy rule base.
2. Calculate the constraint vectors v^{min} and v^{max} using s_1^0, α_1 and α_2.
3. Create the initial population $\mathrm{S}^0 = \{s_1^0, \ldots, s_L^0\}$ where $s_l^0, l = 2, \ldots, L$ are created by constrained random variations around s_1^0, and the partition constraints apply.
4. *Repeat genetic optimization for* $t = 0, 1, 2, \ldots, T-1$:
   (a) Evaluate S^t and obtain J^t.
   (b) Select n_C chromosomes for operation.
   (c) Select n_C chromosomes for deletion.
   (d) Operate on chromosomes acknowledging the search space constraints.
   (e) Implement partition constraints.

(f) Create new population S^{t+1} by substituting the operated chromosomes for those selected for deletion.

5. Select best solution from S^T by evaluating J^T.

V. Proposed fuzzy modeling scheme

We propose a fuzzy modeling approach that combines the modeling, tuning and complexity reduction tools described above. An initial fuzzy model is first obtained from data by fuzzy clustering. Then, the model is successively reduced, simplified and optimized in an iterative fashion. After termination of the iterations, a last GA- based fine tuning is done.

1. *Initialization*: Obtain initial fuzzy model.
2. *Complexity reduction*: Repeat until termination:
   (a) Rule reduction according to user preference after visual inspection of P-QR ranking
   (b) Similarity driven rule base simplification
   (c) GA optimization with redundancy objective: Model accuracy while exploiting redundancy
3. GA fine tuning with transparency objective: Model accuracy with well separated fuzzy sets

In step 2 of the algorithm, user intervention is needed. The rule ordering by the P-QR is visualized for the user to select the number of rules to remove. Step 2 terminates when the rule base can not be further reduced or simplified.

The model accuracy is measured in terms of the *mean squared error* (MSE):

$$MSE = \frac{1}{K}\sum_{k=1}^{K}(y_k - \hat{y}_k)^2\,, \tag{9}$$

where y is the true output and $\hat{y}$ is the model output. In the GA-based optimization, the MSE is combined with a similarity measure. In step 2, similarity is rewarded, that is, the GA tries to emphasize the redundancy in the model. This redundancy is then used to remove unnecessary rules or fuzzy sets in the next iteration. In step 3 of the algorithm, the final fine tuning, similarity among fuzzy sets is penalized to obtain a distinguishable term set for linguistic interpretation. The objective the GA is to minimize the cost function

$$J = (1 + \lambda S^*) \cdot MSE\,, \tag{10}$$

where $S^* \in [0, 1]$ is a the average of the maximum pairwise similarity present in each input, i.e., S^* is an aggregated similarity measure for the total model. The weighting function $\lambda \in [-1, 1]$ determines whether similarity is rewarded ($\lambda < 0$) or penalized ($\lambda > 0$).

VI. Example: Nonlinear plant

We consider the 2^{nd} order nonlinear plant studied by Wang and Yen in [15], [16], [6]:

$$y(k) = g(y(k-1), y(k-2)) + u(k), \text{ with} \tag{11}$$

$$g(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)(y(k-1) - 0.5)}{1 + y^2(k-1)y^2(k-1)}. \tag{12}$$

The goal is to approximate the nonlinear component $g(y(k-1), y(k-2))$ of the plant with a fuzzy model. In [15], 400 simulated data points were generated from the plant model (11). 200 samples of identification data were obtained with a random input signal $u(k)$ uniformly distributed in $[-1.5, 1.5]$, followed by 200 samples of evaluation data obtained using a sinusoid input signal $u(k) = \sin(2\pi k/25)$ (Fig. 1).

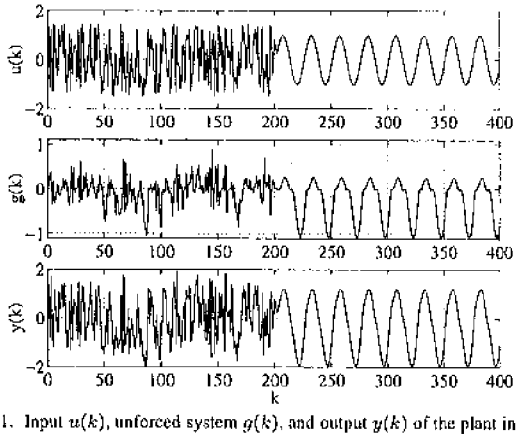

Fig. 1. Input $u(k)$, unforced system $g(k)$, and output $y(k)$ of the plant in (11).

A. Solutions in the literature

We compare our results, with those obtained by the three different approaches described below. The best results obtained in each case are summarized in Table I.

In [15] a GA was combined with a Kalman filter to obtain a fuzzy model of the plant. The antecedent fuzzy sets of 40 rules, encoded by Gaussian membership functions, were determined initially by clustering and kept fixed. A binary GA was used to select a subset of the initial 40 rules in order to produce a more compact rule base with better generalization properties. The consequents of the various models in the GA population were estimated after each generation by the Kalman filter, and an information criterion was used as the evaluation function to balance the trade-off between the number of rules and the model accuracy.

In [16] various information criteria was used to *successively* pick rules from a set of 36 rules in order to obtain a compact, but accurate model. The initial rule base was obtained by partitioning each of the two inputs $y(k-1)$ and $y(k-2)$ by six equally distributed fuzzy sets. The rules were picked in an order determined by an orthogonal transform.

In [6] various orthogonal transforms for rule selection and rule ordering were studied using an initial model with 25 rules. In this initial model, 20 rules were obtained by clustering, while five redundant rules were added to evaluate the selection performance of the studied techniques.

B. Proposed approach

We applied both the modeling approach proposed in Section V and its predecessor, presented in [4], which does not contain the second step (the complexity reduction) of the scheme proposed in Section V. For both methods, TS models with as well singleton as linear consequent functions were studied. The GA was applied with $L = 40$, $n_C = 10$, $\alpha_1 = 25\%$, $\alpha_2 = 25\%$ and $T = 1000$ in the final optimization and $T = 200$ in the complexity reduction step. The threshold for set merging was

TABLE I

FUZZY MODELS FOR THE DYNAMIC PLANT. ALL MODELS ARE OF THE TAKAGI-SUGENO TYPE.

Ref.	No. of rules	No. of sets	Consequent	MSE train	MSE eval
[15]	40 rules (initial)	40 Gaussians (2D)	Singleton	$3.3e^{-4}$	$6.9e^{-4}$
	28 rules (optimized)	28 Gaussians (2D)	Singleton	$3.3e^{-4}$	$6.0e^{-4}$
[16] [1]	36 rules (initial)	12 B-splines	Singleton	$2.8e^{-5}$	$5.1e^{-3}$
	23 rules (optimized)	12 B-splines	Singleton	$3.2e^{-5}$	$1.9e^{-3}$
	36 rules (initial)	12 B-splines	Linear	$1.9e^{-6}$	$2.9e^{-3}$
	24 rules (optimized)	12 B-splines	Linear	$2.0e^{-6}$	$6.4e^{-4}$
[6]	25 rules (initial)	25 Gaussians (2D)	Singleton	$2.3e^{-4}$	$4.1e^{-4}$
	20 rules (optimized)	20 Gaussians (2D)	Singleton	$6.8e^{-4}$	$2.4e^{-4}$
This paper	7 rules (initial)	14 triangulars	Singleton	$1.2e^{-2}$	$3.5e^{-2}$
without step 2	7 rules (optimized)	14 triangulars	Singleton	$1.9e^{-3}$	$7.5e^{-3}$
	5 rules (initial)	10 triangulars	Linear	$3.8e^{-3}$	$2.0e^{-3}$
	5 rules (optimized)	10 triangulars	Linear	$6.1e^{-4}$	$3.0e^{-4}$
This paper	10 rules (initial)	10 triangulars	Singleton	$1.4e^{-2}$	$1.5e^{-2}$
Fig. 2	6 rules (optimized)	5 triangulars	Singleton	$1.4e^{-3}$	$7.6e^{-4}$
	7 rules (initial)	14 triangulars	Linear	$1.8e^{-3}$	$1.0e^{-3}$
Fig. 3	5 rules (optimized)	5 triangulars	Linear	$5.0e^{-3}$	$4.2e^{-4}$

[1] The low MSE on the training data is in contrast to the MSE for the evaluation data which indicates overtraining.

0.5 and 0.8 for removing sets similar to the universal set ("don't care" terms).

Without complexity reduction: First a *singleton* TS model consisting of seven rules was obtained by fuzzy c-means clustering and genetic optimization. The MSE for both training and validation data were comparable, indicating that the initial model is not over-fitted. By GA optimization, the MSE was reduced by 84% from $1.2e^{-2}$ to $1.9e^{-3}$ on the training data, and by 79% from $3.5e^{-2}$ to $7.5e^{-3}$ on the evaluation data.

Then a TS model with linear consequents was considered. Because of the more powerful approximation capabilities of the functional consequents, an initial model of only five rules was constructed by clustering. The MSE for both training and validation data were, as expected, better than for the singleton model. Moreover, the result on the validation data (low frequency signal) is twice as good as on the identification data, indicating the generality of the obtained model. By GA optimization, the MSE was reduced by 84% from $3.8e^{-3}$ to $6.1e^{-4}$ on the training data, and by 85% from $2.0e^{-3}$ to $3.0e^{-4}$ on the evaluation data.

With complexity reduction: The proposed method in Section V including the complexity reduction step (step 2) was considered. Due to the possibility of rule reduction, an initial singleton TS model with as much as 10 fuzzy rules and in total 20 fuzzy sets was constructed by clustering (Fig. 2 *Top*).

During the iterative complexity reduction step, in each iteration the model was sought reduces, simplified and finally optimized by the GA. The model was reduced as follows: (i) simplification reduces from 10 + 10 to 9 + 5 fuzzy sets, (ii) simplification reduces to 7 + 4 sets, (iii) rule reduction removes three rules, resulting in 7 rules and 5 + 4 sets, (iv) simplification reduces to 4 + 2 sets, and (v) one rule was removed. The final model, has only 6 rules, using 3 + 2 fuzzy sets (Fig. 2 *Middle*). The identification and validation results as well as the prediction error, are presented in Fig. 2 *Bottom*. The resulting singleton TS model is compact and has good approximation properties, except in the low region were almost no data was provided. The reduced model with 6 rules and 5 sets is as accurate as the initial model with 7 rules and 14 sets.

Finally a TS model with linear consequents was studied. The initial model was obtained with 7 clusters, resulting in a model with 7 rules and 14 fuzzy sets (Fig. 3 *Top*). The model was reduced as follows: (i) simplification reduces from 7 + 7 to 5 + 5 fuzzy sets, (ii) rule reduction removes two rules resulting in 5 rules and 3 + 4 sets, (iii) simplification reduces to 2 + 4 sets, and (iv) simplification to 2 + 3 sets. The resulting TS model with linear consequents has only 5 rules using 2 + 3 fuzzy sets (Fig. 3 *Middle*). The identification and validation results as well as the prediction error, are presented in Fig. 3 *Bottom*. The approximation properties are better than for the singleton TS model (Fig. 2 *Bottom*). The linear consequent TS model also extrapolates well and the difficult part in the low region is nicely approximated. Once again, the reduced and optimized TS model with 5 rules and 5 sets is comparable in accuracy to the initial TS model with 7 rules and 14 fuzzy sets.

From the results summarized in Table I, we see that the proposed modeling approach is capable of obtaining good results using fewer rules and fuzzy sets than other approaches reported in the literature. Moreover, simple triangular membership functions were used as opposed to cubic B-splines in [16] and Gaussian-type basis functions in [15], [6]. step, not only accurate, but also compact and

VII. CONCLUSION

We have described an approach to construct compact and transparent, yet accurate fuzzy rule-based models from measured input-output data. Several methods for modeling, complexity reduction and optimization are combined in the approach. Fuzzy clustering is first used to obtain an initial rule base. Rule reduction, similarity based simplification and GA-based optimization are then used in an iterative manner to decrease the complexity of the model while maintaining high ac-

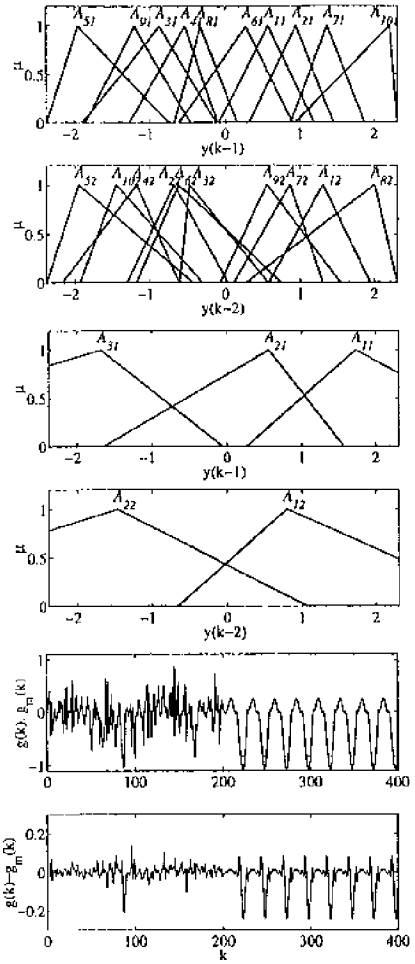

Fig. 2. Singleton consequent TS model; *Top:* Initial model (10 rules, 20 fuzzy sets). *Middle:* Simplified initial model (6 rules, 5 fuzzy sets). *Bottom:* Identification and validation for the optimized fuzzy model.

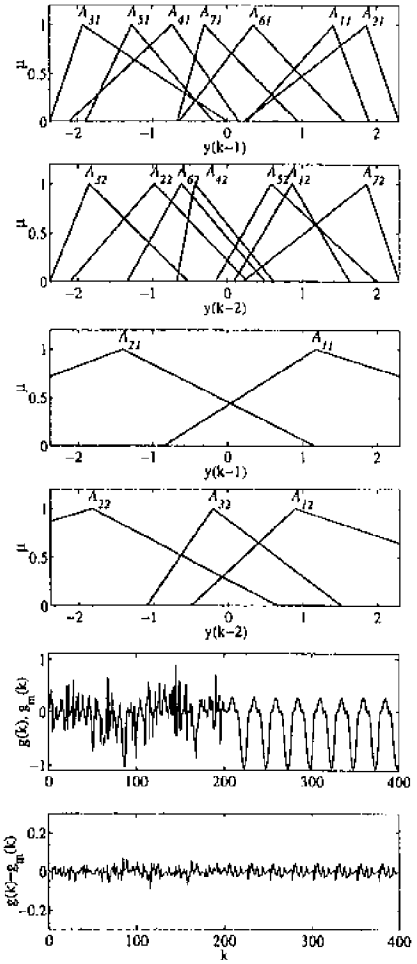

Fig. 3. Linear consequent TS model; *Top:* Initial model (7 rules, 14 fuzzy sets), *Middle:* Simplified initial model (5 rules, 5 fuzzy sets), *Bottom:* Identification and validation for the optimized fuzzy model.

curacy. We successfully applied the proposed algorithm to a problem known from the literature. The accuracy of the obtained models were comparable to the results reported in the literature. However, the obtained models use fewer rules and fuzzy sets than other models reported in the literature.

REFERENCES

[1] R. Babuška, *Fuzzy Modeling for Control*, Kluwer Academic Publishers, Boston, 1998.

[2] M. Setnes and H. Hellendoorn, "Orthogonal transforms for ordering and reduction of fuzzy rules," in *FUZZ-IEEE*, San Antonio, Texas, USA, 2000.

[3] M. Setnes, R. Babuška, U. Kaymak, and H. R. van Nauta Lemke, "Similarity measures in fuzzy rule base simplification," *IEEE Trans. Systems, Man and Cybernetics - Part B*, vol. 28, pp. 376–386, 1998.

[4] M. Setnes and J.A. Roubos, "Transparent fuzzy modeling using fuzzy clustering and GA's," in *NAFIPS*, New York, USA, June 10-12, 1999, pp. 198–202.

[5] J. Hohensohn and J. M. Mendel, "Two-pass orthogonal least-squares algorithm to train and reduce fuzzy logic systems," in *FUZZ-IEEE*, Orlando, USA, 1994, pp. 696–700.

[6] J. Yen and L. Wang, "Simplifying fuzzy rule-based models using orthogonal transformation methods," *IEEE Trans. Systems, Man and Cybernetics - Part B*, vol. 29, pp. 13–24, 1999.

[7] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Systems, Man and Cybernetics*, vol. 15, pp. 116–132, 1985.

[8] J. Yen, C. Liao, B. Lee, and D. Randolph, "A hybrid approach to modeling metabolic systems using genetic algorithm and simplex method," *IEEE Trans. Systems, Man, and Cybernetics - Part B*, vol. 28, pp. 173–191, 1998.

[9] M. Setnes, R. Babuška, and H. B. Verbruggen, "Rule-based modeling: Precision and transparency," *IEEE Trans. Systems, Man and Cybernetics - Part C*, vol. 28, pp. 165–169, 1998.

[10] J. Valente de Oliveira, "Semantic constraints for membership function optimization," *IEEE Trans. Systems, Man and Cybernetics Part A*, vol. 29, pp. 128–138, 1999.

[11] G.C. Mouzouris and J.M. Mendel, "Designing fuzzy logic systems for uncertain environments using a singular-value-qr decomposition method," in *FUZZ-IEEE'96*, New Orleans, USA, September 1996, pp. 295–301.

[12] G.W. Stewart, "Rank degeneracy," *SIAM J. Sci. and Stat. Comput.*, vol. 5, pp. 403–413, 1984.

[13] G.H. Golub, "Numerical methods for solving least squares problems," *Numerical Mathematics*, no. 7, pp. 206–216, 1965.

[14] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Verlag, New York, 2nd edition, 1994.

[15] L. Wang and J. Yen, "Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter," *Fuzzy Sets and Systems*, vol. 101, pp. 353–362, 1999.

[16] J. Yen and L. Wang, "Application of statistical information criteria for optimal fuzzy model construction," *IEEE Trans. Fuzzy Systems*, vol. 6, pp. 362–371, 1998.