Rule Base Reduction: Some Comments on the Use of Orthogonal Transforms

Magne Setnes, *Member, IEEE,* and Robert Babuška

***Abstract*—This paper comments on recent publications about the use of orthogonal transforms to order and select rules in a fuzzy rule base. The techniques are well known from linear algebra, and we comment on their usefulness in fuzzy modeling. The application of rank-revealing methods based on singular value decomposition (SVD) to rule reduction gives rather conservative results. They are essentially subset selection methods, and we show that such methods do not produce an "importance ordering" contrary to what has been stated in literature. The orthogonal least-squares (OLS) method, which evaluates the contribution of the rules to the output, is more attractive for systems modeling. However, it has been shown to sometimes assign high importance to rules that are correlated in the premise. This hampers the generalization capabilities of the resulting model.**

We discuss the performance of rank-revealing reduction methods and advocate the use of a less complex method based on the pivoted QR decomposition. Further, we show how detection of redundant rules can be introduced in OLS by a simple extension of the algorithm. The methods are applied to a problem known from the literature and compared to results reported by other researchers.

***Index Terms*—Fuzzy rule-based modeling, orthogonal transforms, rule reduction.**

I. Introduction

FUZZY MODELS describe systems by establishing relations between the relevant variables in the form of *if-then* rules that are to a certain degree transparent to interpretation and analysis. For approximation tasks, data-driven fuzzy modeling is becoming more and more popular. Most such modeling techniques are driven by the optimization of some cost function and do not consider the complexity and inspectability of the resulting rule base [1]. Consequently, there has been a lot of focus on methods for improving these aspects, considering various issues like local rule behavior [2], statistical information criteria [3], similarities in the rule base [4], [5], conditional clustering [6], [7], and constrained learning [8], to mention some.

One approach that has received much attention in recent literature is the use of orthogonal transforms for ordering and reducing the rules in a rule base [9]–[13]. Orthogonal transforms are well known from statistics and linear algebra, where they are typically applied to subset selection and rank-deficient least-squares problems.

An overview of some orthogonal-based rule reduction methods is given in [12]. All approaches assume the formulation of the modeling exercise as a regression problem, and they can roughly be divided into two groups: the rank revealing ones and those that evaluate the individual contribution of the rules. Rank-revealing methods are typically based on the determination of the effective rank of the rule firing matrix from its singular values. In practice, this is difficult to determine, and the methods are rather conservative with respect to rule-reduction. Also, the claimed "importance ordering" [3], [12] of the rules depends on this estimate and thus often makes no sense. Methods that evaluate the output contribution of the rules to obtain an ordering, like the orthogonal least-squares approach (OLS) [9], [14], are more attractive for systems modeling. However, OLS methods proposed for fuzzy models, [9], [10], do not consider the structure of the rule base in terms of redundant (similar) and correlated rules. Evaluating only the approximation capabilities of the rules, the OLS method often assigns high importance to a set of redundant or correlated rules [12]. This can result in poor generalization capabilities of the model.

In the next section, we discuss the working of rank-revealing reduction methods and advocate the use of the simple pivoted QR decomposition as opposed to the calculation and inspection of the singular values of the firing matrix. The proposed method is computationally simple and can produce a rule ordering without any estimate of the efficient rank. Moreover, it is known to track the singular values of a matrix with enough precision to estimate the effective rank for subset selection [15]. In Section III, we show that detection of redundant and correlated rules can be introduced in OLS-based rule selection by a simple extension to the algorithm. Section IV considers the problem studied in [12], and it is shown that the pivoted QR decomposition and the extended OLS algorithm give better results than other orthogonal methods. Finally, Section V ends the paper with some concluding remarks.

II. Data-Driven Modeling and Rule Reduction

A. Fuzzy Modeling

The most commonly used model for data-driven fuzzy modeling is the Takagi–Sugeno (T–S) fuzzy model [16]. It describes local input-output (I/O) relations using fuzzy rules with consequents that are usually linear combinations of the inputs or simply constant values like

$$R_i\text{: } \mathbf{If}\ x_1 \text{ is } A_{i1}\ \mathbf{and} \ldots \mathbf{and}\ x_n \text{ is } A_{in} \quad \mathbf{then}\ \hat{y}_i = c_i, \quad i = 1, 2, \ldots, M. \tag{1}$$

Manuscript received May 21, 1999; revised May 31, 2001. This work was supported in part by the Research Council of Norway. This paper was recommended by Associate Editor R. Popp.

M. Setnes is with Research and Development, Heineken Technical Services, Zoeterwoude, The Netherlands (e-mail: magne@ieee.org).

R. Babuška is with the Systems and Control Engineering Group, Electrical Engineering Department, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands (e-mail: r.babuska@its.tuldeft.nl).

Publisher Item Identifier S 1094-6977(01)07600-3.

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

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

where $p_i(\boldsymbol{x})$ is the normalized firing strength of the ith rule

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

Given N I/O data pairs $\{\boldsymbol{x}_k, y_k\}$, the model in (2) can be written as a linear regression problem

$$\boldsymbol{y} = \mathbf{P}\theta + \boldsymbol{e} \tag{4}$$

where

$\boldsymbol{y} = [y_1, y_2, \ldots, y_N]^T$ measured outputs;
$\theta = [c_1, c_2, \ldots, c_M]^T$ consequents of the M rules;
$\boldsymbol{e} = [e_1, e_2, \ldots, e_N]^T$ vector of approximation errors.

The matrix $\mathbf{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M] \in \mathbb{R}^{N\times M}$ contains the firing strength of all the M rules for the N inputs $\boldsymbol{x}_k$, where $\boldsymbol{p}_i = [p_{1i}, p_{2i}, \ldots, p_{Ni}]^T$.

Typically, the identification is a two-step approach. First, the fuzzy sets A_{ij} are determined and the firing matrix $\mathbf{P}$ is calculated from (3). In the second step, the rule consequents are determined. This problem is linear in the parameters, and θ can be determined using some least-squares parameter estimation technique [1], [2], [17].

B. Rule Selection with Rank-Revealing Methods

The least-squares solution to the overdetermined parameter estimation problem in (4) satisfies the *normal equations*

$$\mathbf{P}^T\mathbf{P}\theta_{\mathrm{LS}} = \mathbf{P}^T\boldsymbol{y}. \tag{5}$$

The solution θ_{LS} is the one that minimizes the error $\|\boldsymbol{y} - \mathbf{P}\theta\|$. Its determination requires the cross product matrix $\mathbf{P}^T\mathbf{P}$ to be invertible, in other words, the columns of $\mathbf{P}$ must be independent; $\mathbf{P}$ must have rank M. When $\mathbf{P}$ is *rank deficient*, i.e., $r = \mathrm{rank}(\mathbf{P}) < M$, there will be no unique solution to the parameter estimation problem. Such problems are usually solved by means of the *pseudo inverse* $\mathbf{P}^+$ of $\mathbf{P}$

$$\theta' = \mathbf{P}^+\boldsymbol{y}. \tag{6}$$

The pseudo inverse of $\mathbf{P}$ is obtained from the singular value decomposition (SVD) of $\mathbf{P}$

$$\mathbf{P} = \mathbf{U}\Sigma\mathbf{V}^T \tag{7}$$

where $\mathbf{U} \in \mathbb{R}^{N\times N}$ and $\mathbf{V} \in \mathbb{R}^{M\times M}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{N\times M}$ is a diagonal matrix with the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M \geq 0$ in decreasing order as diagonal elements. The pseudo inverse is calculated as

$$\mathbf{P}^+ = \mathbf{V}\Sigma^+\mathbf{U}^T \tag{8}$$

where $\Sigma^+ \in \mathbb{R}^{M\times N}$ is a diagonal matrix with the reciprocals $1/\sigma_1, \ldots, 1/\sigma_r$ of the r *nonzero* singular values as diagonal elements. The number of nonzero singular values σ in the SVD of $\mathbf{P}$ reveals the rank of $\mathbf{P}$. Usually, r is an estimate of the effective rank of $\mathbf{P}$, identified by a (relative) gap $\sigma_r \gg \sigma_{r+1}$ in the singular values. The gap represents a natural point to reduce the dimensionality of the problem by setting the singular values below the gap to zero. Given an estimate $r \leq \mathrm{rank}(\mathbf{P})$, the solution to (6) can be written as

$$\theta' = \sum_{i=1}^{r} \frac{\boldsymbol{u}_i^T\boldsymbol{y}}{\sigma_i}\boldsymbol{v}_i \tag{9}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the ith column of $\mathbf{U}$ and $\mathbf{V}$, respectively. This solution minimizes the error $\|\boldsymbol{y} - \mathbf{P}'\theta'\|$, where $\mathbf{P}'$ is the closest matrix to $\mathbf{P}$ that has rank r

$$\mathbf{P}' = \mathbf{U}\Sigma'\mathbf{V}^T. \tag{10}$$

Here, $\Sigma' \in \mathbb{R}^{N\times M}$ is a diagonal matrix extracted from Σ in (7) by taking the first r singular values $\sigma_1, \ldots, \sigma_r$ as diagonal elements. Indeed, if $r = M = \mathrm{rank}(\mathbf{P})$, then $\theta' = \theta_{\mathrm{LS}}$ and $\mathbf{P}' = \mathbf{P}$.

It was recognized in [11] that replacing $\mathbf{P}$ with $\mathbf{P}'$ is a way of reducing the redundancy among the underlying rules as the redundant rules are associated with near zero singular values. For the application to rule selection, a system $\mathbf{P}\theta$ should be designed, where θ has $r \leq \mathrm{rank}(P)$ nonzero components. The position of the nonzero entries in θ determines which columns in $\mathbf{P}$ are used to approximate the output $\boldsymbol{y}$ and, consequently, which of the $r \leq M$ initial rules should remain in the reduced rule base.

C. Rule Selection with SVD

The problem of picking the most influential columns of $\mathbf{P}$ is known as *subset selection* [15]. An overview of subset selection methods applied to fuzzy rule-based systems was given in [12]. The methods seek to replace $\mathbf{P}\theta$ in (5) with $\mathbf{P}_r\theta_r$, where $\mathbf{P}_r \in \mathbb{R}^{N\times r}$ consist of r columns picked from $\mathbf{P}$. The natural way to determine r is to locate a gap in the singular values of the firing matrix $\mathbf{P}$. One such approach is the *SVD-QR with pivoting* first proposed in [18] and applied to fuzzy systems in [11]. In short, the algorithm works as follows.

1) Calculate the SVD of $\mathbf{P}$ as in (7) and estimate its effective rank $r \leq \mathrm{rank}(\mathbf{P})$ from Σ.
2) Calculate a permutation matrix $\mathbf{\Pi}$ such that the columns of the matrix $\mathbf{P}_r \in \mathbb{R}^{N\times r}$ in
$$\mathbf{P}\mathbf{\Pi} = [\mathbf{P}_r, \mathbf{P}_{M-r}] \tag{11}$$
are independent $(\mathrm{rank}(\mathbf{R}_r) = r)$.
3) Approximate $\boldsymbol{y}$ with $\mathbf{P}\theta$, where
$$\theta = \mathbf{\Pi}\begin{bmatrix}\theta_r \\ 0\end{bmatrix}$$
and $\theta_r \in \mathbb{R}^r$ minimizes $\|\mathbf{P}_r\theta_r - \boldsymbol{y}\|$.

The actual rule selection is the calculation of the permutation matrix $\mathbf{Pi}$ that extracts an independent subset of columns $\mathbf{P}_r$ from among the columns of $\mathbf{P}$, assumed to correspond to the most important rules. A heuristic solution to the problem of obtaining $\mathbf{\Pi}$ is given in [15] as computing the pivoted QR decomposition of the submatrix $[\mathbf{V}_{11}^T\mathbf{V}_{21}^T]$ of singular vectors extracted from $\mathbf{V}$ in (7)

$$\mathbf{V} = \begin{bmatrix}\mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22}\end{bmatrix} \tag{12}$$

where $\mathbf{V}_{11} \in \mathbb{R}^{r\times r}$ and $\mathbf{V}_{21} \in \mathbb{R}^{(M-r)\times r}$. The pivoted QR decomposition determines a permutation matrix such that

$$\mathbf{Q}^T \left[\mathbf{V}_{11}^T \mathbf{V}_{21}^T\right] \mathbf{\Pi} = [\mathbf{R}_{11}\mathbf{R}_{12}]$$

where $\mathbf{Q} \in \mathbb{R}^{r\times r}$ is orthogonal and $\mathbf{R}_{11}$ and $\mathbf{R}_{12}$ form an upper triangular matrix.

If the number of nonsmall singular values $r \geq \text{rank}(\mathbf{P})$ can be properly determined, i.e., there is a well defined gap $\sigma_{r+1}(\mathbf{P}) \ll \sigma_r(\mathbf{P})$, then the subset selection performed by $\mathbf{P\Pi} = [\mathbf{P}_r \mathbf{P}_{M-r}]$ will tend to produce a subset $\mathbf{P}_r$ containing the most important columns (rules) of $\mathbf{P}$. However, often, the singular values tend to decrease smoothly without any clear gap. In such cases, r is determined by counting the number of (close to) zero singular values in the SVD of $\mathbf{P}$, resulting in a conservative rule reduction method that degenerates to a means for detecting equal rules and rules that do not fire. To help in such situations, it has been claimed [2], [3] that "*the smaller are the singular values, the less important are the associated rules*" and that one can use this ordering to construct a reduced fuzzy model of any size $M_s \leq M$ by picking the M_s most important rules according to $\mathbf{\Pi}$. This is different from the idea of subset selection, where a subset of r rules is extracted, and no further importance other than this classification is associated with the order in which they are picked by the permutation matrix $\mathbf{\Pi}$. The claim that this is an importance ordering cannot be proved, and it is far from apparent. First of all, the column pivoting strategy applied to obtain $\mathbf{\Pi}$ is itself a heuristic approach. It tends to work fine in applications, but little is known in theory about its behavior. Moreover, the produced permutation is strongly dependent on the estimate of the effective rank r. If a slightly different (e.g., lower) threshold is used to define small singular values (resulting in e.g., $r' = r - 1$), the permutations (rule order) in $\mathbf{\Pi}$ can change dramatically. This is illustrated in the example in Section II-E.

A method that can produce a permutation order that is independent of the estimate of r is obtained by applying the pivoted QR decomposition directly to $\mathbf{P}$.

D. Pivoted QR Decomposition of the Firing Matrix

The pivoted QR (P-QR) decomposition can be applied directly to $\mathbf{P}$ to obtain a permutation matrix and, for most practical cases, it will reveal its rank [15] at the same time. This is computationally attractive. One can skip the calculation of the SVD and it is not necessary to give an estimate of r to obtain the permutation.

The QR decomposition of $\mathbf{P}$ is given by $\mathbf{P\Pi} = \mathbf{QR}$, where $\mathbf{\Pi} \in \mathbb{R}^{M\times M}$ is a permutation matrix, $\mathbf{Q} \in \mathbb{R}^{N\times M}$ has orthonormal columns, and $\mathbf{R} \in \mathbb{R}^{M\times M}$ is upper triangular. If $\mathbf{P}$ has full rank, then $\mathbf{R}$ is nonsingular (invertible). When $\mathbf{P}$ is (near) rank deficient, it is desirable to select the permutation matrix $\mathbf{\Pi}$ such that the rank deficiency is exhibited in $\mathbf{R}$, having a small lower right block $\mathbf{R}_{kk}$ [19]

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

where $\mathbf{R}_{kk} \in \mathbb{R}^{k\times k}$. It can be shown that for the $M - k+$ first singular value of $\mathbf{P}$, we have $\sigma_{M-k+1}(\mathbf{P}) \leq ||\mathbf{R}_{kk}||_2$. Therefore, if $||\mathbf{R}_{kk}||_2$ is small, then $\mathbf{P}$ has at least k small singular values.

The QR decomposition is uniquely determined by the permutation matrix $\mathbf{\Pi}$, and many techniques have been proposed to compute it. The most well known is the column pivoting strategy [18] which in practice is very efficient in computing a triangular factor $\mathbf{R}$ with a small $||\mathbf{R}_{kk}||$. The pivoting works such that the norm of the first column of $\mathbf{R}_{kk}$ dominates the norm of the other columns. The norm of this column is $|R(kk)|$, and the pivoting strategy can be regarded as a greedy algorithm to make the leading principal submatrices of $\mathbf{R}$ as well conditioned as possible by maximizing their trailing diagonals. Thus, the values $|R(kk)|$ on the diagonal of $\mathbf{R}$, called the R values, are decreasing, and they tend to track the singular values $\sigma_k(\mathbf{P})$ well enough to expose gaps.

The pivoting algorithm favors columns of $\mathbf{R}$ with a large norm, related (through orthogonalization) to the norm of the columns of $\mathbf{P}$. In regular regression problems, this is not necessarily an interesting observation. For a fuzzy rule base, however, the norms of the columns of $\mathbf{P}$ correspond to the firing strength and firing frequency of the rules. Thus, P-QR picks first the most active and least redundant of the remaining rules. This will typically correspond to an ordering according to the generalizing capability of the rules as the most active rules can be assumed to have high generalizing capability, i.e., they describe large regions of the systems state space, or frequently occurring situations.

E. Example: Pivoted QR versus SVD-QR

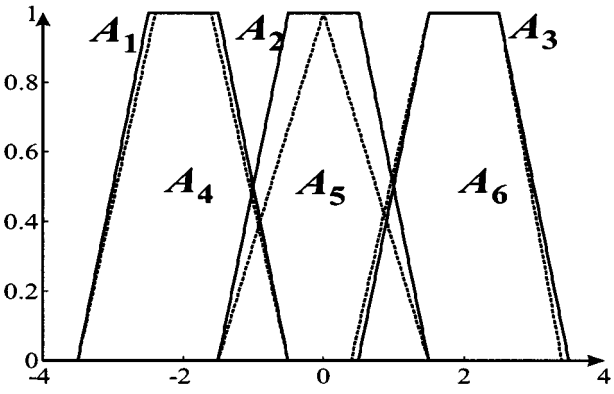
A simple example will illustrate the effect of the estimate of r on the "rule ordering" by the SVD-QR algorithm. We also show the ordering by the P-QR decomposition and how it tracks the singular values.

A simple one-dimensional (1-D) fuzzy partition of three trapezoidal fuzzy sets A_1, A_2, and A_3 is considered. To this partition we add the three more or less redundant sets A_4, A_5, and A_6 as shown in Fig. 1(a).

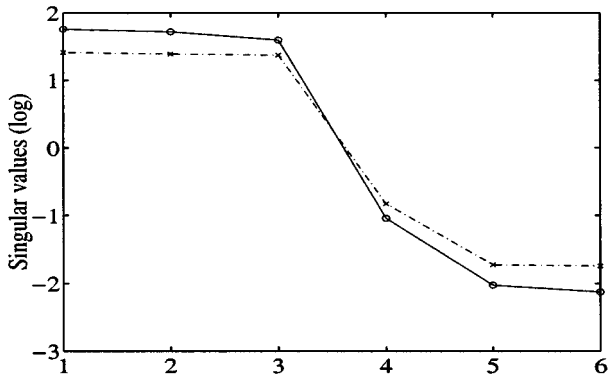
An input data vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_{200}]^T$ of 200 observations evenly spaced in $[-3, 3]$ is constructed, and the 200×6 firing matrix $\mathbf{P}$ is calculated. Both the singular values and the R values of $\mathbf{P}$ are plotted in Fig. 1(b), and it is seen that the R values track the singular values well. Further, we notice that $\mathbf{P}$ has two small (close to zero) singular values, but also a distinct gap after three values (as could be expected with our knowledge of the partition).

We now apply the SVD-QR algorithm several times with varying estimates of r. The results are reported in Table I together with those of the P-QR. The entries i in the table correspond to the order in which the rules R_i are picked from most important at the top till least important at the bottom. The antecedent of the rules R_i are formed by the fuzzy sets A_i shown in Fig. 1(a).

According to the distribution of the singular values, two good choices are $r = 4$ and $r = 3$. However, the resulting "importance ordering" of the SVD-QR differ completely for these two values. For $r = 4$, the algorithm assigns the highest importance to the two rules defined by the fuzzy sets A_2 and A_5 of which one is certainly redundant. In fact, the only reasonable ordering obtained with SVD-QR is the one obtained for $r = 3$. The P-QR



(a)



(b)

Fig. 1. (a) Redundant fuzzy partition of six fuzzy sets and the singular values ($\circ$) and (b) R values ($\times$) of the corresponding 200×6 firing matrix $\mathbf{P}$.

TABLE I
RULE ORDERING BY P-QR AND SVD-QR FOR VARYING ESTIMATES OF r (MOST IMPORTANT AT TOP)

Pivoted QR	SVD-QR					
	$r=6$	$r=5$	$r=4$	$r=3$	$r=2$	$r=1$
2	2	2	2	2	1	6
1	3	5	5	1	6	2
3	5	1	1	3	3	3
5	4	4	3	4	4	4
6	1	3	4	5	5	5
4	6	6	6	6	2	1

does not need any estimate of r, and it produces a reasonable order. As for the singular values, the R values of the P-QR decomposition [see Fig. 1(b)] can help to determine the number of rules to pick. Due to the robust ordering, this estimate is not necessary, and one can use other methods like an information criteria [3] to pick the first M_s rules according to the permutation order produced by the pivoted QR decomposition.

As discussed above, the claimed importance ordering [2], [3] produced by the rank-revealing SVD-QR and related methods can be questioned. These methods have been proposed for subset selection, and they typically require some estimate of the efficient rank to work in a reasonable way. Also, they operate on the information in the rule-firing matrix only. However, the effect of the rule consequents should not always be discarded. In systems modeling, where the measured output is often available, a more useful rule ordering is produced by methods like the OLS method [9], [14] which order the rules based on their contribution to explain the variance of the data to be approximated. This method is discussed in the following.

III. ORTHOGONAL LEAST SQUARES-BASED REDUCTION

The OLS method was first applied to fuzzy systems in [9] to select the most important fuzzy basis functions needed to approximate a data set. The OLS method transform the columns of the firing matrix $\mathbf{P}$ into a set of orthogonal basis vectors in order to inspect the individual contribution of each rule. The Gram–Schmidt orthogonalization is used to perform the orthogonal decomposition $\mathbf{P} = \mathbf{WA}$, where $\mathbf{W}$ is an orthogonal matrix such that $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, and $\mathbf{A}$ is an upper-triangular matrix with unity diagonal elements. Substituting $\mathbf{P} = \mathbf{WA}$ into (4), we have $\boldsymbol{y} = \mathbf{WA}\theta + \boldsymbol{e} = \mathbf{W}\boldsymbol{g} + \boldsymbol{e}$, where $\boldsymbol{g} = \mathbf{A}\theta$. Since the columns $\boldsymbol{w}_i$ of $\mathbf{W}$ are orthogonal, the sum of squares of $y(k)$ can be written as

$$\boldsymbol{y}^T\boldsymbol{y} = \sum_{i=1}^{M} g_i \boldsymbol{w}_i^T \boldsymbol{w}_i + \boldsymbol{e}^T\boldsymbol{e}. \tag{13}$$

The part of the output variance $\boldsymbol{y}^T\boldsymbol{y}/N$ explained by the regressors is $\sum g_i \boldsymbol{w}_i^T\boldsymbol{w}_i/N$. Thus, an error reduction ratio [14] due to an individual rule i can be defined as

$$[\text{err}]^i = \frac{g_i^2 \boldsymbol{w}_i^T \boldsymbol{w}_i}{\boldsymbol{y}^T\boldsymbol{y}}, \quad 1 \le i \le M. \tag{14}$$

This ratio offers a simple means of ordering the rules, and was used in [9] to select a subset of important rules in a forward-regression manner.

In [12], it was concluded that the OLS method may produce an inappropriate subset of fuzzy rules. An explanation for this was sought in the used error reduction ratio (14) as it tries to minimize the fitting error without paying any attention to the model structure. Thus, in the OLS algorithm applied in [9], [10], and [12], it is possible that a rather redundant rule is assigned a high importance because of its contribution to the output. This problem can easily be helped by introducing a check for $\boldsymbol{w}_k^T\boldsymbol{w}_k \le \epsilon$ when selecting the kth most important rule based on its error reduction. Here, $\epsilon > 0$ is some numerical approximation of zero, and the relation $\boldsymbol{w}_i^T\boldsymbol{w}_i = 0$ implies that the corresponding column vector $\boldsymbol{p}_k$ is a linear combination of the column vectors corresponding to the previously selected $k-1$ rules [14]. An extended OLS algorithm for ordering the rules can now be written down that creates a vector $\boldsymbol{O} = [o_1, o_2, \ldots, o_M]^T$ of rule indices where the M rules are ordered in decreasing importance.

- Step 1: *Select the first vector $\boldsymbol{w}_1$ of the orthogonal basis*
  For $1 \le i \le M$,
  set $\boldsymbol{w}_1^{(i)} = \boldsymbol{p}_i$ and calculate the corresponding element of the OLS solution vector

$$g_1^{(i)} = \frac{\left(\boldsymbol{w}_1^{(i)}\right)^T \boldsymbol{y}}{\left(\boldsymbol{w}_1^{(i)}\right)^T \boldsymbol{w}_1^{(i)}}$$

  and the error-reduction ratio

$$[\text{err}]_1^{(i)} = \frac{\left(g_1^{(i)}\right)^2 \left(\boldsymbol{w}_1^{(i)}\right)^T \boldsymbol{w}_1^{(i)}}{\boldsymbol{y}^T\boldsymbol{y}}$$

where $\boldsymbol{p}_i = [p_{ki}, \ldots, p_{Ni}]^T$ is given by the firing strength matrix $\mathbf{P}$.
Find the rule with the largest error reduction ratio

$$o_1 = \arg \max_{1 \le i \le M} \left([\text{err}]_1^{(i)} \right)$$

and select the first basis vector $\boldsymbol{w}_1$ and the first element g_1 of the OLS solution vector

$$\boldsymbol{w}_1 = \boldsymbol{w}_1^{(o_1)} = \boldsymbol{p}_{o_1}$$
$$g_1 = g_1^{(o_1)}.$$

- Step 2: *Select the next basis vectors* $\boldsymbol{w}_k$.
Repeat for $2 \le k \le M$:
For $1 \le i \le M, i \ne o_1, \ldots, i \ne o_{k-1}$, calculate

$$\alpha_{jk}^{(i)} = \frac{\boldsymbol{w}_j^T \boldsymbol{p}_i}{\boldsymbol{w}_j^T \boldsymbol{w}_j}, \quad 1 \le j < k$$
$$\boldsymbol{w}_k^{(i)} = \boldsymbol{p}_i - \sum_{j=1}^{k-1} \alpha_{jk}^{(i)} \boldsymbol{w}_j$$
$$g_k^{(i)} = \frac{\left(\boldsymbol{w}_k^{(i)}\right)^T \boldsymbol{y}}{\left(\boldsymbol{w}_k^{(i)}\right)^T \boldsymbol{w}_k^{(i)}}$$
$$[\text{err}]_k^{(i)} = \frac{\left(g_k^{(i)}\right)^2 \left(\boldsymbol{w}_k^{(i)}\right)^T \boldsymbol{w}_k^{(i)}}{\boldsymbol{y}^T \boldsymbol{y}}.$$

Find the remaining rule with the largest error reduction ratio that is not redundant (see the equation shown at the bottom of the page) and select the kth basis vector $\boldsymbol{w}_k$ and the kth element g_k of the OLS solution vector.

$$\boldsymbol{w}_k = \boldsymbol{w}_k^{(o_k)}$$
$$g_k = g_k^{(o_k)}.$$

For rule selection, if a predetermined number of rules $M_S < M$ is to be selected, step two of the algorithm is ended at $k = M_S$. It is also possible to determine a stopping criterion based on, e.g., the approximation accuracy [10] or the relative contribution of the selected rules [7].

IV. Example

We consider the same example as in [12] to illustrate the working of the pivoted QR decomposition and the extended OLS algorithm.[1] The system under study is a second-order nonlinear plant

$$y(k) = f(y(k-1), y(k-2)) + u(k) \tag{15}$$

[1]See [12] for a discussion and comparison of other methods.

TABLE II
Parameters of the Gaussian Membership Functions

i	center c_{i1}	center c_{i2}	width σ_{i1}	width σ_{i2}
1	**0.0930**	**-0.3630**	**0.7095**	**0.7095**
2	**0.0933**	**-0.3632**	**0.7095**	**0.7095**
3	1.3828	-0.6617	0.6271	0.6271
4	-1.0414	1.5397	0.7969	0.7969
5	**-1.8130**	**-1.6470**	**1.3205**	**1.3205**
6	**-1.8125**	**-1.6469**	**1.3205**	**1.3205**
7	0.7776	-1.1555	0.7800	0.7800
8	0.1898	1.0142	0.6141	0.6141
9	-0.4052	0.2798	0.8099	0.8099
10	**-0.6613**	**-0.4846**	**0.0100**	**0.0100**
11	-0.6613	-0.4846	0.7051	0.7051
12	0.9529	-0.3965	0.6313	0.6313
13	0.7860	0.7723	0.6177	0.6177
14	0.4329	0.1910	0.6652	0.6652
15	**1.2940**	**1.0740**	**0.6474**	**0.6474**
16	**1.2942**	**1.0738**	**0.6474**	**0.6474**
17	0.6801	1.4083	0.6370	0.6370
18	1.2656	0.2698	0.7156	0.7156
19	-0.3846	1.1827	0.6772	0.6772
20	**-1.2642**	**-0.1808**	**0.0100**	**0.0100**
21	-1.2642	-0.1808	0.7907	0.7907
22	-0.9099	-1.1750	0.7728	0.7728
23	-0.1008	-1.1384	0.8046	0.8046
24	-1.1533	0.7037	0.8517	0.8517
25	1.7691	-1.2798	0.8746	0.8746

where

$$f(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)[y(k-1) - 0.5]}{1 + y^2(k-1) + y^2(k-2)}. \tag{16}$$

We want to approximate the nonlinear component f of the plant (the unforced system) with a fuzzy model (1). Twelve hundred simulated data points were generated from the plant model (15). Starting from equilibrium state (0, 0), 1000 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 $u(k) = \sin(2\pi k/25), k = 1001, \ldots, 1200$. The simulated data is shown in Fig. 2(a). The input to the fuzzy model is $\boldsymbol{x}_k = [y(k-1), y(k-2)]$, and the 25 Gaussian membership functions taken from [12] with the parameters shown in Table II are used to partition the input space as illustrated in Fig. 2(b).

$$A_{ij}(x_j(k)) = \exp\left(-\frac{(x_j(k) - c_{ij})^2}{2\sigma_{ij}^2} \right) \quad j = 1, 2, \quad i = 1, \ldots, 25. \tag{17}$$

Each row in Table II is associated with one of the fuzzy rules in the rule base. The first two rows have equivalent membership function parameters. Thus, the first two rules in the rule base will always have nearly the same firing strengths. This implies that there is redundancy between the two rules and removing one of them will not significantly affect the performance of the model.

$$o_k = \begin{cases} \arg\max_{1 \le i \le M, i \ne o_1, \ldots, i \ne o_{k-1}} \left\{ [\text{err}]_1^{(i)} \;\middle|\; \left(\boldsymbol{w}_k^{(i)}\right)^T \boldsymbol{w}_k^{(i)} > \epsilon \right\}, & \text{if } \exists \boldsymbol{w}_k^{(i)}, \left(\boldsymbol{w}_k^{(i)}\right)^T \boldsymbol{w}_k^{(i)} > \epsilon \\ \arg\max_{1 \le i \le M, i \ne o_1, \ldots, i \ne o_{k-1}} \left([\text{err}]_1^{(i)} \right), & \text{otherwise.} \end{cases}$$

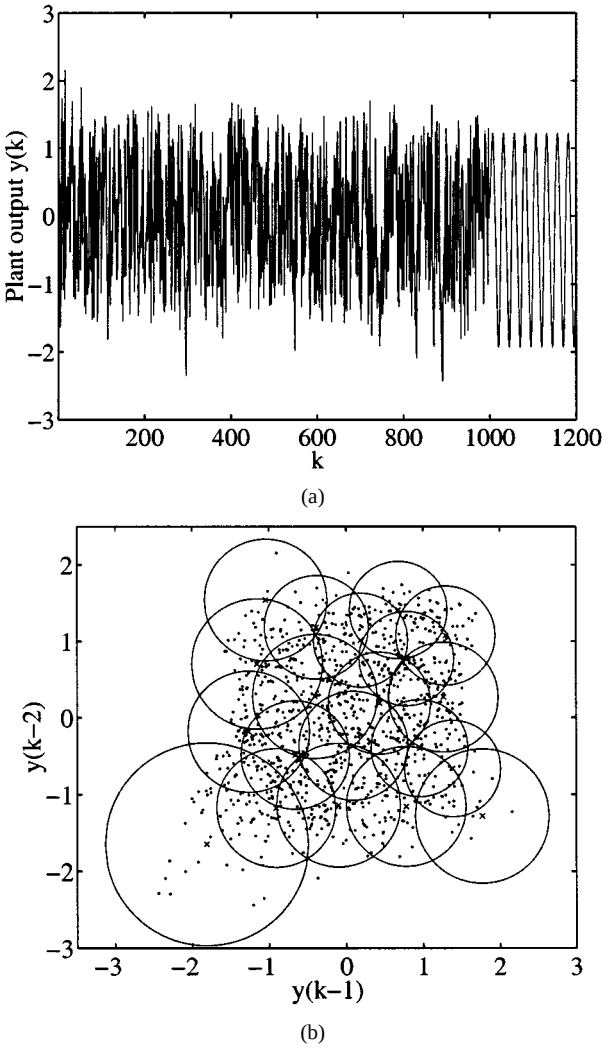


Fig. 2. (a) Simulated output of the plant. (b) Positions and widths of the Gaussian membership functions in the models input space along with the input data.

The same holds for rules 5 and 6 as well as rules 15 and 16. The Gaussian membership functions associated with the rules 10 and 20 are very narrow. This implies that the rules will virtually never fire, and they can thus be removed from the rule base.

From the first 1000 input data points $\boldsymbol{x}_k = [y(k-1), y(k-2)], k = 1, \ldots, 1000$, the 1000×25 firing strength matrix $\mathbf{P}$ is calculated using (3). The singular values of the matrix are shown in Fig. 3 together with the R values of the pivoted QR decomposition. It is seen that the R vlaues track the singular values with considerable fidelity.

A. Rule Subset Selection

From inspecting the log scale plot in Fig. 3, one could conclude that there are anywhere from one till five relatively small singular values. The real scale plot indicates the presence of five near zero singular values, corresponding with our knowledge about the rule base. We now apply the orthogonal transformation-based methods studied above to the problem. The results are reported in Table III, which shows the order in which the rules are picked from the rule base. The SVD-QR and the OLS method are the same as those studied for this problem in [12][2] while the P-QR decomposition and the extended OLS method (E-OLS) are those proposed in this paper.

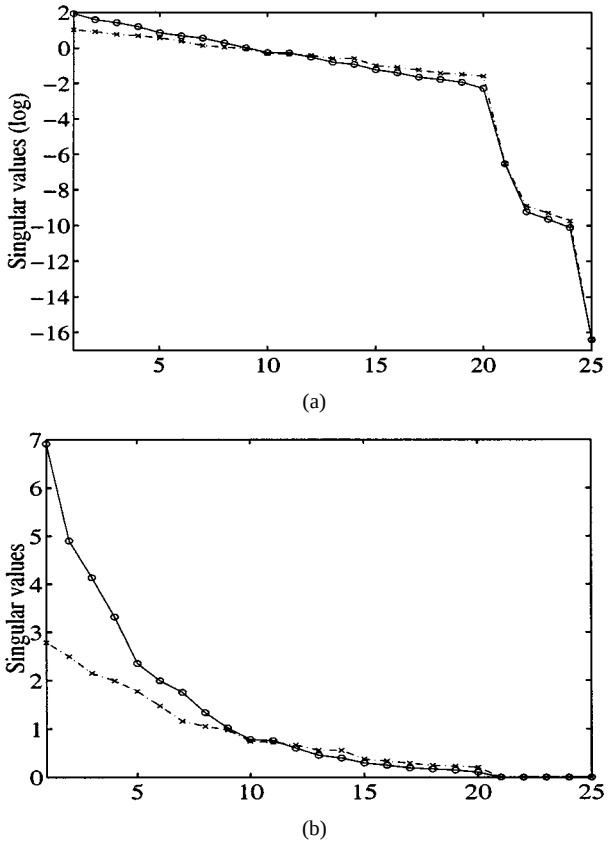


Fig. 3. Singular values ($\circ$) and R values ($\times$) of the corresponding 1000×25 firing matrix $\mathbf{P}$. (a) Log scale. (b) Real scale.

TABLE III
ORDER IN WHICH THE RULES ARE PICKED FROM THE RULE BASE (MOST IMPORTANT AT TOP)

P-QR	SVD-QR	E-OLS	OLS
24	25	5	5
25	4	24	24
6	19	25	25
15	7	16	16
23	3	8	8
8	24	21	21
4	8	23	23
12	13	11	11
11	23	3	3
14	14	22	22
18	21	7	6
19	17	9	7
22	22	19	15
17	18	4	19
7	12	14	4
21	9	18	9
3	11	1	17
13	2	17	13
9	5	13	18
2	16	12	12
20	15	2	1
5	10	6	2
16	20	10	14
1	6	15	10
10	1	20	20

[2]Due to the random nature of the identification data, the rule order by SVD-QR and OLS differs slightly from that in [12].

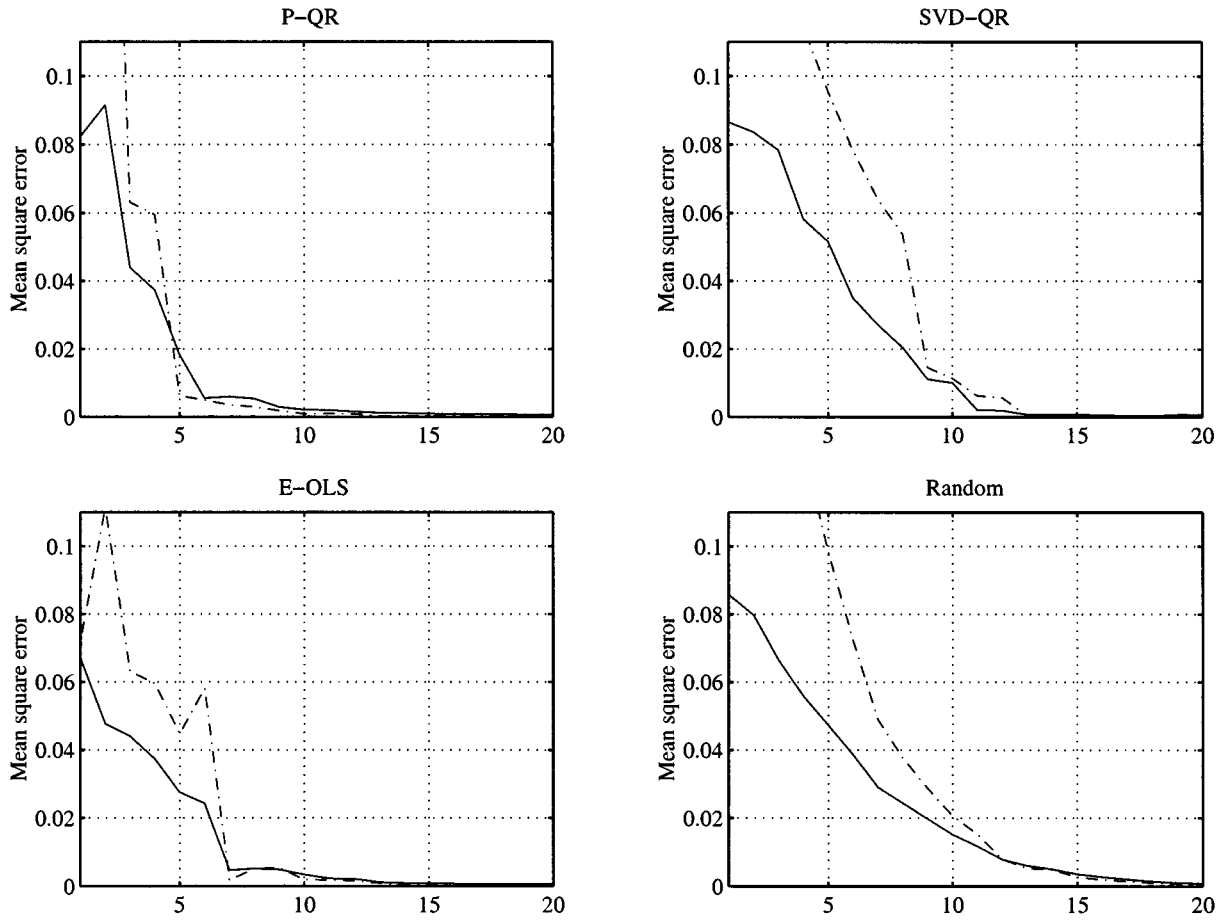


Fig. 4. Performance of models with an increasing number of rules picked according to the different methods. Solid line indicates mean square error (MSE) on training data. Dash-dotted line indicates MSE on evaluation data.

The results show that both rank-revealing methods, the P-QR and the SVD-QR, pick as the least important rules three redundant ones and the two nonfiring ones. In this case, the SVD-QR algorithm was executed with $r = 20$, i.e., it was told that there where $25-20 = 5$ noninfluential rules, according to the inspection of the singular values in Fig. 3. The P-QR method does not need this information, and still produces the correct subset of least important rules for this problem. Further, as concluded in [12], the OLS method correctly sorts out the nonfiring rules (10 and 20) but fails to assign a low importance to one rule from each of the three pairs of redundant rules. This deficiency is not encountered with the E-OLS method proposed in Section III. It produces an importance ordering that successfully detects both the redundant as well as the nonfiring rules.

B. Rule Ordering

We now repeat the modeling exercise several times. According to each of the three methods (P-QR, SVD-QR, and E-OLS)[3], we make 20 models of increasing complexity with the rules picked in the order reported in Table III and evaluate their performances on fitting the training (1000 samples) and evaluation (200 samples) data. For instance, according to the P-QR method, we first make a one-rule model consisting only of rule R_6, then a two-rule model with the rules R_6 and R_{24}, etc., until we have a model of 20 rules. In this exercise, when $r \in [1, 20]$ rules are picked, the corresponding $1000 \times r$ firing matrix $\mathbf{P}_r$ is formed using the training data, and the rule consequents θ are determined by solving the resulting least-squares problem. To verify the methods, for each model complexity, 50 models are made with $r = 1, 2, \ldots, 20$ different rules drawn at random from the total set of 25 rules. The average performance of the random models are recorded. The results are presented in Fig. 4.

In this experiment, the P-QR picks the rules such that they have good generalizing capabilities. It obtains an error on the evaluation data that is below that of the training data for a low number of rules $(r \geq 5)$. This result supports the observation made in Section II-D concerning the pivoting algorithm.

As expected, the E-OLS method has a good performance in fiting the training data with a low number of rules. Unlike the other methods, it uses information about the systems output and tries to fit the training data as well as possible. For a low number of rules, the E-OLS constructs models that are fitting only the training data. However, already from seven rules on, the models show good performance both on the training and the evaluation data.

The worst performing of the studied methods is the SVD-QR. As could be expected from the discussion in Section II-E, the order in which the rules are picked bears no proof of representing any importance order; neither with respect to fitting the training

[3]Since the ordering produced by the E-OLS method is quite similar to that of the OLS, only the models obtained with E-OLS are inspected in this exercise.

data, nor with respect to generalization capabilities. In fact, Fig. 4 shows that its performance is qualitatively close to the average random approach in which the redundant and nonfiring rules were picked with the same probability as all the other rules.

V. Conclusion

We have discussed the principles and the performance of some orthogonal transform-based rule reduction methods proposed in the literature. These methods can be divided into two groups: the rank-revealing ones and the ones that evaluate the contribution of each rule to the output. A representative of the former is the SVD-QR method which was originally developed for subset selection [18]. It has been claimed by researchers that the SVD-QR method can be used to order the rules according to their importance [2], [3]. We have shown in this paper that this is not the case. The SVD-QR behaves strictly like a subset selection method. It only classifies the rules into two sets of influential and noninfluential rules, respectively. Moreover, the success of this classification strongly depends on a correct estimate of the effective rank of the firing matrix.

We propose to use the pivoted QR decomposition for rule selection. Similarly to the SVD-QR, the P-QR method can be used as a subset selection method. We have shown that it can track the singular values in the firing matrix $\mathbf{P}$ well enough to make an estimate of the effective rank without calculating the SVD of $\mathbf{P}$. For rule ordering, the method produces a permutation matrix directly from the firing matrix, and the order in which the rules are picked is not dependent on any estimate of the rank. Moreover, in the studied experiments, this order proved to pick the rules according to their generalizing capabilities while filtering out redundant ones. Finally, it is also computationally less expensive. Thus, when applying rank-revealing methods to select rules according to their ordering as proposed in [2] and [3], the P-QR method is preferable to the SVD-QR and related methods.

The rank-revealing methods consider the partitioning of the input space by the rule antecedents and the influence of the rule consequents is discarded. In systems modeling, when measured output data are available, a more effective and transparent rule ordering is produced by the OLS method [9], [14]. It was concluded in [12] that the OLS does not make any considerations about the structure of the rule base, and sometimes it assigns a high importance to redundant rules. In Section III, we have presented a simple extension to the algorithm that detects rule redundancy. When repeating the experiments from [12], it was shown that the E-OLS method effectively filtered out the redundant rules, making it more applicable to rule selection than the standard OLS method.

References

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

[2] J. Yen, L. Wang, and C. W. Gillespie, "Improving the interpretability of TSK fuzzy models by combining global learning and local learning," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 530–537, Aug. 1998.

[3] J. Yen and L. Wang, "Application of statistical information criteria for optimal fuzzy model construction," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 362–372, June 1998.

[4] C. T. Chao, Y. J. Chen, and T. T. Teng, "Simplification of fuzzy-neural systems using similarity analysis," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, pp. 344–354, June 1996.

[5] M. Setnes *et al.*, "Similarity measures in fuzzy rule base simplification," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 376–386, June 1998.

[6] W. Pedrycz, "Conditional fuzzy clustering in the design of radial basis function neural networks," *IEEE Trans. Neural Netw.*, vol. 9, pp. 601–612, Aug. 1998.

[7] M. Setnes, "Supervised fuzzy clustering for rule extraction," in *Proc. FUZZ-IEEE*, Seoul, Korea, Aug. 1999, pp. 1270–1274.

[8] J. V. de Oliveira, "Semantic constraints for membership function optimization," *IEEE Trans. Syst., Man, Cybern. A*, vol. 29, pp. 128–138, Jan. 1999.

[9] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Trans. Neural Netw.*, vol. 3, pp. 807–813, Oct. 1992.

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

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

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

[13] Y. Yam, P. Baranyi, and C.-T. Yang, "Reduction of fuzzy rule base via singular value decomposition," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 120–132, Apr. 1999.

[14] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 2, pp. 302–309, Apr. 1991.

[15] G. H. Golub and C. F. van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1989.

[16] 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, Jan./Feb. 1985.

[17] R. Babuška, *Fuzzy Modeling for Control*. Boston, MA: Kluwer, 1998.

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

[19] G. W. Stewart, "Rank degeneracy," *SIAM J. Sci. Stat. Comput.*, vol. 5, no. 2, pp. 403–413, 1984.

Magne Setnes (S'96–M'00) was born in 1970 in Bergen, Norway. He received the B.Sc. degree in robotics from the Kongsberg College of Engineering, Kongsberg, Norway, in 1992, the M.Sc. degree in electrical engineering from Delft University of Technology (DUT), Delft, The Netherlands, in 1995, where he also obtained the Degree of Chartered Designer in Information Technology in 1997, and the Ph.D. degree from the Control Laboratory, DUT, in 2001.

He is currently with Research and Development, Heineken Technical Services, Zoeterwoude, The Netherlands. His research interests include fuzzy systems and computational intelligence techniques for modeling, control, and decision making.

Robert Babuška 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 (DUT), Delft, The Netherlands, in 1997.

Currently, he is an Associate Professor with the Systems and Control Engineering Group of the Electrical Engineering Department, DUT. He is an editor of *Engineering Applications of Artificial Intelligence*, and an Area Editor of *Fuzzy Sets and Systems*. He has co-authored more than 35 journal papers and chapters in books, over 100 conference papers, a research monograph *Fuzzy Modeling for Control* (Boston, MA: Kluwer, 1998) and co-edited two books. His main research interests are fuzzy set techniques and neural networks for nonlinear systems identification and control.

Dr. Babuška is serving as an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS.