

Note on Orthogonal Transformation Methods for Simplifying Fuzzy Rule-Based Models

Özer Ciftcioglu

Delft University of Technology, Faculty of Architecture
 Building Technology, 2628 CR Delft, The Netherlands
 o.ciftcioglu@bk.tudelft.nl; Tel:31-15 2784485; Fax:31-15 2784127

Abstract - Simplifying fuzzy rule based models is considered. For the rule selection process both input space and the output space are considered, where both spaces play important role. The method used is principal component analysis complemented with the OLS method for the selection of the rules. In this selection process, first the regression matrix of a common radial basis function (RBF) network is considered. The magnitude of the eigenvalues of the RBF matrix of radial basis function network is not central to the selection. The selection is made according to the energy contribution from the graded principal components, where eigenvectors with low eigenvalues may have relatively more energy contribution to the output depending on the model outputs. In parallel to the above gradation process, the influential basis functions are identified, as they are associated with the graded principal components of higher ranks in this very gradation. This approach is extended to normalized RBF matrix for fuzzy systems with singular value decomposition. The comparative results are presented and the implication and importance of the approach is pointed out.

I. INTRODUCTION

The selection of a set of important fuzzy rules from a given rule base is an important decision-making process for effective fuzzy-rule-based modeling. In a fuzzy model a balance between reducing the fitting error and increasing the model complexity is essential for a satisfactory model, that is both accurate and transparent. Several researchers have applied orthogonal transformation methods for selecting important fuzzy rules from a given rule base [1-5]. A comparative study of these methods is presented by Yen and Wang [6]. This paper aims at to point out how one can still improve the rule selection process by using orthogonal methods jointly. The orthogonal methods involved are *orthogonal least squares* (OLS), *singular value decomposition* (SVD), and *principal component analysis* (PCA). The organisation of the paper is as follows. Section two gives a brief description of a fuzzy modeling, which is a special case of Takagi-Sugeno fuzzy modeling dealt with in this research. Section two gives a brief description of orthogonal least squares method. This method is central to the present work due to its relevance to radial basis functions (RBF) network, which is closely related to fuzzy modeling being considered. Section three describes the orthogonal methods applied in this work jointly to enhance the efficacy of these methods for fuzzy rule selection. Section four presents the comparative results obtained. This is followed by the conclusions in section five.

II. FUZZY MODELING

A fuzzy model is a set of If-Then rules that associates inputs to outputs. In fuzzy logic terms the inputs and outputs are rule antecedents, and rule consequents, respectively. There are two major type of fuzzy modeling. These are Mamdani type model and Takagi-Sugeno type model. The latter differs from the preceding model especially from the rule-consequents viewpoint. Namely, in Takagi-Sugeno type modeling rule consequents are linear functions of inputs rather than fuzzy sets.

A. Takagi-Sugeno Model with Constant Consequent

Takagi-Sugeno [7] type fuzzy modeling consists of a set of fuzzy rules of n input variables with a local input-output relation in a linear form.

$$R_i : \text{If } x_1 \text{ is } A_{i1} \text{ and } \dots x_n \text{ is } A_{in} \quad (1)$$

$$\text{Then } y_i = a_i x + b_i, \quad i = 1, 2, \dots, m$$

where R_i is the i th rule, $x = [x_1, x_2, \dots, x_n]^T \in \Xi$ is the vector of input variables; $A_{i1}, A_{i2}, \dots, A_{in}$ are fuzzy sets and y_i is the rule output; m is the number of rules. The estimated output y of the model is calculated through the weighted average of the rule consequents of the form

$$y = \frac{\sum_{i=1}^m \beta_i(x) y_i}{\sum_{i=1}^m \beta_i(x)} \quad (2)$$

In (2), $\beta_i(x)$ is the degree of activation of the i th rule

$$\beta_i(x) = \prod_{j=1}^n \mu_{A_{ij}}(x_j), \quad i = 1, 2, \dots, m \quad (3)$$

where $\mu_{A_{ij}}(x_j) \rightarrow [0, 1]$ is the membership function of the fuzzy set A_{ij} at the input (antecedent) of R_i . A special case of the Takagi-Sugeno model is formed if only the constant term in the linear consequent is retained, namely b_i . In this case, the fuzzy model becomes equivalent to radial basis function (RBF) network in a neural network (NN) context under lenient conditions [8,9]. As to fuzzy systems, the Gaussian radial basis functions play the role of membership functions:

$$\mu_{ij} = \exp\left(-\frac{(x_j - c_{ij})^2}{2\sigma_{ij}^2}\right) \quad (4)$$

where c_{ij} and σ_{ij} are the *center* and *width* of Gaussian function, respectively. In this fuzzy model the output is computed by (2) which takes the form

$$y = \sum_{i=1}^M p_i b_i \quad (5)$$

where p_i is the normalized firing strength of the i th rule which is defined by

$$p_i = \frac{\prod_{j=1}^n \exp\left(-\frac{(x_j - c_{ij})^2}{2\sigma_{ij}^2}\right)}{\sum_{i=1}^M \prod_{j=1}^n \exp\left(-\frac{(x_j - c_{ij})^2}{2\sigma_{ij}^2}\right)} \quad (6)$$

To form the fuzzy system model from the data set with N data samples, given by

$$X = [x_1, x_2, \dots, x_N]^T, \quad y = [y_1, y_2, \dots, y_N]^T \quad (7)$$

where each data sample has a dimension of n ($N \gg n$). First the structure is determined, and afterwards the parameters of the structure are identified. The number of rules characterizes the structure of a fuzzy system.

From (6), the fuzzy model as a matrix equation is given by

$$y = P \vartheta + e \quad (8)$$

where P is $m \times m$ firing strength matrix; ϑ is the $m \times 1$ parameter vector. It is interesting to note that, the formulation given by (8), is the general regression formalism in a linear system modeling where ϑ is $m \times 1$ model parameter vector; P is $m \times m$ regressor matrix; y is regressant or system output vector in a general multivariable model. The solution of this linear model is well established. Referring to (5), the RBF network structure is central to the fuzzy model subject to application of orthogonal decomposition methods. Therefore, the RBF network is briefly described below for describing the principal component analysis (PCA) for the determination of the basis function centers and thereby corresponding basis functions. Based on this analysis, the application of singular value decomposition (SVD) will be described for fuzzy rule selection.

B. Radial Basis Function Network

Considering the function approximation by the radial basis functions $\phi(x, c)$, an approximation to a function $f(x)$ by radial basis function network is carried out by

$$y = f(x) = \sum_{i=1}^M w_i \phi(\|x - c_i\|) + e \quad (9)$$

where w_i are weights; M is the number of basis functions; x is the sample vector; c_j is the RBF center vector; e is the model error; $\phi(x, c)$ is the basis function generally based on the Euclidean distance metric defined by

$$\|x - c_j\|^2 = (x - c_j)^T (x - c_j) \quad (10)$$

so that

$$\phi = \exp\{-\|x - c_j\|^2 / \sigma_j^2\} \quad (11)$$

where σ_j is the j -th width parameter that determines the effective support of the j -th basis function. The symmetric matrix Φ formed by the elements

$$\phi_{ij} = \exp\{-\|x_i - c_j\|^2 / \sigma_j^2\} \quad (12)$$

is referred to as RBF matrix which is used to train the network. For a set of input and output pairs, the approximation model in matrix form can be expressed as

$$y = \Phi w + e \quad (13)$$

where y , is the desired output vector which is $[N \times q]$ matrix in the multi-output case where q is the number of outputs; Φ is the regression matrix that consists of regressor vectors and plays the role of P in (8), after normalisation similar to that given in (6); w is the parameter vector or output matrix which is $M \times q$ matrix in the multi-output case; e is the error vector.

C. The Orthogonal Least Squares Method

The orthogonal least squares (OLS) method [10-11] makes orthogonal decomposition of Φ . This is generally accomplished by Gram-Schmidt [12] orthogonalisation procedure:

$$\Phi = RA \quad (14)$$

where A is an upper-diagonal matrix and R is an $N \times M$ matrix with orthogonal columns r_i ($i=1, \dots, M$) such that

$$R^T R = H, \quad H = \begin{bmatrix} h_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & h_M \end{bmatrix} \quad (15)$$

H is $M \times M$ diagonal matrix. With the definitions

$$R = \Phi A^{-1} \quad (16)$$

and,

$$w_R = Aw \quad (17)$$

The matrix equation (13) in this case takes the form

$$y = R w_R + e \quad (18)$$

which has the solution for the estimation of the transformed parameter vector w_R as

$$\hat{w}_R = H^{-1} R^T y \quad \text{or,} \quad \hat{w}_R = \frac{r_i^T y}{r_i^T r_i}, \quad i = 1, \dots, M \quad (19)$$

By means of orthogonalisation process the traditional Gram-Schmidt computes one column of A at a time and orthogonalizes R at the same time. In terms of energy transmitted from input to the output, we can write the following algebraic balance equation for zero-mean output vector:

$$y^T y = \sum_{i=1}^M w_R^2 r_i^T r_i + e^T e \quad (20)$$

Normalizing by $D^T D$ we obtain the relative energy contribution from each basis function, i.e. from each regressor to the output as

$$z_i = \frac{w_R^2 r_i^T r_i}{y^T y} \quad (i = 1, \dots, M) \quad (21)$$

which is defined as error reduction ratio.

III. ORTHOGONAL TRANSFORMATION METHODS

A. RBF network training

Considering (13), in place of Gram-Schmidt orthogonalisation, the orthogonal decomposition can be

accomplished by principal component analysis (PCA) where $M=N$ and Φ is $[N \times N]$ a symmetric positive definite matrix; N is the number of data samples. By PCA decomposition, we write

$$\Phi = Q \Lambda Q^T \quad (22)$$

In (22) Q is a matrix ($Q^T=Q^{-1}$) whose columns are the eigenvectors of RBF matrix Φ ; Λ is the eigenvalue matrix. For the number of hidden layer nodes equal to the data samples the error term in (13) vanishes. In this case, from (13), we obtain

$$y = Q \Lambda Q^T w = Q(\Lambda Q^T w) = Q w_R \quad (23)$$

where w_R is the weight matrix corresponding to Q . Its relation to the weight matrix w which corresponds Φ is given by

$$w = (\Lambda Q^T)^{-1} w_R = (Q \Lambda^{-1}) w_R \quad (24)$$

Note that in the Gram-Schmidt orthogonalisation procedure, the orthogonalisation and the ensuing selection of the RBF center is accomplished at the same time. However in the PCA case orthogonalisation and the selection of the orthogonal vectors is performed in two consecutive steps. Therefore in the case the columns of the orthogonal matrix in (22) is obtained after shuffling according to the descending order of magnitude of the corresponding eigenvalues, this information is given by the permutation matrix and this should be taken into account for tracing the graded sequence of the basis function centers. This is executed in the algorithm as follows. Let i represents the indices of the sequence obtained from the OLS algorithm and the array $ipvt$ contains the indices of the sequence obtained from the permutation matrix. Then, the indices for the sequence of the basis functions is computed from $ipvt(i)$, in the algorithm. For a reduced number of basis function centers selected from the graded sequence the OLS algorithm can be used second time for the weight determination of the RBF network with these centers.

The PCA based orthogonalisation in RBF network training described above has major advantages. Especially in cases with a higher number of data samples, the Gram-Schmidt based orthogonalisation provides numerical instability next to exponentially increase of execution time. This situation occurs especially in complex information modeling applications for pattern recognition purposes, for instance in data mining research. Secondly, from the graded sequence of the eigenvectors, the most important ones can sequentially be selected. The number of the selected eigenvectors m forms the number of hidden layer nodes of the RBF network. This number is optimal in the representation of the RBF regression matrix. The use of principal components of Φ is much more convenient for two major reasons. First, it is faster for modeling the data because the classical Gram-Schmidt orthogonalisation process in the OLS algorithm is replaced by the principal components. Second, it is most effective in influential center gradation and hence effective in data modeling due to optimality of this particular orthogonalisation process. Namely, by selection of m principal components, the matrix norm

$\|_m \Phi - \Phi\|$ is minimized over all matrices $_m \Phi$ with rank m [13]. In other words, the regression matrix Φ is best represented with m selected principal components. This favourable representation is reflected in the favourable multivariable function approximation by the RBF network.

B. Simplifying Fuzzy Rule-Based Model

In the preceding subsection the selection of the RBF centers are described as an effective training method of a RBF network. From the fuzzy modeling viewpoint the regression matrix Φ in (13) is normalized to give a firing matrix form as shown in (6). In this case the $[N \times m]$ dimensional firing matrix P is subjected to singular value decomposition (SVD) where N is the number of samples; m is the number of rules:

Theorem: Let $A \in \mathfrak{R}^{m \times n}$ be a matrix of rank $r \leq \min(m, n)$. Then, there are two orthogonal matrices $U \in \mathfrak{R}^{m \times m}$ and $V \in \mathfrak{R}^{n \times n}$ and a diagonal matrix $S \in \mathfrak{R}^{m \times n}$, such that $U^T A V = S$ and $A = U S V^T$. The diagonal elements of S may be ordered with decreasing values, r of them are positive, the others are zero:

$$S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r, 0, \dots, 0) = \begin{bmatrix} S^{r \times r} & 0 \\ 0 & 0 \end{bmatrix}. \quad \text{The diagonal}$$

elements of S are called singular values of A . The singular values of A are uniquely defined, but not the orthogonal matrices U and V .

Let the SVD of $P \in \mathfrak{R}^{N \times m}$ be given by

$$P = U S V^T \quad (25)$$

Then, the fuzzy model is given by

$$y = P w_F + e \quad (26)$$

yields

$$y = U S V^T w + e = U S (V^T w) + e = U S w_F + e \quad (27)$$

where $w_F = V^T w$, and therefore

$$w = (V^T)^{-1} w_F \quad (28)$$

which corresponds to (8) for fuzzy model. Based on this result, similar to the processing presented in the preceding subsection the $N \times m$ matrix US is subjected to OLS algorithm in place of Q matrix in the RBF network case. In this process the permutation matrix is obtained by SVD-QR with column pivoting algorithm as suggested by Mouzouris and Mendel in [1]. This algorithm is summarized below as follows.

- Compute the SVD of P as $P=USV^T$ and check the singular values in $S=\text{diag}(\sigma_1, \sigma_2, \dots, \sigma_L)$ for the rank of P
- Determine the number of fuzzy partitions r where $r \leq \text{rank}(P)$

- Partition V as $V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}$, where $V_{11} \in \mathfrak{R}^{r \times r}$, and $V_{21} \in \mathfrak{R}^{r \times (L-r)}$ and form $V^T = [V_{11}^T \ V_{21}^T]$

- Apply QR with column pivoting algorithm to V and obtain the permutation matrix $P \in \mathfrak{R}^{L \times L}$

- e) Form $Q^T [V_{11}^T \ V_{21}^T] P = [R_{11} \ R_{12}]$ where $Q \in \mathcal{R}^{r \times r}$ orthogonal and $QQ^T = I$ and $R_{11} \in \mathcal{R}^{r \times r}$ is upper triangular
- f) Define $[A_1 \ A_2] \equiv AP$ where $L \times r$ matrix A_i is the desired subset of $L \times L$ matrix A

The permutation matrix P having been determined, this is combined with the graded sequence obtained from OLS using $N \times m$ matrix US to trace the graded sequence of the fuzzy rules, with a construction $ipvt(i)$ in the algorithm as described in the preceding subsection.

IV. COMPARATIVE STUDY

The comparative study is performed by means of the data obtained from a second order non-linear plant model [6]. 1000 data samples are used. $y(k-1)$ and $y(k-2)$ are used as the input variables and the number of fuzzy rules are arbitrarily set to 25. The Gaussian functions defined in (4) are used to express the membership functions of $y(k-1)$ and $y(k-2)$. The centers c_{ij} and the widths σ_{ij} of the Gaussian membership functions are predetermined using a k-means clustering algorithm and a nearest-neighbor heuristic, respectively [14]. A comparative study with the orthogonal transformation methods, e.g., the OLS algorithm, eigenvalue decomposition (ED), singular value decomposition with column pivoting (SVD-QR), total least squares (TLS), direct singular value decomposition (D-SVD) for clustering is reported earlier [6]. These results are repeated in Table I where 20 rules are retained and five rules are eliminated in the rule base which have altogether 25 rules.

TABLE 1
COMPARATIVE RESULTS [6]

Rules	i	OLS	ED	SVD-QR	TLS	D-SVD
Retained rules	1	5	25	25	25	2
	2	24	4	4	17	5
	3	25	7	7	8	4
	4	16	19	19	12	8
	5	8	3	3	3	3
	6	21	24	24	24	11
	7	23	8	8	18	13
	8	11	23	23	7	9
	9	3	14	14	22	7
	10	22	13	13	23	25
	11	6	21	21	4	19
	12	7	18	18	21	15
	13	19	17	17	19	24
	14	4	22	22	13	12
	15	15	12	12	9	22
	16	14	9	9	14	23
	17	9	11	11	16	21
	18	12	2	2	6	18
	19	2	16	16	2	17
	20	18	5	5	11	14

Eliminated rules	21	1	6	10	20	20
	22	13	15	15	15	6
	23	17	1	20	10	16
	24	10	20	6	5	1
	25	20	10	1	1	10

The OLS algorithm and singular value decomposition with column pivoting (SVD-QR) are briefly given in the preceding section. The methods of eigenvalue decomposition (ED), total least squares (TLS), and direct singular value decomposition (D-SVD) are given below for the sake of completeness of the description of the case study.

A. Eigenvalue Decomposition Method

This method is used in [15] to construct reduced RBF networks and Volterra series polynomials. Namely, instead of considering the original linear equation in (8), the following equation is considered:

$$\Phi_{pp} \theta = \Phi_{py}$$

where $\Phi_{pp} = P^T P \in \mathcal{R}^{m \times m}$ and $\Phi_{py} = P^T y \in \mathcal{R}^m$

are called the correlation matrix and cross-correlation vector, respectively. The eigenvalue decomposition of

$$\Phi_{pp} = V S V^T$$

is performed and from the descending order of the eigenvalues the entries of θ in (8) is determined. Selecting the entries of θ corresponding to r number of larger eigenvalues of V relative to $M-r$ smaller ones. The resulting r -dimensional parameter vector $\theta^{(r)} = [\theta_1, \theta_2, \dots, \theta_r]$, was solved from the reduced normal equation

$$\Phi_{pp}^{(r)} \theta^{(r)} = \Phi_{py}^{(r)}$$

B. Total Least Squares

In this method the linear equation of the form $Ax = b$ is considered. It is straightforward to derive the algebraic condition that guarantees the existence of at least one solution: $rank[Ab] = rank[A]$

However, often this condition is not satisfied due to measurement inaccuracies, limited machine precision, simplification or approximation of the original problem, nonlinearities etc. The solution is to extract a least squares solution x' from the following minimization procedure:

$$x': \|Ax' - b\| = \min, \|At - b\|$$

Then the total linear least squares problem formulation is to find x' such that $A'x' = b'$ [16-18]:

$$\|[A' \ b'] - [A \ b]\|_F = \min_{B, z; rank\{Bz\} = rank\{B\}} \|[Bz] - [Ab]\|_F$$

where F indicates the Frobenius norm. Stated in this way, the total least squares solution is obtained from the SVD of the extended matrix $[A \ b]$ [19]. In the present case the extended matrix becomes $[P \ y]$.

C. Direct SVD Method

The direct SVD method is the name of the method where as result of SVD process also the entries to the fuzzy rules are obtained at the same time based on the descending order of the

singular values. Generally this information is given by the permutation matrix as mentioned in the preceding section.

D. Comparative Studies

Using the same data in [6], the results of this research obtained from the same analysis is given in Table II, where in contrast with Table I, there is one additional column indicating the results obtained from the joint orthogonal method namely OLS-SVD, introduced by the present research. This is the rightmost column in Table II.

TABLE II
COMPARATIVE RESULTS FROM THE PRESENT RESEARCH

Rules	i	ols	ed	svd-qr	tls	d-svd	ols-svd
Retained rules	1	5	25	25	8	4	4
	2	24	4	4	17	25	7
	3	25	7	7	12	6	13
	4	21	19	19	13	9	3
	5	23	3	3	22	18	24
	6	19	8	8	7	23	14
	7	16	24	24	3	17	19
	8	11	13	13	21	21	17
	9	8	23	23	25	7	21
	10	3	14	14	4	19	23
	11	22	17	17	24	16	25
	12	6	22	22	18	24	6
	13	4	21	21	23	3	11
	14	7	18	18	19	2	8
	15	15	12	12	9	22	12
	16	13	9	9	14	13	10
	17	9	11	11	16	11	16
	18	18	2	2	2	8	22
	19	12	5	5	6	12	18
	20	1	16	16	11	14	2
Eliminated rules	21	2	15	6	1	10	5
	22	14	6	15	5	5	1
	23	17	1	20	10	15	9
	24	10	10	10	20	1	15
	25	20	20	1	15	20	20

The comparison of Table I and Table II indicates approximately the same results for five mutual orthogonal methods applied, i.e., *OLS algorithm*, *eigenvalue decomposition (ED)*, *singular value decomposition with column pivoting (SVD-QR)*, *total least squares (TLS)* and *direct singular value decomposition (D-SVD)*. The minor differences are attributed to numerical round-off errors. The results from these five orthogonal transformation methods are apparently consistent in both cases. It is interesting to note that, OLS method gives a slightly different sequence. This may be attributed to two different reasons as explained below.

The first reason is due to the Gram-Schmidt orthogonalisation. Namely, the objective of the OLS method is to find the smallest subset of a fixed original basis. For this, the possibilities available to the procedure are restricted to the various combinations of the original basis vectors where these are the principal components. The OLS algorithm is effectively trying to approximate these principal components

as closely as possible. This attitude may be good enough for the data set at hand for a compact energy representation which means fewer rules in the model. However, this may be disadvantageous to represent the general characteristics of the model that it should have. This different attitude in the orthogonalisation procedure manifests itself clearly in the eliminated rules.

The second reason is due to DC component at the output. It should be pointed out that in this research, to compare the results with the earlier ones reported, in the OLS algorithm, the DC component is not removed. Although it is relatively small, it can be significant in the configuration of the rules. Especially for the rules lower in the gradation, the energy contribution to output is rather small, meaning that the DC component can have much significance relative to them. The DC removal of the output in OLS training is important. An additional node in the OLS algorithm can take this care of. The slightly different performance of OLS deserves a further investigation to gain insight into the matter.

The important observation from results of the OLS-SVD method in Table 2 is the replacement of rule 9 with the rule 10 so that the rule 9 is eliminated. This is in contrast with the outcomes of other methods. Also, the method has its own gradation for the rules, although the eliminated rules almost corroborates with those given by the rest of the methods. To identify the implication of this new gradation of rules requires new research in this direction. However, the important implications of this new approach is presented in the next section.

V. CONCLUSIONS

In a fuzzy model, a balance between reducing the fitting error and increasing the model complexity is essential. In the majority of the previous works, the input space is the essential concern for rule selection. In this work there are two essential points. First, for the rule selection process both the output space of the fuzzy model as well as the input space are considered. Investigation on the clustering process using input-output training samples together in connection with RBF network training has been reported [20]. Also, the OLS and total least squares methods both include the output space for rule selection. However, in these applications output space has a relatively marginal role in contrast to the present work where output space has a primary role. The work addresses two different issues. These are RBF network training by OLS and rule selection for a fuzzy model by OLS. In the RBF network training, since the regression matrix is non-negative definite and symmetric, the PCA analysis is applied first, which is complemented by OLS afterwards. By means of this approach, a higher energy compaction by PCA is obtained. This implies that, the global energy characteristics of the data set are better observed in the model development. The main characteristic of this novel approach is the replacement of the Gram-Schmidt orthogonalisation by the eigenvalue decomposition. The

gradation of the components is accomplished according to their contribution to the output on the basis of output variance reduction, so that output space is explicitly plays the role in rule selection. Note that in this selection process the magnitude of the eigenvalues of the rule base is not central to the selection. Principal components with low eigenvalues may however have relatively more energy contribution to the output depending on the data set at hand. The case corroborates the result that selecting principal components on the basis of variance reduction does not necessarily imply that the corresponding component is unimportant in the regression [13]. Following above gradation process for the eigenvalues, the influential rules are selected according their rank in this gradation by integrating the information from the permutation matrix into this process. Referring to energy compaction, another research along this line is reported earlier [21]. Next to efficient training of RBF network, this method is important in data mining applications where the number of data samples may be high in number. In this context, for some situations it may have significance for fuzzy systems as well. Using soft data from a building technological design the application of the present method in data mining context is presented in another work for determining the functional attribute relations between selected pairs of design variables [22]. Concerning the fuzzy rule selection, the same method above is applied where SVD replaced PCA, because of the peculiarity of the firing matrix in this case. SVD method is complemented with the OLS method for the selection of the rules. Namely, by SVD the orthogonal transformation of the data samples is accomplished and the orthogonal vectors are scaled by the singular values. The OLS algorithm accomplishes the gradation of the scaled orthogonal vectors, and the graded rules are obtained by the help of SVD-QR column pivoting method which is integrated to the results obtained from the OLS method for a final outcome of rule selection. The comparative studies reported here are not enough to determine the exact added value of the orthogonal transformation method introduced. However, the new method, has desirable features in the orthogonal transformation process and therefore most appealing. It basically gives compatible results with the results of general orthogonal transformation methods and it preserves its own properties and merits at the same time. These properties and merits are subject to exploration for enhanced performance in fuzzy modeling applications.

REFERENCES

- [1] G. C. Mouzouris and J. M. Mendel, "Designing fuzzy logic systems for uncertain environments using a singular-value-QR decomposition method," in *Proc. 5th IEEE Int Conf. Fuzzy Systems*, New Orleans, LA, Sep.1996, pp.295-301
- [2] L. X Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least squares learning," *IEEE Trans. Neural Networks*, Vol.3, pp.807-814, 1992
- [3] J. Yen and L. Wang, "An SVD-based fuzzy model reduction strategy," in *Proc. 5th IEEE Int. Conf. Fuzzy Systems*, New Orleans, LA, Sept. 1996, pp.835-841
- [4] J. Yen and L. Wang, "Application of orthogonal transformation in the construction of reduced fuzzy models," in *Proc. 7th IFSA World. Congress*, Prague, Czech Republic, June 1997, Vol.2, pp.342-347
- [5] J. Yen and L. Wang, "Simplification of fuzzy rule based systems using orthogonal transformation," in *Proc. 6th IEEE Int. Conf. Fuzzy Systems*, Barcelona, Spain, July 1997
- [6] J. Yen and L. Wang, "Simplifying of fuzzy rule-based models using orthogonal transformation methods," *IEEE Trans. Systems, Man and Cybernetics-Part B: Cybernetics*, Vol.29, No.1, Feb.1999, pp.13-24
- [7] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, pp.116-132, 1985
- [8] J. S. R. Jang and C. T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Trans. Neural Networks*, vol.4, pp.156-158, Jan.1993
- [9] K. J. Hunt, R. Haas and R. Murray-Smith, "Extending the functional equivalence of radial basis function networks and fuzzy inference systems," *IEEE Trans. Neural Networks*, vol.7, no.3, pp.776-781, May 1996 [8] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol.50, no.5, pp. 1873-1896, 1989
- [10] S. Chen, C. F. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis functions network," *IEEE Trans. Neural Networks*, vol.2, pp.302-309, March 1991
- [11] S. Chen, P. M. Grant and C. F. N. Cowan, "Orthogonal least-squares algorithm for training multi-output radial basis function networks," *IEE proceedings-F*, vol.139, no.6, pp.378-384, December 1992
- [12] G. Strang, *Linear Algebra and Its Applications*, New York: Academic 1980
- [13] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York, 1986
- [14] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol.1, pp.281-294, 1989
- [15] K. C. Nisbet and B. Mulgrew and S. McLaughlin, "Reduced state methods in nonlinear prediction," *Signal Processing*, vol.28, pp.37-49, 1996
- [16] S. van Huffel, "Analysis of the total least squares problem and its use in parameter estimation," *Doct. Diss. ESAT, Electrical Eng. Dept., Kath. UnivLeuven*, 1987
- [17] G. H. Golub, C. F. van Loan, "*Matrix Computations*". North Oxford Academic/ Johns Hopkins University Press, 1983
- [18] S. van Huffel, J. vandewalle and J. Star, "The total least squares problem: Formulation, algorithm and applications," *Proc. IEEE ISCAS Conf.*, Montreal, May 1984, pp.328-331
- [19] J. Vandewalle and B. de Moor, "A variety of applications of SVD in identification and signal processing," *SVD and Signal Processing: Algorithms, Application and Architectures*, E.F. Deprettere (Ed.), Elsevier Science Pub., North Holland, 1988, pp.43-91
- [20] Z. Uykan et al., "Analysis of Input-Output Clustering for Determining Centers of RBFN," *IEEE Trans. Neural networks*, vol.11, no.4, July 2000
- [21] O. Ciftcioglu, "Studies on the complexity reduction with orthogonal transformation," in *Proc. IEEE International Conference on Fuzzy Systems*, May 12-17, 2002, Honolulu, pp.1476-1481
- [22] O. Ciftcioglu, "Functional Attribute relations by data modeling and application for knowledge discovery," *Proc. The 8th World Multiconference on Systemics, Cybernetics and Informatics*, July 18-21, 2004, Orlando, Florida, USA