

SVD-Based Complexity Reduction to TS Fuzzy Models

Péter Baranyi, Yeung Yam, *Senior Member, IEEE*, Annamária R. Várkonyi-Kóczy, *Senior Member, IEEE*, Ron J. Patton, *Member, IEEE*, Pál Michelberger, and Masaharu Sugiyama

Abstract—One of the typical important criteria to be considered in real-time control applications is the computational complexity of the controllers, observers, and models applied. In this paper, a singular value decomposition (SVD)-based complexity reduction technique is proposed for Takagi Sugeno (TS) fuzzy models. The main motivation is that the TS fuzzy model has exponentially growing computational complexity with the improvement of its approximation property through, as usually practiced, increasing the density of antecedent terms. The reduction technique proposed here is capable of defining the contribution of each local linear model included in the TS fuzzy model, which serves to remove the weakly contributing ones as according to a given threshold. Reducing the number of models leads directly to the computational complexity reduction. This work also includes a number of numerical and application examples.

Index Terms—Anytime systems, complexity reduction, fuzzy rule base reduction, singular value decomposition (SVD), TS fuzzy model.

I. INTRODUCTION

THE complexity of today's system embedding control, measurement, monitoring, and diagnosis techniques have increased to such a degree that designers and engineers would have difficulties in the application of "classical" design methods and tools. On the other hand, growing demands are arising in the reliability, safety, robustness, adaptability, and

low-cost operations of this new generation of engineering systems. To overcome these problems, the experts have turned their attention from the often used conventional model-based approaches to the more "nonconventional" types. As a result, artificial-intelligence-based and soft computing methods have gained ground in both fundamental research and application [16]. In this paper, we focus on the complexity problem of applying fuzzy logic to model-based approaches in fault diagnosis. However, the validity of the methods presented here and their consequences are more general. They can be adopted in more general modeling, diagnosis, and control systems.

System models are aimed at capturing relationships between measured variables and system component parameters. In model-based systems, the *a priori* knowledge is directly represented as a built-in model capable of following changes in the supervised system or its environment for optimal performance. Fault Detection and Isolation (FDI) methods often employ failure models to establish the relationship between measurements and a preenumerated set of faults [21]. However, burdened by their limited capability, the identification is restricted to only the usual and preknown faults. To overcome this disadvantage, more general observer-based functional models can be employed to describe the system behavior and the fault detection or isolation can then be based on the analysis of the reported deviations in the context of the given model [3]. The signals representing the inconsistencies between the model and the actual system being monitored are called residuals. The problem with this method is that any modeling error will affect the fault diagnosis performance, i.e., performance will depend not only on the presence of faults but on modeling uncertainty as well [11]. All these point to the need for more precise and accurate modeling and robust techniques.

Fuzzy tools has been proven to be advantageous in the above applications due to their robustness against uncertain and inaccurate knowledge, shortage of relevant information, lack of exact mathematical model(s), and errors in the measurements. Furthermore, they can mimic and implement the actions of human experts without accurate mathematical models and are easily interpretable. They seem to result in a real breakthrough especially in nonlinear problems, but their use is restricted by their exponential complexity. In this regard, there exist methods that offer partial solutions to the problem, e.g., the supervisory control ([8], [15]) approach reduces the models at a lower level to yield simpler structures for observers and controllers using Takagi-Sugeno (TS) models. TS fuzzy observers are used particularly because of their advantages in making the error dynamics independent of the parameters of the system. Due to

Manuscript received June 19, 2001; revised December 10, 2001. Abstract published on the Internet January 9, 2002. This work was supported by the Hungarian National Science Research Funds (OTKA) F030056, T030655, and T026254 and the British-Hungarian Intergovernmental Scientific Co-operation Fund GB 37/98. The work of P. Baranyi was supported by the Zoltán Magyary Foundation.

P. Baranyi is with the Integrated Intelligent Systems, Japanese-Hungarian Laboratory, Department of Telecommunication and Telematics, Budapest University of Technology and Economics, Budapest H-1111, Hungary (e-mail: baranyi@ttt-202.ttt.bme.hu).

Y. Yam is with the Department of Automation and Computer Aided Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: yyam@acae.cuhk.edu.hk).

A. R. Várkonyi-Kóczy is with the Integrated Intelligent Systems Japanese-Hungarian Laboratory, Department of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest H-1111, Hungary (e-mail: koczy@mit.bme.hu).

R. J. Patton is with the School of Engineering, The University of Hull, Hull, HU6 7RX, U.K. (e-mail: r.j.patton@eng.hull.ac.uk).

P. Michelberger is with the Control and Intelligent Systems Engineering Research Group, Department of Research Group for Mechanics, Hungarian Academy of Sciences, Budapest University of Technology and Economics, Budapest H-1111, Hungary.

M. Sugiyama is with the Integrated Intelligent Systems Japanese-Hungarian Laboratory, Research Institute of Manufacturing Information Technology, Gifu 509-0108, Japan (e-mail: sugi@gifu-irtc.go.jp).

Publisher Item Identifier S 0278-0046(02)02878-2.

their relevancy in the present work as well, a brief discussion on the TS fuzzy model and its computational complexity problem is given below.

TS fuzzy models are fuzzy models where multiple models are applied to the fuzzy inference systems. The inputs of the inference system coming from the environment (the fuzzy inference variables) are in fact quantitative variables. The qualitative properties of this variable, however, are derived in order to combine linear models into a new nonlinear dynamic structure under the multiple-model fuzzy inference modeling strategy of Takagi-Sugeno [27]. In this way, the TS fuzzy model structure can be analyzed [29] and designed to closely resemble the true nonlinear input-output and state variable behavior of the real process. The TS multiple-model scheme has been used both for the feedback control [28], [30], [31] and for the design of observers for FDI [32]. Both are based on the principle of *parallel distributed compensation* [33], [34]. By extension, the FDI structure is also based upon the TS multiple-model system. This can be viewed as either a nonlinear observer or as a (in a restricted sense) fuzzy observer (because of the use of the fuzzy inference variable). The use of a TS fuzzy model offers a way to describe nonlinear dynamics using local linear models [27]. Thus, according to the TS model, a nonlinear dynamic system can be linearized around a number of operating points. Each local linear model represents the local system behavior around a certain operating point. The global system behavior is described by a fuzzy fusion of all the linear model outputs. The model is described by fuzzy IF-THEN rules, which represent local linear relations of the nonlinear system. Appropriately chosen operating points, i.e., the number of local linear models and the size of the corresponding regions used in the supervision system, can guarantee the asymptotic stability of the dynamic system [5]. However, there can be a serious limitation on the applicability of such control schemes because the computational complexity of the system increases exponentially with the number of models, and finding the minimum number of the necessary models would be of great importance. In other words, despite all of the above advantages, applications of TS fuzzy models are restricted by model complexity. The problem becomes even more acute because of the contradictory requirements in performance. On one hand, there is an effort to use a great number of fuzzy rules or a great number of local, linear fuzzy models to ensure a good approximation with enough precision to guarantee the observer's stability. On the other hand, designers and experts are tempted to reduce the number of the applied models and rule-bases to keep the complexity of the system and the costs of the operations as low as possible. Unfortunately, there is no standardized framework regarding the design, optimality, reducibility, and partitioning of a fuzzy rule set and thus the success of finding the optimal tradeoff is only by chance. Fuzzy rule bases describing system behavior, which often is the case, contain redundant, weakly contributing, or outright inconsistent components [22]. Techniques offering tools to extract the more pertinent elements and to filter out unnecessary or weakly contributing parts of a given rule set are, hence, highly desirable. Among the wide variety of known complexity reduction techniques that may contribute in overcoming the above burden, we focus in this work on the singular value decomposition (SVD)

technique because it also offers an excellent extension for "anytime" operations [35]. This paper is hence an attempt to investigate a TS fuzzy observer and model from the complexity and possible complexity reduction point of view.

In computer-based monitoring and diagnostic systems, operations should be performed under prescribed response time conditions. Given that sufficient computational power is provided, the achievable processing speed is highly influenced by the procedures, timing, and data access conditions of the processing itself. It may be inevitable even in the case of extremely careful design to get into situations where the shortage of necessary data and/or processing time becomes serious. Such situations may result in a critical breakdown of the monitoring and/or diagnostic systems [25]. The concept of "anytime" processing tries to handle the case of too many abrupt changes and their consequences in larger scale embedded systems [2]. The idea is that, if there is a temporal shortage of computational power and/or there is a loss of some data, the actual operation can be continued to maintain the overall performance "at a lower price," i.e., information processing based on algorithms and/or models of reduced complexity should provide outputs of acceptable quality to continue the operation of the full monitoring system. The accuracy of the process will be temporarily lowered but possibly still sufficient to produce data for qualitative evaluations and supporting decisions. Consequently, "anytime" algorithms provide short response time and are very flexible with respect to the available input information and computational power.

The TS fuzzy model-based approaches combined with the SVD technique are excellent tools for "anytime" operations [35]. By using SVD, not only the "sequence" of the rules is defined but also the extent to which they contribute to the mapping. To cope with the limits arising in the system or in its environment determined by the computational need of the remaining truncated model, we can appropriately abandon the less significant parts of the rule base online. This can be conducted based on the singular values, which also manage to give the approximation error of the process [4], [22]–[24]. The key ideas of this paper are based on the SVD fuzzy rule base reduction technique initialized in [1], [4], and [22]–[24]. The transformation of the antecedent sets via SVD decomposition, which is extended here to local linear models, is introduced in [4], [22], and [23]. Presumably, the SVD technique here can be replaced by other orthogonal techniques as in [24]. Extension of [24] to multi-dimensional cases may also be conducted in a similar fashion as high-order SVD [4], [8], [23].

In the following, we present the exact and nonexact SVD-based model reduction methods for TS fuzzy models to decrease model complexity and thus to open up possibility for their application in extremely large systems or in such systems where serious temporal limitations may occur in the computing and/or timing conditions during the operations. The present work focuses on product-sum-gravity inference. However, algorithms introduced here can readily be extended to other fuzzy inference techniques by replacing the product inference form with generalized tuneable operators introduced in, for instance, the works of Rudas [17]–[20]. This paper is organized as follows. Section II gives the definitions of the parameters adopted for later development. Section III gives a brief

overview on the TS fuzzy models and observers. Section IV investigates the computational complexity of such systems. Section V presents the SVD-based complexity reduction methods. Section VI presents the performance of the reduction. Section VII shows the possible extension of the present method to the multi-dimensional case. Section VIII gives the examples to illustrate the strength of the present methods and, finally, Section IX gives the conclusions.

II. DEFINITIONS

This section introduces the elementary definitions and concepts utilized in later sections. Before starting with the definitions, we note the following on the notations to be utilized. To facilitate the distinction between the types of given quantities, scalar values are denoted by lowercase letters $\{a, b, \dots; \alpha, \beta, \dots\}$; column vectors and matrices are given by bold-face letters as $\{\mathbf{a}, \mathbf{b}, \dots\}$ and $\{\mathbf{A}, \mathbf{B}, \dots\}$ respectively, matrix $\mathbf{0}$ contains zero values only; multi-way matrices correspond to capital letters as $\{A, B, \dots\}$. The transpose of matrix \mathbf{A} is denoted as \mathbf{A}^T . A subscript is consistently used for a lower order of a given structure, e.g., the elements of matrix \mathbf{A} are marked by the row-column number of i, j and symbolized as $(\mathbf{A})_{i,j} = a_{i,j}$. Throughout this work, the i th column vector of \mathbf{A} is denoted as \mathbf{a}_i , i.e., $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \dots]$. To enhance the overall readability, characters $i, j, l, n, o, u, v, \dots$ denote indices (counters) and $I, J, L, N, O, U, V, \dots$ are reserved for the upper bounds of the indices, unless stated otherwise. $\mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ is the vector space of real valued $(I_1 \times I_2 \times \dots \times I_N)$ multi-way matrices. Fuzzy sets are denoted by capital letters, e.g., A .

Definition 1: (n -mode matrix of A) Assume an N -way matrix $A \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$. The n -mode matrix $\underline{\mathbf{A}}_{(n)} \in \mathfrak{R}^{I_n \times J}$, $J = \prod_l I_l$ contains all the vectors in the n th dimension of matrix A . The ordering of the vectors is arbitrary; this ordering shall, however, be consistently used later on. $(\underline{\mathbf{A}}_{(n)})_j$ is called the j th n -mode vector. Note that any matrix of which the columns are given by n -mode vectors $(\underline{\mathbf{A}}_{(n)})_j$ can evidently be restored to be matrix A . The restoring can be executed even when some rows of $\underline{\mathbf{A}}_{(n)}$ are discarded since the value of I_n has no role in the ordering of $(\underline{\mathbf{A}}_{(n)})_j$.

Definition 2: (n -mode matrix partition) Assume N -way matrix $A \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$. The n -mode partitions of A are $B_l \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_{n-1} \times J_l \times I_{n+1} \times \dots \times I_N}$ denoted as $A = [B_1 B_2 \dots B_L]_n$, where $I_n = \sum_l J_l$.

Definition 3: (n -mode matrix product) The n -mode product of $A \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ by a matrix $\mathbf{U} \in \mathfrak{R}^{J \times I_n}$, denoted by $A \times_n \mathbf{U}$, is an $(I_1 \times I_2 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N)$ matrix of which the entries are given by $A \times_n \mathbf{U} = B$, where $B_{(n)} = \mathbf{U} \cdot \mathbf{A}_{(n)}$. For more explanation of the above definitions, see [14] and [39].

III. THE TS FUZZY MODEL

This section discusses the basics of the TS fuzzy model. Without loss of generality, a fuzzy observer is given as an example. This exemplified observer taken from [5] and [36]–[38] consists of a number of local linear models designed to estimate the system state vector and is discussed here from a complexity

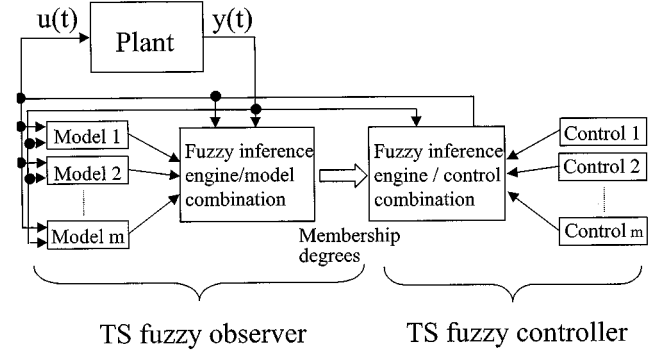


Fig. 1. TS fuzzy model utilized to observer and controller.

point of view. Fig. 1 shows the block diagram of a TS fuzzy observer from [5] and [13], together with a possible TS fuzzy controller. Since they are the same in the sense of computational complexity, the reduction technique to be discussed here can be performed on both the TS fuzzy observer and the controller.

For the fuzzy observer design, it is assumed that the fuzzy system model is locally observable. Using the idea of parallel distributed compensation (PDC) [26], a linear time-invariant observer can be associated with each rule of the TS fuzzy model. Let us suppose that the models are varying with respect to a fuzzy variable ω , which yields the following fuzzy rules [5], [13], [37]:

$$\text{IF } \omega \text{ is fuzzy set } A_v \text{ THEN model } \mathbf{M}_v \quad (1)$$

with $v = 1, \dots, V$ and V is the number of fuzzy terms, namely, the number of local linear models. Equation (1) involves only one variable ω . The case of multi-dimensional parameter space will be briefly discussed in Section VII. \mathbf{M}_v in (2) is defined for the observers as

$$\left. \begin{aligned} \dot{\hat{\mathbf{x}}}(t) &= \underline{\mathbf{A}}_v \hat{\mathbf{x}}(t) + \underline{\mathbf{B}}_v \mathbf{u}(t) + \underline{\mathbf{L}}_v (\mathbf{y}(t) - \hat{\mathbf{y}}(t)) \\ \hat{\mathbf{y}}(t) &= \underline{\mathbf{C}}_v \hat{\mathbf{x}}(t) \end{aligned} \right\} \Rightarrow \mathbf{M}_v. \quad (2)$$

Let us define arbitrary shaped fuzzy sets $A_v : \mu_{A_v}(\omega) = f_v(\omega)$. The TS fuzzy model approximation based on the combination of the local models is then

$$\hat{\mathbf{M}}(\omega) = \sum_{v=1}^V f_v(\omega) \mathbf{M}_v. \quad (3)$$

The structure of the TS fuzzy model based approximation is depicted in Fig. 2, which shows that the output of each linear model is contributed by function $f_v(\omega)$ to the output. In order to have general discussion, let us consider the following form:

$$\mathbf{M}_v \Rightarrow \begin{bmatrix} \mathbf{z}_{1,v} \\ \mathbf{z}_{2,v} \\ \vdots \\ \mathbf{z}_{L,v} \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{B}}_{v,1,1} & \underline{\mathbf{B}}_{v,1,2} & \dots & \underline{\mathbf{B}}_{v,1,U} \\ \underline{\mathbf{B}}_{v,2,1} & \underline{\mathbf{B}}_{v,2,2} & & \underline{\mathbf{B}}_{v,2,U} \\ \vdots & & \ddots & \\ \underline{\mathbf{B}}_{v,L,1} & \underline{\mathbf{B}}_{v,L,2} & \dots & \underline{\mathbf{B}}_{v,L,U} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_U \end{bmatrix} \quad (4)$$

where L denotes the number of rows in the model (i.e., the number of equations describing the model) and U indicates how many terms there are in the rows of the equations (2 and 3, respectively, in the case of (2)). Vector \mathbf{x}_u consists of the model inputs and state vectors. Further, in the case of (2), $\mathbf{x}_1 = \mathbf{x}(t)$,

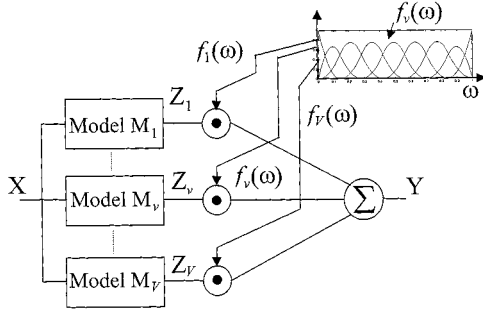


Fig. 2. Structure of the TS fuzzy model. The number of local linearized models and function $f_v(\omega)$ is V .

$\underline{x}_2 = \underline{u}(t)$, and $\underline{x}_3 = \underline{y}(t) - \hat{\underline{y}}(t)$, the outputs of the v th local linear model are $\underline{z}_{1,v} = \dot{\underline{x}}_v(t)$ and $\underline{z}_{2,v} = \underline{y}_v(t)$. The coefficient matrices of (4) are then: $\underline{\mathbf{B}}_{v,1,1} = \underline{\mathbf{A}}_v$, $\underline{\mathbf{B}}_{v,1,2} = \underline{\mathbf{B}}_v$, $\underline{\mathbf{B}}_{v,1,3} = \underline{\mathbf{L}}_v$ and $\underline{\mathbf{B}}_{v,2,1} = \underline{\mathbf{C}}_v$, $\underline{\mathbf{B}}_{v,2,2} = \mathbf{0}$ and $\underline{\mathbf{B}}_{v,2,3} = \mathbf{0}$, i.e., some of the matrices $\underline{\mathbf{B}}_{v,l,u} \in \mathfrak{R}^{O_l \times I_u}$ contain only zero elements. Let us form the three-way matrices $B_{l,u} \in \mathfrak{R}^{O_l \times I_u \times V}$ from $\underline{\mathbf{B}}_{v,l,u} \in \mathfrak{R}^{O_l \times I_u}$. Substituting (4) into (3), the following is obtained.

Definition 4: General form of the weighted combination of models:

$$\underline{y}_l(t) = \sum_{v=1}^V f_v(\omega) \underline{z}_{l,v}(t) = \sum_{v=1}^V f_v(\omega) \sum_{u=1}^U \underline{\mathbf{B}}_{v,l,u} \underline{x}_u(t). \quad (5)$$

Finally, for further notation, let the weighting functions $f_v(\omega)$ be contained in row vector $\underline{\mathbf{f}}(\omega)$.

IV. COMPLEXITY INVESTIGATION

This section investigates the computational complexity of (5). The investigation considers only the number of multiplication operations in calculating the output. Let us write (5) in the form of

$$y_{i,l}(t) = \sum_{v=1}^V f_v(\omega) \sum_{u=1}^U \sum_{j=1}^{I_u} b_{i,j,v,l,u} x_{j,u}(t) \quad (6)$$

where $b_{j,i,v,l,u}$ stands for the (j, i) th element of matrix $\underline{\mathbf{B}}_{v,l,u} \in \mathfrak{R}^{O_l \times I_u}$, $y_{i,l}(t)$ is the i th element of output vector $\underline{y}_l(t)$, and, similarly, $x_{j,u}(t)$ is the j th element of $\underline{x}_u(t)$. According to (6), we have the following Lemma.

Lemma 1: The computational complexity of (5) grows exponentially with the number of models. Considering the multiplication operation only, the computational requirement is characterized as

$$P = V \left(\sum_{l=1}^L \sum_{u=1}^U O_l I_u + \sum_{l=1}^L O_l + C_p \right) \quad (7)$$

where C_p indicates the number of multiplications during the calculation of an $f(\omega)$. To arrive at (7), one notes that calculating $\underline{\mathbf{B}}_{v,l,u} \underline{x}_u$ needs $O_l \cdot I_u$ multiplications. Furthermore, there are U number of terms $\underline{\mathbf{B}}_{v,l,u} \underline{x}_u$ in the rows of (4) which leads to the term $\sum_{l=1}^L \sum_{u=1}^U O_l I_u$. This is multiplied by V for the calculations of the outputs of the V models. Moreover, the output is given by the linear combination of the V outputs of models,

which yields the term $V \cdot \sum_{l=1}^L O_l$. Finally, the term $V \cdot C_p$ is the calculation of the $f(\omega)$'s. Lemma 1 shows that the computational complexity explodes with the number of local linear models. As pointed out in the introduction, this gives rise to difficulties in our pursuit of good approximation and hence system performance.

V. COMPLEXITY REDUCTION

The main objective of this section is to propose a method capable of reducing the complexity of the TS fuzzy model approximation. The aim here is to decrease V , namely, to find the minimal number of local linear models and corresponding antecedents. A subsequent goal is to present a method capable of decreasing O_l and I_u , which is in the dominant growing part of (7). Actually, O_l and I_u are the number of elements in the state, input, and output vectors defined by the application at hand and hence cannot be reduced directly. However, the TS fuzzy model approximation can be transformed into a subspace where O_l and I_u may become $O_l^r < O_l$ and $I_u^r < I_u$. Output calculations performed in this subspace (which we will term minimal computational space) may then require less computational efforts, after which the result can be projected back to the original space for the final output. If the transformation is reversible, then the reduction is exact in the sense that the output of the original model is the same as the reduced one, otherwise a reduction error is produced. This idea is performed in Method 1 of this section. Method 2 in this section is aimed at finding the minimal number of models and the corresponding weighting functions. The reduction is exact if the original model has redundant local linear models; if it does not, then the reduction can be increased at the price of nonexact reduction. The more reduction we aim for, the larger approximation error may occur. Therefore, the algorithms in this section have an error-controllable property, which can help us with executing the reduction technique according to a predefined reduction error tolerance, which safely conserves the system performance. The main concept of the reduction is defined in the following theorem.

Theorem 1: Equation (5) can always be transformed into the following form:

$$\begin{aligned} \underline{y}_l(t) &= \underline{\mathbf{A}}_l \sum_{v=1}^{V^r} f_v^r(\omega) \underline{z}_{l,v}^r(t) \\ &= \underline{\mathbf{A}}_l \sum_{v=1}^{V^r} f_v^r(\omega) \sum_{u=1}^U \underline{\mathbf{B}}_{v,l,u}^r \underline{\mathbf{C}}_u^T \underline{x}_u(t) \end{aligned} \quad (8)$$

where “ r ” denotes “reduced, the size of $\underline{\mathbf{A}}_l$, $\underline{\mathbf{C}}_u$, and $\underline{\mathbf{B}}_{v,l,u}^r$ are $O_l \times O_l^r$, $I_u \times I_u^r$, and $O_l^r \times I_u^r$, respectively. Further, $\forall l: O_l^r \leq O_l, \forall u: I_u^r \leq I_u$ and the number of models is reduced to $V^r \leq V$. The proof can be derived readily from the following Methods 1 and 2. Before starting with the algorithms, let us briefly digress and represent (4) in different ways. Let the multi-way matrix $G_l \in \mathfrak{R}^{O_l \times (\sum_u I_u) \times V}$ be given by the form of n -mode matrix partition as: $G_l = [B_{l,1} \ B_{l,2} \ \dots \ B_{l,U}]_2$. This actually fits the coefficient matrices in the two-dimensional arrangement and, as a result, (4) can be written as

$$\underline{z}_l = (G_l \times_2 [\underline{x}_1^T \ \underline{x}_2^T \ \dots \ \underline{x}_U^T])_{(1)} \quad (9)$$

where $\underline{\mathbf{Z}}_i \in \mathfrak{R}^{O_i \times V}$ is understood as $\underline{\mathbf{Z}}_i = [\mathbf{z}_{i,1} \ \mathbf{z}_{i,2} \ \cdots \ \mathbf{z}_{i,V}]$. Let multi-way matrix $H_u \in \mathfrak{R}^{(\sum_i O_i) \times I_u \times V}$ be given as: $H_u = [B_{1,u} \ B_{2,u} \ \cdots \ B_{L,u}]_1$. Thus, (4) can also be given in the form of

$$\begin{bmatrix} \underline{\mathbf{Z}}_1 \\ \underline{\mathbf{Z}}_2 \\ \vdots \\ \underline{\mathbf{Z}}_L \end{bmatrix} = ([H_1 \ H_2 \ \cdots \ H_U]_{2 \times 2} [\underline{\mathbf{x}}_1^T \ \underline{\mathbf{x}}_2^T \ \cdots \ \underline{\mathbf{x}}_U^T])_{(1)}$$

The reduction is conceptually obtained by the singular value-based decomposition of (5) and by the truncation of the zero or smallest singular values with their corresponding decomposed parts. Hence, we have the following definition.

Definition 5: Singular Value Based Reduction (SVDR). Let us suppose that matrix $\underline{\mathbf{B}}_{(n_1 \times n_2)} = [b_{i,j}]$ is given. Applying singular value decomposition yields

$$\underline{\mathbf{B}} = \underline{\mathbf{U}} \underline{\mathbf{D}} \underline{\mathbf{V}}^T = \left[\underline{\mathbf{U}}^r \mid \underline{\mathbf{U}}^d \right] \begin{bmatrix} \underline{\mathbf{B}}^r & \underline{\mathbf{0}} \\ \underline{\mathbf{0}} & \underline{\mathbf{B}}^d \end{bmatrix} \left[\underline{\mathbf{V}}^r \mid \underline{\mathbf{V}}^d \right]^T. \quad (9)$$

Matrices $\underline{\mathbf{U}}$ and $\underline{\mathbf{V}}$ are orthogonal. Matrix $\underline{\mathbf{D}}$ contains the singular values in decreasing magnitude, as diagonal elements. The zero or the smallest singular values (smaller than singular value threshold T_0 , say) can be discarded to yield a simpler system. Let $\underline{\mathbf{B}}^r$ contain the retained and $\underline{\mathbf{B}}^d$ contain the discarded singular values. Let the result of SVDR be: $\hat{\underline{\mathbf{B}}} = \underline{\mathbf{U}}^r \underline{\mathbf{B}}^r \underline{\mathbf{V}}^{rT}$.

For a more detailed description of SVD, see [6], [9] and [10]. Having defined SVDR, the following reduction techniques can then be proposed for the TS fuzzy model approximations in accordance with Theorem 1. First, we discuss the case when only zero singular values are discarded in SVDR. Then, an error bound of the more general reduction case is discussed in Remark 2.

Method 1 (Projection to Minimal Computational Space):

- I) Determination of matrices $\underline{\mathbf{A}}_i$, namely, the reduction of O_i . Let $\underline{\mathbf{S}}_i = (G_i)_{(1)}$. Then, applying SVDR to $\underline{\mathbf{S}}_i$ yields $\underline{\mathbf{S}}_i = \underline{\mathbf{A}}_i \cdot \underline{\mathbf{D}}_i \cdot \underline{\mathbf{V}}_i = \underline{\mathbf{A}}_i \cdot \underline{\mathbf{S}}_i'$. Matrix $\underline{\mathbf{S}}_i' \in \mathfrak{R}^{O_i^r \times V \cdot \sum_u I_u}$ can be restored to multi-way matrix $G_i' \in \mathfrak{R}^{O_i^r \times (\sum_u I_u) \times V}$.
- II) Determination of matrices $\underline{\mathbf{C}}_u$, namely, the reduction of I_u . Let us construct $H_u' \in \mathfrak{R}^{(\sum_i O_i) \times I_u \times V}$ which is resulted by $H_u' = [B'_{1,u} \ B'_{2,u} \ \cdots \ B'_{L,u}]_1$, where multi-way matrices $B'_{i,u}$ are defined by the result of step I), thus, $G_i' = [B'_{i,1} \ B'_{i,2} \ \cdots \ B'_{i,U}]_2$. Then let $\underline{\mathbf{M}}_u = (H_u')_{(2)}$ whereupon SVDR is performed: $\underline{\mathbf{M}}_u = \underline{\mathbf{C}}_u \cdot \underline{\mathbf{D}}_u \cdot \underline{\mathbf{V}}_u' = \underline{\mathbf{C}}_u \cdot \underline{\mathbf{M}}_u'$. Matrix $\underline{\mathbf{M}}_u'$ can be restored according to $\underline{\mathbf{M}}_u'' = (H_u'')_{(2)}$, in order to define matrices $B''_{i,u} \in \mathfrak{R}^{O_i \times I_u^r \times V}$ via $H_u'' = [B''_{1,u} \ B''_{2,u} \ \cdots \ B''_{L,u}]_1$. Let us assume that the values of O_i and I_u are decreased. Consequently, matrices $\underline{\mathbf{A}}_i$ and $\underline{\mathbf{C}}_u$ transform the input vectors $\underline{\mathbf{x}}_u$

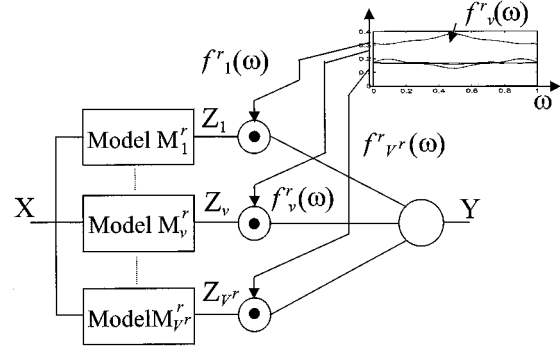


Fig. 3. Compressed model structure.

into a subspace as: $\underline{\mathbf{x}}_u^r = \underline{\mathbf{C}}_u^T \cdot \underline{\mathbf{x}}_u$, then the output values can be calculated in the reduced space as

$$\mathbf{M}_v^r \Rightarrow \begin{bmatrix} \mathbf{z}_{1,v}^r \\ \mathbf{z}_{2,v}^r \\ \vdots \\ \mathbf{z}_{L,v}^r \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{B}}_{v,1,1}^r & \underline{\mathbf{B}}_{v,1,2}^r & \cdots & \underline{\mathbf{B}}_{v,1,U}^r \\ \underline{\mathbf{B}}_{v,2,1}^r & \underline{\mathbf{B}}_{v,2,2}^r & & \underline{\mathbf{B}}_{v,1,U}^r \\ \vdots & & \ddots & \\ \underline{\mathbf{B}}_{v,L,1}^r & \underline{\mathbf{B}}_{v,L,2}^r & \cdots & \underline{\mathbf{B}}_{v,L,U}^r \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^r \\ \mathbf{x}_2^r \\ \vdots \\ \mathbf{x}_U^r \end{bmatrix}$$

which requires less computational effort than in the original case (4) since the size of the coefficient matrices are decreased. The resulting output vector is the combination of the outputs of the local models (5): $\underline{\mathbf{y}}_i^r(t) = \sum_{v=1}^V f_v(\omega) \mathbf{z}_{i,v}^r(t)$, which also requires less calculation than the original model because the size of vector $\underline{\mathbf{y}}_i^r(t)$ is less than the size of vector $\underline{\mathbf{y}}_i(t)$, though in fact this latter reduction is not as significant as the computational reduction in $\mathbf{z}_{i,v}^r$. Output $\underline{\mathbf{y}}_i(t)$ resulted by combining the outputs of the local linear models (5) should be projected back to the original space as: $\underline{\mathbf{y}}_i(t) = \underline{\mathbf{A}}_i \underline{\mathbf{y}}_i^r(t)$. The following method serves to pinpoint the minimal number of linearized models.

Method 2 (Determination of the Minimal Number of Linear Models): Determination of V^r . Let us form matrix $\underline{\mathbf{K}} = [(B_{1,1})_{(3)} \ (B_{1,2})_{(3)} \ \cdots \ (B_{L,U})_{(3)}]$. Then, executing SVDR on matrix $\underline{\mathbf{K}}$, we obtain: $\underline{\mathbf{K}} = \underline{\mathbf{T}} \underline{\mathbf{D}}'' \underline{\mathbf{V}}'' = \underline{\mathbf{T}} \underline{\mathbf{K}}'$. $\underline{\mathbf{T}}$ has the size of $V \times V^r$. Matrix $\underline{\mathbf{K}}'$ can be restored to matrices $\underline{\mathbf{B}}_{v,l,u}^r$ according to $\underline{\mathbf{K}}' = [(B'_{1,1})_{(3)} \ (B'_{1,2})_{(3)} \ \cdots \ (B'_{L,U})_{(3)}]$. The reduced weighting functions are transformed by matrix $\underline{\mathbf{T}}$ as

$$\underline{\mathbf{f}}^r(\omega) = \underline{\mathbf{f}}(\omega) \underline{\mathbf{T}}. \quad (10)$$

The order of performing Method 1 or 2 is arbitrary. The results obtained by both methods are matrices $\underline{\mathbf{A}}_i$, $\underline{\mathbf{C}}_u$, $\underline{\mathbf{B}}_{v,l,u}^r$, and $\underline{\mathbf{f}}^r(\omega)$ in full accordance with Theorem 1. The number of models is V^r in the reduced form. The reduced form is depicted in Fig. 3.

Remark 1: Fuzzy logic theory has restrictions to the functions of fuzzy sets. The results of Method 1 and 2 do not guarantee that the resulted weighting functions can be interpreted as fuzzy sets. If the reduced form is not only for saving computational cost, but for further studies in fuzzy operation as well, then the reduced weighting functions should accommodate the pertaining characterizations. Further transformations may hence be required. For instance, to obtain matrix $\underline{\mathbf{T}}$ in such a way

that the reduced functions are bounded by [0,1]. Some tools, such as *nonnegativeness*, *sum normalization*, and *set normalization*, have been developed in [22] and [23]. These transformations are attached to SVDR to ensure that the results are interpretable as fuzzy sets, furthermore to maintain Ruspini-partition and to have normalized sets according to the concept discussed by Dubois *et al.* [7]. For more details about these transformations, see [4], [22] and [23]. In conclusion, the computational cost of the algorithm may be reduced in the final implementation, which serves our main goal, but the price to pay is that the interpretability of the fuzzy sets may be degraded. This also makes an interesting point in fuzzy theory—how to represent and extract a rule-base in different ways. In the worst case, it may be that the resulted antecedents are very weakly informative as far as fuzzy linguistic interpretation is concerned.

Remark 2: If not only zero singular values are discarded then the effectiveness of the reduction is improved, however, reduction error is obtained. The error resulting from SVDR is bounded by the sum of the discarded singular values [4], [22], [23]. The final error of the model approximation depends on the type of weighting functions applied. In this regard, several cases of the antecedents are discussed in [4]. Generally, it can be said that if the weighting functions satisfy the Ruspini-partitions, which is maintained in the above transformations, then the maximum final error is the sum of the discarded singular values controllable during the execution of Method 1 and 2. For more details of SVD reduction error, see preliminary works [4], [22].

VI. INVESTIGATION OF COMPLEXITY REDUCTION

This section compares the computational complexity of the original (5) and the reduced (8) forms. The comparison is done for two cases of the weighting functions $f_v(\omega)$. The first case assumes that the transformed weighting functions can be analytically given in the same form as the original ones. For instance, if the antecedent sets are given in the form of $f_v(\omega) = a_v \sin(c\omega + \gamma_v)$, then the result of (10) can be given in the same form as $f_v^r(\omega) = a_v^r \sin(c\omega + \gamma_v^r)$. Another example is that the weighting functions are piecewise linear or given in a general form of

$$f_v(\omega) = \sum_{m=1}^M g_{v,m} \varphi_m(\omega) \quad (11)$$

where the weighting functions are expressed in terms of the basis functions $\varphi_m(\omega)$. The second case has weighting functions where the transformed antecedents cannot be analytically simplified.

For the first case, where the transformed sets are expressible in the same form as the original one, we have the following results.

Lemma 2: The computational complexity of (8) is

$$P_1^r = V^r \left(\sum_{l=1}^L \sum_{u=1}^U O_l^r I_u^r + \sum_{l=1}^L O_l^r + C_p \right) + \sum_{l=1}^L O_l^r O_l + \sum_{u=1}^U I_u^r I_u \quad (12)$$

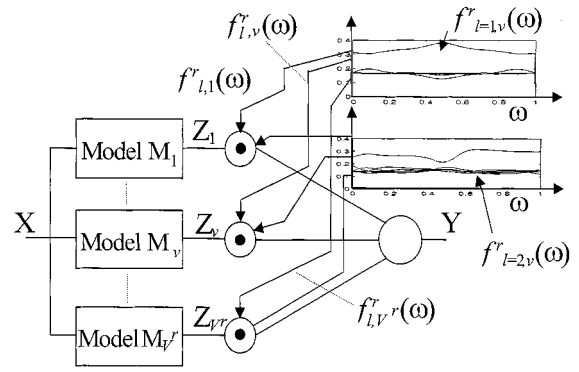


Fig. 4. Reduction where each equation in the model has its own weighting functions.

where the additional terms, compared to (7), come from the transformation to the reduced space.

For the second case, where the transformed antecedents cannot be analytically simplified, we have the following.

Lemma 3: The computational complexity of (8) is

$$P_2^r = V^r \left(\sum_{l=1}^L \sum_{u=1}^U O_l^r I_u^r + \sum_{l=1}^L O_l^r \right) + \sum_{l=1}^L O_l^r O_l + \sum_{u=1}^U I_u^r I_u + C_p V + V^r V \quad (13)$$

where $V^r V$ indicates extra computation to result the values of the reduced weighting functions from the values of the original weighting function [see (10)].

Hence, with $P_1^r \leq P_2^r$, we have both (12) and (13) showing complexity reduction over (7) except for some extreme cases, where the number of model inputs and outputs are 2 or 3. As a matter of fact, different $\underline{\mathbf{T}}_l$ can be generated for each equation for the reduced model, which yields a different set of weighting functions for each value of l . The benefit of having $\underline{\mathbf{T}}_l$ is that the size of a common $\underline{\mathbf{T}}$ may be larger than the size of any $\underline{\mathbf{T}}_l$ and the value of V^r is in the dominant exponentially growing term of (12) and (13). The disadvantage is that using $\underline{\mathbf{T}}_l$, though it may be smaller than $\underline{\mathbf{T}}$, does bring extra calculation since each l needs the calculation of a different set of weighting functions. In this light, Method 2 can be modified in such a way that $\underline{\mathbf{K}}_l = [(B_{l,1})_{(3)} \ (B_{l,2})_{(3)} \ \cdots \ (B_{l,U})_{(3)}]$ is first constructed, whereupon the execution of SVDR is carried out as $\underline{\mathbf{K}}_l' = \underline{\mathbf{T}}_l \underline{\mathbf{K}}_l'$. Coefficient matrices are then defined by $\underline{\mathbf{K}}_l' = [(B_{l,1}')_{(3)} \ (B_{l,2}')_{(3)} \ \cdots \ (B_{l,U}')_{(3)}]$ and the membership functions by $\underline{\mathbf{f}}_l^r(\omega) = \underline{\mathbf{f}}(\omega) \underline{\mathbf{T}}_l$. Consequently, to improve the reduction, one has to check whether the application of $\underline{\mathbf{T}}_l$ would lead to further reduction or not. The structure in this case of using different $\underline{\mathbf{T}}_l$ s is depicted in Fig. 4.

VII. MULTIVARIABLE CASE

This section briefly discusses a possible direction to extend the proposed reduction technique if the input parameter space is N -dimensional. In this case, the rules in a TS fuzzy model are:

IF ω_1 is A_{1v_1} AND ω_2 is A_{2v_2} \cdots AND ω_N is A_{N,v_N} , THEN model $\mathbf{M}_{v_1 v_2 \dots v_N}$.

Applying the product inference form leads to

$$\underline{\mathbf{y}}_l(t) = \sum_{v_1 v_2 \dots v_N}^{V_1 V_2 \dots V_N} \prod_n f_{n, i_n}(\omega_n) \sum_{u=1}^U \underline{\mathbf{B}}_{v_1 v_2 \dots v_N, l, u} \underline{\mathbf{x}}_u(t) \quad (14)$$

where $f_{n, j}(\omega_n)$ of row vector $\underline{\mathbf{f}}_n(\omega_n)$ is the j th membership function defined on the n th input universe. Equation (14) can be written as

$$\underline{\mathbf{y}}_l(t) = \sum_{u=1}^U B_{l, u} \times_2 \underline{\mathbf{x}}_u^T(t) \times_3 \underline{\mathbf{f}}_1(\omega_1) \times_4 \underline{\mathbf{f}}_2(\omega_2) \cdots \times_{N+2} \underline{\mathbf{f}}_N(\omega_N). \quad (15)$$

Hence, reduction is to be of the form

$$\underline{\mathbf{y}}_l^r(t) = \underline{\mathbf{A}}_l \left(\sum_{u=1}^U B_{l, u}^r \times_2 \underline{\mathbf{x}}_u^T(t) \underline{\mathbf{C}}_u \times_3 \underline{\mathbf{f}}_1^r(\omega_1) \times_4 \underline{\mathbf{f}}_2^r(\omega_2) \cdots \times_{N+2} \underline{\mathbf{f}}_N^r(\omega_N) \right)$$

where $\underline{\mathbf{f}}_n^r = \underline{\mathbf{f}}_n \underline{\mathbf{T}}_n$ in full accordance with (8) and (10). In this case, Method 2 can be performed on all dimensions of multiway matrix $B_{l, u} \in \mathbb{R}^{O_l \times I_u \times V_1 \times V_2 \cdots \times V_N}$, except the first and the second one in order to obtain transformation matrices $\underline{\mathbf{T}}_1, \underline{\mathbf{T}}_2, \dots, \underline{\mathbf{T}}_N$ for the antecedent sets of each input. Also, Method 1 can be applied to the first and second dimensions to determine $\underline{\mathbf{A}}$ and $\underline{\mathbf{C}}$. A detailed extension and examples of the multidimensional case will be given in future works.

VIII. EXAMPLES

In this section, three examples are given. The first two demonstrates the use of the proposed exact reduction technique. The third illustrates application of the technique to a TS fuzzy observer for controlling an induction motor. It is worth mentioning that the examples all use triangular-shaped fuzzy sets (resulting in a rough bilinear approximation) for simplicity, where any observation fires a maximum of two antecedents with nonzero membership values. The reduced forms will also have the same property. In this situation, it is therefore enough to consider only those rule pairs in both the original and the reduced form which are fired with nonzero values. This implies that the computational reduction is questionable. In general, computational reduction is expected when, for instance, all antecedents are fired in the original and reduced rule bases, as is typical when a B-spline basis is used for antecedents. In this regard, the antecedents in the examples can actually be replaced by a B-spline base. This would considerably improve the approximation.

Example 1: Let the local linear model be such that there is only one equation in each model and only one term in each equation, namely, $L = U = 1$. The number of models is $V = 3$. Let $O = I = 3$ and the three models be given by $\underline{\mathbf{z}}_v(t) = \underline{\mathbf{B}}_v \underline{\mathbf{x}}(t)$. They are to be combined according to the actual values of the corresponding weighting functions $f_v(\omega)$ [see (5)], which are triangularly shaped as depicted in Fig. 5. Let the elements of the coefficient matrices be $b_{i, j, v} = 2ijv - ij - iv - jv - 3i - 2j - v + 8$ which imitates an approximation of a nonlinear system. In the first step of Method 1, the two nonzero singular values are: 24.2036 and 4.71003. Here, five decimal places are used to enhance the precision. For matrix $\underline{\mathbf{A}}$, we discard the zero

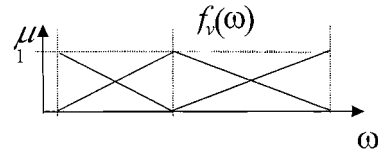


Fig. 5. Example 1: weighting functions.

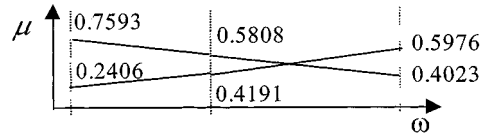


Fig. 6. Example 1: transformed antecedents $V^r = 2$, which are in a Ruspini-partition.

TABLE I
EXAMPLE 1: RESULT OF THE PROPOSED REDUCTION

$\underline{\mathbf{A}}$		$\underline{\mathbf{T}}$		$\underline{\mathbf{C}}$	
0.0301	0.7058	0.0301	-0.9046	0.0301	-0.9046
-0.4349	-0.3797	0.3941	-0.4218	0.3941	-0.4218
-0.8999	0.1529	0.9108	0.0608	0.9108	0.0608

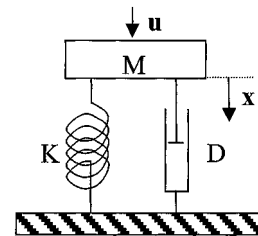


Fig. 7. Example 2: mass-spring-damper system.

singular value and keep the two nonzero singular values. For matrix $\underline{\mathbf{C}}$, two nonzero singular values are also obtained and kept. They are 23.2313 and 8.2647. We then proceed to find a common transformation matrix $\underline{\mathbf{T}}$ by Method 1, with the SVDR extended by NN and SN transformations to obtain fuzzy sets in Ruspini-partition (see [4], [22], [23], and Fig. 6). As a result, matrices $\underline{\mathbf{A}} \in \mathbb{R}^{3 \times 2}$, $\underline{\mathbf{C}} \in \mathbb{R}^{3 \times 2}$, and $\underline{\mathbf{T}} \in \mathbb{R}^{3 \times 2}$ are given in Table I.

The transformed antecedents are depicted in Fig. 6. The computational complexity reduction is now considered. Calculating $\underline{\mathbf{z}}_v(t) = \underline{\mathbf{B}}_v \underline{\mathbf{x}}(t)$, $v = 1..3$ in the original model requires $3 \times 3 \times 3 = 27$ multiplications. Transforming the input and output between the reduced and the original space requires $2 \times 3 \times 2$ multiplications. Calculation in the reduced space requires $2 \times 2 \times 2$ multiplications. The reduced form requires 20 operations in total. If the membership values of the reduced antecedents are transformed from the original weighting functions, then 3×2 extra operations are needed. In this case, the number of operations is 26, which is less than in the original model approximation.

Example 2: This example, taken from [34], involves the design of a simple nonlinear mass-spring-damper mechanical system as depicted in Fig. 7. The main goal of the example here is to approximate the mass-spring-damper mechanical system (like a dynamically unknown one) by a TS fuzzy model over a fuzzy partition. The partition used is quite dense so as

to achieve a good approximation. Then, the proposed method is applied for reduction and to obtain a minimal form. The differential equations of the mechanical system are analytically given in the minimal form of a TS fuzzy model as well in order to evaluate the effectiveness of the reduction. The goal here is to show that the resulting minimal form is actually the same, in the complexity sense, as the one to train the dense rule base in the first place. The example as given in [34] utilizes a dynamics model varying in a 2-D parameter space. Here, the model is slightly modified to yield variation in only one-dimensional (1-D) parameter space. The model is a spring-mass-damper system with nonlinear/uncertain stiffness coefficient, damping coefficient and forcing function:

$$m \cdot \ddot{x} + g(x, \dot{x}) + k(x) = \phi(\dot{x}) \cdot u \quad (16)$$

where m is the mass, u is the force, $k(x)$ is the nonlinear or uncertain term with respect to the spring $g(x, \dot{x})$ is the nonlinear or uncertain term with respect to the damper, and $\phi(\dot{x})$ is the nonlinear term with respect to the input term. It is assumed that $g(x, \dot{x}) = d(c_1x + c_2\dot{x}^3)$, $k(x) = c_3x + c_4x^3$, $\phi(\dot{x}) = 1 + c_5\dot{x}^3$ and also that $x \in [-a, a]$, $\dot{x} \in [-b, b]$, and $a, b > 0$. In [34], the parameters are set as follows: $m = 1$, $d = 1$, $c_1 = 0.01$, $c_2 = 0.1$, $c_3 = 0.01$, $c_4 = 0.67$, $c_5 = 0$, $a = 1.5$, and $b = 1.5$. Here, we let $c_4 = 0$ in order to have a simpler model, varying in only 1-D parameter space. Equation (16) then becomes

$$\ddot{x} = -0.1\dot{x}^3 - 0.02x + u. \quad (17)$$

The nonlinear term is $-0.1\dot{x}^3$. Let us proceed further in the same way as in [34] and give a TS fuzzy model of (17) with minimal number, namely, two fuzzy rules. Given that \dot{x} satisfies the following conditions:

$$\begin{cases} -0.225\dot{x} \leq -0.1\dot{x}^3 \leq \dot{x} \cdot 0 & x \geq 0 \\ 0 \cdot \dot{x} \leq -0.1\dot{x}^3 \leq -0.225\dot{x} & x < 0 \end{cases}$$

the nonlinear term can thus be represented by the following upper and lower bounds: $-0.1\dot{x}^3 = f_1(\dot{x})\dot{x} \cdot 0 - (1 - f_1(\dot{x})) \cdot 0.225\dot{x}$, where $f_v(\dot{x}) \in [0, 1]$, $V = 2$. This leads to fuzzy sets $F_1^a : f_1^a(\dot{x}) = 1 - (\dot{x}^2/2.25)$ ("a" meaning that this function is obtained analytically) and $F_2^a : f_2^a(\dot{x}) = \dot{x}^2/2.25$. The antecedent functions are as depicted in Fig. 9. The following rules are thus obtained analytically: IF $\dot{x}(t)$ is F_1^a THEN $\ddot{x} = -0.02x + u$; IF $\dot{x}(t)$ is F_2^a THEN $\ddot{x} = -0.225\dot{x} - 0.02x + u$. With matrix representation, we obtain: IF $\dot{x}(t)$ is F_v^a THEN $\dot{\mathbf{x}}(t) = \underline{\mathbf{A}}_v \mathbf{x}(t) + \underline{\mathbf{B}}_v \mathbf{u}(t)$, $V = 2$, where $\underline{\mathbf{A}}_1 = \begin{bmatrix} 0 & -0.02 \\ 1 & 0 \end{bmatrix}$, $\underline{\mathbf{B}}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\underline{\mathbf{A}}_2 = \begin{bmatrix} -0.225 & -0.02 \\ 1 & 0 \end{bmatrix}$, $\underline{\mathbf{B}}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Consequently, the nonlinear model can be described by a TS fuzzy model, which has only two rules. The next step is to approximate the model (17) with a dense rule base, after which we can assume that (17) is unknown and then go about generating a minimum rule base by the present technique based only on the dense rules. The idea here is to show that the resulting minimal rule base would be similar to or the same as (17) in the first place.

We aim to use a high-density fuzzy partition to achieve a dense rule base of a good approximation. Let the interval $\dot{x} \in$

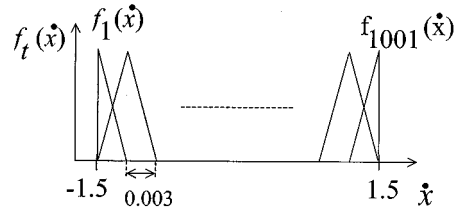


Fig. 8. Example 2: dense fuzzy partition to achieve a good approximation.

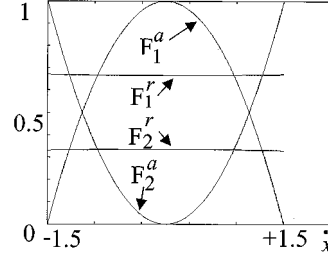


Fig. 9. Example 2: the analytical and the reduced antecedent sets. $x' = \dot{x}$.

$[-1.5, 1.5]$ be divided by 1001 triangular shaped fuzzy sets, as in Fig. 8. We sample the dynamic system at points $\dot{x} = -1.5 + (v-1)0.003$, which imitates the result of an identification algorithm, e.g., a gradient descent algorithm. The dense TS fuzzy model as obtained is then: IF $\dot{x}(t)$ is F_v THEN $\ddot{x} = a_v\dot{x} + b_vx + c_vu$, where $a_v = -0.1(-1.5 + (v-1)0.003)^2$, $b_v = -0.02$ and $c_v = 1$, which is in matrix form: IF $\dot{x}(t)$ is F_v THEN $\dot{\mathbf{x}}(t) = \underline{\mathbf{A}}_v \mathbf{x}(t) + \underline{\mathbf{B}}_v \mathbf{u}(t)$, $V = 1001$. Executing Method 2 on matrices $\underline{\mathbf{A}}_v$ in MATLAB (note that matrices $\underline{\mathbf{B}}_v$ are constant) yields three nonzero singular values of 31.734 509 156 680 5, 2.120 618 090 212 44 and 2.285 539 931 266 82e-016. The last one is numerically equivalent to zero. In this case, we can keep the two nonzero singular values, which means that two local linear models are sufficient for the same approximation of the original 1001. It turns out that in this case, the matrix $\underline{\mathbf{T}}$ as obtained has negative values, which implies that the resulting functions by (10) are not interpretable as fuzzy sets, though the result is acceptable in the sense of computational complexity reduction. Further application of the NN and SN transformation of [4] and [22] then yields an SN and NN matrix $\underline{\mathbf{T}}$ which guarantees that (10) yields fuzzy sets in the Ruspini-partition. The resulting SN and NN sets are depicted in Fig. 9. The reduced TS fuzzy model is: IF $\dot{x}(t)$ is F_v^r THEN $\dot{\mathbf{x}}(t) = \underline{\mathbf{A}}_v \mathbf{x}(t) + \underline{\mathbf{B}}_v \mathbf{u}(t)$, $V = 2$, where

$$\begin{aligned} \underline{\mathbf{A}}_1 &= \begin{bmatrix} -13.403 23 & -0.02 \\ 1 & 0 \end{bmatrix}, \underline{\mathbf{B}}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \text{and} \\ \underline{\mathbf{A}}_2 &= \begin{bmatrix} 26.580 01 & -0.02 \\ 1 & 0 \end{bmatrix}, \underline{\mathbf{B}}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \end{aligned} \quad (18)$$

Note that if the identification process includes noise then we may obtain some extra, relatively smaller (according to the noise) nonzero singular values. Keeping only two of them has some filtering effect on the noise. Since the original fuzzy sets are triangular, as depicted in Fig. 8, the columns of matrix $\underline{\mathbf{T}}$ actually contain the membership values [4], [22]. Using polynomial approximation to the 1001 values of both columns,

(19) is obtained, shown at the bottom of the page. Indeed, substituting (19) into (18), the original model (17) is obtained in a matrix form.

Consequently, the two fuzzy rules as obtained by Method 2 from the 1001 rules are in full accordance with the analytically derived TS fuzzy model. They constitute a transformed rule set similar to the one using F_1^a and F_2^a . Analytically, they are shown to be equivalent to the original model.

Example 3: The main idea in the application of [5], [12], and [36]–[38] is to have a set of local linear models at various operating points with respect to a certain variable, which can then be used as tool for either fault detection or isolation. The overall nonlinear model is approximated based on the local linear models in the form of a TS fuzzy model. The induction motor under investigation constitutes a very nonlinear system because of the coupling effect between rotor and stator phases where the coefficients of the matrices vary with the rotor position and the motor speed. For detailed development and analysis, see [5], [12], [37], and [38]. The performance of the proposed observer here is assessed using Matlab/Simulink. The FDI subsystem forms part of a fault-tolerant control system to regulate the torque and flux in the induction motor.

The sensor fault-tolerant control scheme is run in real time using measurements taken from a hardware testing, which consists of a three phases AC induction motor driving a dc motor as load. The three-phase voltages are supplied from a 100 KHz PWM-inverter. The induction motor is a 0.2-kW three-phase four-pole machine. The overall system consists of a pulsewidth modulation (PWM) inverter, which provides the necessary power to drive the machine. As the torque and flux cannot be measured they are estimated using a TS fuzzy observer. Faults due to gain and bias changes, noise and interference pick up, clipping or slew-rate limits, signal processing effects, and disconnections due to continuous and intermittent loss of measurement can occur in the system. A FDI/reconfiguration process based on the fuzzy observers performs the detection and isolation of faults and takes the necessary action to maintain the controller performance. The dynamics of the induction motor observer is described by TS fuzzy rules [see (1)]. Specifically, the induction motor can be modeled by the following bilinear differential equations:

$$\begin{aligned} \dot{\mathbf{x}} &= \underline{\mathbf{A}}(\omega)\mathbf{x} + \underline{\mathbf{B}}\mathbf{u} = \underline{\mathbf{A}}\mathbf{x} + \underline{\mathbf{N}}(\omega)\mathbf{x} + \underline{\mathbf{B}}\mathbf{u} \\ \mathbf{y} &= \underline{\mathbf{C}}\mathbf{x} \end{aligned}$$

where $\mathbf{x} = [I_{ds} I_{qs} I_{dr} I_{qr}]^T$, $\mathbf{y} = [I_{ds} I_{dr}]^T$, $\mathbf{u} = [V_{ds} V_{qs} V_{dr} V_{qr}]^T$, and V and I denote the current and voltage, respectively. Comprehensive analysis of the exampled motor and its differential equations are given in [5], [12], and [36]–[38]. The speed ω is referred to as the bilinear input. This model is derived based on three main assumptions, namely: ideal air gap flux distribution, linear magnetization character-

istic, and zero temperature coefficient in the windings. In [5], [12],[37], and [38], the model is comprised of five local linear models as $\hat{\mathbf{M}}(\omega) = \sum_{v=1}^5 f_v(\omega)\mathbf{M}_v$. For simplicity, let us focus only on matrix $\underline{\mathbf{A}}(\omega)$. Thus, $\mathbf{M}(\omega) \Rightarrow x(t) = \underline{\mathbf{A}}(\omega)x(t)$. Given five matrices, $\underline{\mathbf{A}}_v$ is shown as

$$\begin{aligned} \underline{\mathbf{A}}_1 &= \begin{bmatrix} -459.2987 & 0 & 233.3444 & 0 \\ 0 & -459.2987 & 0 & 233.3444 \\ 427.7981 & 0 & -250.5265 & 0 \\ 0 & 427.7981 & 0 & -250.5265 \end{bmatrix} \\ \underline{\mathbf{A}}_2 &= \begin{bmatrix} -459.2987 & -81.8650 & 233.3444 & -87.8931 \\ 81.8650 & -459.2987 & 87.8931 & 233.3444 \\ 427.7981 & 87.8931 & -250.5265 & 94.3650 \\ -87.8931 & 427.7981 & -94.3650 & -250.5265 \end{bmatrix} \\ \underline{\mathbf{A}}_3 &= \begin{bmatrix} -459.2987 & -130.9840 & 233.3444 & -140.6289 \\ 130.9840 & -459.2987 & 140.6289 & 233.3444 \\ 427.7981 & 140.6289 & -250.5265 & 150.9840 \\ -140.6289 & 427.7981 & -150.9840 & -250.5265 \end{bmatrix} \\ \underline{\mathbf{A}}_4 &= \begin{bmatrix} -459.2987 & -212.8490 & 233.3444 & -228.5220 \\ 212.8490 & -459.2987 & 228.5220 & 233.3444 \\ 427.7981 & 228.5220 & -250.5265 & 245.3490 \\ -228.5220 & 427.7981 & -245.3490 & -250.5265 \end{bmatrix} \\ \underline{\mathbf{A}}_5 &= \begin{bmatrix} -459.2987 & -343.8330 & 233.3444 & -369.1509 \\ 343.8330 & -459.2987 & 369.1509 & 233.3444 \\ 427.7981 & 369.1509 & -250.5265 & 396.3330 \\ -369.1509 & 427.7981 & -396.3330 & -250.5265 \end{bmatrix} \\ \underline{\mathbf{T}} &= \begin{bmatrix} 0.5967 & 0.4033 \\ 0.6302 & 0.3698 \\ 0.6502 & 0.3498 \\ 0.6837 & 0.3163 \\ 0.7372 & 0.2628. \end{bmatrix} \end{aligned}$$

These five matrices are of full rank. No reduction is thus obtained with Method 1. Method 2 utilized with NN and SN transformation results in the above matrix $\underline{\mathbf{T}}$ in transforming the antecedent sets. The reduced antecedent sets are interpretable as fuzzy sets defined in Ruspini-partition. This implies that the use of only two models is enough to the same approximation. The computational reduction theoretically is about 50% based on (13) with $C_p = 1$. The reduction is exact, which implies that the system performance is not degraded compared to the results in [5], [12], [37], and [38]. In order to check the real computational reduction, the output values of the original model were generated 10 000 times. At the same time, the reduced form was calculated in parallel and was executed 14 843 times. It is concluded that the computational time of the reduced model was 67.3701% of the original model (the difference between the theoretical and the measured reduction is that the real value of C_p is larger than one and hence is dominant because of the small number of local linear models.

$$\begin{aligned} F_1^r : f_1^r(\dot{x}) &= (0.002\,501\,047\,170\,33..)\dot{x}^2 + (0.664\,778\,651\,145\,95..); \\ F_2^r : f_2^r(\dot{x}) &= 1 - f_1^r(\dot{x}) \end{aligned} \tag{19}$$

IX. CONCLUSION

This paper proposes an SVD-based computational complexity reduction technique capable of finding the linear dependence among local linear models. SVD is utilized here to define the contribution of the local models in a decreasing order, which allows the removal of weak or noncontributing components according to a given error threshold. The error tolerance depends on how robust the system at hand is with respect to approximation error. The key idea here goes back to the original idea of applying an SVD reduction technique [22]–[24]: given that there is no formal and proper procedures in establishing a fuzzy system, how can one be sure that a given, rather complex system is already in a simple form? The SVD reduction approach basically allows that one does as good a job as one can in coming up with a fuzzy model utilizing dense fuzzy rule partition in order to achieve fine approximation. Then one applies the proposed methods and sees if any reduction is possible. The methods proposed here are not merely restricted to reduction of TS fuzzy models, but are applicable to any kind of model approximation techniques where the investigated weighting function based formula (5) is applied. Having said the above, an important point should be noted regarding the computational reduction here. In some cases, more of the reduced rules will be fired simultaneously for a given input than in the case of the original rule base. In such cases, the advantage of smaller number of rules may be offset by an increase in the computation load. Even so, computational issues aside, it is still interesting from the perspective of knowledge acquisition to see how a given rule set can be replaced or closely replaced by a set of fewer rules. Our future plan is to formulate a detailed extension of the present methods to multidimensional parameter space and analyze the adaptive property of the reduced models.

REFERENCES

- [1] P. Baranyi and Y. Yam, "Singular value-based approximation with non-singleton fuzzy rule base," in *Proc. 7th Int. Fuzzy Systems Association World Congr. (IFS'97)*, Prague, 1997, pp. 127–132.
- [2] J. C. Baron and C. Geffroy, *Embedded System Applications*, G. Motet, Ed. Norwell, MA: Kluwer, 1997.
- [3] G. Biswas, R. Kapadia, and X. W. Yum, "Combined qualitative quantitative steady state diagnosis of continuous-valued systems," *IEEE Trans. Syst., Man Cybernet.*, vol. 27, pp. 167–185, Mar. 1997.
- [4] P. Baranyi and Y. Yam, "Fuzzy rule base reduction," in *Fuzzy IF-THEN Rules in Computational Intelligence: Theory and Applications*, D. Ruan and E. Kerre, Eds. Norwell, MA: Kluwer, 2000, ch. Chapter 7, pp. 135–160.
- [5] J. Chen and R. J. Patton, *Robust Model Based Fault Diagnosis for Dynamic Systems*. Norwell, MA: Kluwer, 1999.
- [6] J. J. Dongara, C. B. Moler, J. R. Bunch, and G. W. Stewart, *Linpack Users' Guide*. Philadelphia, PA: SIAM, 1979.
- [7] D. Dubois and H. Prade, "What are fuzzy rules and how to use them," *Fuzzy Sets Syst.*, vol. 84, pp. 169–185, 1996.
- [8] P. M. Frank and B. Kuipel, "Fuzzy supervision and application to lean production," *Int. J. Syst. Sci.*, vol. 24, no. 10, pp. 1935–1944, 1993.
- [9] B. S. Garbow, J. M. Boyle, J. J. Dongara, and C. B. Moler, *Matrix Eigensystem Routines-Eispack Guide Extension*. New York: Springer-Verlag, 1993.
- [10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: The John Hopkins Univ. Press, 1989.
- [11] R. J. Patton and J. Chen, "Observer-based fault detection and isolation: Robustness and applications," *Contr. Eng. Practice*, vol. 5, no. 5, pp. 671–682, 1997.
- [12] R. J. Patton, J. Chen, and C. J. Lopez-Toribio, "Fuzzy observers for nonlinear dynamic systems fault diagnosis," in *Proc. 37th IEEE Conf. Decision and Control*, vol. 1, 1998, pp. 84–89.
- [13] R. J. Patton, R. Clark, Frank, and M. Paul, *Issues of Fault Diagnosis for Dynamic Systems*, R. J. Patton, R. Clark, and P. M. Frank, Eds. New York: Springer-Verlag, 2000.
- [14] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Applicat.*, vol. 21, no. 4, pp. 1253–1278, Apr. 2000.
- [15] D. A. Linkens and M. F. Abbod, "Supervisory intelligent control using fuzzy logic hierarchy," *Trans. Inst. Meas. Control*, vol. 15, no. 3, pp. 112–132, 1993.
- [16] GY. Román and A. R. Várkonyi-Kóczy, "On the consequences of the complexity of tasks in measurement," *Engineering*, vol. 4, no. 1, pp. 53–62, Mar. 1998.
- [17] I. J. Rudas and O. Kaynak, *New Types of Generalized Operations. Computational Intelligence: Soft Computing and Fuzzy-Neuro Integration with Applications*. ser. Series F, O. Kaynak, L. A. Zadeh, B. Türkmen, and I.J Rudas, Eds: Springer NATO ASI Series, 1998, vol. 192, Computer and Systems, Sciences, pp. 128–156.
- [18] I. J. Rudas and M. O. Kaynak, "Entropy-based operations on fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 33–40, Feb. 1998.
- [19] —, *Min. Max. Fuzziness Generalized Operators Fuzzy Sets Syst.*, 1998, vol. 98, pp. 83–94.
- [20] I. J. Rudas, "Toward the generalization of t-operators: A distance-based approach," *J. Inform. Org. Sci.*, vol. 23, no. 2, pp. 149–166, 1999.
- [21] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. C. Tenekedzis, "Failure diagnosis using discrete-event models," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 1555–1575, Sept. 1995.
- [22] Y. Yam, P. Baranyi, and Ch.-T. Yang, "Reduction of fuzzy rule base via singular value decomposition," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 120–132, Apr. 1999.
- [23] Y. Yam, "Fuzzy approximation via grid point sampling and singular value decomposition," *IEEE Trans. Syst., Man, Cybernet.*, vol. 27, pp. 933–951, 1997.
- [24] J. Yen and L. Wang, "Simplifying fuzzy rule-based models using orthogonal transformation methods," *IEEE Trans. Syst., Man, Cybernet.*, pt. B, vol. 29, pp. 13–24, Feb. 1999.
- [25] A. R. Várkonyi-Kóczy and T. Kovácsáhy, "Anytime algorithms in embedded signal processing systems," in *Proc. IX Eur. Signal Processing Conf. EUSIPCO-98*, vol. 1, Rhodes, Greece, Sept. 8–11, 1998, pp. 169–172.
- [26] O. H. Wang, K. Tanaka, and M. F. P. Griffin, "Parallel distributed compensation of nonlinear systems by Takagi and Sugeno fuzzy models," in *Proc. FUZZ-IEEE/FES'95*, 1995, pp. 531–538.
- [27] T. Takagi and N. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybernet.*, vol. SMC-15, no. 1, pp. 116–132, 1985.
- [28] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets Syst.*, vol. 45, no. 2, pp. 135–156, 1992.
- [29] —, "Stability analysis of fuzzy systems using Lyapunov's direct method," in *Proc. NAFIPS'90*, 1990, pp. 133–136.
- [30] O. H. Wang, K. Tanaka, and M. F. P. Griffin, "An approach to fuzzy control of nonlinear systems: Stability and design issues," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 1, pp. 14–23, 1996.
- [31] K. Tanaka, T. Ikeda, and O. H. Wang, "Fuzzy regulators and fuzzy observers: Relaxed stability conditions and LMI-based design," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 2, pp. 250–265, 1998.
- [32] R. J. Patton, J. Chen, and C. J. Lopez-Toribio, "Fuzzy observers for nonlinear dynamic systems fault diagnosis," *Proc. 37th IEEE Conf. Decision and Control*, vol. 1, pp. 84–89, 1998.
- [33] O. H. Wang, K. Tanaka, and M. F. P. Griffin, "Parallel distributed compensation of a nonlinear systems by Takagi and Sugeno fuzzy models," in *Proc. FUZZ-IEEE/IFES'95*, 1995, pp. 531–538.
- [34] K. Tanaka, T. Ikeda, and O. H. Wang, "Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: Quadratic stabilisability, H_∞ control theory and linear matrix inequalities," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 1, pp. 1–13, 1996.
- [35] A. R. Várkonyi-Kóczy, A. Ruano, P. Baranyi, and O. Takács, "Anytime information processing based on fuzzy and neural network models," in *Proc. 2001 IEEE Instrumentation and Measurement Technology Conf.*, vol. 2, Budapest, Hungary, May 21–23, 2001, pp. 1247–1252.
- [36] R. J. Patton, C. J. Lopez-Toribio, and F. J. Uppal, "Artificial intelligence approaches to fault diagnosis," *Int. J. Appl. Math. Comput. Sci.*, vol. 9, no. 3, 1999.
- [37] C. J. Lopez-Toribio, R. J. Patton, and S. Daley, "Takagi-Sugeno fuzzy fault tolerant control of an induction motor," *NeuroComputing Applicat. J. Special Issue on N-F Systems*, 1999.

- [38] J. Chen, R. J. Patton, and C. J. Lopez-Toribio, "Fault diagnosis and fault-tolerant estimation of a rail traction system via fuzzy observers," *Trans. Inst. Meas. Control*, vol. 21, no. 1, pp. 14–20, 1999.
- [39] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "An introduction to independent component analysis," *J. Chemometrics*, vol. 14, pp. 123–149, 2000.



Péter Baranyi was born in Hungary in 1970. He received the M.Sc. degree in electrical engineering, the M.Sc. degree in education of engineering sciences, and the Ph.D. degree from the Technical University of Budapest, Budapest, Hungary, in 1994, 1995, and 1999, respectively.

He held research positions at The Chinese University of Hong Kong (1996 and 1998), The University of New South Wales (1997), Australia, CNRS LAAS Institute, Toulouse, France (1996), and Gifu Research Institute, Japan (2000–2001). His research interests

include fuzzy and neural network techniques.

Dr. Baranyi received the Youth Prize of the Hungarian Academy of Sciences and the International Dennis Gábor Award in 2000. He is the Secretary General of the Hungarian Society of the International Fuzzy Systems Association. He was a founding member of the Integrated Intelligent Systems Japanese-Hungarian Laboratory.



Yeung Yam received the B.S. degree from The Chinese University of Hong Kong, Hong Kong, in 1975, the M.S. degree in physics from the University of Akron, OH, and the M.S. and Sc.D. degrees in aeronautics and astronautics from Massachusetts Institute of Technology, Cambridge, in 1979 and 1983, respectively.

He is currently an Associate Professor in the Department of Automation and Computer-Aided Engineering, The Chinese University of Hong Kong. Before joining the university in 1992, he was with the

Control Analysis Research Group of the Guidance and Control Section at the Jet Propulsion Laboratory, Pasadena, CA. His research interests include dynamics modeling and control, system identification, fuzzy design, analysis and approximation.



Annamária R. Várkonyi-Kóczy (M'95–SM'97) was born in Budapest, Hungary, in 1957. She received the M.Sc.E.E., M.Sc.M.E.-T., and Ph.D. degrees from the Technical University of Budapest (now the Budapest University of Technology and Economics), Budapest, Hungary, in 1981, 1983, and 1996, respectively.

She was a Researcher for the Research Institute for Telecommunication in Budapest for six years, followed by four years with the Hungarian Academy of Science, where she was a Research Associate in

the group of Engineering Mechanics, transferred to the Technical University of Budapest. Since 1991, she has been with the Department of Measurement and Information Systems of the same university, currently as an Associate Professor. Her research interests include digital signal processing, uncertainty handling, soft computing, anytime and hybrid techniques in complex measurement, diagnostics, and control systems.

Dr. Várkonyi-Kóczy is the Vice Chair of the Hungarian Fuzzy Association (member of the IFSA) and a member of the European Association for Signal Processing (EURASIP), Hungarian Academy of Engineers, John von Neumann Computer Society (Hungary), and Measurement and Automation Society (Hungary).



Ron J. Patton was born in Peru in 1949. He received three degrees in electrical and electronic engineering and control systems from Sheffield University, Sheffield, U.K.

He has worked in the hospital service in medical physics and was a founding member of the Electronics Laboratory at the Royal Free Hospital, London, U.K., in 1967–1968. During 1973–1974, he worked at the BBC Research Department, Kingswood Warren, Surrey. After completing his Ph.D. studies in 1976, he worked for GEC

Electrical Projects, Rugby, U.K., and Sheffield City Polytechnic on dynamic ship positioning control systems. He became a Lecturer at Sheffield Hallam University in 1978 and moved to the new Electronics Department at York in 1981 where he focused on fault diagnosis and aerospace control systems, with promotion to Senior Lecturer in 1987. In 1995, he was appointed Professor of Control and Intelligent Systems Engineering at the University of Hull, Hull, U.K., where his research interests are fault-tolerant control and the use of AI/soft computing methods for fault diagnosis in control systems. He is the chair of the IFAC Technical Committee SAFEPROCESS. He has published more than 250 papers and five books on eigenstructure assignment for control systems design, fault diagnosis, and fault-tolerant control.



Pál Michelberger was born in Hungary in 1930. He received the M.Sc., Ph.D., and D.Sc. degrees from the Technical University of Budapest, Budapest, Hungary, in 1952, 1960, and 1970, respectively.

He has held several positions at the Technical University of Budapest: Professor (1968–present), Vice Dean (1968–71), Dean (1985–present), and Rector (1990–1994) of the Faculty of Transportation Engineering. His research interests include vehicle design, structure analysis, vehicle dynamics, and applied mechanics. He is or has served as

Editor-in-Chief of *Acta Technica Hung* (1986–2000), *Periodica Polytechnica* (1985–1991), and *Vehicles, Agricult. Machines* (1983–1999). He is a member of the Editorial Boards of *Vehicle System Dynamics* (1984–present), *International Journal of Vehicle Design* (1978–present), *Periodica Polytechnica* (1991–1995), and *Technika* (1996–present).

Dr. Michelberger has served as vice president (1978–1992) and president (1992–1994) of FISITA. He is a member of the Directory Board of IAVSD (1977–1981 and 1983–present), Academy Europae (1993–present), the International Academy of Higher Education (Moscow) (1994–present), IGIP (1993–present), Sigma Xi, the Russian Academy of Transport (Sankt-Petersburg) (1992–present), and the European Academy of Sciences and Arts (Salzburg) (1998–present). He is an honorary member of VDI (Verein Deutscher Ingenieure, Düsseldorf) and an Honorary Scholar. Since 1990, he has been an ordinary member of the Hungarian Academy of Sciences.



Masaharu Sugiyama received the B.E. and M.E. degrees in electrical engineering from University of Gifu, Gifu, Japan, in 1970 and 1973, respectively.

He has been a Director of the Information System Division, Gifu Research Institute of Manufacturing Information Technology, Gifu, Japan, since 1999. His research interests include watermarking, virtual reality, and related topics.

Mr. Sugiyama is a member of the Institute of Electronics, Information and Communication Engineers.