

TRADEOFF BETWEEN APPROXIMATION ACCURACY AND COMPLEXITY: HOSVD BASED COMPLEXITY REDUCTION

Péter BARANYI^{*1}, Sándor MIZIK*, Péter VÁRLAKI** and Pál MICHELBERGER*

*Research Group for Mechanics of the Hungarian Academy of Sciences
Budapest University of Technology and Economics
H-1521 Budapest, Hungary
Tel: 36 1 463 1758
Fax: 36 1 463 1763
e-mail: baranyi@tnt-202.tnt.bme.hu

** Department Automobile Engineering
Budapest University of Technology and Economics
H-1521 Budapest, Hungary

Received: October 1, 2001

Abstract

Higher Order Singular Value Decomposition (HOSVD) based complexity reduction method is proposed in this paper to polytopic model approximation techniques. The main motivation is that the polytopic model has exponentially growing computational complexity with the improvement of its approximation property through, as usually practiced, increasing the density of local linear models. The reduction technique proposed here is capable of defining the contribution of each local linear model, which serves to remove the weakly contributing ones according to a given threshold. Reducing the number of local models leads directly to the complexity reduction. The proposed reduction can also be performed on TS fuzzy model approximation method. A detailed illustrative example of a non-linear dynamic model is also discussed. The main contribution of this paper is the multi-dimensional extension of the SVD reduction technique introduced in the preliminary work [1]. The advantage of this extension is that the HOSVD based technique of this paper can be applied to polytopic models varying in a multi-dimensional parameter space unlike the reduction method of [1] which is designed for one dimensional parameter space.

Keywords: polytopic model, TS fuzzy model, complexity reduction, singular value decomposition (SVD - HOSVD).

1. Introduction

As a result of the dramatic and continuing growth in computer power, and the advent of very powerful algorithms (and associated theory) for *convex optimisation*, we can now solve very rapidly many *convex optimisation* problems for which no traditional 'analytic' or 'closed form' solutions are known (or likely to exist). Indeed, the solution to many *convex optimisation* problems can now be computed in

¹and Integrated Intelligent Systems Japanese-Hungarian Laboratory

a time which is comparable to the time required to evaluate a ‘closed-form’ solution for a similar problem. This fact has far-reaching implications for engineers: it changes our fundamental notion of what we should consider as a solution to a problem. In the past, a ‘solution to a problem’ generally meant a ‘closed-form’ or ‘analytic’ solution. There are ‘analytical solutions’ to a few very special cases in the wide variety of problems in systems and control theory, but in general non-linear problems cannot be solved. A control engineering problem that reduces to solving two algebraic Riccati equations is now generally regarded as ‘solved’. A control engineering problem that reduces to solving even a large number of convex algebraic Riccati inequalities (a problem which has no analytic solution) should also be regarded as “solved”, even though there is no analytic solution. A number of problems that arise in Systems and Control such as optimal matrix scaling, digital filter realization, interpolation problems that arise in system identification, robustness analysis and statefeedback synthesis via Lyapunov functions, can be reduced to a handful of standard *convex* and *quasiconvex problems* that involve *matrix inequalities*. Extremely efficient interior point algorithms have recently been developed for and tested on these standard problems; further developments of algorithms for these standard problems are in an area of active research.

The notion of convex combination of a finite set of points gets considerable relevance in the context of dynamic systems if ‘points’ become systems. Various model approximation techniques are based on the convex combination of local models. The combination is usually defined by basis functions which express the local dominance of the linear local models. One of these approximation methods is called Polytopic Model Approximation (PMA) and utilized to describe linear parametrically varying or, hence, linear time variant systems where the parameters vary in time. Regarding the explicit form of PMA the *Takagi-Sugeno* Model Approximation (TSMA), which emerged in the field of softcomputing, was introduced in the same fundamental form. The use of the PMA and TSMA techniques is motivated by the fact that they can easily be cast or recast as *convex optimisation* problems that involve *Linear Matrix Inequalities* (LMIs) and the controller design and stability analysis can, hence, be done in this framework [19, 20]. Therefore, once the model approximation is given in the form of PMA or TSMA then the controller design and the Lyapunov stability analysis of a non-linear system reduces to solve an LMI problem as outlined above.

Despite the advantage discussed above the use of PMA and TSMA is practically limited. The reason for this is that although the PMA and TSMA theoretically have the universal approximation property achieving a good system approximation, PMA and TSMA suffer from exponential complexity in respect to approximation accuracy. Let a brief digression be taken here, in order to see the contradiction between approximation accuracy and complexity. In 1900, D. Hilbert listed 23 conjectures, hypotheses concerning unsolved problems which he considered would be the most important ones to solve by the mathematicians of the 20th century. According to the 13th conjecture there exist such continuous multi-variable functions, which cannot be decomposed as the finite superposition of continuous functions of less variables. In 1957 ARNOLD disproved this hypothesis [27], moreover, in

the same year, KOLMOGOROV [9] proved a general representation theorem with a constructive proof, where the functions in the decomposition were one dimensional. KOLMOGOROV's representation theorem was further improved by several authors (SPRECHER [17] and LORENTZ [13]). In 1980, DE FIGUEIREDO showed that KOLMOGOROV's theorem could be generalized for multi-layer feedforward neural networks, and these could, hence, be considered as universal approximators. From the late '80s several authors proved that different types of neural network possessed the universal approximation property. Similar results have been established from the early '90s in fuzzy theory. These results [5, 10, 26] claim that different fuzzy reasoning methods are capable of approximating an arbitrary continuous function on a compact domain with any specified accuracy. As a result, softcomputing techniques were considered as universal approximators in general. Regarding the explicit form the PMA and STMA techniques share the same advantage.

In spite of these remarkable advantages the neural network model as well as fuzzy approximation, further PMA and TSMA have exponential complexity in terms of the number of variables shown by KÓCZY and HIROTA (1997) [8]. It means that in the neural network context, the number of units, or in the context of PMA and TSMA the number of local linear models (we say building units) grows exponentially as the approximation error tends to zero. This exponentiality cannot be eliminated, so the universal approximation property of these uncertainty based approaches cannot be exploited straightforwardly for practical purposes. *Fig. 1* shows the relation between the number of building units and approximation accuracy.

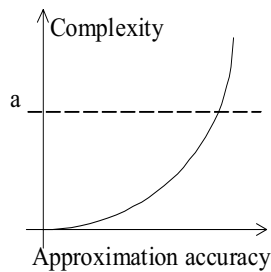


Fig. 1. Tradeoff between complexity and approximation accuracy. 'a' indicates the available computation capacity in a real application.

Moreover, for some special approximation techniques (PMA and TSMA) it is shown, that if the number of the building units is bounded, the resulting set of functions is nowhere dense in the space of approximated functions (TIKK, 1999) [21]. According to the opinion of some researchers [21], analogous results should hold for most fuzzy and neural systems. The mutually contradicting results naturally raise the question to what extent the model approximation should be accurate concerning the available computational cost. From the practical point of view it is

enough to achieve an ‘acceptably’ good approximation, where the given problem determines the factor of acceptability in terms of ε . Hence the task is to find a possible tradeoff between the specified accuracy and the number of building units.

Recently, several approaches have applied orthogonal transformation methods to find the minimal number of building units in a given approximation. For instance, in 1999 YEN and WANG [24] investigated various techniques such as orthogonal least-squares, eigenvalue decomposition, SVD-QR with column pivoting method, total least square method and direct SVD method. SVD based fuzzy approximation technique was initialised by YAM in 1997 [22], which directly finds the minimal number of building units from sampled values. Shortly after, this technique was introduced as SVD reduction of the building units and structure decomposition [1, 2, 3, 4, 23]. An extension of YEN and WANG’s work [24] to multi-dimensional cases may also be conducted in a similar fashion as the higher order SVD reduction technique proposed in the papers [1, 2, 3, 4, 22, 23]. SVD is not merely used as a way of reduction of fuzzy rule bases. A brief enumeration of some opportunities offered by SVD, the development of which was started by BELTARMI about 200 years ago as discussed by STEWART (1993) [14] and which became one of the most fruitful tools in linear algebra, gives ideas about its promising role in complexity reduction in general. The key idea of using SVD in complexity reduction is that the singular values can be applied to decompose a given system and indicate the degree of the significance of the decomposed parts. Reduction is conceptually obtained by the truncation of those parts, which have weak or no contribution at all to the output, according to the assigned singular values. This advantageous feature of SVD is used in this paper to extract a given model approximation and discard those local linear models, namely, building units, which have no significant role in the overall system according to a given approximation accuracy. However, reducing the number of building units does not imply the computational cost reduction in all cases since the computation also depends on the number of overlapping basis functions, see later. Therefore, as a subsequent aim, a detailed investigation is given in the aspect of the computational time reduction in this paper.

The concept of this paper is based on the above outlined ideas [1, 2, 3, 4, 22, 23]. Presumably, the SVD technique in this paper as well as in the papers [1, 2, 3, 4, 22, 23] can be replaced by other orthogonal techniques investigated by YEN and WANG [24]. The present work constitutes a detailed investigation of the preliminary approaches outlined in the work [1] and gives a possible solution to the complexity problem analysed above. The algorithms proposed here are mostly developed in the papers [22, 23], but are restructured in terms of tensor description in order to facilitate further developments. Concepts of HOSVD are investigated in tensor forms in the works of LATHAUWER *et al.* (2000 and 2001) [11, 12], COMON (1994) [6] and SWAMI *et al.* (1996) [18].

Before starting with the discussion of the proposed method let a brief digression be taken here to outline the motivation of the above characterized reduction technique in vehicle engineering and transportation design. Static and dynamic stresses in commercial vehicles were outlined in the work of MICHELBERGER *et al.* (1976) [15]. Shortly after the load analysis of commercial vehicles was intro-

duced by HORVÁTH et al. in 1981 [7]. The fundamental equation of motion of a linear system of discrete mass points and rigid bodies is given in the paper of HORVÁTH et al. that is described by the following differential equation derived in the paper by MICHELBERGER et al. (1976) [16]:

$$\mathbf{M}\ddot{\mathbf{y}} + \mathbf{K}\dot{\mathbf{y}} + \mathbf{S}\mathbf{y} = \mathbf{G}\mathbf{f}_v^h(t) + \mathbf{D}\dot{\mathbf{f}}_v^h(t), \quad (1)$$

where matrix \mathbf{M} is a mass matrix comprising point-like masses and adequately transformed principal moments of inertia; matrix \mathbf{K} indicates the damping acting on mass points (rigid bodies); \mathbf{S} is a stiffness matrix; \mathbf{D} is damping matrix applied on the road surface as constraint co-ordinates due to vehicle components (tyres); \mathbf{G} stiffness matrix applied like \mathbf{D} ; \mathbf{y} is vertical displacement of discrete mass points (rotation of rigid bodies around the centroid); $\mathbf{f}_v^h(t)$ is the function of excitation by speed v of a road type h . A sequence of equation (1) is defined for various cases as:

$$\mathbf{M}_i\ddot{\mathbf{y}} + \mathbf{K}_i\dot{\mathbf{y}} + \mathbf{S}_i\mathbf{y} = \mathbf{G}_j\mathbf{f}_v^h(t) + \mathbf{D}_j\dot{\mathbf{f}}_v^h(t), \quad i + 1 \dots n, \quad j = 1 \dots m.$$

The i -th model on the left side is assigned to the j -th model on the right side depending on the investigated case: $j = \text{assign}(i)$. Thus, the number of models is $m \times n$. Furthermore, these assigned models are combined based on the calculation of weighted-average to form special models for various kinds of investigation:

$$\begin{aligned} \mathbf{M}_o\ddot{\mathbf{y}} + \mathbf{K}_o\dot{\mathbf{y}} + \mathbf{S}_o\mathbf{y} &= \mathbf{G}_o\mathbf{f}_v^h(t) + \mathbf{D}_o\dot{\mathbf{f}}_v^h(t) \\ &\Rightarrow \sum_{i=1}^n w_{m,i}(p)\mathbf{M}_i\ddot{\mathbf{y}} + w_{k,i}(p)\mathbf{K}_i\dot{\mathbf{y}} + w_{s,i}(p)\mathbf{S}_i\mathbf{y} \\ &= \sum_{\substack{i=1 \dots n, \\ j=\text{assign}(i)}} w_{g,j}(p)\mathbf{G}_j\mathbf{f}_v^h(t) + w_{d,j}(p)\mathbf{D}_j\dot{\mathbf{f}}_v^h(t). \end{aligned} \quad (2)$$

The target what for (2) is determined is indicated by p that is for instance $p = 1$ if (2) is used for investigating the stability, or $p = 2$ if energy requirement is estimated by (2), a different (2) is defined when the commercial comfort is assayed; etc. In order to achieve good approximation the model points are usually increased (m and n) which leads to extremely complex calculation of (2). The main goal is the reduction of the complexity, namely, to find the minimal m and n in respect to p in order to simplify the model. Another aspect is that once the reduction is done, which sets free further calculation capacity, we have a chance to put new model points, in order to improve the approximation.

In conclusion the main contribution of this paper is the multi-dimensional extension of the SVD reduction technique of [1]. Work [1] can be applied to reduce the complexity of those polytopic or TS fuzzy models which vary in a one-dimensional parameter space. This paper replaces the SVD with HOSVD in [1], which leads to a reduction technique capable of operating on multi-dimensional polytopic or TS fuzzy models.

2. Basic Definitions

This section is devoted to introduce some elementary definitions and concepts utilized in the further developments. Before starting with the definitions, some comments are enumerated on the notation to be utilized. To facilitate the distinction between the types of given quantities, they will be reflected by their representation: scalar values are denoted by lower-case letters $\{a, b, \dots\}$; column vectors and matrices are given by bold-face letters as $\{\mathbf{a}, \mathbf{b}, \dots\}$ and $\{\mathbf{A}, \mathbf{B}, \dots\}$ respectively. Tensors correspond to capital letters as $\{A, B, \dots\}$, tensor 1 contains values 1 only. The transpose of matrix \mathbf{A} is denoted as \mathbf{A}^T . Subscript is consistently used for a lower order of a given structure. E.g. an element of matrix \mathbf{A} is defined by row-column number i, j symbolized as $(\mathbf{A})_{i,j} = a_{i,j}$. Systematically, 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, \dots are in the meaning of indices (counters), I, J, \dots are reserved to denote the index upper bounds, unless stated otherwise. $\mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ is the vector space of real valued $(I_1 \times I_2 \times \dots \times I_N)$ -tensors. Letter N serves to denote the number of variables of the space where the coefficient matrices of the model are approximated.

DEFINITION 1 (*n*-mode matrix of tensor *A*) Assume an N -th order tensor $A \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$. The n -mode matrix $A_{(n)} \in \mathfrak{R}^{I_n \times J}$, $J = \prod_k I_k$ contains all the vectors in the n -th dimension of tensor A . The ordering of the vectors is arbitrary in $A_{(n)}$, this ordering shall, however, be consistently used later on. $(\mathbf{A}_{(n)})_j$ is called a j -th n -mode vector.

Note that any matrix the columns of which are given by n -mode vectors $(\mathbf{A}_{(n)})_j$ can evidently be restored to be tensor A . The restoring can be executed even in case when some rows of $\mathbf{A}_{(n)}$ are discarded since the value of I_n has no role in the ordering of $(\mathbf{A}_{(n)})_j$ [11, 12].

DEFINITION 2 (*n*-mode sub-tensor of tensor *A*) Assume an N -th order tensor $A \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$. The n -mode sub-tensor $A_{i_n=\alpha}$ contains elements $a_{i_1, i_2, \dots, i_{n-1}, \alpha, i_{n+1}, \dots, i_N}$.

DEFINITION 3 (*n*-mode tensor partition) Assume an N -th order tensor $A \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$. n -mode partitions of tensor 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 $A = [B_1 \ B_2 \ B_L]_n$, where $I_n = \sum_l J_l$.

DEFINITION 4 (Scalar product) The scalar product $\langle A, B \rangle$ of two tensors $A, B \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ is defined as $\langle A, B \rangle \stackrel{\text{def}}{=} \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} b_{i_1 i_2 \dots i_N} a_{i_1 i_2 \dots i_N}$.

DEFINITION 5 (Orthogonality) Tensors the scalar product of which equals 0 are mutually orthogonal.

DEFINITION 6 (Frobenius norm) The Frobenius norm of a tensor A is given by $\|A\| \stackrel{\text{def}}{=} \sqrt{\langle A, A \rangle}$.

DEFINITION 7 (*n*-mode matrix-tensor product) The *n*-mode product of tensor $A \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ by a matrix $\mathbf{U} \in \mathfrak{R}^{J \times I_n}$, denoted by $A \times_n \mathbf{U}$ is an $(I_1 \times I_2 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N)$ -tensor the entries of which are given by $A \times_n \mathbf{U} = \mathbf{B}$, where $B_{(n)} = \mathbf{U} \cdot A_{(n)}$. Let $A \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \dots \times_N \mathbf{U}_N$ be noted for brevity as $A \otimes_{n=1} \mathbf{U}_n$.

THEOREM 1 (Matrix singular value decomposition (SVD)) Every real valued $(I_1 \times I_2)$ -matrix \mathbf{F} can be written as the product of $\mathbf{F} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T = \mathbf{S} \times_1 \mathbf{U} \times_2 \mathbf{V}$, in which

1. $\mathbf{U} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_{I_1}]$ is a unitary $(I_1 \times I_1)$ -matrix,
2. $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_{I_2}]$ is a unitary $(I_2 \times I_2)$ -matrix,
3. \mathbf{S} is an $(I_1 \times I_2)$ -matrix with the properties of
 - (i) pseudodiagonality:

$$\mathbf{S} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{\min(I_1, I_2)})$$

- (ii) ordering: $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(I_1, I_2)} \geq 0$.

The σ_i are singular values of \mathbf{F} and the vectors \mathbf{U}_i and \mathbf{V}_i are respectively an *i*-th left and an *i*-th right singular vector.

There are major differences between matrices and higher-order tensors when rank properties are concerned. These differences directly affect the way an SVD generalization could look like. As a matter of fact, there is no unique way to generalize the rank concept. In this paper we restrict the description to *n*-mode rank only.

DEFINITION 8 (*n*-mode rank of tensor) The *n*-mode rank of A , denoted by $R_n = \text{rank}_n(A)$, is the dimension of the vector space spanned by the *n*-mode vectors as $\text{rank}_n(A) = \text{rank}(A_{(n)})$.

THEOREM 2 (Higher Order SVD (HOSVD)) Every tensor $A \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ can be written as the product $A = S \otimes_{n=1} \mathbf{U}_n$, in which

1. $\mathbf{U}_n = [\mathbf{u}_{1,n} \ \mathbf{u}_{2,n} \ \dots \ \mathbf{u}_{I_n,n}]$ is a unitary $(I_n \times I_n)$ -matrix called *n*-mode singular matrix.
2. tensor $S \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$ whose subtensors $S_{i_n=\alpha}$ have the properties of
 - (i) all-orthogonality: two subtensors $S_{i_n=\alpha}$ and $S_{i_n=\beta}$ are orthogonal for all possible values of *n*, α and β : $\langle S_{i_n=\alpha}, S_{i_n=\beta} \rangle = 0$ when $\alpha \neq \beta$,
 - (ii) ordering: $\|S_{i_n=1}\| \geq \|S_{i_n=2}\| \geq \dots \geq \|S_{i_n=I_n}\| \geq 0$ for all possible values of *n*.

The Frobenius norm $\|S_{i_n=i}\|$, symbolized by $\sigma_i^{(n)}$, are *n*-mode singular values of A and the vector $\mathbf{u}_{i,n}$ is an *i*-th singular vector. S is called as core tensor.

More detailed discussion of matrix SVD and HOSVD is given in [11, 12]. A detailed algorithm of HOSVD is given in papers [1, 2, 3, 4, 22, 23]. Graphical illustrations of the above definitions are given in [11, 23].

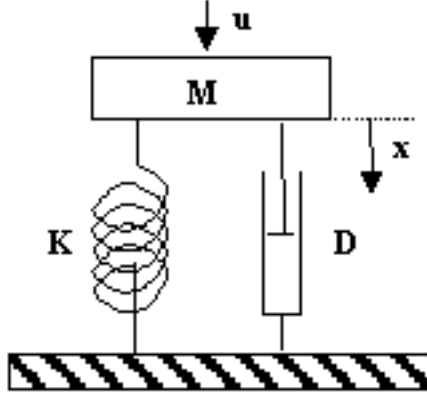


Fig. 2. Mass-spring-damper system

3. Polytopic and TS Model Approximation

This section is intended to discuss the fundamental form of PMA and TSMA. Consider a parametrically varying dynamical system

$$\begin{aligned}\frac{d\mathbf{x}}{dt}(t) &= \mathbf{A}(p)\mathbf{x}(t) + \mathbf{B}(p)\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}(p)\mathbf{x}(t) + \mathbf{D}(p)\mathbf{u}(t)\end{aligned}$$

with input $\mathbf{u}(t)$, output $\mathbf{y}(t)$ and state $\mathbf{x}(t)$. Suppose that its system matrix

$$\mathbf{S}(p) = \begin{pmatrix} \mathbf{A}(p) & \mathbf{B}(p) \\ \mathbf{C}(p) & \mathbf{D}(p) \end{pmatrix} \quad (3)$$

is a parametrically varying object which for any parameter p can be written as a *convex combination* of the V system matrices $\mathbf{S}_1, \dots, \mathbf{S}_V$ (as a matter of fact, p may depend on the state vector as well, $\mathbf{S}(p)$ can hence be viewed as a time varying object if p is considered as $p(t)$). This means that there exists a function $\alpha_v : \mathfrak{R} \rightarrow [0, 1]$ such that for any p we have that

$$\mathbf{S}(p) = \sum_{v=1}^V \alpha_v(p) \mathbf{S}_v,$$

where $\sum_v^V \alpha_v(p) = 1$ and

$$\mathbf{S}_v = \begin{pmatrix} \mathbf{A}_v & \mathbf{B}_v \\ \mathbf{C}_v & \mathbf{D}_v \end{pmatrix}$$

are constant system matrices. In particular, this implies that the system matrices $\mathbf{S}(p)$ belong to convex hull of $\mathbf{S}_1, \dots, \mathbf{S}_V$, i.e. $\mathbf{S}(p) \in co(\mathbf{S}_1, \dots, \mathbf{S}_V)$. Such models

are called polytopic linear differential inclusions and arise in the wide variety of modelling problems. Consequently, the system is approximated by a model, which consists of a number of local linear models assigned to regions defined by basis functions $\alpha_v(p)$. In the case of TSMA $\alpha_v(p)$ represent the membership functions of the antecedent fuzzy sets. In multi-variable case, when the system is varying in a multi-dimensional vector space P , rectangular gridding is utilized to define the local linear models and the corresponding basis functions. The system matrix $\mathbf{S}(\mathbf{p})$, where $\mathbf{p} \in \mathfrak{R}^N$, is approximated as:

$$\hat{\mathbf{S}}(\mathbf{p}) = \sum_{v_1=1}^{V_1} \sum_{v_2=2}^{V_2} \cdots \sum_{v_N=1}^{V_N} \prod_{n=1}^N \alpha_{n,v_n}(p_n) \mathbf{S}_{v_1, v_2, \dots, v_N}, \quad (4)$$

where p_n are the elements of vector \mathbf{p} . Along in the same line as above

$$\forall n, v : \alpha_{n,v}(p_n) \in [0, 1] \quad (5)$$

and $\forall n : \sum_{v=1}^{V_n} \alpha_{n,v}(p_n) = 1$ which implies that

$$1 = \sum_{v_1=1}^{V_1} \sum_{v_2=2}^{V_2} \cdots \sum_{v_N=1}^{V_N} \prod_{n=1}^N \alpha_{n,v_n}(p_n). \quad (6)$$

In order to have a general technique to mixed problems with various performance specifications, let a multi-channel system description be discussed, where the system is given as:

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_q \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{A}(\mathbf{p}) & \mathbf{B}_1(\mathbf{p}) & \cdots & \mathbf{B}_2(\mathbf{p}) & \mathbf{B}(\mathbf{p}) \\ \mathbf{C}_1(\mathbf{p}) & \mathbf{D}_1(\mathbf{p}) & \cdots & \mathbf{D}_{1q}(\mathbf{p}) & \mathbf{E}_1(\mathbf{p}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{C}_q(\mathbf{p}) & \mathbf{D}_{1q}(\mathbf{p}) & \cdots & \mathbf{D}_{qq}(\mathbf{p}) & \mathbf{E}_q(\mathbf{p}) \\ \mathbf{C}(\mathbf{p}) & \mathbf{F}_1(\mathbf{p}) & \cdots & \mathbf{F}_q(\mathbf{p}) & \mathbf{D}(\mathbf{p}) \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_q \\ \mathbf{u} \end{pmatrix}, \quad (7)$$

where $\mathbf{w}_j \rightarrow \mathbf{v}_j$ are the channels on which we want to impose certain robustness and/or performance objectives. To facilitate the further development let the notation of (7) be simplified in a systematic form as:

$$\begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_K \end{pmatrix} = \begin{pmatrix} \mathbf{B}_{1,1}(\mathbf{p}) & \mathbf{B}_{1,2}(\mathbf{p}) & \cdots & \mathbf{B}_{1,L}(\mathbf{p}) \\ \mathbf{B}_{2,1}(\mathbf{p}) & \mathbf{B}_{2,2}(\mathbf{p}) & \cdots & \mathbf{B}_{2,L}(\mathbf{p}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_{K,1}(\mathbf{p}) & \mathbf{B}_{K,2}(\mathbf{p}) & \cdots & \mathbf{B}_{K,L}(\mathbf{p}) \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_L \end{pmatrix} = \mathbf{S}(\mathbf{p}) \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_L \end{pmatrix}, \quad (8)$$

where K denotes the number of rows in the model (8) (i.e. the number of equations describing the model), and L indicates how many terms are in the rows of the equations, for instance, these are 2 in (3). Vector $\mathbf{x}_l \in \mathfrak{R}^{l_l}$ consists of the model

input and state vectors, where I_l denotes the number of ‘input’ elements in \mathbf{x}_l . Vector $\mathbf{z}_k \in \mathfrak{R}^{O_k}$ contains the output values of the k -th row in (8), where O_k denotes the number of ‘output’ values in \mathbf{z}_k . This implies that the size of $\mathbf{B}_{k,l}(\mathbf{p})$ is $O_k \times I_l$. For example, describing (3) by (8) results in: $\mathbf{x}_1(t) = \mathbf{x}(t)$, $\mathbf{x}_2(t) = \mathbf{u}(t)$ and the outputs of the model are $\mathbf{z}_1(t) = \dot{\mathbf{x}}(t)$ and $\mathbf{z}_2(t) = \mathbf{y}(t)$. Coefficient matrices become: $\mathbf{B}_{1,1}(\mathbf{p}) = \mathbf{A}(\mathbf{p})$, $\mathbf{B}_{1,2}(\mathbf{p}) = \mathbf{B}(\mathbf{p})$, $\mathbf{B}_{2,1}(\mathbf{p}) = \mathbf{C}(\mathbf{p})$ and $\mathbf{B}_{2,2}(\mathbf{p}) = \mathbf{D}(\mathbf{p})$. Let (8) be substituted into (4), then we obtain:

$$\hat{\mathbf{B}}_{k,l}(\mathbf{p}) = \sum_{v_1=1}^{V_1} \sum_{v_2=2}^{V_2} \cdots \sum_{v_N=1}^{V_N} \prod_{n=1}^N \alpha_{n,v_n}(p_n) \mathbf{B}_{v_1,v_2,\dots,v_N,k,l},$$

which can be reformulated in terms of tensors as:

$$\hat{\mathbf{B}}_{k,l}(\mathbf{p}) = \left(B_{k,l} \otimes_n \mathbf{m}_n(p_n) \right)_{(N+1)} \quad \text{or} \quad \mathbf{S}(\mathbf{p}) = \left(S \otimes_n \mathbf{m}_n(p_n) \right)_{(N+1)},$$

where row vector $\mathbf{m}_n(p_n) \in \mathfrak{R}^{V_n}$ contains basis functions $\alpha_{n,v_n}(p_n)$ and the $N+2$ -dimensional coefficient tensor $B_{k,l} \in \mathfrak{R}^{V_1 \times V_2 \times \dots \times V_N \times O_k \times I_l}$ is constructed from matrices $\mathbf{B}_{v_1,v_2,\dots,v_N,k,l} \in \mathfrak{R}^{O_k \times I_l}$. Tensor $S \in \mathfrak{R}^{V_1 \times V_2 \times \dots \times V_N \times \sum_k O_k \times \sum_l I_l}$ is constructed from system matrices $\mathbf{S}_{v_1,v_2,\dots,v_N}$. The first N dimensions of $B_{k,l}$ are assigned to the dimensions of the parameter space P . The next two ones are assigned to the output and input vectors, respectively.

4. Complexity Investigation

This section investigates the computation complexity of PMA and TSMA. The output values are calculated by (8) as:

$$\mathbf{z}_k = \left(\sum_l \left[B_{k,l} \otimes_n \mathbf{m}_n(p_n) \right] \times_{N+2} \mathbf{x}_l^T \right)_{(N+1)}. \quad (9)$$

LEMMA 1 (Complexity explosion) *The computational complexity of PMA and TSMA techniques grows exponentially with the number of basis functions, dimension of the parameter space and the size of the model coefficients. Considering the number of multiplications the computational requirement is characterised as:*

$$P = \prod_n V_n \left(\sum_k \sum_l O_k I_l + \sum_k O_k \right) + C_p \sum_n V_n, \quad (10)$$

where C_p indicates the number of multiplications during the calculation of a basis function.

To arrive at (10), one notes that calculating the output of one linear local model to a given input needs $\sum_k \sum_l O_k I_l$ multiplications. The number of the local linear models is $\prod_n V_n$. The outputs of the $\prod_n V_n$ local linear models are weighted by the basis functions, which implies $\prod_n V_n \cdot \sum_k O_k$ further multiplications. $C_p \sum_n V_n$ indicates the calculation of the basis functions, where C_p represents the number of multiplications in the calculation of one basis function. Consequently, (10) shows that increasing the density of the basis functions in pursue of good approximation leads to the explosion of the number of local linear models (building units) fully according to the paper by KÓCZY and HIROTA (1997) [8].

5. Key Concept of HOSVD Based Reduction

This section briefly discusses the fundamentals of HOSVD in the sense of complexity reduction. Many reduction properties of the HOSVD of higher-order tensors are investigated in the related literature. Let us briefly summarize those, which have prominent roles in this paper. In multi-linear algebra as well as in matrix algebra, the FROBENIUS norm is unitary invariant. As a consequence, the fact, that the squared FROBENIUS norm of a matrix equals the sum of its squared singular values, can be generalized.

PROPERTY 1 (Approximation) *Let the HOSVD of A be given as in Theorem 2 and let the n -mode rank of A be equal to R_n . Define a tensor \hat{A} by discarding singular values $\sigma_{I'_n+1}^{(n)}, \sigma_{I'_n+2}^{(n)}, \dots, \sigma_{R_n}^{(n)}$ for given values of I'_n , i.e. set the corresponding parts of S equal to zero. Then we have:*

$$\|A - \hat{A}\|^2 \leq \sum_{n=1}^N \left(\sum_{i_n=I'_n+1}^{R_n} (\sigma_{i_n}^{(n)})^2 \right). \quad (11)$$

This property is the higher-order equivalent of the link between the SVD of a matrix and its best approximation in a least-squares sense, by a matrix of lower rank. The situation is, however, quite different for tensors. By discarding the smallest n -mode singular values, one obtains a tensor \hat{A} with n -mode rank of I'_n . Unfortunately, this tensor is, in general, not the best possible approximation under the given n -mode rank constraints [11]. Nevertheless, the ordering implies that the main components of A are mainly concentrated in the part corresponding to low values of the indices. Consequently, if $\sigma_{I'_n}^{(n)} \gg \sigma_{I'_n+1}^{(n)}$, where actually I'_n corresponds to the numerical rank of A then the smaller n -mode singular values are not significant, which implies their discarding. In this case, the obtained \hat{A} is still considered as a good approximation of A . According to the special terms in this topic the following naming has emerged [22, 23]:

DEFINITION 9 (Exact/non-exact reduction) Assume an N -th order tensor $A \in \mathfrak{R}^{I_1 \times I_2 \times \dots \times I_N}$. **Exact** reduced form $A = A^r \underset{n}{\otimes} \mathbf{U}_n$, where ‘ r ’ denotes ‘reduced’, is defined by tensor $A^r \in \mathfrak{R}^{I_1^r \times I_2^r \times \dots \times I_N^r}$ and n -mode singular matrices $\mathbf{U}_n \in \mathfrak{R}^{I_n \times I_n^r}$, $\forall n : I_n^r \leq I_n$ which are the results of Theorem 2, where only the zero singular values and the corresponding singular vectors are discarded. **Non-exact** reduced form $\hat{A} = A^r \underset{n}{\otimes} \mathbf{U}_n$ is obtained if not only zero singular values and the corresponding singular vectors are discarded.

6. SVD Based Complexity Reduction of PMA and TSMA

The main objective of the complexity reduction proposed in this section is twofold, which is introduced via two methods. Method 1 is aimed to minimize values V_n , which means the decrease of the size of tensor $B_{k,l}$ in the first N dimensions. This leads to the minimal number of local linear models. The reduction conducts HOSVD on tensor $B_{k,l}$ to root out linear dependences by truncating zero or small singular values. First an exact reduction is discussed in this section. Increasing the effectiveness of the reduction by discarding non-zero singular values in HOSVD, reduction error is obtained which will be bounded in Remark 2 at the end of this section. A further aim of the reduction to be treated in Method 2 is to decrease values O_k and I_l which also appear in the dominant term of (10). The numbers of input and output values are defined by the application at hand, which implies that Q_k and I_l cannot be directly decreased. Similarly to [1] the key idea of reducing these values can be viewed as the transformation of the whole approximation to a smaller computational space. The input values are also projected in each state step of the modelled system and the output values are calculated in the reduced computational space. Finally, the output values are transformed back to the original space. The reduction is based on executing SVD reduction to the coefficient matrices. As a matter of fact exact reduction cannot be obtained in this step if the coefficient matrices are full in rank, which is usually guaranteed by modelling processes. Non-exact reduction is, however, still possible at the price of reduction error. Note that in this case the approximated coefficient matrices will not be in full rank, which is not acceptable in various theorems of control design. In order to have a completed view, both reduction possibilities are discussed in the next part. First let us characterise the concept and the goal of the reduction by the following Theorem 3:

THEOREM 3 (complexity reduction) Eq. (9) can always be transformed into the following form:

$$\mathbf{z}_k = \left(\sum_l \left[B_{k,l}^r \underset{n}{\otimes} \mathbf{m}_n^r(p_n) \right] \times_{N+1} \mathbf{A}_k \times_{N+2} \mathbf{x}_l^T \mathbf{C}_l \right)_{(N+1)},$$

which is equivalent to

$$\mathbf{z}_k = \mathbf{A}_k \left(\sum_l \left[B_{k,l}^r \otimes_n \mathbf{m}_n^r(p_n) \right] \times_{N+2} \mathbf{x}_l^T \mathbf{C}_l \right)_{(N+1)}, \quad (12)$$

where the size of $B_{k,l}^r \in \mathfrak{R}^{V_1^r \times V_2^r \times \dots \times V_N^r \times O_k^r \times I_l^r}$ may be reduced as $\forall n : V_n^r \leq V_n, O_k^r \leq O_k$ and $I_l^r \leq I_l$.

$\mathbf{m}_n^r(p_n) \in \mathfrak{R}^{V_n^r}$ consists of the new basis functions. The number of the basis functions on the n -th universe is V_n^r . $\mathbf{A}_k \in \mathfrak{R}^{O_k \times O_k^r}$ and $\mathbf{C}_l \in \mathfrak{R}^{I_l \times I_l^r}$ are applied to transform the inputs and the outputs between the reduced and the original computational space, see later at Method 2.

The proof of the Theorem 3 can readily be derived from the following Methods 1 and 2. Before starting with the introduction of the methods, let us have a brief digression and represent the calculation of values \mathbf{z}_k of PMA in respect to \mathbf{x}_l in two different ways, similarly to [1]. Let tensor $G_k \in \mathfrak{R}^{V_1 \times V_2 \times \dots \times V_N \times O_k \times (\sum_l I_l)}$ be given by the form of $G_k = [B_{k,1} \ B_{k,2} \ \dots \ B_{k,L}]_{N+2}$. The output value \mathbf{z}_k of the approximation in respect to \mathbf{x}_k is:

$$\mathbf{z}_k = \left(\left[G_k \otimes_n \mathbf{m}_n(p_n) \right] \times_{N+2} [\mathbf{x}_1^T \ \mathbf{x}_2^T \ \dots \ \mathbf{x}_L^T] \right)_{(N+1)}.$$

The second way utilizes matrix $H_l \in \mathfrak{R}^{V_1 \times V_2 \times \dots \times V_N \times (\sum_k O_k) \times I_l}$ constructed by $H_l = [B_{1,l} \ B_{2,l} \ \dots \ B_{K,l}]_{N+1}$. The output of the TS fuzzy model is:

$$\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_K \end{bmatrix} = \left(\left[[H_1 \ H_2 \ \dots \ H_L]_{N+2} \otimes_n \mathbf{m}_n(p_n) \right] \times_{N+2} [\mathbf{x}_1^T \ \mathbf{x}_2^T \ \mathbf{x}_L^T] \right)_{(N+1)}. \quad (13)$$

The first method shows how to find the minimal number of local linear models.

Method 1 (Determination of the minimal values of V_1, V_2, \dots, V_N)

Applying HOSVD (Theorem 2) to the $N + 2$ -dimensional system tensor S ($S = [G_1 \ G_2 \ \dots \ G_K]_{N+1}$ in such a way that the SVD is executed only on dimensions $1 \dots N$ yields:

$$S = S^r \otimes_n \mathbf{T}_n, \quad (14)$$

where ‘ r ’ denotes ‘reduced’. Tensors $B_{k,l}^r \in \mathfrak{R}^{V_1^r \times V_2^r \times \dots \times V_N^r \times O_l \times I_l}$ are found as the partitions of S^r ($S^r = [G_1^r \ G_2^r \ \dots \ G_K^r]_{N+1}$ and $G_k^r = [B_{k,1}^r \ B_{k,2}^r \ \dots \ B_{k,L}^r]_{N+2}$). If singular values are discarded then the size of $B_{k,l}^r \in \mathfrak{R}^{V_1^r \times V_2^r \times \dots \times V_N^r \times O_l \times I_l}$ is less

than the size of $B_{k,l} \in \mathfrak{R}^{V_1 \times V_2 \times \dots \times V_N \times O_l \times I_l}$, so, $\forall n : V_n^r \leq V_n$, which is the key point of the reduction. Thus for (14) we obtain

$$B_{k,l} = B_{k,l}^r \underset{n}{\otimes} \mathbf{T}_n. \quad (15)$$

The new basis functions are constructed as

$$\mathbf{m}_n^r(p_n) = \mathbf{m}_n(p_n) \mathbf{T}_n. \quad (16)$$

Consequently, (9) can be written in the reduced form by substituting (15) and (16) into (9) which yields:

$$\mathbf{z}_k = \left(\sum_l \left[B_{k,l}^r \underset{n}{\otimes} \mathbf{m}_n^r(p_n) \right] \times_{N+2} \mathbf{x}_l^T \right)_{(N+1)},$$

which is in full accordance with the Theorem 3 of complexity reduction. The objective of Method 2 is to decrease O_k and I_l .

Method 2 (Determination of the minimal computational space)

1. Determination of matrices \mathbf{A}_k , namely, the reduction of O_k .

Let $\mathbf{R}_k = (G_k)_{\mathbf{1}}$. Applying SVD, with discarding zero singular values, to \mathbf{R}_k yields:

$$\mathbf{R}_k = \mathbf{A}_k \cdot \mathbf{D}_k \cdot \mathbf{V}_k = \mathbf{A}_k \mathbf{R}'_k.$$

Matrix $\mathbf{R}'_k \in \mathfrak{R}^{O_l^r \times \prod_n V_n \cdot \sum_l I_l}$ can be restored to tensor

$$G'_k \in \mathfrak{R}^{V_1 \times V_2 \times \dots \times V_N \times O_k^r \times (\sum_l I_l)}.$$

2. Determination of matrices \mathbf{C}_l , namely, the reduction of I_l .

Let tensor $H'_k \in \mathfrak{R}^{V_1 \times V_2 \times \dots \times V_N \times \sum_k O_k \times I_l}$ be constructed as

$$H'_k = [B'_{1,l} B'_{2,l} \dots B'_{K,l}]_{N+1},$$

where tensors $B'_{k,l}$ are defined according to the result

$$G'_k = [B'_{k,1} B'_{k,2} \dots B'_{k,L}]_{N+2}$$

by step 1. Then let $\mathbf{M}_l = (H'_l)_{(N+2)}$ whereupon executing SVD yields (where zero singular values are discarded):

$$\mathbf{M}_l = \mathbf{C}_l \cdot \mathbf{D}'_l \cdot \mathbf{V}'_l = \mathbf{C}_l \cdot \mathbf{M}'_l.$$

Matrix \mathbf{M}_l^r defines tensors $B_{k,l}^r \in \mathfrak{N}^{V_1 \times V_2 \times \dots \times V_N \times O_k^r \times I_l^r}$ according to $\mathbf{M}_l^r = (H_l^r)_{(N+2)}$ and $H_l^r = [B_{1,l}^r \ B_{2,l}^r \ \dots \ B_{K,l}^r]_{N+1}$.

The results of Method 2 are \mathbf{A}_k and \mathbf{C}_l . \mathbf{C}_l is applied to transform the input values \mathbf{x}_l to a reduced space as: $\mathbf{x}_l^r = \mathbf{C}_l^T \cdot \mathbf{x}_l$. The output is calculated in the reduced computational space as:

$$\mathbf{z}_k^r = \left(\sum_l \left[B_{k,l}^r \otimes_n \mathbf{m}_n(p_n) \right] \times_{N+2} (\mathbf{x}_l^r)^T \right)_{(N+1)}.$$

The output \mathbf{z}_k^r is projected to the original space by $\bar{\mathbf{z}}_k^r \mathbf{A}_k \mathbf{z}_k^r$, which is in full accordance with the Theorem 3 of complexity reduction.

The ordering of executing Method 1 and 2 is arbitrary. In the following, some important issues and interpretability problems of the results are discussed.

Remark 1 The functions in (16) obtained by Method 1 may not be interpretable as basis functions, which satisfy (5), since the transformation using matrix \mathbf{T}_n may result in functions with negative values. Another crucial point is that the resulted basis functions do not guarantee the normality, which means that (6) may not be equal to 1. This fact would destroy the whole reduction concept since calculating (6) with the new basis may get far from 1. However, if only the saving of computational cost of final implementation is in purpose and the conditions (5) and (6) of the basis do not have to be accommodated, then (12) is directly applicable. If the reduced form is for further studies in fuzzy theory and/or Lyapunov stability analysis, then the reduced basis functions should accommodate additional characterization pertaining to specific operations. This may require further transformations. To obtain matrices \mathbf{T}_n in such a way that the reduced basis functions are bounded by [0, 1] and hold (6), *Non-Negativeness* (NN) and *Sum-Normalisation* (SN) transformation techniques are developed by YAM in [22, 23]. If the SVD is accompanied by these transformations then the resulted functions fulfil (5) and (6), and remain interpretable as antecedent fuzzy sets in the case of TSMA. This leads to the theoretically correct use of (9) and (8).

Remark 2 An advantage of the proposed algorithm is that it has error controllable property, i.e. if the HOSVD is executed in non-exact mode then the original and the reduced approximation differ and the difference can be estimated during executing the reduction technique. In Section 5 it is shown that discarding non-zero singular values results in reduction error, which can be bounded by (11). Works [1, 2, 3, 4, 22, 23] bound the maximum reduction error by the sum of the discarded singular values. As a matter of fact, the reduction errors of the proposed methods also depend on the basis functions applied. In this regard various cases of basis functions are discussed in [2]. Generally speaking, it can be said that if the original basis holds (6) then the maximum final model approximation error is the sum of the discarded singular values, which can be controlled during executing Methods 1 and 2. For more details about the error bound of SVD reduction see works [1, 2, 3, 4, 22, 23].

Remark 3 Method 1 may result in basis functions, which cannot be analytically simplified and hence their shapes are rather complicated and their computational loads may be greater than that of the original ones. Observing (10) it is concluded that C_p is not in the dominant part of (10) which implies that this computational increase of the new basis is dispensable in comparison to the exponential feature of the dominant term. In the worst case, the original functions are calculated first, after this the values of the reduced basis are simply determined by (16) in each step of the system. Therefore, the worst case is bounded by

$$P = \prod_n V_n^r \left(\sum_k \sum_l O_k^r I_l^r + \sum_k O_k^r \right) + C_p \sum_n V_n + \sum_n V_n V_n^r + \sum_k O_k O_k^r + \sum_l I_l I_l^r, \quad (17)$$

where the extra term $V_n V_n^r$ indicates the extra computational load of calculating the values of the basis functions on the n -th universe. $\sum_k O_k O_k^r$ and $\sum_l I_l I_l^r$ are from the computation requirement of the transformation between the original and the reduced computational spaces. Consequently, the effectiveness of the reduction is

$$\eta = \frac{\prod_n V_n^r (\sum_k \sum_l O_k^r I_l^r + \sum_k O_k^r) + C_p \sum_n V_n + \sum_n V_n V_n^r + \sum_k O_k O_k^r + \sum_l I_l I_l^r}{\prod_n V_n (\sum_k \sum_l O_k I_k + \sum_k O_k) + C_p \sum_n V_n}.$$

In the case of a dense griding or higher dimensional parameter space its dominant expresses the effectiveness of the reduction in the sense of computational complexity:

$$\eta \approx \frac{\prod_n V_n^r (\sum_k \sum_l O_k^r I_l^r + \sum_k O_k^r)}{\prod_n V_n (\sum_k \sum_l O_k I_k + \sum_k O_k)}.$$

Remark 4 Method 1 could be modified in such a way that the reduction results in one approximation for each row or column of (2) like in [1]. Furthermore, one basis system could be resulted for each coefficient tensor $B_{k,l}$. The advantage of the separately executed reduction of each $B_{k,l}$ in Method 1 is that the size of some $B_{k,l}^r$ may become less, while computing (14) the sizes of all partitions $B_{k,l}$ of S^r are the same in the first N dimensions. This is due to the fact that the n -mode rank of tensor $B_{k,l}$ is less than or equal to the n -mode rank of tensor S in (14). In the worst case its maximum could be $\min(\sum_{k,l} \text{rank}_n(B_{k,l}), \text{rows}((S)_{(n)}))$. Consequently, replacing S in (14) with $B_{k,l}$, the following is obtained:

$$B_{k,l} = B_{k,l}^r \otimes_n \mathbf{T}_{n,k,l}$$

and according to (16) the new basis systems are: $\mathbf{m}_{n,k,l}^r(p_n) = \mathbf{m}_n(p_n) \mathbf{T}_{n,k,l}$, where the basis defined by $\mathbf{m}_{n,k,l}^r(p_n)$ is assigned to the approximation of $B_{k,l}$. Again, the

benefit is that the size of each $B_{k,l}^r$ in the present modified case is less than or equal to the common size of $B_{k,l}^r$ resulted by Method 1. As a matter of fact, the calculation cost of the basis may increase since one basis system should be calculated for each $B_{k,l}$, however, this extra calculation is not included in the exponentially dominant part of (10) and (17). This pin-pointing of the reduction is burdened by the fact, that one has to check, whether performing the reduction for each coefficient tensor separately would yield a better computational reduction or not.

7. Example

This example, taken from [28] and [1], is a design for a simple non-linear mass-spring-damper mechanical system depicted in *Fig. 2*. The main goal of this example is to approximate the mass-spring-damper mechanical system (like a dynamically unknown one) by PMA or TSMA over a dense approximation grid. The reason for applying dense approximation grid is the goal of achieving a small approximation error. Then the example performs the proposed reduction technique to find the minimal number of local linear models. The differential equations of the mechanical system are analytically derived into the minimal form of PMA or TSMA, as well, in order to evaluate the effectiveness of the proposed reduction technique. The goal here is to show that the minimal form resulted by the proposed methods from training data is the same, in the sense of complexity, as the analytically derived PMA or TSMA model.

First let us discuss the dynamic model from the example of [28]. It is assumed that the stiffness coefficient of the spring, the damping coefficient of the damper, and the input term have non-linearity or uncertainty

$$m \cdot \ddot{x} + g(x, \dot{x}) + k(x) = \phi(\dot{x}) \cdot u, \quad (18)$$

where m is the mass and u stands for the force. $k(x)$ is the non-linear or uncertain term with respect to the spring. $g(x, \dot{x})$ is the non-linear or uncertain term with respect to the damper. $\phi(\dot{x})$ is the non-linear term with respect to the input term. Assume that $g(x, \dot{x}) = d(c_1x + c_2\dot{x}^3)$, $k(x) = c_3x + c_4x^3$, and $\phi(\dot{x}) = 1 + c_5\dot{x}^3$. Furthermore, assume that $x \in [-a, a]$, $\dot{x} \in [-b, b]$ and $a, b > 0$. The above parameters are set as follows [25]: $m = 1$, $d = 1$, $c_1 = 0.01$, $c_2 = 0.1$, $c_3 = 0.01$, $c_4 = 0.67$, $c_5 = 0$, $a = 1.5$, and $b = 1.5$. *Eq. (18)* then becomes:

$$\ddot{x} = -0.1\dot{x}^3 - 0.02x - 0.67x^3 + u. \quad (19)$$

The non-linear terms are $-0.1\dot{x}^3$ and $-0.67x^3$. Let us proceed further in the same way as done in [28] and give a PMA or TSMA of (19) with minimal number of basis functions. x and \dot{x} have the following conditions:

$$\begin{cases} -1.5075x \leq -0.67x^3 \leq 0 \cdot x & x \geq 0 \\ 0 \cdot x \leq -0.67x^3 \leq -1.5075x & x < 0 \end{cases}$$

and

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

This fact means that the non-linear term can be represented by the upper and the lower bounds: $-0.67x^3 = f_{1,1}(x)x \cdot 0 - (1 - f_{1,1}(x)) \cdot 1.5075x$ and $-0.1\dot{x}^3 = f_{2,1}(\dot{x})\dot{x} \cdot 0 - (1 - f_{2,1}(\dot{x})) \cdot 0.225\dot{x}$, where $f_{n,v_n}(\dot{x}) \in [0, 1]$, $V_n = 2$. This leads to basis functions $f_{1,1}^a(x) = 1 - \frac{x^2}{2.25}$, ('a' means that the function is obtained analytically), $f_{1,2}^a(x) = \frac{x^2}{2.25}$; $f_{2,1}^a(\dot{x}) = 1 - \frac{\dot{x}^2}{2.25}$; $f_{2,2}^a(\dot{x}) = \frac{\dot{x}^2}{2.25}$. The basis functions are depicted in Fig. 4. Thus, the following assignments are obtained analytically:

$$\ddot{x} = -0.1\dot{x}^3 - 0.02x - 0.67x^3 + u = \sum_{i=1}^2 \sum_{j=1}^2 f_{1,i}^a(x) f_{2,j}^a(\dot{x}) \phi_{i,j},$$

where $\phi_{1,1} = -0.02x + u$, $\phi_{1,2} = -0.225\dot{x} - 0.02x + u$, $\phi_{2,1} = -1.5275x + u$ and $\phi_{2,2} = -0.225\dot{x} - 1.5275x + u$. This approximation in matrix representation takes the form:

$$\mathbf{A}(x, \dot{x}) = \sum_{i=1}^2 \sum_{j=1}^2 f_{1,i}^a(x) f_{2,j}^a(\dot{x}) \mathbf{A}_{i,j}^a; \quad \mathbf{B}(x, \dot{x}) = \sum_{i=1}^2 \sum_{j=1}^2 f_{1,i}^a(x) f_{2,j}^a(\dot{x}) \mathbf{B}_{i,j}^a, \quad (20)$$

where

$$\begin{aligned} \mathbf{A}_{1,1}^a &= \begin{bmatrix} 0 & -0.02 \\ 1 & 0 \end{bmatrix}, & \mathbf{B}_{1,1}^a &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, & \mathbf{A}_{1,2}^a &= \begin{bmatrix} -0.225 & -0.02 \\ 1 & 0 \end{bmatrix}, \\ \mathbf{B}_{1,2}^a &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, & \mathbf{A}_{2,1}^a &= \begin{bmatrix} 0 & -1.5275 \\ 1 & 0 \end{bmatrix}, & \mathbf{B}_{2,1}^a &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\ \mathbf{A}_{2,2}^a &= \begin{bmatrix} -0.225 & -1.5275 \\ 1 & 0 \end{bmatrix}, & \mathbf{B}_{2,2}^a &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \end{aligned}$$

The analytically obtained PMA, consisting of four local linear models, exactly represents the non-linear system. The model has two basis functions in each parameter dimension, which is sufficient for the approximation. The next step is to approximate the model (18) with a dense approximation grid, after which we can assume that (18) is unknown and then go about generating a minimum sized PMA by the present reduction technique. Let intervals $\dot{x}, x \in [-1.5, 1.5]$ be divided by 400 triangular shaped basis functions (or, in other words, by first order B-spline basis), see Fig. 3.

Therefore the approximation is computed as:

$$\hat{\mathbf{A}}(x, \dot{x}) = \sum_{i=1}^{400} \sum_{j=1}^{400} f_{1,i}(x) f_{2,j}(\dot{x}) \mathbf{A}_{i,j}; \quad \hat{\mathbf{B}}(x, \dot{x}) = \sum_{i=1}^{400} \sum_{j=1}^{400} f_{1,i}(x) f_{2,j}(\dot{x}) \mathbf{B}_{i,j}. \quad (21)$$

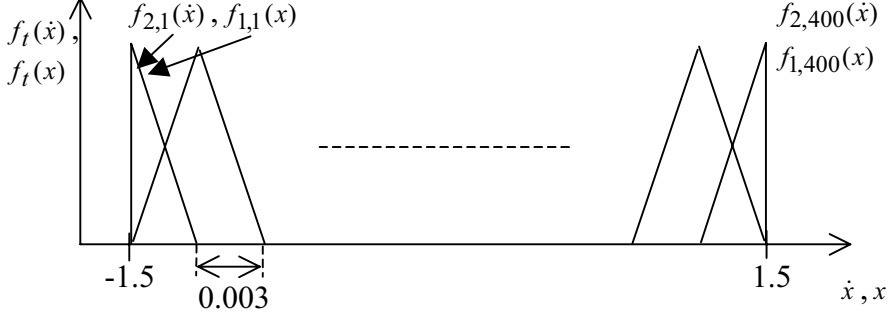


Fig. 3. Dense basis to achieve a good approximation

We sample the dynamic system over the approximation grid points defined by $x_i = -1.5 + (i - 1)3/400$ and $\dot{x}_j = -1.5 + (j - 1)3/400$, which imitates the result of an identification algorithm like in [1]. Thus the dense approximation becomes

$$\ddot{x} = \sum_{i=1}^{400} \sum_{j=1}^{400} f_{1,i}(x) f_{2,j}(\dot{x}) (a_{i,j} \dot{x} + b_{i,j} x + c_{i,j} u), \quad (22)$$

where $a_{i,j} = -0.1(-1.5 + (j - 1)3/400)^2$, $b_{i,j} = -0.02 - 0.67(-1.5 + (i - 1)3/400)^2$, and $c_{i,j} = 1$. The matrix form in (21) can easily be generated from (22). Executing Method 2 on matrices $\mathbf{A}_{i,j}$, namely, on tensor $A \in \mathfrak{R}^{400 \times 400 \times 2 \times 2}$ (note that matrices $\mathbf{B}_{i,j}$ are constant) results in two non-zero singular values to the first dimension such as 461.6404... and 156.5663... and after performing SN and NN transformation (see Remark 1) two non-zero singular values are obtained to the second dimension, such as 100.8708... and 1.8970....

$$\begin{aligned} \mathbf{A}_{1,1}^r &= \begin{bmatrix} -169.5952205449 \dots & -2.871864441639 \dots \\ 1 & 0 \end{bmatrix}, \\ \mathbf{A}_{1,2}^r &= \begin{bmatrix} 338.965358635779 \dots & -2.871864441665 \dots \\ 1 & 0 \end{bmatrix}, \\ \mathbf{A}_{2,1}^r &= \begin{bmatrix} -169.595220544832 \dots & 3.895957360524 \dots \\ 1 & 0 \end{bmatrix}, \\ \mathbf{A}_{2,2}^r &= \begin{bmatrix} 338.965358635644 \dots & 3.895957360558 \dots \\ 1 & 0 \end{bmatrix}. \end{aligned} \quad (23)$$

This means that two basis functions on each dimension are sufficient for the same approximation, which is in full accordance with the analytical polytopic model design. Further the resulted basis functions maintain (5) and (6). The main conclusion

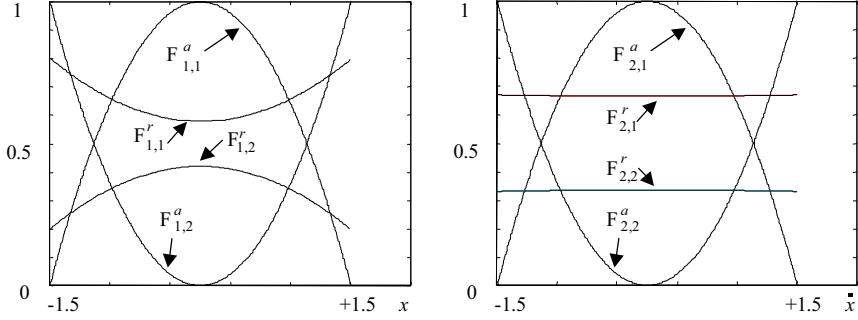


Fig. 4. Basis functions of the original approximation via analytical derivation and the basis extracted from training data by HOSVD reduction.

is that the PDC design (or any further LMI analysis) can be restricted to the resulted four linear local models. Let us proceed further and determine the basis functions. The new membership functions inherit the piece-wise linear property of the original triangular shaped membership functions. We approximate the break points of the pieces, (which are actually the elements in the columns of \mathbf{T}_n [2]), by a polynomial fitting, which results in:

$$\begin{aligned} f_{1,1}^r(x) &= \alpha_1 + \beta_1 x^2, & f_{1,2}^r(x) &= 1 - f_{1,1}^r(x), \\ f_{2,1}^r(\dot{x}) &= \alpha_2 + \beta_2 \dot{x}^2, & f_{2,2}^r(\dot{x}) &= 1 - f_{2,1}^r(\dot{x}), \end{aligned} \quad (24)$$

where $\alpha_1 = 0.57861413538877\dots$, $\beta_1 = 0.09899787842876\dots$, $\alpha_2 = 0.66651913756641\dots$ and $\beta_2 = 1.966334082795784 \cdot 10^{-4}$. The membership functions are depicted in Fig. 4. Let us take a brief digression here and show via linear transformations that the model obtained in (24) is a variant form of (20). The analytically derived basis functions can be transformed to the reduced basis as (in the following steps the equivalency is understood in numerical sense):

$$\mathbf{m}_1^r = \mathbf{m}_1^a \mathbf{T}_1 \quad \text{and} \quad \mathbf{m}_2^r = \mathbf{m}_2^a \mathbf{T}_2,$$

where $\mathbf{m}_1^r(x) = [f_{1,1}^r(x) \ f_{1,2}^r(x)]$, $\mathbf{m}_2^r(\dot{x}) = [f_{2,1}^r(\dot{x}) \ f_{2,2}^r(\dot{x})]$, $\mathbf{m}_1^a(x) = [f_{1,1}^a(x) \ f_{1,2}^a(x)]$ and $\mathbf{m}_2^a(\dot{x}) = [f_{2,1}^a(\dot{x}) \ f_{2,2}^a(\dot{x})]$. The transformation matrices are:

$$\mathbf{T}_1 = \begin{bmatrix} 0.57861413538877\dots & 0.42138586461123\dots \\ 0.80135936185347\dots & 0.19864063814653\dots \end{bmatrix}$$

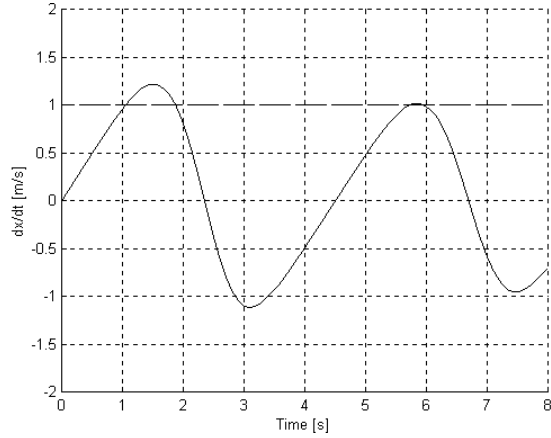
$$\mathbf{T}_2 = \begin{bmatrix} 0.66651913756641\dots & 0.33348086243359\dots \\ 0.66696156273504\dots & 0.33303843726497\dots \end{bmatrix}.$$

In the same way $A^r = A^a \times_1 \mathbf{T}_1^{-1} \times_2 \mathbf{T}_2^{-1}$, where coefficient tensors $A^r \in \mathfrak{N}^{2 \times 2 \times 2 \times 2}$ and $A^a \in \mathfrak{N}^{2 \times 2 \times 2 \times 2}$ are respectively constructed from matrices $\mathbf{A}_{i,j}^r$ and $\mathbf{A}_{i,j}^a$. Consequently,

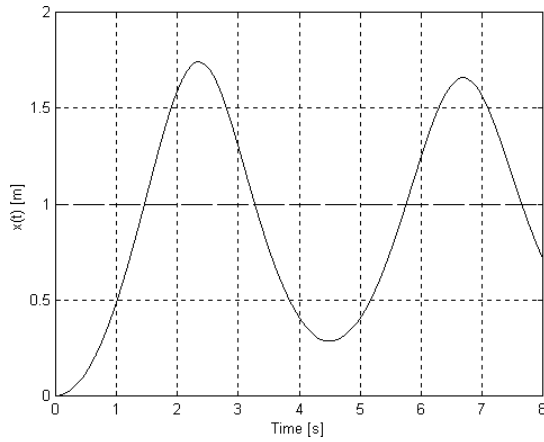
$$\begin{aligned} A^r \times_1 \mathbf{f}_1^r(x) \times_2 \mathbf{f}_2^r(x) &= A^a \times_1 \mathbf{T}_1^{-1} \times_2 \mathbf{T}_2^{-1} \times_1 (\mathbf{f}_1^a(x) \mathbf{T}_1) \times_2 (\mathbf{f}_2^a(x) \mathbf{T}_2) \\ &= A^a \times_1 (\mathbf{f}_1^a(x) \mathbf{T}_1 \mathbf{T}_1^{-1}) \times_2 (\mathbf{f}_2^a(x) \mathbf{T}_2 \mathbf{T}_2^{-1}) = A^a \times_1 \mathbf{f}_1^a(x) \times_2 \mathbf{f}_2^a(x). \end{aligned}$$

We can conclude that the two models are equivalent to the model given by differential equations. Equivalency of the models is understood here in numerical sense i.e. the difference between the outputs of the models to the same inputs is under $\varepsilon < 10^{-12}$. *Fig. 5* shows the response of the analytically derived and the reduced rule base in the case of step change. We can observe that the output signals are equivalent.

In conclusion let the above outlined example be summarised in *Fig. 6*. Block 1 illustrates the model given by differential equations. Block 2 is achieved via an analytical derivation of the differential equations of Block 1. Block 3 shows a polytopic model consisting of four linear local models and the corresponding basis functions. Having the polytopic model the frameworks of LMI based approaches to solve Lyapunov criteria can be applied. This way between Block 1 and Block 2 cannot, however, be done easily in a general case. It may need accurate and complicated mathematical solutions including human intuition. In the present example we have chosen a simple model, in order to show the derivation via simple analytical solution as well. Another way, which is proposed in this paper, is to go and generate an approximation of the analytically given model of Block 1 by a universal approximation technique such as tensor product based polytopic method, the determination of which does not need human intuition, but computation power only. In a general case the polytopic model tends to have the universal property while its complexity goes to infinity. Consequently, Block 3 represents a polytopic model approximation of the differential equations of Block 1 with acceptable small approximation error. The price to pay for the acceptable approximation accuracy is the extremely high complexity explosion of the polytopic model (160000 linear local models in the present example), which may burden the use of LMI based controller design frameworks (in order to design a controller in the present simple example already 160000 equations should be solved by LMI). The complexity reduction of the model is, hence, highly desired. Block 4 is a polytopic model which is the reduced form of the polytopic model of Block 3. This contains only four rules fully according to the result in Block 2, which shows that four models should be enough. The reduction is done by the proposed HOSVD reduction technique. This model has the same approximation error as the model in Block 3. Block 5 shows that there exists a solution for polytopic model that is exactly the same as the differential equations in Block 1. The final polytopic model in Block 5 is generated from the reduced model via approximating the reduced basis functions by polynomials. This final step is motivated by the result in Block 2.



a)



b)

Fig. 5. Response of the reduced (depicted by solid line) and the analytically derived (depicted by dash dotted line) models to step change (depicted by dashed line). Figures a) and b) respectively show \dot{x} and x . The curves of the reduced and the analytically derived models precisely cover each other in both figures.

8. Conclusion

In this paper we have argued that the identification of PMA and TSMA models from training data needs consideration of an important feature between data fitness and model complexity. We emphasize the importance of these features by pointing out that PMA and TSMA models with large number of local models may encounter

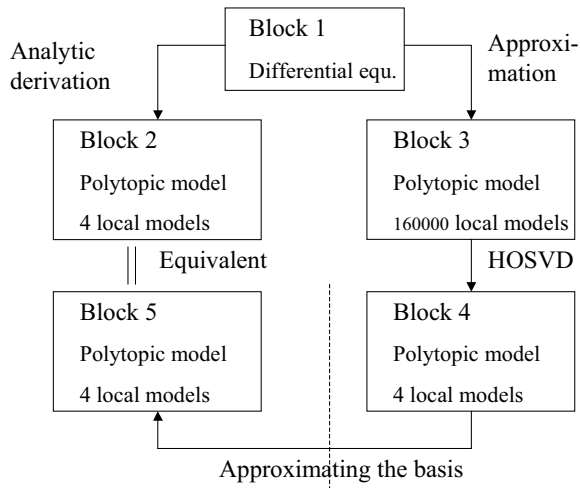


Fig. 6. Structure of the example

the risk of having an approximation capable of fitting training data well, but incapable of running on satisfactorily low computational cost. In order to help the developments of PMA and TSMA models to strive for balance between the two conflicting modelling objectives, we introduce a HOSVD based PMA and TSMA model reduction technique. Using the proposed method, we have demonstrated the application of HOSVD to a dynamic system approximation. The main contribution of this paper is that the proposed approach is the expounded form of the single variable SVD based reduction technique of [1] to multi-variable cases.

Acknowledgement

P. Baranyi is supported by 2. Magyar scholarship. This work is supported by OTKA F 30056 and FKFP 180/2001.

References

- [1] BARANYI, P. – YAM, Y. – VÁRKONYI-KÓCZY, A. R. – PATTON, R. J. – MICHELBERGER, P. – SUGIYAMA, M., SVD Based Reduction of TS Model Approximation. *IEEE Trans. Industrial Electronics* (2001) (accepted).
- [2] BARANYI, P. – YAM, Y., Fuzzy Rule Base Reduction. Chapter 7 of *Fuzzy IF-THEN Rules in Computational Intelligence: Theory and Applications* Eds., D. Ruan and E.E. Kerre, Kluwer, 2000, pp. 135–160.
- [3] BARANYI, P. – YAM, Y., Singular Value-Based Approximation with Takagi-Sugeno Type Fuzzy Rule Base. *6th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97)*, Barcelona, Spain, 1997, pp. 265–270.

- [4] BARANYI, P. – YAM, Y., Singular Value-Based Approximation with Non-Singleton Fuzzy Rule Base. *7th Int. Fuzzy Systems Association World Congress (IFSA'97)* Prague, 1997, pp. 127–132.
- [5] CASTRO, J. L., Fuzzy Logic Controllers are universal approximators. *IEEE Trans. on SMC.*, **25** (1995), pp. 629–635.
- [6] COMON, P., Independent Component Analysis, a New Concept? *Signal Processing, Special Issue on Higher Order Statistics*, **36** (1994), pp. 287–314.
- [7] HORVÁTH, S. – KERESZTES, A. – MICHELBERGER, P. – SZEIDL, L., Preliminary Load Analysis of Commercial Vehicles, *Periodica Polytechnica, Technical University of Budapest, Transportation Engineering*, **9** (1981), No. 2.
- [8] KÓCZY, L. T. – HIROTA, K., Size Reduction by Interpolation in Fuzzy Rule Bases, *IEEE Trans. SMC*, **27** (1997), pp. 14–25.
- [9] KOLMOGOROV, A. N., On the Preservation of Continuous Functions of Many Variables by Superpositions of Continuous Functions of one Variable and Addition. *Dokl. Akad. Nauk SSSR*, **114** (1957), pp. 953–956.
- [10] KOSKO, B., Fuzzy Systems as Universal Approximators. *Proc. of the IEEE Int. Conf. On Fuzzy Systems*, San Diego, 1992, pp. 1153–1162.
- [11] LATHAUWER, L. D. – MOOR, B. D. – VANDEWALLE, J., A Multi Linear Singular Value Decomposition. *SIAM Journal on Matrix Analysis and Applications*, **21** No. 4. (2000), pp. 1253–1278.
- [12] LATHAUWER, L. D. – MOOR, B. D. – VANDEWALLE, J., An Introduction to Independent Component Analysis. *Jour. Chemometrics*, **14** (2001), pp. 123–149.
- [13] LORENTZ, G. G., *Approximation of Functions*, Holt, Reinhard and Winston, New York, 1966.
- [14] STEWART, G. W., On the Early History of the Singular Value Decomposition. *SIAM Rev.*, **35** (1993), No. 4, Dec. pp. 551–566.
- [15] MICHELBERGER, P. – FUTÓ, P. – KERESZTES, A., *Acta Technica Hung.*, **83** (1–2) p. 93.
- [16] MICHELBERGER, P. – FERENCZI, M. – ÁGOSTON, A. – ÚJHELYI, Z., *Periodica Polytechnica, Technical University of Budapest, Transportation Engineering*, **4** (1976), p. 161.
- [17] SPRECHER, D. A., On the Structure of Continuous Functions of Several Variables. *Trans. Amer. Math. Soc.*, **115** (1965), pp. 340–355.
- [18] SWAMI, A. – GIANNAKIS, G., Editorial. Higher-Order Statistics, *Signal Processing, Special Issue Higher Order Statistics*, **53** (1996), Nos. 2–3, pp. 89–91.
- [19] TANAKA, K. – IKEDA, T. – WANG, H. O., Fuzzy Regulators and Fuzzy Observers: Relaxed Stability Conditions and LMI-Based Design. *IEEE Trans. Fuzzy Systems*, **6** (2) (1998), pp. 250–265.
- [20] TANAKA, K. – SUGENO, M., Stability Analysis and Design of Fuzzy Control Systems, *Fuzzy Sets and Systems*, **45** (2) (1992), pp. 135–156.
- [21] TIKK, D., On Nowhere Denseness of Certain Fuzzy Controllers Containing Prerestricted Number of Rules, *Tatra Mountains Mathematical Publications*, **16** (1999), pp. 369–377.
- [22] YAM, Y., Fuzzy Approximation via Grid Point Sampling and Singular Value Decomposition, *IEEE Trans. SMC*, **27** (1997), pp. 933–951.
- [23] YAM, Y. – BARANYI, P. – YANG, C. T., Reduction of Fuzzy Rule Base via Singular Value Decomposition, *IEEE Trans. Fuzzy Systems*, **7** No. 2 (1999), pp. 120–132.
- [24] YEN, J. – WANG, L., Simplifying Fuzzy Rule-based Models Using Orthogonal Transformation Methods. *IEEE Trans. SMC*, **29**: Part B, No. 1, (1999), pp. 13–24.
- [25] WANG, H. O. – TANAKA, K. – GRIFFIN, M. F. P., An Approach to Fuzzy Control of Non-Linear Systems: Stability and Design Issue, *IEEE Trans. Fuzzy Systems*, **4** (1) (1996), pp. 14–23.
- [26] WANG, L. X., Fuzzy Systems are Universal Approximators, *Proc. of the IEEE Int. Conf. On Fuzzy Systems*, San Diego, 1992, pp. 1163–1169.
- [27] ARNOLD, V. I., On Functions of Three Variables, *Doklady Akademii Nauk USSR*, **114** (1957), pp. 679–681.
- [28] TANAKA, K. – IKEDA, T. – WANG, H. O., Robust Stabilization of a Class of Uncertain Nonlinear Systems via Fuzzy Control: Quadratic Stabilizability, H Infinite Control Theory, and Linear Matrix Inequalities, *IEEE Trans. On Fuzzy Systems*, **4** No. 1, 1996.