Singular Value Based Model Approximation

[1]P.BARANYI, [2]A. VÁRKONYI-KÓCZY, [1]P.VÁRLAKI, [1]P. MICHELBERGER and [3]R.J. PATTON
[1]Mechanical Research Group of Hungarian Academy of Sciences
[2]Dept. Measurement and Information Systems, Technical University of Budapest, Hungary
[3]Control and Intelligent Systems Research Group, University of Hull,
Cottingham Road,Hull, HU6 7RX, UK

Abstract: - This paper is motivated by the fact that though weighted component based non-linear model approximation techniques are popular engineering tools, their utilisation is being restricted by their exponential complexity caused by the number of components. Even in case when the components are generated by some expert operator, the approximations usually have redundant or weakly contributing components resulting in exponentially growing unnecessary calculation. The main objective of this paper is to propose a complexity reduction technique capable of finding the minimal number of components of a given approximation.

Key-Words: - Model approximation, model reduction, calculation complexity, singular value decomposition,

1 Introduction

The calculation power offered by computers considerably gained the role of numerical techniques of smoothing procedures, which represents the function to be approximated by linear combination of m known basis functions $w(\underline{\mathbf{x}})$ such as: $f(\underline{\mathbf{x}}) \approx \hat{f}(\underline{\mathbf{x}}) = \sum_{v=1}^{m} w_v(\underline{\mathbf{x}}) p_v$ ([8] page 273). One of the mostly adapted approaches to determine p_i is the least squares of basis expansions. Various techniques have been proposed to choose a basis to achieve an excellent approximation using comparatively small value of m. For instance, Fourier series, polynomial bases, regression spline bases and Wavelet bases, or explicitly localized smoothing methods such as kernel and local polynomial smoothing are to serve this purpose [7]. In some engineering aspects the basis functions themselves can become interesting descriptors of the dates from a substantive point of view, like in a probabilistic extension or even in the case of fuzzy logic that takeovers the use of tight mathematical framework to indicate the semantic and linguistic meaning of the approximation. The rapidly increasing complexity of ordinary systems to be controlled forces engineers to face the inevitable changing of points p_i into various type components (models, behaviours, knowledge-bases etc.) that may be complex in itself. This paper focuses on the widely adopted approximation of non-linear models in a given parameter space. Regarding the limited size of this paper, instead of selecting from the uncountable number of publications let us mention only various topics where the mentioned concepts are popular: behaviour fusion techniques in the topic of behaviour based control, multi objective behaviour based control, multi objective decision making, robot guiding based on superposition of behaviours, fuzzy and B-spline approximations [2,3].

Despite the widely spreading use of the above techniques, they are however strongly restricted by their exponential complexity problem, namely, their calculation complexity requirement grows exponentially with the number of model points (see section 3). The calculation time is a crucial quest in real time applications. The main problem is that there is no standardised framework regarding the design, optimality, reducibility for defining the model points in general. The model points, be it generated from expert operators or by some training or identification schemes, may contain redundant, weakly contributing, or outright inconsistent model points. Moreover, in pursuit of good approximation, one may be tempted to overly assign the number of model points, thereby resulting in problems of computation time and storage space. A model approximation, hence, has two important objectives. One is to achieve a good approximation. The other is to reduce the number of model points. The main difficulty is that these two objectives are contradictory. A formal approach to extracting the more pertinent model points, hence, is highly desirable. The present paper is an attempt in this direction. The reduction technique proposed in this paper finds the minimal model points and their new weighting, which defines the same approximation. The proposed method finds the minimal common basis for model points. Another way to reduce the calculation complexity is to find a minimal common space for the equations defined at all model points. In this case the input values are transformed into the reduced space and the calculation of the output values is done in the reduced equation space with smaller computation effort. The real output values are finally transformed back to the original space. An important advantage of the proposed reduction technique is that there is a formal measure to filtering out not only the redundant, but the weakly contributing components as well. This implies that the degree of reduction can be applied according to the maximum acceptable error of the model approximation. The key idea of SVD based reduction applied in this paper is proposed by Yeung Yam for fuzzy approximation in 1996 [1].

2 Definitions

This section gives the definitions of some widely adopted non-linear model approximation techniques what the reduction is proposed for.

Definition*: Model M. In this paper the following model M is investigated:*

$$\left.\begin{aligned}\dot{\underline{\mathbf{x}}}(t) &= \underline{\underline{\mathbf{A}}}\underline{\mathbf{x}}(t)+\underline{\underline{\mathbf{B}}}\underline{\mathbf{u}}(t)\\ \underline{\mathbf{y}}(t) &= \underline{\underline{\mathbf{C}}}\underline{\mathbf{x}}(t)+\underline{\underline{\mathbf{D}}}\underline{\mathbf{u}}(t)\end{aligned}\right\}\Rightarrow \mathbf{M}\quad, \qquad (3)$$

where the numbers of elements in vectors $\underline{\mathbf{u}}$, $\underline{\mathbf{x}}$, $\underline{\mathbf{y}}$ *are* n_u, n_x *and* n_y*, respectively.*

Definition*: Non-linear model* $\mathbf{M}\Rightarrow\mathbf{M}(\underline{\mathbf{p}})$ *in the parameter space P. Let us suppose that the model defined in (3) is non-linear in the* n *dimensional parameter space P, namely, matrices* $\underline{\underline{\mathbf{A}}}$, $\underline{\underline{\mathbf{B}}}$, $\underline{\underline{\mathbf{C}}}$ *and* $\underline{\underline{\mathbf{D}}}$ *are non-linear in respect of* $\underline{\mathbf{p}}=[p_i]$, $i=1..n$ *such as:*

$$\underline{\underline{\mathbf{A}}}=\begin{vmatrix} f^a_{1,1}(\underline{\mathbf{p}}) & \cdots & f^a_{1,n_x}(\underline{\mathbf{p}})\\ \vdots & \ddots & \vdots\\ f^a_{n_x,1}(\underline{\mathbf{p}}) & \cdots & f^a_{n_x,n_x}(\underline{\mathbf{p}})\end{vmatrix}$$

and similarly $\underline{\underline{\mathbf{B}}}=\left\lfloor f^b_{i,j}(\underline{\mathbf{p}})\right\rfloor$, $\underline{\underline{\mathbf{C}}}=\left\lfloor f^c_{i,j}(\underline{\mathbf{p}})\right\rfloor$ *and* $\underline{\underline{\mathbf{D}}}=\left\lfloor f^d_{i,j}(\underline{\mathbf{p}})\right\rfloor$. *(4)*

Therefore

$$\left.\begin{aligned}\dot{\underline{\mathbf{x}}}(t) &= \underline{\underline{\mathbf{A}}}(\underline{\mathbf{p}})\underline{\mathbf{x}}(t)+\underline{\underline{\mathbf{B}}}(\underline{\mathbf{p}})\underline{\mathbf{u}}(t)\\ \underline{\mathbf{y}}(t) &= \underline{\underline{\mathbf{C}}}(\underline{\mathbf{p}})\underline{\mathbf{x}}(t)+\underline{\underline{\mathbf{D}}}(\underline{\mathbf{p}})\underline{\mathbf{u}}(t)\end{aligned}\right\}\Rightarrow \mathbf{M}(\underline{\mathbf{p}})\ .$$

Let us suppose that model $\mathbf{M}(\underline{\mathbf{p}})$ is identified at some certain points $\underline{\mathbf{p}}_v$ ($=1..m$, where m is the words the non- $\mathbf{M}(\underline{\mathbf{p}})$ is known and linearised at points $\underline{\mathbf{p}}_v$.

Definition*: Model point* $\mathbf{M}_v=\mathbf{M}(\underline{\mathbf{p}}_v)$. *Let* $\mathbf{M}_v$ *be the v-th model point of non-linear model M as:*

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

$$=\left.\begin{aligned}\dot{\underline{\mathbf{x}}}(t) &= \underline{\underline{\mathbf{A}}}_v\underline{\mathbf{x}}(t)+\underline{\underline{\mathbf{B}}}_v\underline{\mathbf{u}}(t)\\ \underline{\mathbf{y}}(t) &= \underline{\underline{\mathbf{C}}}_v\underline{\mathbf{x}}(t)+\underline{\underline{\mathbf{D}}}_v\underline{\mathbf{u}}(t)\end{aligned}\right\}\Rightarrow\mathbf{M}_v\quad, \qquad (5)$$

where $\underline{\underline{\mathbf{A}}}_v=\left\lfloor a_{v,i,j}=f^a_{i,j}(\underline{\mathbf{p}}_v)\right\rfloor$, $\underline{\underline{\mathbf{B}}}_v=\left\lfloor b_{v,i,j}=f^b_{i,j}(\underline{\mathbf{p}}_v)\right\rfloor$, $\underline{\underline{\mathbf{C}}}_v=\left\lfloor c_{v,i,j}=f^c_{i,j}(\underline{\mathbf{p}}_v)\right\rfloor$ *and* $\underline{\underline{\mathbf{D}}}_v=\left\lfloor d_{v,i,j}=f^d_{i,j}(\underline{\mathbf{p}}_v)\right\rfloor$.

Having the identified model points $\mathbf{M}_v$ the non-linear model is approximated in the whole parameter space:

$$\left.\begin{aligned}\dot{\underline{\mathbf{x}}}(t) &= \hat{\underline{\underline{\mathbf{A}}}}(\underline{\mathbf{p}})\underline{\mathbf{x}}(t)+\hat{\underline{\underline{\mathbf{B}}}}(\underline{\mathbf{p}})\underline{\mathbf{u}}(t)\\ \underline{\mathbf{y}}(t) &= \hat{\underline{\underline{\mathbf{C}}}}(\underline{\mathbf{p}})\underline{\mathbf{x}}(t)+\hat{\underline{\underline{\mathbf{D}}}}(\underline{\mathbf{p}})\underline{\mathbf{u}}(t)\end{aligned}\right\}\Rightarrow\hat{\mathbf{M}}(\underline{\mathbf{p}})=f\left(\mathbf{M}_1,\cdots,\mathbf{M}_m,\underline{\mathbf{p}}\right)\ .$$

The approximation is done like any function approximation technique since all elements contained in matrices $\underline{\underline{\mathbf{A}}}$, $\underline{\underline{\mathbf{B}}}$, $\underline{\underline{\mathbf{C}}}$ and $\underline{\underline{\mathbf{D}}}$ of model M are functions in space P see (4). According to the introduction let us define the following approximations:

Definition*: Weighted Combination of Model points (WCM). Model approximation based on the weighted combination of given model points. The weighting is determined by weighting functions* $w(.)$ *defined for each given model point.*

$$\hat{\mathbf{M}}(\underline{\mathbf{p}})=\sum_{v=1}^{m}w_v(\underline{\mathbf{p}})\mathbf{M}_v=\sum_{v=1}^{m}w_v(\underline{\mathbf{p}})\mathbf{M}(\underline{\mathbf{p}}_v)\ . \qquad (6)$$

3 Complexity investigation

The main objective of this section is to show that the defined approximation techniques have exponential complexity problem. According to the introduction, the exponential explosion of the complexity is investigated in respect of the number of model points and the number of elements in vectors $\underline{\mathbf{x}}$, $\underline{\mathbf{y}}$ and $\underline{\mathbf{u}}$. Let us omit the calculation effort requirement of add, but consider the product operation in the followings.

Lemma 1*. The calculation complexity of WCM grows exponentially with the number of model points, n_u, n_x and n_y (see 3). The calculation power requirement is proportional to the number of product operation in (3 and 6):*

$$P_{WCM} = m(n_x^2 + n_x n_u + n_y n_x + n_y n_u + T_w), \quad (9)$$

where T_w indicates the calculation effort needed for weighting function $w_t(\underline{\mathbf{p}})$.

4 Reduction

This section is to propose a method capable of generating a minimal form of WCM (6) in the sense that the resultant algorithm utilises the minimal number of basis functions and model points for the same approximation. This section considers only exact reductions. Inexact reduction will be treated at the end of this section. The main concept of the reduction is based on singular value decomposition (SVD) [8]. Methods of numerical computation of SVD can be found in [4,5,6]. Note that, in the following algorithms any kinds of minimal matrix decomposition can be used instead of SVD. The SVD has important role in the case of inexact reduction since it provides the importance of the components. Therefore the following algorithms will be introduced applying SVD. To effectuate an easier understanding, the reduction algorithm will be introduced in two steps. The first outlines an off-line transformation of the whole model, hence, the model points into a common minimal space. The benefit of the transformation is that the calculation of the model $\mathbf{M}^{tr}$ ("tr" means "transformed") in the reduced space requires reduced calculation effort. As a matter of fact, the input values of the model is on-line transformed into the reduced space as well:

$$\underline{\mathbf{x}}^{tr}(t) = \underline{\underline{\mathbf{N}}}_1^{\mathrm{T}} \underline{\mathbf{x}}(t) \quad \text{and} \quad \underline{\mathbf{u}}^{tr}(t) = \underline{\underline{\mathbf{N}}}_2^{\mathrm{T}} \underline{\mathbf{u}}(t),$$

where the sizes of $\underline{\underline{\mathbf{N}}}_1$ and $\underline{\underline{\mathbf{N}}}_2$ are $n_x \times n_x^r$ and $n_u \times n_u^r$, respectively, and $n_x^r \le n_x$, $n_u^r \le n_u$ (subscript "r" means "reduced"). The calculation of $\dot{\underline{\mathbf{x}}}^{tr}(t)$ and $\underline{\mathbf{y}}^{tr}(t)$, hence, is done in the reduced space and finally the output values are transformed into the original space:

$$\dot{\underline{\mathbf{x}}}(t) = \underline{\underline{\mathbf{N}}}_3 \dot{\underline{\mathbf{x}}}^{tr}(t) \quad \text{and} \quad \underline{\mathbf{y}}(t) = \underline{\underline{\mathbf{N}}}_4 \underline{\mathbf{y}}^{tr}(t),$$

where the sizes of $\underline{\underline{\mathbf{N}}}_3$ and $\underline{\underline{\mathbf{N}}}_4$ are $n_x \times n_d^r$ and $n_y \times n_y^r$, respectively, and $n_d^r \le n_x$, $n_y^r \le n_y$. Consequently, the model points $\mathbf{M}_v^{tr}$, hence, model $\mathbf{M}_v$ are defined in the reduced space as:

$$\left.\begin{aligned} \dot{\underline{\mathbf{x}}}(t) &= \underline{\underline{\mathbf{N}}}_3 \left(\underline{\underline{\mathbf{A}}}_v^{tr} \underline{\underline{\mathbf{N}}}_1^{\mathrm{T}} \underline{\mathbf{x}}(t) + \underline{\underline{\mathbf{B}}}_v^{tr} \underline{\underline{\mathbf{N}}}_2^{\mathrm{T}} \underline{\mathbf{u}}(t)\right) \\ \underline{\mathbf{y}}(t) &= \underline{\underline{\mathbf{N}}}_4 \left(\underline{\underline{\mathbf{C}}}_v^{tr} \underline{\underline{\mathbf{N}}}_1^{\mathrm{T}} \underline{\mathbf{x}}(t) + \underline{\underline{\mathbf{D}}}_v^{tr} \underline{\underline{\mathbf{N}}}_2^{\mathrm{T}} \underline{\mathbf{u}}(t)\right) \end{aligned}\right\} \Rightarrow$$

$$\Rightarrow \left.\begin{aligned} \dot{\underline{\mathbf{x}}}^{tr}(t) &= \underline{\underline{\mathbf{A}}}_v^{tr} \underline{\mathbf{x}}^{tr}(t) + \underline{\underline{\mathbf{B}}}_v^{tr} \underline{\mathbf{u}}^{tr}(t) \\ \underline{\mathbf{y}}^{tr}(t) &= \underline{\underline{\mathbf{C}}}_v^{tr} \underline{\mathbf{x}}^{tr}(t) + \underline{\underline{\mathbf{D}}}_v^{tr} \underline{\mathbf{u}}^{tr}(t) \end{aligned}\right\} \Rightarrow \mathbf{M}_v^{tr}, \quad (11)$$

where the sizes of $\underline{\underline{\mathbf{A}}}^{tr}$, $\underline{\underline{\mathbf{B}}}^{tr}$, $\underline{\underline{\mathbf{C}}}^{tr}$ and $\underline{\underline{\mathbf{D}}}^{tr}$ are $n_d^r \times n_x^r$, $n_d^r \times n_u^r$, $n_y^r \times n_x^r$ and $n_y^r \times n_u^r$, respectively. The transformation may lead to calculation reduction regarding Lemma 1 and 2, see (9).

The second step finds the minimal number of model points $\underline{\underline{\mathbf{M}}}_v^r$, $v = 1..m^r$, $m^r \le m$, and their corresponding weighting functions from the transformed model points $\mathbf{M}_v^{tr}$. Let us assume the above discussed reduction steps in the followings:

Theorem 1*: Model points $\mathbf{M}_v$ can always be transformed into a common minimal space as:*

$$\left.\begin{aligned} \dot{\underline{\mathbf{x}}}^{tr}(t) &= \underline{\underline{\mathbf{A}}}_v^{tr} \underline{\mathbf{x}}^{tr}(t) + \underline{\underline{\mathbf{B}}}_v^{tr} \underline{\mathbf{u}}^{tr}(t) \\ \underline{\mathbf{y}}^{tr}(t) &= \underline{\underline{\mathbf{C}}}_v^{tr} \underline{\mathbf{x}}^{tr}(t) + \underline{\underline{\mathbf{D}}}_v^{tr} \underline{\mathbf{u}}^{tr}(t) \end{aligned}\right\} \Rightarrow \mathbf{M}_v^{tr},$$

where the sizes of $\underline{\underline{\mathbf{A}}}^{tr}$, $\underline{\underline{\mathbf{B}}}^{tr}$, $\underline{\underline{\mathbf{C}}}^{tr}$ and $\underline{\underline{\mathbf{D}}}^{tr}$ are $n_d^r \times n_x^r$, $n_d^r \times n_u^r$, $n_y^r \times n_x^r$ and $n_y^r \times n_u^r$, respectively, and $n_x^r \le n_x$, $n_u^r \le n_u$, $n_d^r \le n_x$, $n_y^r \le n_y$. Matrices $\underline{\underline{\mathbf{N}}}_1$, $\underline{\underline{\mathbf{N}}}_2$, $\underline{\underline{\mathbf{N}}}_3$ and $\underline{\underline{\mathbf{N}}}_4$ can *always be found in such way that the result of using models* $\mathbf{M}_v$ *and* $\mathbf{M}_v^{tr}$ *are equivalent if* $\underline{\mathbf{x}}^{tr}(t)=\underline{\underline{\mathbf{N}}}_1^{\mathrm{T}}\underline{\mathbf{x}}(t)$, $\underline{\mathbf{u}}^{tr}(t)=\underline{\underline{\mathbf{N}}}_2^{\mathrm{T}}\underline{\mathbf{u}}(t)$, $\underline{\dot{\mathbf{x}}}(t)=\underline{\underline{\mathbf{N}}}_3\underline{\dot{\mathbf{x}}}^{tr}(t)$ *and* $\underline{\mathbf{y}}(t)=\underline{\underline{\mathbf{N}}}_4\underline{\mathbf{y}}^{tr}(t)$.

Theorem 2*: The function (6) can always be transformed into the following form:*

$$\hat{\mathbf{M}}(\underline{\mathbf{p}})=\sum_{v=1}^{m} w_v(\underline{\mathbf{p}})\mathbf{M}_v=\sum_{v=1}^{m^r} w_v^r(\underline{\mathbf{p}})\mathbf{M}_v^r \qquad (12)$$

where $m^r \le m$, *and* $\underline{\underline{\mathbf{M}}}_v^r$ *is generated form* $\mathbf{M}_v^{tr}$ *, so it still has the reduced size:*

$$\mathbf{M}_v^r \Rightarrow \begin{cases} \underline{\dot{\mathbf{x}}}^{tr}(t)=\underline{\underline{\mathbf{A}}}_v^r\underline{\mathbf{x}}^{tr}(t)+\underline{\underline{\mathbf{B}}}_v^r\underline{\mathbf{u}}^{tr}(t) \\ \underline{\mathbf{y}}^{tr}(t)=\underline{\underline{\mathbf{C}}}_v^r\underline{\mathbf{x}}^{tr}(t)+\underline{\underline{\mathbf{D}}}_v^r\underline{\mathbf{u}}^{tr}(t) \end{cases},$$

where the sizes of $\underline{\underline{\mathbf{A}}}^r$, $\underline{\underline{\mathbf{B}}}^r$, $\underline{\underline{\mathbf{C}}}^r$ *and* $\underline{\underline{\mathbf{D}}}^r$ *are the same as the sizes of* $\underline{\underline{\mathbf{A}}}^{tr}$, $\underline{\underline{\mathbf{B}}}^{tr}$, $\underline{\underline{\mathbf{C}}}^{tr}$ *and* $\underline{\underline{\mathbf{D}}}^{tr}$ *, respectively.*

Proof: The proof of Theorem 1, 2 can be inferred from Method 1, 2 to be introduced in the followings. Method 1 is for finding the minimal space for the model approximation. Method 2 is to define the minimal number of model points and weighting functions. First of all let us define the singular value decomposition [5,6,7]:

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

$$\underline{\underline{\mathbf{B}}}=\underline{\underline{\mathbf{A}}}_1\underline{\underline{\mathbf{D}}}\underline{\underline{\mathbf{A}}}_2^{\mathrm{T}}=[\underline{\underline{\mathbf{A}}}_1^r\,|\,\underline{\underline{\mathbf{A}}}_1^d]\begin{bmatrix}\underline{\underline{\mathbf{B}}}^r & \underline{\underline{\mathbf{O}}}\\ \underline{\underline{\mathbf{O}}} & \underline{\underline{\mathbf{B}}}^d\end{bmatrix}[\underline{\underline{\mathbf{A}}}_2^r\,|\,\underline{\underline{\mathbf{A}}}_2^d]^{\mathrm{T}} \qquad (14)$$

Here, matrix $\underline{\underline{\mathbf{O}}}$ *contains zero elements, matrices* $\underline{\underline{\mathbf{A}}}_1$ *and* $\underline{\underline{\mathbf{A}}}_2$ *are orthogonal, and diagonal matrix* $\underline{\underline{\mathbf{D}}}$ *contains the singular values in decreasing magnitude. The zero or the smallest of singular values (smaller than singular value threshold T0, say) can be discarded to yield a simpler system.*

Matrix $\underline{\underline{\mathbf{B}}}^r$ *thus contains the retained singular values and* $\underline{\underline{\mathbf{B}}}^d$ *contains the discarded ones. The result of SVDR is therefore:* $\hat{\underline{\underline{\mathbf{B}}}}=\underline{\underline{\mathbf{A}}}_1^r\underline{\underline{\mathbf{B}}}^r\underline{\underline{\mathbf{A}}}_2^{r\,\mathrm{T}}=\underline{\underline{\mathbf{A}}}_1^r\underline{\underline{\mathbf{B}}}'$.

Definition*:* $\underline{\underline{\mathbf{H}}}=layout\left(d,\underline{\underline{\mathbf{L}}}_1,\cdots,\underline{\underline{\mathbf{L}}}_z\right)$. *Matrices* $\underline{\underline{\mathbf{L}}}_k$, $k=1..z$, *are n dimensional with the size of* $e_{k,1}\times\cdots\times e_{k,n}$. $\underline{\underline{\mathbf{L}}}_k$ *must hold that* $\forall k: e_{k,d}=e_d$ *.The layout function generates two dimensional matrix* $\underline{\underline{\mathbf{H}}}$ *by placing all the vectors of the d-th dimension of n matrices* $\underline{\underline{\mathbf{L}}}_k$ *as column vector into* $\underline{\underline{\mathbf{H}}}$ *with the size of* e_d. *So, the size of* $\underline{\underline{\mathbf{H}}}=\lfloor\underline{\mathbf{h}}_o\rfloor$ *is* $e_d\times\sum_{k=1}^{z}\prod_{i=1,i\ne d}^{n} e_{k,i}$ *. The ordering of the vectors* $\underline{\mathbf{h}}_o$ *is arbitrary.*

Example: Let us suppose that the size of three dimensional matrices $\underline{\underline{\mathbf{L}}}_k=\lfloor l_{k,r,s,v}\rfloor$, is $(e_1\times e_2\times e_3)$. Let us layout matrices $\underline{\underline{\mathbf{L}}}_k$ for the second dimension: $\underline{\underline{\mathbf{H}}}=layout\left(2,\underline{\underline{\mathbf{L}}}_1,\cdots,\underline{\underline{\mathbf{L}}}_z\right)$. The resulted $\underline{\underline{\mathbf{H}}}_{(e_2\times e_1 e_3)}=\lfloor\underline{\mathbf{h}}_o\rfloor$, $o=1..e_1e_3$, where $\underline{\mathbf{h}}_o=\lfloor h_{o,s}\rfloor$ and $h_{o,s}=l_{k,r,s,v}$, $s=1..e_2$. The ordering of the vectors $\underline{\mathbf{h}}_o$ is arbitrary: $o\Leftrightarrow(k,r,v)$, for instance $o=(k-1)e_1e_3+(r-1)e_3+v$.

Definition*:* $(\underline{\underline{\mathbf{L}}}_1,\cdots,\underline{\underline{\mathbf{L}}}_z)=rebuild\left(\underline{\underline{\mathbf{H}}}\right)$.

Function $rebuild\left(\underline{\underline{\mathbf{H}}}\right)$ *is the inverse of* $\underline{\underline{\mathbf{H}}}=layout\left(d,\underline{\underline{\mathbf{L}}}_1,\cdots,\underline{\underline{\mathbf{L}}}_z\right)$ *and results in matrices* $\underline{\underline{\mathbf{L}}}_k$, $k=1..z$. *It requires the ordering* $o\Leftrightarrow(k,r,v)$ *applied in the layout function. So, function* $rebuild\left(\underline{\underline{\mathbf{H}}}\right)$ *generates n-dimensional matrices* $\underline{\underline{\mathbf{L}}}_k$ *from the column vectors of two dimensional matrix* $\underline{\underline{\mathbf{H}}}=\lfloor\underline{\mathbf{h}}_o\rfloor$ *based on the ordering of the vectors used in* $\underline{\underline{\mathbf{H}}}=layout\left(d,\underline{\underline{\mathbf{L}}}_1,\cdots,\underline{\underline{\mathbf{L}}}_z\right)$.

Example: Let us suppose $\underline{\underline{\mathbf{H}}}_{(e_2 \times e_1 e_3)} = \lfloor \underline{\mathbf{h}}_o \rfloor$ is given, where $\underline{\mathbf{h}}_o = [h_{o,1} \ \cdots \ h_{o,e_2}]^{\mathrm{T}}$. Executing $\underline{\underline{\mathbf{L}}}_k = rebuild(\underline{\underline{\mathbf{H}}})$, $k = 1..z$, with $o \Leftrightarrow (k,r,v)$, matrices $\underline{\underline{\mathbf{L}}}_k = \lfloor l_{k,r,s,v} \rfloor$, are obtained, where $l_{k,r,s,v} = h_{o,s}$.

***Definition**:* $(\underline{\underline{\mathbf{N}}}, \underline{\underline{\mathbf{L}}}_1^r, \cdots, \underline{\underline{\mathbf{L}}}_z^r) = reduct(d, \underline{\underline{\mathbf{L}}}_1, \cdots, \underline{\underline{\mathbf{L}}}_z)$. *This function reduces the size of n-dimensional* $e_{k,1} \times \cdots \times e_{k,n}$ *sized matrices* $\underline{\underline{\mathbf{L}}}_k$, $k = 1..z$, *in d-th dimension.* $\underline{\underline{\mathbf{L}}}_k$ *must hold that* $\forall k : e_{k,d} = e_d$. *The results of function* $reduct(d, \underline{\underline{\mathbf{L}}}_1, \cdots, \underline{\underline{\mathbf{L}}}_z)$ *are matrices* $\underline{\underline{\mathbf{N}}}$ *and* $\underline{\underline{\mathbf{L}}}_1^r, \cdots, \underline{\underline{\mathbf{L}}}_z^r$ *("r" denotes reduced), where the sized of* $\underline{\underline{\mathbf{N}}}$ *is* $e_d \times e_d^r$, $e_d^r \le e_d$ *and the size of* $\underline{\underline{\mathbf{L}}}_k^r$ *are* $c_{k,1} \times \cdots \times c_{k,n}$, *where* $\forall k, i, i \ne d : c_{k,i} = e_{k,i}$ *and* $\forall k : c_{k,d} = e_d^r$. *Let us define the algorithm for* $reduct(d, \underline{\underline{\mathbf{L}}}_1, \cdots, \underline{\underline{\mathbf{L}}}_z)$ *as:*

1) $\underline{\underline{\mathbf{H}}} = layout(d, \underline{\underline{\mathbf{L}}}_1, \cdots, \underline{\underline{\mathbf{L}}}_z)$, *the size of* $\underline{\underline{\mathbf{H}}}$ *is* $e_d \times \sum_{k=1}^{z} \prod_{i=1, i \ne d}^{n} e_{k,i}$;

2) Applying SVDR on matrix $\underline{\underline{\mathbf{H}}}$ *the following reduction is obtained:* $\hat{\underline{\underline{\mathbf{H}}}} = \underline{\underline{\mathbf{N}}}\underline{\underline{\mathbf{H}}}'$ *; where the size of* $\underline{\underline{\mathbf{N}}}$ *and* $\underline{\underline{\mathbf{H}}}'$ *are* $e_d \times e_d^r$ *and* $e_d^r \times \sum_{k=1}^{z} \prod_{i=1, i \ne d}^{n} e_{k,i}$, *respectively, where* $e_d^r \le e_d$.

3) Matrices $\underline{\underline{\mathbf{L}}}_k^r$, $k = 1..z$, *are determined by function* $rebuild(\underline{\underline{\mathbf{H}}}')$.

The main goal of this Method 1 is to find the transformed form that is defined in Theorem 1 see (11). For convenient notation let us form three dimensional matrix: $\underline{\underline{\mathbf{A}}} = \lfloor a_{v,i,j} \rfloor$, from matrices $\underline{\underline{\mathbf{A}}}_v = \lfloor a_{v,i,j} \rfloor$ defined in (5). In the same way let us define three dimensional matrices $\underline{\underline{\mathbf{B}}}$, $\underline{\underline{\mathbf{C}}}$ and $\underline{\underline{\mathbf{D}}}$.
Finding common minimal space for the model points (algorithm for Theorem 1).

Method 1: I) Determination of $\underline{\underline{\mathbf{N}}}_1$ and $\underline{\underline{\mathbf{N}}}_2$:
Let $(\underline{\underline{\mathbf{N}}}_1, \underline{\underline{\mathbf{A}}}', \underline{\underline{\mathbf{C}}}') = reduct(3, \underline{\underline{\mathbf{A}}}, \underline{\underline{\mathbf{C}}})$. The sizes of $\underline{\underline{\mathbf{N}}}_1$, $\underline{\underline{\mathbf{A}}}'$ and $\underline{\underline{\mathbf{C}}}'$ are $n_x \times n_x^r$, $m \times n_x \times n_x^r$ and $m \times n_y \times n_x^r$, respectively.

Let $(\underline{\underline{\mathbf{N}}}_2, \underline{\underline{\mathbf{B}}}', \underline{\underline{\mathbf{D}}}') = reduct(3, \underline{\underline{\mathbf{B}}}, \underline{\underline{\mathbf{D}}})$. The sizes of $\underline{\underline{\mathbf{N}}}_2$, $\underline{\underline{\mathbf{B}}}'$ and $\underline{\underline{\mathbf{D}}}'$ are $n_u \times n_u^r$, $m \times n_x \times n_u^r$ and $m \times n_y \times n_u^r$, respectively.

II) Determination of $\underline{\underline{\mathbf{N}}}_3$ and $\underline{\underline{\mathbf{N}}}_4$:
Let $(\underline{\underline{\mathbf{N}}}_3, \underline{\underline{\mathbf{A}}}^{tr}, \underline{\underline{\mathbf{B}}}^{tr}) = reduct(2, \underline{\underline{\mathbf{A}}}', \underline{\underline{\mathbf{B}}}')$. The sizes of $\underline{\underline{\mathbf{N}}}_3$, $\underline{\underline{\mathbf{A}}}^{tr}$ and $\underline{\underline{\mathbf{B}}}^{tr}$ are $n_x \times n_d^r$, $m \times n_d^r \times n_x^r$ and $m \times n_d^r \times n_u^r$, respectively.

Let $(\underline{\underline{\mathbf{N}}}_4, \underline{\underline{\mathbf{C}}}^{tr}, \underline{\underline{\mathbf{D}}}^{tr}) = reduct(2, \underline{\underline{\mathbf{C}}}', \underline{\underline{\mathbf{D}}}')$. The sizes of $\underline{\underline{\mathbf{N}}}_4$, $\underline{\underline{\mathbf{C}}}^{tr}$ and $\underline{\underline{\mathbf{D}}}^{tr}$ are $n_y \times n_y^r$, $m \times n_y^r \times n_x^r$ and $m \times n_y^r \times n_u^r$, respectively.

Note that $n_d^r \le n_x$, $n_x^r \le n_x$, $n_y^r \le n_y$ and $n_u^r \le n_u$ that is in full accordance with Theorem 1.

The elements of two-dimensional matrices $\underline{\underline{\mathbf{A}}}^{tr}_{v,(n_d^r \times n_x^r)} = \lfloor a^{tr}_{v,i,j} \rfloor$ defined in $\mathbf{M}_v^{tr}$, see (11), are determined from $\underline{\underline{\mathbf{A}}}^{tr} = \lfloor a^{tr}_{v,i,j} \rfloor$. Matrices $\underline{\underline{\mathbf{B}}}^{tr}_{v,(n_d^r \times n_u^r)}$, $\underline{\underline{\mathbf{C}}}^{tr}_{v,(n_y^r \times n_x^r)}$ and $\underline{\underline{\mathbf{D}}}^{tr}_{v,(n_d^r \times n_x^r)}$ are defined in the same way.

Finding the minimal number of model in the minimal space for WCM (Algorithm for Theorem 2).

Method 2: For this purpose let $(\underline{\underline{\mathbf{T}}}, \underline{\underline{\mathbf{A}}}^r, \underline{\underline{\mathbf{B}}}^r, \underline{\underline{\mathbf{C}}}^r, \underline{\underline{\mathbf{D}}}^r) = reduct(1, \underline{\underline{\mathbf{A}}}^{tr}, \underline{\underline{\mathbf{B}}}^{tr}, \underline{\underline{\mathbf{C}}}^{tr}, \underline{\underline{\mathbf{D}}}^{tr})$.

The size of matrix $\underline{\underline{\mathbf{T}}}$ is $m \times m^r$. The size of matrices $\underline{\underline{\mathbf{A}}}^r, \underline{\underline{\mathbf{B}}}^r, \underline{\underline{\mathbf{C}}}^r$ and $\underline{\underline{\mathbf{D}}}^r$ are $m^r \times n_d^r \times n_x^r$, $m^r \times n_d^r \times n_u^r$, $m^r \times n_y^r \times n_x^r$ and $m^r \times n_y^r \times n_u^r$, respectively.

Functions $w_t^r(\underline{\mathbf{p}})$ are transformed from $w_t(\underline{\mathbf{p}})$ by $\underline{\underline{\mathbf{T}}}$ as:

$$\left[w_1^r(\underline{\mathbf{p}}) \quad \cdots \quad w_{m^r}^r(\underline{\mathbf{p}}) \right] = \left[w_1(\underline{\mathbf{p}}) \quad \cdots \quad w_m(\underline{\mathbf{p}}) \right] \underline{\underline{\mathbf{T}}} \qquad (15)$$

Consequently, the number of model points and weighting functions are reduced: $m^r \le m$.

The elements of two-dimensional matrices $\underline{\underline{\mathbf{A}}}^r_{v,(n_d^r \times n_x^r)} = \left\lfloor a^r_{v,i,j} \right\rfloor$ defined in $\mathbf{M}_v^r$, see (12), are defined from $\underline{\underline{\mathbf{A}}}^r = \left\lfloor a^r_{v,i,j} \right\rfloor$. Matrices $\underline{\underline{\mathbf{B}}}^r_{v,(n_d^r \times n_u^r)}$, $\underline{\underline{\mathbf{C}}}^r_{v,(n_y^r \times n_x^r)}$ and $\underline{\underline{\mathbf{D}}}^r_{v,(n_d^r \times n_x^r)}$ are defined in the same way.

If non-zero singular values are discarded then the reduced model approximation differs from the original one, however, the effectiveness of the reduction is increased. This difference is the reduction error. In order to save calculation effort, it would be useful if the final reduction error could be estimated during the reduction process. Therefore an error controllable reduction technique is highly desirable. This motivates the main objective of our next publication that is to show the error bound of the reduction algorithms. The key point of the error estimation is based on the sum of the discarded singular values.

5 Conclusion

The complexity problem of non-linear model approximations techniques motivates the algorithms proposed in this paper. These algorithms utilize SVD that helps with defining the importance of the model components, which fact offers the forming of an error controllable inexact reduction. The algorithms are used off-line to compress the information in the models, hence, decrease the dimensionality. The compressed form is used on-line in the applications.

References:

[1] Yeung Yam: “Fuzzy Approximation via Grid Point Sampling and Singular Value Decomposition” IEEE Tr. S.M.C. Vol. 27, No. 6, 1997, pp. 933-951.

[2] L.Wang, R. Langari, and J.Yen ”Principal components, B-splines, and fuzzy systems reduction” in Fuzzy Logic for the applications to Complex systems, W. Chiang and J. Lee, Eds. Singapure: World Scientific, 1996, pp. 255-259.

[3] M. Mizumoto, “ Fuzzy controls by product-sum-gravity method” Advancement of Fuzzy Thory and Systems in China and Japan, Eds. Liu and Mizumoto, International Academic Publishers, c1.1.-c.1.4. 1990

[4] G.H.Golub and C.F.Van Loan “Matrix Computations” 2nd ed., The John Hopkins University press, baltimore, MD, 1989

[5] J.J. Dongara, C.B. Moler, J.R. Bunch and G.W. Stewart “Linpack Users’ Guide” SIAM, 1979.

[6] B.S. Garbow, J.M. Boyle, J.J. Dongara and C.B. Moler “Matrix Eigensystem Routines-Eispack Guide Extension, Springer, new York, 1993.

[7] J.O. Ramsay and B.W. Silverman “Functional Data Analaysis” Springer series in Statistic. Springer ISBN 0-387-94956-9

[8] Chen J & Patton R J, Robust Model Based Fault Diagnosis For Dynamic Systems, Kluwer Academic Publishers, 1999, ISBN 0-7923-8411-3