

Dimensionality Reduction of Volterra Kernels by Tensor Decomposition using Higher-Order SVD

Urszula Libal, John S. Baras and Karl H. Johansson

Abstract—The paper proposes a practical method for a significant dimensionality reduction of Volterra kernels, defining a discrete nonlinear model of a signal by Volterra series of higher order. In system identification of Volterra series, the Volterra kernels and nonlinear inputs of the system can be described by super-symmetrical tensors. The reduction of their dimensionality is obtained by a tensor decomposition technique called Higher Order Singular Value Decomposition (HOSVD). The main contribution of the paper is a cascade learning algorithm for the system identification based on residuals of least squares minimization. Numerical examples for Volterra system of order four are used to illustrate the approach.

I. INTRODUCTION

A. Motivation

Volterra series and Wiener series are one of the most universal nonlinear system models [15]. At the same time, the Volterra series model has a simple homogeneous power series structure similar to Taylor series expansion (albeit in function space) and a more compact form comparing to the Wiener series. The resulting Volterra kernels, included in the model, are tensors of increasing ranks. The Volterra system model of order M and memory N is fully described by a set of kernels, which are tensors of sizes: 1 (scalar), N (vector), N^2 (matrix), N^3 (cube), N^4 (4-way tensor), and so on till a tensor of size N^M (M -way tensor of dimension N). That causes the need for lowering the dimensionality of the higher-order tensors incorporated in the model, to reduce the number of coefficients we need to identify. Reducing the complexity of the system model is of paramount importance to the design of controllers and control algorithms. This is even more important for nonlinear systems as well as for systems that include learning components, like for example in adaptive control. Reducing the dimension of the kernels in the Volterra series model is a much better approximation than just using the associated functional series expansion in function space, because like in the simpler linear case, the dimensionality reduction implies reduction in the structure of the nonlinear system, like which inputs affect more significantly certain outputs, that cannot be captured by just the norm of the input function (which underlies the truncation of

the Taylor series). In the easier linear case eigenvalue- and/or singular value decomposition can be directly associated with reduced models based on keeping the most significant input-output subspaces rather than relying on the norm size of the input vector. Thus the aim of the paper is to develop a methodology for more meaningful reduction of nonlinear system models based on a structural decomposition of the input output relations based on keeping the most significant such relations as revealed by the proposed singular value decomposition of the associated Volterra series kernels.

B. Related Work

Chronologically, the oldest identification scheme is cross-correlation method [10], [13], [14], using the property that every Volterra functional is orthogonal to all Wiener functionals [21] of higher orders. The discrete analogue of the Wiener series is orthogonalized for Gaussian white noise input with zero mean [12]. This assumption not always can be fulfilled due to limitations of applications. The new approach to Wiener and Volterra system identification was a polynomial kernel regression [3], [4]. The other modification of the original cross-correlation method [10] was based on the idea of differentiating the variance of the kernels for various orders, called multiple-variance method [11].

The Fast Orthogonal Algorithm and the Exact Orthogonal Algorithm [9] also perform orthogonalization, but allow for arbitrary system inputs. There are also attempts to solve the identification problem by statistical learning approaches and kernel methods [5], based on minimizing the empirical error.

For the problem of lowering the dimension of higher-order tensors multiple solutions have been developed, leading to tensor decompositions. Currently the most popular tensor decomposition method is the so called Canonical Polyadic Decomposition (CPD) [7]. This decomposition was named also PARAFAC [6], CANDECOMP [1] and Tensor Rank Decomposition. The method represents tensors ${}^m\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times \dots \times N_m}$ of order m as a linear combination of r rank-1 tensors (vectors)

$${}^m\mathcal{T} = \sum_{i=1}^r \lambda_i \mathbf{a}_i^{(1)} \otimes \mathbf{a}_i^{(2)} \otimes \dots \otimes \mathbf{a}_i^{(m)}, \quad (1)$$

where $\lambda_i \in \mathbb{R}$ is a scalar, and factor matrices $\mathbf{a}_i^{(m)} \in \mathbb{R}^{N_m}$.

The oldest tensor decomposition method is the Tucker Decomposition [17], [18], [19], which was designed to decompose a tensor ${}^3\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ of order 3 in the following way

$${}^3\mathcal{T} = (U_1, U_2, U_3) \otimes {}^3\mathcal{S} \quad (2)$$

The authors are with Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm 100 44, Sweden. Email: {urli, baras, kallej}@kth.se

U. Libal is also with the Signal Processing Systems Department, Wroclaw University of Science and Technology, Wroclaw 50-370, Poland. Email: urszula.libal@pwr.edu.pl

J. S. Baras is also with the Institute of Systems Research and the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742, USA. Email: baras@isr.umd.edu

to a core tensor ${}^3\mathcal{S} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and orthogonal factor matrices U_1, U_2 and U_3 of matching sizes. Its generalization for tensors ${}^m\mathcal{T} \in \mathbb{R}^{N_1 \times N_2 \times \dots \times N_m}$ of order m higher than 3 is called Higher Order Singular Value Decomposition (HOSVD) [2], and has an analogous form

$${}^m\mathcal{T} = (U_1, U_2, \dots, U_m) \otimes {}^m\mathcal{S}. \quad (3)$$

The last method is explained in detail in later sections of the paper.

C. Main Contribution

The main contribution of this paper is a sequential learning algorithm for a nonlinear system identification based on a Volterra series model. The Volterra series model of higher-order can be described by kernels in the form of super-symmetric tensors. Using that specific representation of Volterra kernels, we were able to modify the standard Higher-Order Singular Value Decomposition (HOSVD) algorithm. The new approach decreases the number of calculations necessary to perform HOSVD of m -way super-symmetric tensors by m -times. The presented identification schema utilizes HOSVD, which is reducing tensor dimensions for kernels of order $m = 3$ and higher at each step of the multistage procedure.

The paper is divided into eight sections. In Section II the most popular definitions of nonlinear Volterra series system models are introduced in two cases: a full model of higher order with kernels as super-symmetric tensors, and a reduced lower triangular model with a cropped range of indexes. In Section III we present a sketch of the cascade learning algorithm for system identification. The details of the HOSVD in the general case and for super-symmetric tensors are described in Section IV. The practical cascade algorithm for identification of Volterra series system model of higher-order with parallel reduction of dimensionality of particular tensors by the HOSVD is proposed in Section V. The space complexity of the Volterra kernel and its decomposed version is analyzed in Section VI. At the end of the paper, we describe our simulation results and conclusions.

II. NONLINEAR VOLTERRA SERIES MODELS

The Volterra series represents one of the earliest nonlinear system model dated back to the end of XIX century [20]. It can be interpreted as a polynomial representation of a system (in input-output function spaces) with increasing degree of nonlinearity, similar to a power series expansion by the Taylor series (for functions), but with memory effect. The first-order Volterra series is just a linear approximation of the input-output map characterized by a typical transfer function or impulse response of the linear systems.

The models, approach and algorithms of this paper can be extended to handle multi-input multi-output nonlinear systems. In the interest of simplicity, especially since the indexing becomes much more complex, we investigate in this paper single input single output systems, with excitation x_t and response y_t at discrete time instances t . We define

the discrete Volterra series system model ${}^M y_t$ of order M and finite memory N in the following way:

$$\begin{aligned} {}^M y_t &= h_0^x + \sum_{i_1=1}^N h_1^x(i_1) x_{t-i_1} \\ &+ \sum_{i_1=1}^N \sum_{i_2=1}^N h_2^x(i_1, i_2) x_{t-i_1} x_{t-i_2} \\ &+ \dots \\ &+ \sum_{i_1=1}^N \dots \sum_{i_M=1}^N h_M^x(i_1, \dots, i_M) x_{t-i_1} \dots x_{t-i_M}. \end{aligned} \quad (4)$$

We should mention that the constant $h_0^x = \mathbf{E}y_t$ (in the stochastic input case) can be easily estimated, and is often removed from the assumed models for simplicity. The main concern of this paper is dimensionality reduction connected to memory N . It means that the response of the system uses N past excitations $x_{t-1}, x_{t-2}, \dots, x_{t-N}$. The multi-dimensional transfer functions $h_1^x(i_1), h_2^x(i_1, i_2), \dots, h_M^x(i_1, \dots, i_M)$ are called the Volterra kernels of order $1, 2, \dots, M$, respectively – see Fig. 1. Each of the kernels $h_m^x(i_1, \dots, i_m)$ is a super-symmetrical m -way tensor of dimension N , producing the same tensor for all possible index permutations. The total number of tensor elements is equal to

$$\underbrace{N \times \dots \times N}_{m \text{ times}} = N^m,$$

which might be a huge number, depending on the memory size N and kernel order m .

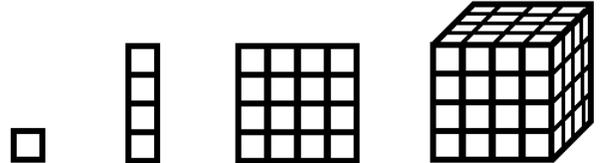


Fig. 1. The Volterra kernels of orders $m = 0, 1, 2, 3$ and dimension $N = 4$ are: a scalar h_0^x , a vector ${}^1\mathcal{H}_x$, a matrix ${}^2\mathcal{H}_x$, and a cube ${}^3\mathcal{H}_x$

Due to the multiplication alternation, a large part of the sum elements reoccurs. A common approach is to use instead reduced lower triangular kernels $h_m^{x,\Delta}(i_1, \dots, i_m)$ with a clipped range of indexes. The reduced model has the following form:

$$\begin{aligned} {}^M y_t &= h_0^{x,\Delta} + \sum_{i_1=1}^N h_1^{x,\Delta}(i_1) x_{t-i_1} \\ &+ \sum_{i_1=1}^N \sum_{i_2=i_1}^N h_2^{x,\Delta}(i_1, i_2) x_{t-i_1} x_{t-i_2} \\ &+ \dots \\ &+ \sum_{i_1=1}^N \dots \sum_{i_M=i_{M-1}}^N h_M^{x,\Delta}(i_1, \dots, i_M) x_{t-i_1} \dots x_{t-i_M}. \end{aligned} \quad (5)$$

In order to preserve the super-symmetry of the kernel tensors, we will consider in the following only the model (4).

III. IDENTIFICATION SCHEME

We denote the Volterra series of order M as a sum of $M + 1$ components, to simplify the notation. Let

$$\begin{aligned} {}^M y_t &= \sum_{m=0}^M y_t^{(m)} \\ &= h_0^x + \sum_{m=1}^M \sum_{i_1=1}^N \cdots \sum_{i_m=1}^N h_m^x(i_1, \dots, i_m) x_{t-i_1} \cdots x_{t-i_m} \\ &= h_0^x + \sum_{m=1}^M \langle {}^m \mathcal{H}_x, {}^m \mathcal{X} \rangle_F, \end{aligned} \quad (6)$$

where $\langle \cdot, \cdot \rangle_F$ denotes the Frobenius inner product of two tensors of the same size. For a non-reduced model (4), the Volterra kernel of order m is a m -way tensor of dimension N , or simply a Dm array of N^m cells

$${}^m \mathcal{H}_x = (h_{m; i_1, \dots, i_m}^x)_{i_1, \dots, i_m=1}^{N, \dots, N} \in \mathbb{R}^{N^m}, \quad (7)$$

where all indexes i_1, \dots, i_m run from 1 till N . The m -th degree homogeneous functional of input observations x_{t-i} creates the m -way super-symmetric tensor

$${}^m \mathcal{X} = \mathbf{x}^{\otimes m} = (x_{t-i_1} \cdots x_{t-i_m})_{i_1, \dots, i_m=1}^{N, \dots, N} \in \mathbb{R}^{N^m}. \quad (8)$$

The subscript $\otimes m$ denotes the symmetric outer product, as defined in [16].

The identification scheme is a sequential MSE procedure. We minimize the following residuals for the initial steps

$$r_t^{(1)} = y_t - y_t^{(0)} = y_t - h_0^x, \quad (9)$$

$$r_t^{(2)} = y_t - y_t^{(0)} - y_t^{(1)} = r_t^{(1)} - \langle {}^1 \mathcal{H}_x, {}^1 \mathcal{X} \rangle_F, \quad (10)$$

$$r_t^{(3)} = y_t - y_t^{(0)} - y_t^{(1)} - y_t^{(2)} = r_t^{(2)} - \langle {}^2 \mathcal{H}_x, {}^2 \mathcal{X} \rangle_F, \quad (11)$$

and generally

$$r_t^{(m+1)} = r_t^{(m)} - y_t^{(m)}, \quad (12)$$

where

$$\begin{aligned} y_t^{(m)} &= \langle {}^m \mathcal{H}_x, {}^m \mathcal{X} \rangle_F \\ &= \sum_{i_1=1}^N \cdots \sum_{i_m=1}^N h_m^x(i_1, \dots, i_m) x_{t-i_1} \cdots x_{t-i_m}. \end{aligned} \quad (13)$$

The identification of the Volterra series system model of order M in a sequential way, defined by (9)-(12), leads to minimizing the residuals in $(M + 1)$ -steps. In each step, we obtain the kernel estimates: a constant scalar \hat{h}_0^x in the first step, a vector ${}^1 \hat{\mathcal{H}}_x$ in the second step, a matrix ${}^2 \hat{\mathcal{H}}_x$, a cube ${}^3 \hat{\mathcal{H}}_x$, and a 4-way tensor ${}^4 \hat{\mathcal{H}}_x$ in following steps, and so on.

IV. HIGHER-ORDER SINGULAR VALUE DECOMPOSITION

A. HOSVD Algorithms

Algorithm 1 presents the proposed general procedure of reducing the dimension of an m -way tensor ${}^m \mathcal{T}$, without necessarily preserving the property of super-symmetry.

In the super-symmetric case, every unfolding (flattening) of a tensor ${}^m \mathcal{T}$ to a matrix A_j gives the same matrix of

Algorithm 1 HOSVD of m -way tensor ${}^m \mathcal{T}$

Input: ${}^m \mathcal{T} \in \mathbb{R}^{N^m}$

Output: ${}^m \hat{\mathcal{T}} \in \mathbb{R}^{N^m}$ and ${}^m \mathcal{S} \in \mathbb{R}^{K_1 \times K_2 \times \cdots \times K_m}$

LOOP

1: **for** $j = 1$ to m **do**

2: Unfold the tensor ${}^m \mathcal{T}(i_1, \dots, i_m)$ to a matrix A_j with N rows and $N^{(m-1)}$ columns, where the j th tensor's index i_j becomes the row index of the matrix.

3: Perform SVD of A_j , keep the left-hand side orthogonal matrix U_j and matrix Σ_j with singular values $\sigma_j = (\sigma_{j,1}, \dots, \sigma_{j,N})$ on the diagonal in descending order, $\Sigma_j \leftarrow \text{diag}(\sigma_j)$.

4: Number of remaining dimensions $K_j \leftarrow N$.

LOOP

5: **for** $k = 1$ to N **do**

6: **if** $(\sigma_{j,k} \leq \epsilon)$ **then**

7: Number of remaining dimensions $K_j \leftarrow K_j - 1$.

8: **end if**

9: **end for**

10: Truncate $U_j \leftarrow U_j(:, 1 : K_j)$, dropping all singular values $\sigma_{j,k} \leq \epsilon$.

11: **end for**

12: Calculate m -way tensor $\mathcal{U} \leftarrow (U_1, U_2, \dots, U_m)$.

13: Calculate core tensor ${}^m \mathcal{S} \leftarrow \mathcal{U}^T \otimes {}^m \mathcal{T}$.

14: Calculate tensor estimate ${}^m \hat{\mathcal{T}} \leftarrow \mathcal{U} \otimes {}^m \mathcal{S}$.

15: **return** ${}^m \hat{\mathcal{T}}$ and ${}^m \mathcal{S}$.

size $N \times N^{(m-1)}$. So for every choice of the tensor index $j \in \{1, 2, \dots, m\}$, which will become the row index of the matrix during the flattening of the tensor, we obtain the same matrix A_j . Then, we can reduce the number of calculations by removing the outer loop in Algorithm 1. As a result, the calculations of the HOSVD for an m -way tensor ${}^m \mathcal{T}$ are m -times faster.

Algorithm 2 was used by us to calculate the HOSVD of an m -way super-symmetric tensor ${}^m \mathcal{T} = \mathbf{x}^{\otimes m}$, obtained from tensor product of delayed system inputs $\{x_{t-1}, x_{t-2}, \dots, x_{t-N}\}$. The initial dimension of tensor $\mathbf{x}^{\otimes m}$ was N , and after decomposition and removing small singular values, its dimension is lowered to K . K is the number of left (not deleted) singular values on the diagonal of the matrix Σ . As a result, the initial tensor ${}^m \mathcal{T}$ has N^m cells, and the recovered tensor ${}^m \hat{\mathcal{T}}$ has also N^m cells.

During the procedure we obtain the reduced core tensor ${}^m \mathcal{S} = \mathbf{s}^{\otimes m}$, which is also an m -way super-symmetric tensor, but of lower dimension K ($K \leq N$), and it has only K^m cells. The dimension is not reduced (i.e. it is $K = N$), only if there were no zero singular values and no non-zero singular values were dropped after decomposition.

B. Singular Values and Reconstruction Error

The matrix A_j (a flattened version of the tensor) is factorized by Singular Value Decomposition (SVD) to the form

$$A_j = U_j \Sigma_j V_j^T. \quad (14)$$

Algorithm 2 HOSVD of m -way super-symmetric tensor ${}^m\mathcal{T}$

Input: ${}^m\mathcal{T} \in \mathbb{R}^{N^m}$ **Output:** ${}^m\hat{\mathcal{T}} \in \mathbb{R}^{N^m}$ and ${}^m\mathcal{S} \in \mathbb{R}^{K^m}$

- 1: Unfold the tensor ${}^m\mathcal{T}(i_1, \dots, i_m)$ to a matrix A with N rows and $N^{(m-1)}$ columns, where the 1st tensor's index i_1 becomes the row index of the matrix.
 - 2: Perform SVD of A , keep the left-hand side orthogonal matrix U_1 and matrix Σ with singular values $\sigma = (\sigma_1, \dots, \sigma_N)$ on the diagonal in descending order, $\Sigma \leftarrow \text{diag}(\sigma)$.
 - 3: Number of remaining dimensions $K \leftarrow N$.
LOOP
 - 4: **for** $k = 1$ to N **do**
 - 5: **if** $(\sigma_k \leq \epsilon)$ **then**
 - 6: Number of remaining dimensions $K \leftarrow K - 1$.
 - 7: **end if**
 - 8: **end for**
 - 9: Truncate $U_1 \leftarrow U_1(:, 1 : K)$, dropping all singular values $\sigma_i \leq \epsilon$.
 - 10: Calculate m -way tensor $\mathcal{U} \leftarrow (U_1, U_1, \dots, U_1)$.
 - 11: Calculate core tensor ${}^m\mathcal{S} \leftarrow \mathcal{U}^T \otimes {}^m\mathcal{T}$.
 - 12: Calculate tensor estimate ${}^m\hat{\mathcal{T}} \leftarrow \mathcal{U} \otimes {}^m\mathcal{S}$.
 - 13: **return** ${}^m\hat{\mathcal{T}}$ and ${}^m\mathcal{S}$.
-

The diagonal matrix Σ_j consists of the singular values in descending order. Let us drop all singular values σ_k smaller than given ϵ . All corresponding columns of matrix U_j and rows of Σ_j are deleted. Then, the error between A_j and the reconstructed \hat{A}_j is equal to

$$\text{err}_\epsilon = \sum_{k \in \{\sigma \leq \epsilon\}} \sigma_k^2 < \#\{\sigma \leq \epsilon\} \cdot \epsilon^2, \quad (15)$$

where the number of the deleted singular values is

$$\#\{\sigma \leq \epsilon\} = N - K, \quad (16)$$

while the number of the remaining singular values is K . In the application of HOSVD presented in the paper for the Volterra kernels ${}^m\mathcal{H}$, the number K is becoming a reduced dimension of the core tensor ${}^m\mathcal{S}$.

C. Super-symmetric Tensor Case

The m -th degree homogeneous functional ${}^m\mathcal{X} = \mathbf{x}^{\otimes m}$ is an m -way super-symmetric tensor [8]. The consequence is that also the corresponding Volterra kernel of order m has to be super-symmetric, and we denote it as ${}^m\mathcal{H}_x = \mathbf{h}^{\otimes m}$. Using the HOSVD, we can transform tensor $\mathbf{x}^{\otimes m}$ to a super-symmetric core tensor $\mathbf{s}^{\otimes m}$ by calculating the tensor product of the tensors $\mathbf{x}^{\otimes m}$ and \mathcal{U}^T . The reason to perform HOSVD is dimensionality reduction of a tensor with accompanying small estimation error due to dropping some of the singular

values. We have

$${}^M y_t = h_0^x + \sum_{m=1}^M < {}^m \mathcal{H}_x, {}^m \mathcal{X} >_F \quad (17)$$

$$= h_0^x + \sum_{m=1}^M < \mathbf{h}_x^{\otimes m}, \mathbf{x}^{\otimes m} >_F \quad (18)$$

$$\approx h_0^x + \sum_{m=1}^M < \mathbf{h}_s^{\otimes m}, \mathbf{s}^{\otimes m} >_F \quad (19)$$

$$= h_0^x + \sum_{m=1}^M < {}^m \mathcal{H}_s, {}^m \mathcal{S} >_F \quad (20)$$

All orthogonal matrices U_j , $j = 1, 2, \dots, m$, are linear transformations, and preserve inner product. For super-symmetric tensors, all the left-hand side matrices U_j of the SVD of matrix A_j are identical. We mentioned before that all flattened versions A_j of super-symmetric tensor produce always the same matrix. In case, we do not drop any singular values, the Frobenius inner product will retain the same value

$$< \mathbf{h}_x^{\otimes m}, \mathbf{x}^{\otimes m} >_F = < \mathbf{h}_s^{\otimes m}, \mathbf{s}^{\otimes m} >_F \quad (21)$$

for $\mathbf{s} = U_1^T \mathbf{x}$ and $\mathbf{h}_s = U_1^T \mathbf{h}_x$.

V. VOLTERRA KERNELS DIMENSIONALITY REDUCTION

We present a cascade learning algorithm (see Algorithm 3) performing system identification of Volterra series of a fixed order M and memory N using a set of input and output signals $\{x_{t-1}, x_{t-2}, \dots, x_{t-N}; y_t\}$ for $t = 1, \dots, T$. The whole procedure is performed in $(M+1)$ -steps. In the initial three steps, kernels of lower orders $m = 0, 1$ and 2 are recovered without any reduction of kernel size: in the first step – a constant \hat{h}_0^x , in the second step – a vector ${}^1\hat{\mathcal{H}}_x$, and in the third step – a matrix ${}^2\hat{\mathcal{H}}_x$. The proposed method is based on dimensionality reduction by the HOSVD of the tensors of order $m = 3$ and higher, described in the previous section. It is worth noting that what is called in literature the dimension N of a tensor is, in our case, the memory N of the Volterra series system model.

The dimensionality reduction is firstly performed for an m -way super-symmetric tensor ${}^m\mathcal{X} = \mathbf{x}^{\otimes m}$ of dimension N , resulting from the use of delayed inputs measurements $\{x_{t-1}, x_{t-2}, \dots, x_{t-N}\}$, leading to an m -th degree homogeneous functional. The HOSVD allows for dropping singular values of zero or near-zero values as in standard SVD, and the number K of not-rejected singular values is the new dimension of the decomposed and reduced tensor ${}^m\hat{\mathcal{X}}$. The size of the tensor ${}^m\hat{\mathcal{X}}$ induces the identical size of the corresponding Volterra kernel tensor ${}^m\mathcal{H}_x$. So as a result, the reduction is done for both tensors at each step. Next, using the least squares technique, we estimate the corresponding lower-dimensional kernel ${}^m\hat{\mathcal{H}}_x$, according to the equation

$$r_t^{(m+1)} = r_t^{(m)} - < {}^m \mathcal{H}_x, {}^m \hat{\mathcal{X}} >_F. \quad (22)$$

In the next step, we use the residual $r_t^{(m+1)}$ obtained as a result of the minimization in previous step.

Algorithm 3 Volterra kernels dimensionality reduction

Input: inputs $x_{t-1}, x_{t-2}, \dots, x_{t-N}$ and outputs y_t for $t = 1, \dots, T$

Output: the Volterra kernels ${}^m\mathcal{H}_x$ for $m = 0, 1, \dots, M$ (Order 0)

- 1: Calculate $h_0^x \leftarrow \mathbf{E}y_t$.
 - 2: Calculate residual $r_t^{(1)} \leftarrow y_t - h_0^x$. (Order 1)
 - 3: Prepare ${}^1\mathcal{X} \leftarrow \mathbf{x}$.
 - 4: Obtain ${}^1\mathcal{H}_x$ by minimizing $r_t^{(2)} = r_t^{(1)} - \langle {}^1\mathcal{H}_x, {}^1\mathcal{X} \rangle_F$.
 - 5: Calculate residual $r_t^{(2)} \leftarrow y_t - y_t^{(2)}$. (Order 2)
 - 6: Prepare ${}^2\mathcal{X} \leftarrow \mathbf{x}^{\otimes 2}$.
 - 7: Obtain ${}^2\mathcal{H}_x$ by minimizing $r_t^{(3)} = r_t^{(2)} - \langle {}^2\mathcal{H}_x, {}^2\mathcal{X} \rangle_F$.
 - 8: Calculate residual $r_t^{(3)} \leftarrow y_t - y_t^{(3)}$. (Order $m = 3, \dots, M$)
 - 9: **for** $m = 3$ to M **do**
 - 10: Prepare ${}^m\mathcal{X} \leftarrow \mathbf{x}^{\otimes m}$.
 - 11: Perform the HOSVD (Algorithm 2) for ${}^m\mathcal{X}$, and obtain a reduced-size tensor ${}^m\hat{\mathcal{X}}$.
 - 12: Obtain ${}^m\mathcal{H}_x$ by minimizing $r_t^{(m+1)} = r_t^{(m)} - \langle {}^m\mathcal{H}_x, {}^m\hat{\mathcal{X}} \rangle_F$.
 - 13: Calculate residual $r_t^{(m+1)} \leftarrow y_t - y_t^{(m+1)}$.
 - 14: **end for**
 - 15: **return** all ${}^m\mathcal{H}_x$ for $m = 0, 1, \dots, M$.
-

VI. SPACE COMPUTATIONAL COMPLEXITY FOR VOLTERRA KERNELS

The discrete Volterra series system model of order M and finite (dimension) memory N , given by (4), uses the super-symmetric kernels ${}^m\mathcal{H}_x$, where $m \leq M$. The total number of the parameters (the amount of memory cells) of the kernel $h_m^x(i_1, \dots, i_m)$ in the full model is

$$\#P_{\square} = N^m \sim O(N^m), \quad (23)$$

where m is the order of the kernel. Due to the super-symmetry, the parameters with permuted indexes share the same values, causing that the data stored in such a kernel is redundant.

For the reduced model (5), using the lower triangular kernels $h_m^{x,\Delta}(i_1, \dots, i_m)$ with a clipped range of indexes, the number of parameters describing each kernel is significantly lower, and equal

$$\#P_{\Delta} = \frac{(N+m-1)!}{(m)!(N-1)!} \sim O(N^m), \quad (24)$$

which is the number of all m -element combinations of indexes from N -element set, with repetitions.

The proposed method of the Volterra kernels decomposing by HOSVD leads to even more meaningful reduction of parameters needed to store the same data. As shown in Algorithm 2, the Volterra kernel ${}^m\mathcal{H}_x \in \mathbb{R}^{N^m}$ of order m can be decomposed to two components: a m -way tensor

$\mathcal{U} = (U_1, U_1, \dots, U_1)$ and a core tensor ${}^m\mathcal{S}$. But in fact, from numerical point of view, we only need to remember one orthogonal matrix $U_1 \in \mathbb{R}^{N \times K}$ and the core tensor ${}^m\mathcal{S} \in \mathbb{R}^{K^m}$. If we do not truncate the matrix U_1 , and leave all N singular values of matrix Σ intact, then we can re-obtain the exact original Volterra kernel in the following way

$${}^m\mathcal{H} = \mathcal{U} \otimes {}^m\mathcal{S}. \quad (25)$$

But in that case, we would use $N^2 + N^m$ parameters to store the decomposed tensor, which is higher number than for the original full Volterra kernel.

However, the goal of this decomposition is to keep only valuable data, considering the reconstruction error (15). If we truncate the orthogonal matrix U_1 to K -columns (see Algorithm 2), what means leaving only the K -most significant singular values, then we can reconstruct the estimate ${}^m\hat{\mathcal{H}}_x$ of Volterra kernel by the tensor product

$${}^m\hat{\mathcal{H}} = \mathcal{U} \otimes {}^m\mathcal{S}, \quad (26)$$

where this time \mathcal{U} consists of truncated matrices $U_1 \in \mathbb{R}^{N \times K}$. The synthesis of the estimate ${}^m\hat{\mathcal{H}}_x$ is using only

$$\#P_{HO} = NK + K^m \sim O(K^m) \quad (27)$$

parameters, and $K \in \{1, 2, 3, \dots, N\}$.

The amount of memory space required to store the full information about a Volterra kernel is presented in Fig. 2. The two standard approaches (full super-symmetrical kernel and reduced triangular kernel) use $O(N^m)$ space, where N is the tensor dimension (memory of the Volterra series system model) and m is the order of the kernel. The space complexity for the decomposed by HOSVD and truncated kernel depends on the reduced dimension K , and is $O(K^m)$, which can be meaningfully smaller if $K < N$.

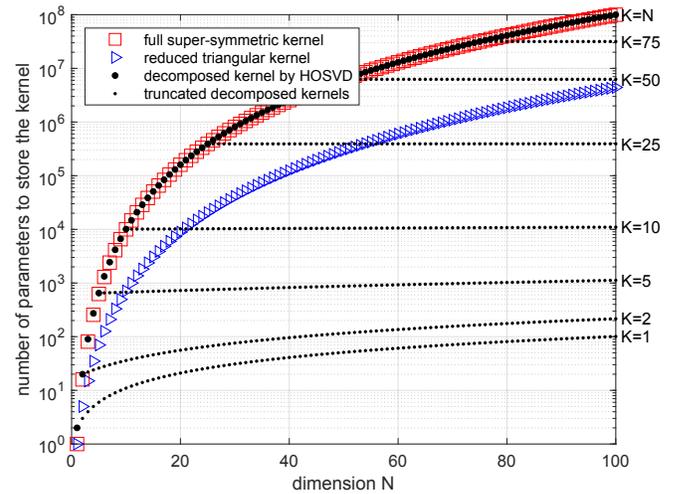


Fig. 2. Space complexity (the amount of cells) for different models of Volterra kernel ${}^4\mathcal{H}_x$ of order $m = 4$ for various dimensions N .

VII. SIMULATIONS

We performed a series of simulations, trying to identify Volterra kernel of order $m = 4$. In each simulation, we assumed a fixed Volterra kernel ${}^m\mathcal{H}_x$ of the system with the following super-symmetrical entries:

$${}^4\mathcal{H}_x(i_1, i_2, i_3, i_4) = f(i_1) \otimes f(i_2) \otimes f(i_3) \otimes f(i_4), \quad (28)$$

where

$$f(i) = \frac{N - i + 1}{N}, \quad i \in \{1, 2, 3, \dots, N\}. \quad (29)$$

The example of the Volterra kernel ${}^4\mathcal{H}_x$ for dimension $N = 6$ is presented, in sliced form, in Fig. 3. The color scale is ranging from 0 – deep blue to 1 – bright yellow.

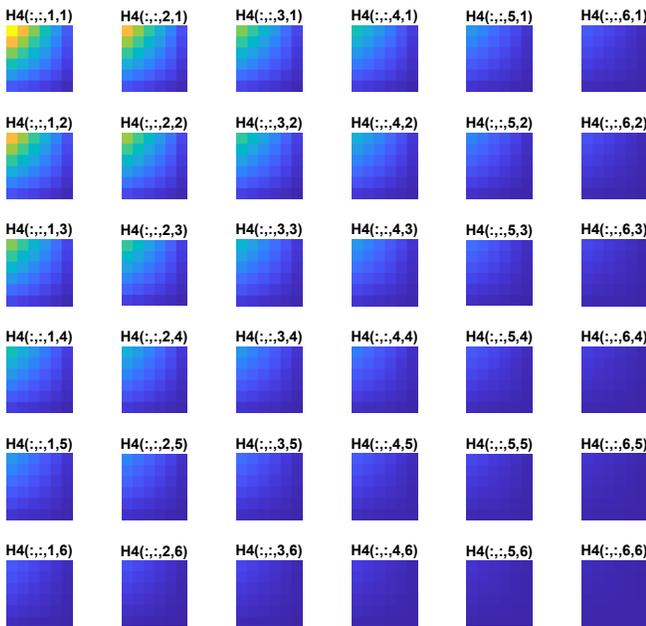


Fig. 3. The original Volterra kernel ${}^4\mathcal{H}_x$ for $N = 6$.

Then, a set of input and output signals $\{x_{t-1}, x_{t-2}, \dots, x_{t-N}; y_t\}$ was generated for the assumed kernel and $t = 1, \dots, T$, where the number of samples T varied from 2,500 to 50,000. Next step was to recover the Volterra kernel, using only the prepared set of delayed inputs and outputs. The performance of the identification technique was measured by the Frobenius norm error $\|{}^m\mathcal{H}_x - {}^m\hat{\mathcal{H}}_x\|_F^2$ between the original assumed Volterra kernel and the recovered Volterra kernel. We used a discrete Gaussian white noise with zero mean and variance equal $A = 9$ as the input signals. But it is worth mentioning that these examples are not an assumption for the proposed method, and almost any signal could serve as an input signal.

The input signal was transmitted through the Volterra series system with non-zero entries only for the prepared kernel ${}^4\mathcal{H}_x$ of order $m = 4$. In this way, we received the output signal $\{y_t\}$, which together with the input signal $\{x_t\}$ served in the learning process. For minimization, we used the least

squares technique with LASSO regularization to minimize the number of non-zero coefficients in the kernels. The reconstructed tensors of order 4 containing the identified by the HOSVD Volterra kernels ${}^m\hat{\mathcal{H}}_x$ (sliced versions) are shown in Figs. 4 and 5. The kernel estimation error measured by the Frobenius norm for various numbers of samples and reduced dimension (reduced number of singular values) to $K = 2, 3, 4, 5$ are presented in Table I.

TABLE I
KERNEL ESTIMATION ERROR $\|{}^m\mathcal{H}_x - {}^m\hat{\mathcal{H}}_x\|_F^2$ FOR ORDER $m = 4$
AND DIMENSION $N = 6$

No. sv	No. samples				
	2,500	5,000	12,500	25,000	50,000
N = 6	0.0162	0.0055	$0.4739e^{-3}$	$0.1842e^{-3}$	$0.1166e^{-3}$
K = 5	0.0162	0.0055	$0.4739e^{-3}$	$0.1842e^{-3}$	$0.1166e^{-3}$
K = 4	0.0163	0.0055	$0.4739e^{-3}$	$0.1842e^{-3}$	$0.1166e^{-3}$
K = 3	0.0163	0.0055	$0.4739e^{-3}$	$0.1842e^{-3}$	$0.1166e^{-3}$
K = 2	0.0162	0.0055	$0.4739e^{-3}$	$0.1842e^{-3}$	$0.1166e^{-3}$

The number of dropped singular values can have implicit impact on the identification error value. However, for the assumed Volterra kernel (28) with values around 1 for small indexes i_1, i_2, i_3, i_4 , and decreasing quickly (polynomially of degree 4) toward zero value along with the indexes increase, with only one significant singular value ($\sigma = [6.3897, 9.8102e^{-16}, 1.7700e^{-16}, 2.3336e^{-17}, 9.5131e^{-32}, 1.1969e^{-47}]$), we observe that the identification error values are almost the same for different numbers of singular values. The kernel estimate based on full decomposition components (for $K = N = 6$) and the kernel estimates reconstructed with the truncated matrices U_1 to $K = 5, 4, 3$ or 2 rows, which corresponds with the number of singular values, present the same reconstruction quality. The greatest impact has the number of samples for which the learning sequence was generated, with the identification error descending along with the increase of the number of samples.

VIII. CONCLUSIONS AND FUTURE WORKS

The paper introduces a sequential learning algorithm for the identification of nonlinear systems represented via Volterra series. The kernels of the Volterra series system model are super-symmetric tensors, and using that feature we modified the standard HOSVD (Algorithm 1) by boosting the calculation time by m -times for a Volterra kernel of order m (Algorithm 2).

The HOSVD is a factorization technique designed for higher order tensors, using the well known Singular Value Decomposition (SVD) for the unfolded tensor representation resulting from the chosen dimension. We proposed a method of identifying Volterra kernels of a system with known (measurable) inputs and outputs. The advantage of the proposed method, in contrary to other well known methods as cross-correlation method for Wiener and Volterra systems, is that we do not assume any particular properties of input signals.

The input signals can be chosen arbitrarily. The presented method can be easily applied, thanks to its simplicity, to reduce the dimension of tensors representing Volterra kernels of higher orders.

The proposed method for a particular kernel, using tensor decomposition and fast matrix operations, can lead to boosting the learning time for identification of Volterra series. For Volterra series models of higher orders, there is need for applying a sequential procedure identifying gradually the kernels with control over the accompanying error, due to dropping of some of singular values. The proposed cascade identification procedure (Algorithm 3) reduces Volterra kernels dimensions for kernels of order $m = 3$ and higher.

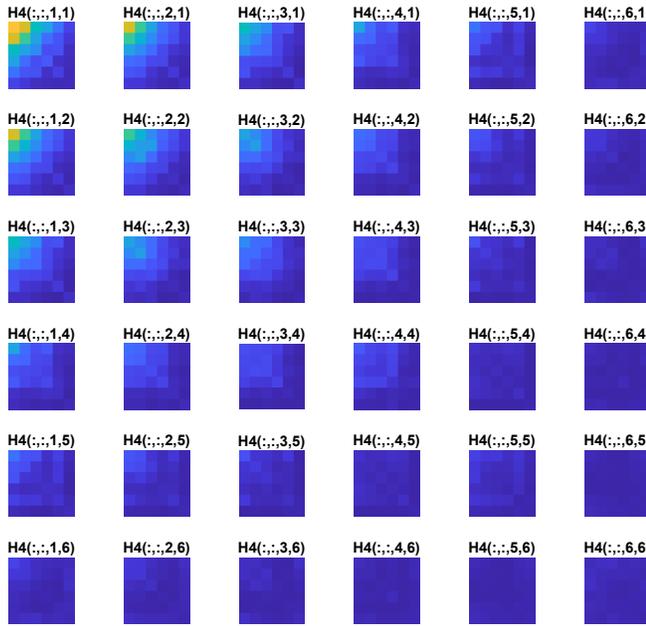


Fig. 4. Reconstructed kernel ${}^4\hat{\mathcal{H}}_x$ for $N = 6, K = 2, T = 5,000$.

REFERENCES

- [1] J. D. Carroll, J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart-Young' decomposition". *Psychometrika*, Vol. 35, No. 3, pp. 283-319, 1970.
- [2] L. De Lathauwer, B. De Moor, J. Vandewalle, "A Multilinear Singular Value Decomposition". *SIAM Journal on Matrix Analysis and Applications*, Vol. 21, No. 4, pp. 1253-1278, 2000.
- [3] M. O. Franz and B. Schölkopf, "Implicit estimation of Wiener series". In A. Barros, J. Principe, J. Larsen, T. Adali, and S. Douglas (Eds.), *Proc. 2004 IEEE Signal Processing Society Workshop, Vol. XIV of Machine Learning for Signal Processing*, pp. 735-744. IEEE, New York, 2004.
- [4] M. O. Franz and B. Schölkopf, "Implicit Wiener series for higher-order image analysis," In L. K. Saul, Y. Weiss, and L. Bottou (Eds.), *Advances in Neural Information Processing Systems*, Vol. 17, pp. 465-472 Cambridge, MA. MIT Press, 2005.
- [5] M. O. Franz and B. Schölkopf, "A Unifying View of Wiener and Volterra Theory and Polynomial Kernel Regression," in *Neural Computation*, Vol. 18, No. 12, pp. 3097-3118, Dec. 2006.
- [6] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis", *UCLA Working Papers in Phonetics*, Vol. 16, pp. 1-84, 1970.
- [7] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products". *Journal of Mathematics and Physics*, Vol. 6, pp. 164-189, 1927.

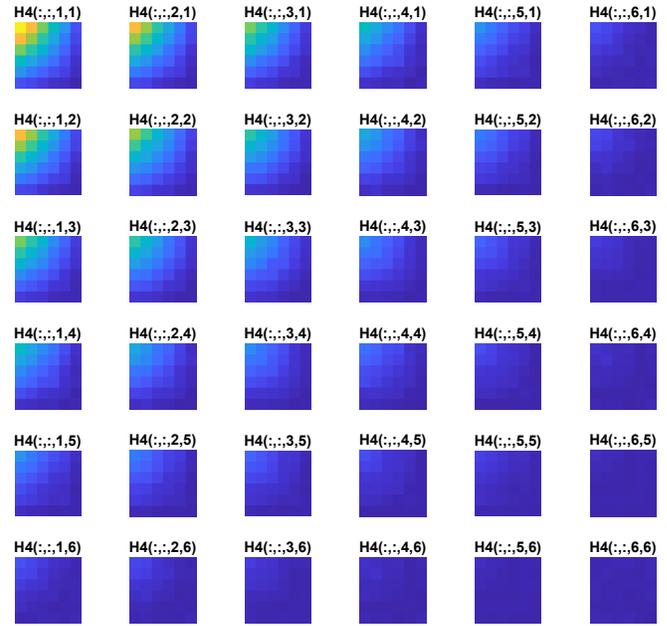


Fig. 5. Reconstructed kernel ${}^4\hat{\mathcal{H}}_x$ for $N = 6, K = 2, T = 12,500$.

- [8] T. G. Kolda, B. W. Bader. "Tensor Decompositions and Applications", *SIAM Review*, Vol. 51, No. 3, pp. 455-500, 2009.
- [9] M. J. Korenberg, S. B. Bruder, P. J. McIlroy, "Exact orthogonal kernel estimation from finite data records: extending Wiener's identification of nonlinear systems", *Ann. Biomed. Eng.*, Vol.16, pp. 201-214, 1988.
- [10] Y. W. Lee and M. Schetzen, "Measurement of the Wiener kernels of a non-linear system by crosscorrelation", *Intl. J. Control* 2, pp. 237-254, 1965.
- [11] S. Orcioni, "Improving the approximation ability of Volterra series identified with a cross-correlation method", *Nonlinear Dynamics*, Vol.78, pp. 2861-2869, 2014.
- [12] G. Palm and T. Poggio, "Stochastic identification methods for non-linear systems: an extension of the Wiener theory", *SIAM Journal of Applied Mathematics*, Vol. 34, No. 3, pp. 524-534, May 1978.
- [13] M. Schetzen, "A theory of nonlinear system identification", *Intl. J. Control* 20(4), pp. 577-592, 1965.
- [14] M. Schetzen, *The Volterra and Wiener theories of nonlinear systems*, Malabar: Krieger, 1980.
- [15] J. Schoukens and L. Ljung, "Nonlinear Systems Identification – A User-Oriented Road Map," *IEEE Control Systems Magazine*, pp. 28-99, December 2019.
- [16] S. Sherman, T. G. Kolda. "Estimating Higher-Order Moments Using Symmetric Tensor Decomposition", *arXiv:1911.03813*, submitted for publication, 2019.
- [17] L. R. Tucker, "Implications of factor analysis of three-way matrices for measurement of change". In C. W. Harris (Ed.), *Problems in Measuring Change*. Madison, Wis.: Univ. Wis. Press.: pp. 122-137, 1963.
- [18] L. R. Tucker, "The extension of factor analysis to three-dimensional matrices". In N. Frederiksen and H. Gulliksen (Eds.), *Contributions to Mathematical Psychology*. New York: Holt, Rinehart and Winston: pp. 109-127, 1964.
- [19] L. R. Tucker, "Some mathematical notes on three-mode factor analysis", *Psychometrika*, Vol. 31, No. 3, pp. 279-311, September 1966.
- [20] V. Volterra, "Sopra le funzioni che dipendono de altre funzioni", *Rend. R. Accademia dei Lincei 2^o Sem.*, pp. 97-105, 141-146 and 153-158, 1887.
- [21] N. Wiener, *Nonlinear problems in random theory*, Wiley, New York, 1958.