PARAMETRIC COMPLEXITY REDUCTION OF VOLTERRA MODELS USING TENSOR DECOMPOSITIONS

Gérard FAVIER and Thomas BOUILLOC

Laboratoire I3S, University of Nice Sophia Antipolis, CNRS,
Les Algorithmes- Bât. Euclide B, 2000 Route des lucioles, B.P. 121 - 06903 Sophia Antipolis Cedex, France
favier@i3s.unice.fr, bouilloc@i3s.unice.fr

ABSTRACT

Discrete-time Volterra models play an important role in many application areas. The main drawback of these models is their parametric complexity due to the huge number of their parameters, the kernel coefficients. Using the symmetry property of the Volterra kernels, these ones can be viewed as symmetric tensors. In this paper, we apply tensor decompositions (PARAFAC and HOSVD) for reducing the kernel parametric complexity. Using the PARAFAC decomposition, we also show that Volterra models can be viewed as Wiener models in parallel. Simulation results illustrate the effectiveness of tensor decompositions for reducing the parametric complexity of cubic Volterra models.

1. INTRODUCTION

Finite-dimensional discrete-time Volterra models, also called truncated Volterra series expansions or nonrecursive polynomial models, can be used for approximating any fading memory nonlinear system [4]. These models have been used in various fields of application : echo and noise cancellation [1], [23] loudspeaker system linearization [15], modeling, equalization and predistortion of nonlinear communication channels [2], [3], [6], [12], modeling of physiological systems [7], control of nonlinear processes [10].

Volterra models that can be viewed as a nonlinear extension of the finite impulse response (FIR) linear model, are interpretable in terms of multidimensional convolutions. They possess two interesting properties : linearity with respect to their parameters, the kernel coefficients, and guaranteed stability in the bounded-input bounded-output (BIBO) sense. Their main drawback is the huge number of parameters needed to characterize their kernels. There are different ways for reducing the parametric complexity of Volterra models. One approach consists in expanding the Volterra kernels using orthonormal basis functions like Laguerre functions [5], [11], [18], [22], or general orthonormal bases, i.e. orthonormal basis functions characterized by multiple poles [17]. Another approach [20] consists in representing the Volterra system in a parallel-cascade form resulting from the singular value decomposition of an unfolded matrix representation of the kernel. In [19], the authors propose to reduce the parametric complexity by using a tensor product basis approximation, four methods being proposed for choosing the approximation basis. The approach proposed in this paper consists in considering the Volterra kernels as tensors and using tensor decompositions. This approach was introduced for the first time in [16].

As any Volterra kernel of order equal to or higher than two can be replaced by a symmetric kernel, we propose, in this paper, to use two tensor decompositions, the so called PARAllel FACtor (PARAFAC) decomposition [14] and Higher Order Singular Value Decomposition (HOSVD) [8], for decomposing symmetric kernels.
The rest of this paper is organized as follows. In section 2, we briefly present the Volterra model and indicate how its kernels can be put in a symmetric form. In section 3, we recall the two used tensor decompositions and we describe the algorithms allowing to estimate the matrix factors of these decompositions. In the case of a separable cubic kernel, we propose an analytic solution for computing the vector factor of the PARAFAC decomposition. Using this PARAFAC decomposition, we also show that Volterra models can be viewed as Wiener models in parallel. Simulation results are shown in section 4 to illustrate the significant parametric complexity reduction of cubic Volterra kernels in using tensor decompositions. Finally, in section 5, we conclude the paper in drawing some perspectives for this work.

Notations : Scalars, vectors, matrices and tensors are respectively written as lower-case $(a,b,\ldots)$, bold lower-case $(\mathbf{a},\mathbf{b},\ldots)$, bold upper-case$(\mathbf{A},\mathbf{B},\ldots)$, and blackboard $(\mathbb{A},\mathbb{B},\cdots)$ letters. $\mathbf{A}^T$, $\mathbf{A}^H$ and $\mathbf{A}^\dagger$ denote respectively transpose, transconjugate (or Hermitian transpose) and Moore-Penrose pseudo-inverse of $\mathbf{A}$. The operators $diag(.)$ and $diag_i(.)$ form a diagonal matrix from its vector argument and from the i^{th} row of its matrix argument respectively, whereas the operator $vec(.)$ forms a column vector by stacking the columns of its matrix argument. The vector $\mathbf{A}_{i.}$ (resp. $\mathbf{A}_{.j}$) denote the i^{th} row (resp. j^{th} column) of $\mathbf{A}$. The outer, Kronecker and Khatri-Rao products are respectively denoted by $\circ$, $\otimes$ and $\odot$. The Khatri-Rao (column-wise Kronecker) product of matrices $\mathbf{A}$ and $\mathbf{B}$ with respective dimensions $M \times R$ and $N \times R$ is the matrix of dimensions $MN \times R$ defined as : $\mathbf{A} \odot \mathbf{B} = (\mathbf{A}_{.1} \otimes \mathbf{B}_{.1} \cdots \mathbf{A}_{.R} \otimes \mathbf{B}_{.R})$.

2. VOLTERRA MODELS

An P^{th}-order Volterra model for a causal, stable, finite memory, single-input single output (SISO) system is described by the following input-output relation :

$$\begin{aligned} y(k) &= h_0 + \sum_{p=1}^{P} \sum_{m_1=1}^{M_p} \cdots \sum_{m_p=1}^{M_p} h_p(m_1,\cdots,m_p) \prod_{i=1}^{p} u(k-m_i) \\ &= h_0 + \sum_{p=1}^{P} y_p(k) \end{aligned} \quad (1)$$

where $u(k)$ and $y(k)$ denote respectively the input and output signals, P is the nonlinearity degree of the Volterra model, M_p is the memory length of the p^{th}-order homogeneous term $y_p(k)$, and $h_p(m_1,\cdots,m_p)$ is a coefficient of the p^{th}-order kernel. This coefficient being characterized by p indices, it can be viewed as an element of a tensor $\mathbb{H}_p \in \mathscr{K}^{M_p\times M_p\times\cdots\times M_p}$, of order p, with $\mathscr{K}=\mathscr{R}$ or $\mathscr{C}$, depending on whether the kernel coefficients are real-valued or complex-valued. The p^{th}-order kernel is characterized by M_p^p coefficients. As each permutation of the indices $m_1,\cdots,m_p$ corresponds to the same product $\prod_{i=1}^{p} u(k-m_i)$ of delayed inputs, we can sum all the coefficients associated with these permutations to get a symmetric kernel defined as :

$$h_{p,sym}(.) = \frac{1}{p\,!}\sum_{\pi(.)} h_p(m_{\pi(1)},\cdots,m_{\pi(p)})$$

where $(\pi(1),\cdots,\pi(p))$ denotes a permutation of $(1,\cdots,p)$. The number of independent coefficients contained in the symmetric p^{th}-order kernel is equal to $C_p^{M_p+p-1}$.

The p^{th}-order kernel is said to be separable if it can be expressed as the product of p first-order kernels :

$$h_p(m_1,\cdots,m_p) = \Pi_{i=1}^{p} h_{m_i}^{(i)} \tag{2}$$

where $h_{m_i}^{(i)}$ represents the m_i^{th} element of the first-order kernel $\mathbf{h}^{(i)}$. If the separable kernel is also symmetric, then the p first-order kernels $\mathbf{h}^{(i)}$ are identical and (2) becomes :

$$h_p(m_1,\cdots,m_p) = \Pi_{i=1}^{p} h_{m_i}$$

3. TENSOR DECOMPOSITIONS

We first recall some definitions relative to tensors. Then, we present the so called PARAFAC and HOSVD decompositions.

3.1 Some definitions

A tensor, also called a multi-way array, of order P and dimensions $M_1\times M_2\times\cdots\times M_P$, is a mathematical object described by means of P indices, each index being associated with a coordinate axis, also called a mode or a way, and M_p representing the dimension of $\mathbb{H}$ along its p^{th}-mode. An P^{th}-order tensor $\mathbb{H}\in\mathscr{K}^{M_1\times M_2\times\cdots\times M_P}$ is characterized by $\prod_{p=1}^{P} M_p$ scalar coefficients $h_{m_1,\cdots,m_p}\in\mathscr{K}$, $m_p=1,2,\cdots,M_p$, $p=1,\cdots,P$.

The tensor $\mathbb{H}$ is said to be symmetric if its elements $h_{m_1m_2\cdots m_p}$ do not change under any permutation of their indices.

The mode-p slice of $\mathbb{H}$ is the $(P-1)^{th}$-order tensor obtained by fixing the mode-p index, and the m_p^{th} mode-p slice is denoted by $\mathbb{H}_{\cdots m_p\cdots}\in\mathscr{K}^{M_1\times\cdots\times M_{p-1}\times M_{p+1}\times\cdots\times M_P}$.

The mode-p vectors (or fibers) are the M_p-dimensional vectors obtained from $\mathbb{H}$ by varying the index m_p, with the other indices fixed : $\mathbb{H}_{m_1\cdots m_{p-1}\bullet m_{p+1}\cdots m_P}$.

For a third-order tensor $\mathbb{H}\in\mathscr{K}^{I\times J\times K}$ with entries h_{ijk}, the mode-p slices corresponding to $p=1,2$ and 3, are the matrices respectively denoted by $\mathbf{H}_{i..}$, $\mathbf{H}_{.j.}$ and $\mathbf{H}_{..k}$, and called horizontal, lateral and frontal slices of $\mathbb{H}$.

An important operation consists in matricizing (or unfolding) a tensor, i.e. transforming the tensor into a matrix. The unfolded representation of $\mathbb{H}\in\mathscr{K}^{M_1\times M_2\times\cdots\times M_P}$ along the mode-p is denoted by $\mathbf{H}_p\in\mathscr{K}^{M_p\times\left(M_{p+1}\cdots M_P M_1\cdots M_{p-1}\right)}$, and the column vectors of $\mathbf{H}_p$ are the mode-p vectors of $\mathbb{H}$. The rank of $\mathbf{H}_p$, denoted by $R_p=r(\mathbf{H}_p)$, is called the mode-p rank of $\mathbb{H}$, i.e. the dimension of the vector space spanned by the mode-p vectors.

For a third-order tensor $\mathbb{H}\in\mathscr{K}^{I\times J\times K}$, its three unfolded matrix representations obtained in columnwise stacking its matrix slices, are defined as :

$$\mathbf{H}_1 = [\ \mathbf{H}_{..1}\ \cdots\ \mathbf{H}_{..K}\ ]\in\mathscr{K}^{I\times JK} \tag{3}$$

$$\mathbf{H}_2 = [\ \mathbf{H}_{1..}\ \cdots\ \mathbf{H}_{I..}\ ]\in\mathscr{K}^{J\times KI} \tag{4}$$

$$\mathbf{H}_3 = [\ \mathbf{H}_{.1.}\ \cdots\ \mathbf{H}_{.J.}\ ]\in\mathscr{K}^{K\times IJ} \tag{5}$$

Another important operation used for computing the HOSVD is the mode-p product of a tensor $\mathbb{H}\in\mathscr{K}^{M_1\times M_2\times\cdots\times M_P}$ by a matrix $\mathbf{U}\in\mathscr{K}^{N_p\times M_p}$, denoted by $\mathbb{H}\times_p\mathbf{U}$ and defined as the tensor $\mathbb{J}\in\mathscr{K}^{m_1\times\cdots\times M_{p-1}\times N_p\times M_{p+1}\times\cdots\times M_P}$ such that :

$$\begin{aligned}&(\mathbb{H}\times_p\mathbf{U})_{m_1\cdots m_{p-1}n_pm_{p+1}\cdots m_P}\\ &=\sum_{m_p=1}^{M_p} h_{m_1\cdots m_{p-1}m_pm_{p+1}\cdots m_P}u_{n_pm_p}\end{aligned}$$

This mode-p product can be expressed in terms of mode-p unfolded representations as $\mathbf{J}_p=\mathbf{U}\mathbf{H}_p$.

3.2 The PARAFAC decomposition

The PARAFAC decomposition of an Pth-order tensor $\mathbb{H}\in\mathscr{K}^{M_1\times M_2\times\cdots\times M_P}$ can be written in scalar form as :

$$h_{m_1m_2\cdots m_P} = \sum_{r=1}^{R}\prod_{p=1}^{P} a_{m_pr}^{(p)} \quad m_p=1,\cdots,M_p \tag{6}$$

where $a_{m_pr}^{(p)}$ is an entry of the matrix factor $\mathbf{A}^{(p)}\in\mathscr{K}^{M_p\times R}$, $p=1,\cdots,P$, and R is the rank of the tensor. Equation (6) can be rewritten as : $\mathbb{H}=\sum_{r=1}^{R}\mathbf{A}_{.r}^{(1)}\circ\mathbf{A}_{.r}^{(2)}\circ\cdots\circ\mathbf{A}_{.r}^{(P)}$ showing that PARAFAC corresponds to a decomposition of the P^{th}-order tensor of rank R, into a sum of R rank-one tensors, i.e. a sum of R outer products of P vectors.

We can note that for an P^{th}-order rank-one tensor $\mathbb{H}$, (6) becomes :

$$h_{m_1m_2\cdots m_P} = \prod_{p=1}^{P} a_{m_p}^{(p)}, \quad m_p=1,\cdots,M_p \tag{7}$$

where $a_{m_p}^{(p)}$ is an entry of the vector factor $\mathbf{a}^{(p)}\in\mathscr{K}^{M_p\times 1}$. By comparing (7) with (2), we can conclude that the PARAFAC decomposition of a rank-one Volterra kernel is equivalent to that of a separable kernel.

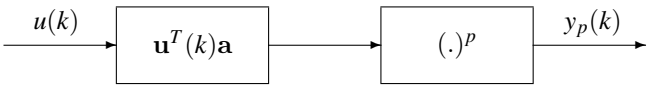

FIGURE 1 – Realization of the p^{th}-order homogeneous term of a Volterra model as a Wiener model. Case of a rank-one kernel.

In the case of an P^{th}-order symmetric tensor $\mathbb{H}$, of rank R, the P matrix factors $\mathbf{A}^{(p)}$, $p = 1, \cdots, P$, are all identical and equal to $\mathbf{A}$. Similarly, for an P^{th}-order rank-one symmetric tensor $\mathbb{H}$, the vector factors $\mathbf{a}^{(p)}$, $p = 1, \cdots, P$, are all identical and equal to $\mathbf{a}$.

Use of the PARAFAC decomposition of a symmetric rank-one p^{th}-order Volterra kernel allows to rewrite the p^{th}-order homogeneous term $y_p(k)$ in (1) as follows :

$$\begin{aligned} y_p(k) &= \sum_{m_1=1}^{M_p} \cdots \sum_{m_p=1}^{M_p} h_p(m_1, \cdots, m_p) \prod_{i=1}^{p} u(k-m_i) \\ &= \sum_{m_1=1}^{M_p} \cdots \sum_{m_p=1}^{M_p} \prod_{i=1}^{p} a_{m_i} u(k-m_i) = \left(\mathbf{u}^T(k)\mathbf{a}\right)^p \quad (8) \end{aligned}$$

where $\mathbf{u}^T(k) = [\ u(k-1) \quad \cdots \quad u(k-M_p)\ ] \in \mathscr{K}^{1\times M_p}$ is the linear regression vector associated with this kernel of memory length M_p, and $\mathbf{a}^T = \left[\ a_1 \quad \cdots \quad a_{M_p}\ \right] \in \mathscr{K}^{1\times M_p}$ contains the coefficients of the vector factor (generator vector) of the PARAFAC decomposition of the p^{th}-order Volterra kernel. We deduce that (8) can be viewed as the output of a Wiener model obtained in concatenating a FIR linear model of memory length M_p with a memoryless nonlinearity of degree p as illustrated in Fig.1.
For a symmetric p^{th}-order Volterra kernel of rank r_p, (8) becomes :

$$y_p(k) = \sum_{r=1}^{r_p} \left(\mathbf{u}^T(k)\mathbf{A}_{.r}\right)^p \quad (9)$$

The p^{th}-order homogeneous term can therefore be carried out in parallelizing r_p Wiener models, each one being associated with a column of the matrix factor of the kernel PARAFAC decomposition.
So, the Volterra model output (1) can be obtained as the sum of a constant term h_0, and the outputs of $\sum_{p=1}^{P} r_p$ Wiener models in parallel, as shown in Fig. 2 for a cubic Volterra model, where $\mathbf{A}_{.r}^{(p)}$ denotes the r^{th} column of the matrix factor of the p^{th}-order kernel PARAFAC decomposition, for $p = 2, 3$, and $\mathbf{A}_{.1}^{(1)} = [h_1(1) \cdots h_1(M_1)]$, $\mathbf{u}_p^T(k) = [u(k-1) \cdots u(k-M_p)]$, $p = 1, 2$ and 3.

Now, we consider the problem of determining the matrix factor $\mathbf{A}$ of the PARAFAC decomposition of an P^{th}-order symmetric Volterra kernel. The proposed estimation method is the conditional least squares (CLS) algorithm. To simplify

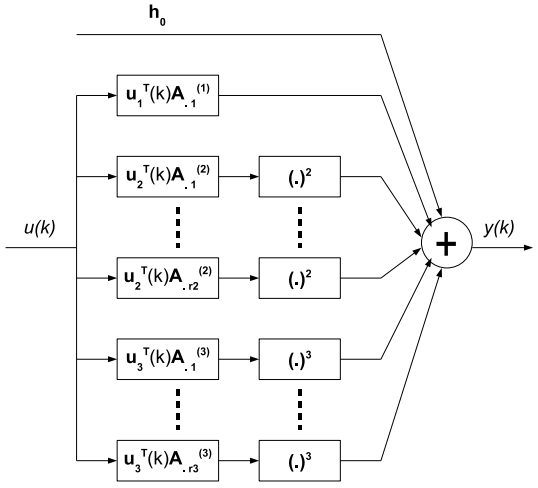

FIGURE 2 – Realization of a cubic Volterra model as Wiener models in parallel.

the presentation, we consider the PARAFAC decomposition of a third-order symmetric kernel $\mathbb{H} \in \mathscr{K}^{M\times M\times M}$:

$$h_{m_1 m_2 m_3} = \sum_{r=1}^{R} \prod_{p=1}^{3} a_{m_p r}, \quad m_p = 1, \cdots, M, \quad p = 1, 2, 3.$$

In this case, the unfolded matrix representations (3)-(5) are identical and can be expressed in terms of the matrix factor $\mathbf{A}$ as :

$$\mathbf{H}_i = \mathbf{A}\left(\mathbf{A} \odot \mathbf{A}\right)^T \in \mathscr{K}^{M\times M^2}, \quad i = 1, 2, 3. \quad (10)$$

We propose to determine the matrix factor $\mathbf{A}$ in minimizing the following conditional LS cost function :

$$\min_{\mathbf{A}} \left\| \mathbf{H}_1 - \mathbf{A}(\mathbf{A}_{t-1} \odot \mathbf{A}_{t-1})^T \right\|_F^2 \quad (11)$$

where t and $\|.\|_F$ denote respectively the iteration number and the Frobenius norm.

The CLS algorithm is summarized as follows :

1. Randomly initialize $\mathbf{A}$ and set $t = 0$.
2. Increment t and compute $\mathbf{B}_t = \mathbf{A}_{t-1} \odot \mathbf{A}_{t-1}$ and $\mathbf{A}_t = \mathbf{H}_1 \left(\mathbf{B}_t^{\dagger}\right)^T$.
3. Return to step 2 until convergence.

The convergence test consists in detecting if an estimated parameter variation between two consecutive iterations or the model fit error calculated in using the tensor reconstructed from the estimated parameters, becomes smaller than a predefined threshold. In practice, to improve the convergence, the CLS algorithm that enforces the symmetry to the solution is applied after a transient period during which the classical alternating least squares (ALS) is used for estimating the three matrix factors without enforcing the symmetry. More efficient algorithms like the enhanced line search (ELS)[21] or the Levenberg Marquardt one can also

be used.

In the case of a symmetric rank-one cubic Volterra kernel, it is also possible to use an analytic solution. Indeed, we have :

$$h_{ijk} = a_i a_j a_k \quad i,j,k = 1,\cdots,M \tag{12}$$

and consequently $h_{iii} = a_i^3$, $i = 1,\cdots,M$. When the tensor is real, there exists only one real solution $a_i = \sqrt[3]{h_{iii}}$, $i = 1,\cdots,M$. In presence of noise, we propose the following analytic solution :

1. For $i = 1,\cdots,M$:
   – Compute $a_j^{(i)} = \frac{h_{jii}}{h_{iii}^{2/3}}$, $j = 1,\cdots,M$.
2. Compute the average value $a_j = \sum_{i=1}^{M} a_j^{(i)}$, $j = 1,\cdots,M$.

In the complex case, the analytic solution is a little bit more complicated to calculate due to the fact that now there exists three possible complex cubic roots $\sqrt[3]{h_{iii}}$. Due to a lack of space, we do not detail the calculation of the corresponding analytic solution.

3.3 The HOSVD

Given the symmetric P^{th}-order tensor $\mathbb{H} \in \mathscr{K}^{M\times M\times\cdots\times M}$, the algorithm for computing its HOSVD is :

1. Compute the SVD of $\mathbf{H}_1$, the mode-1 unfolded representation of $\mathbb{H}$: $\mathbf{H}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^H$, where $\mathbf{U}$ is an $M \times M$ unitary matrix the columns of which span the column space of $\mathbf{H}_1$.
2. Compute the core tensor as : $\mathbb{C} = \mathbb{H} \times_1 \mathbf{U}^H \times_2 \mathbf{U}^H \times \cdots \times_P \mathbf{U}^H$.
3. Compute the HOSVD of $\mathbb{H}$ as :

$$\mathbb{H} = \mathbb{C} \times_1 \mathbf{U} \times_2 \mathbf{U} \times \cdots \times_P \mathbf{U}. \tag{13}$$

Note that, in the case of a real tensor, the transconjugation (H) is replaced by the tranposition (T), and $\mathbf{U}$ is orthogonal.

As well known, the best rank-K approximation of a rank-R matrix, with $K < R$, is obtained by truncating its SVD $\left(\mathbf{U}\mathbf{\Sigma}\mathbf{U}^H\right)$, i.e its reduced SVD : $\mathbf{U}^{(K)}\mathbf{\Sigma}^{(K)}\mathbf{U}^{(K)^H} = \sum_{k=1}^{K} \sigma_k \mathbf{U}_{.k}\mathbf{U}_{.k}^H$, with $\sigma_1 \geq \cdots \geq \sigma_K \geq \cdots \geq \sigma_R > 0$, $\mathbf{U}^{(K)}$ being formed with the left singular vectors associated with the K largest singular values. This property is not valid for tensors of order higher than two. An optimal rank-$(R_1,R_2,\cdots,R_K)$ approximation of an P^{th}-order tensor $\mathbb{H} \in \mathscr{K}^{M_1\times M_2\times\cdots\times M_P}$ was proposed in [9]. However, the computation of this best rank approximation being quite time consuming and providing results very similar to the ones obtained from a simple truncation, we apply this last solution to a symmetric P^{th}-order rank-R tensor, as summarized below :

1. For a given rank $K < R$, compute the reduced SVD of $\mathbf{H_1}$: $\mathbf{U}^{(K)}\mathbf{\Sigma}^{(K)}\mathbf{U}^{(K)^H}$, where $\mathbf{U}^{(K)}$ is the $M \times K$ column-orthonormal matrix.
2. Compute the reduced core tensor as :

$$\mathbb{C}^{(K)} = \mathbb{H} \times_1 \mathbf{U}^{(K)^H} \times_2 \mathbf{U}^{(K)^H} \times \cdots \times_P \mathbf{U}^{(K)^H} \in \mathscr{K}^{K\times K\times\cdots\times K} \tag{14}$$

3. Compute the truncated HOSVD of $\mathbb{H}$ as :

$$\mathbb{H} = \mathbb{C}^{(K)} \times_1 \mathbf{U}^{(K)} \times_2 \mathbf{U}^{(K)} \times \cdots \times_P \mathbf{U}^{(K)}. \tag{15}$$

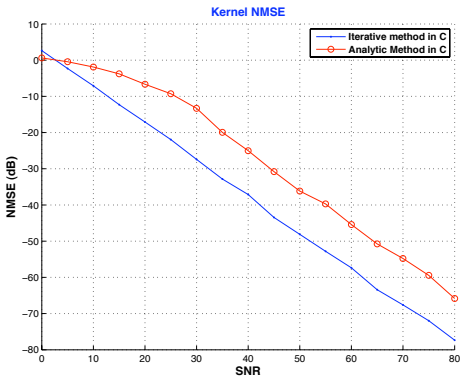

FIGURE 3 – Kernel NMSE versus SNR for the analytic and iterative methods in the case of complex separable kernels.

4. SIMULATION RESULTS

In this section, we present some simulation results for comparing the effectiveness of the PARAFAC decomposition and the truncated HOSVD for reducing the parametric complexity of cubic Volterra kernels. Two kinds of kernels were simulated and, for each kind, 300 kernels were randomly generated.

Two performance criteria are considered for the comparison :

– Number of parameters of the PARAFAC/HOSVD decompositions, and associated complexity reduction rate (CRR) calculated in % as :

$$CRR = 100\frac{N - N_d}{N} \tag{16}$$

where N and N_d represent respectively the number of components contained in the original kernel, and in the factors of its decomposition.

– Kernel NMSE (Normalized mean-square error) calculated in dB as :

$$NMSE = 10\log_{10}\left(\frac{\|\mathbb{H} - \mathbb{H}_{est}\|_F^2}{\|\mathbb{H}\|_F^2}\right) \tag{17}$$

where $\mathbb{H}_{est}$ is the kernel reconstructed from the estimated parameters of its decomposition.

1) Case of random symmetric complex separable kernels generated from their PARAFAC decomposition.

The experiment consists in randomly generating 500 vector factors $\mathbf{a}$ of dimension 15×1. Then, the kernels are constructed using the PARAFAC model. For each SNR value, 100 different noise sequences are added to the simulated kernels. The results are displayed in Fig.3. The iterative method gives better results than the analytic one, with a kernel NMSE very close to the SNR.

2) Case of random symmetric real non-separable kernels.

A set of 500 random rank-3 symmetric tensors were simulated from their PARAFAC decomposition, without addi-

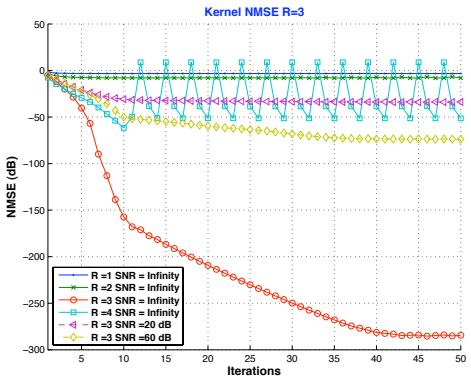

FIGURE 4 – Kernel NMSE versus iterations for CLS algorithm with random rank-3 kernels.

SNR / K	NMSE (dB) 20 dB	60 dB	∞	Number of parameters	CRR (%)
1	−3.4	−3.4	−3.4	63	99.32
2	−8.3	−8.3	−8.3	134	98.55
3	−34.5	−74.5	−295.7	216	97.67
4	−33.4	−73.4	−297.1	316	96.59

TABLE 1 – NMSE and CRR for the truncated HOSVD of random rank-3 kernels.

tive noise. Results are plotted in Fig. 4, for the CLS algorithm, and the NMSE and CRR obtained with the truncated HOSVD and PARAFAC decomposition are given respectively in Tables 1 and 2.

From the simulation results, we can conclude that the CLS algorithm only works when the rank of the decomposition is equal to the rank of the tensor. In this case, the NMSE decreases when the SNR increases. We make the same observation with the HOSVD. Moreover, we can conclude that the truncated HOSVD provides better results than the CLS algorithm.

SNR / R	NMSE (dB) 20 dB	60 dB	∞	Number of parameters	CRR (%)
1	−3.3	−3.3	−3.3	63	99.32
2	−8.1	−8.0	−8.0	126	98.64
3	−33.7	−73.9	−284.4	189	97.96
4	−21.4	−54.9	−51.3	252	97.28

TABLE 2 – NMSE and CRR for the CLS algorithm with random rank-3 kernels.

5. CONCLUSION

In this paper, two tensor decompositions, the so called PARAFAC and truncated HOSVD, have been used for reducing the parametric complexity of Volterra models. The truncated HOSVD presents the advantage to be simple to implement and of non-iterative type. The CLS algorithm is also easy to implement, but, due to its iterative nature, its convergence is often slow and even not always ensured. Another way to determine the PARAFAC matrix factors consists in applying an extended Kalman filter to the input-output relation associated with the Volterra-PARAFAC model [13]. A comparison with methods based on the development of Volterra kernels on orthonormal basis functions is also under study.

REFERENCES

[1] O. Agazzi, D. Messerschmitt, and D. Hodges. *IEEE Tr. Commun.*, 30(11) :2421–2433, Nov 1982.

[2] S. Benedetto and E. Biglieri. *IEEE J. Select. Areas Commun.*, 1(1) :57–62, Jan 1983.

[3] S. Benedetto, E. Biglieri, and R. Daffara. *IEEE Tr. Aerosp. Electron. Syst.*, 15(4) :494–507, July 1979.

[4] S. Boyd and L. Chua. *IEEE Trans. Circ. and Syst. CAS*, 32(11) :1150–1161, Nov 1985.

[5] R. Campello, G. Favier, and W. Amaral. *Automatica*, 40(5) :815–822, 2004.

[6] C.-H. Cheng and E. J. Powers. *IEEE Tr. Signal Proc.*, 49(1) :147–163, Jan. 2001.

[7] K. H. Chon, N. H. Holstein-Rathlou, D. J. Marsh, and V. Z. Marmarelis. *IEEE Tr. Neural Networks*, 9(3) :430–435, May 1998.

[8] L. De Lathauwer, B. De Moor, and J. Vandewalle. *SIAM J. Matrix Analysis and Appl.*, 21(4) :1253–1278, 2000.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle. *SIAM J. Matrix Analysis and Appl.*, 21(4) :1324–1342, 2000.

[10] F. Doyle III, R. Pearson, and B. Ogunnaike. *Springer-Verlag*, 2002.

[11] G. Dumont and Y. Fu. *Int.J. of Adaptive Control and Signal Proc.*, 7 :367–382, 1993.

[12] C. Eun and E. J. Powers. *IEEE Tr. Signal Proc.*, 45(1) :223–227, Jan. 1997.

[13] G. Favier and T. Bouilloc. Submitted to GRETSI 2009.

[14] R. Harshman. *UCLA working papers in phonetics*, 16 :1–84, 1970.

[15] Y. Kajikawa. *EUSIPCO, Lausanne, Switzerland*, 2008.

[16] A. Khouaja and G. Favier. *EUSIPCO, Wien, Austria*, pages 1903–1906, 2004.

[17] A. Kibangou, G. Favier, and M. Hassani. *Signal Process.*, 85 :2371–2385, 2005.

[18] V. Z. Marmarelis. *Annals of Biomedical Eng.*, 21 :573–589, 1993.

[19] R. D. Nowak and B. D. Van Veen. *IEEE Trans. on Signal Processing*, 44(1) :36–50, Jan. 1996.

[20] T. M. Panicker and V. J. Mathews. *IEEE Trans. on Signal Processing*, 46(10) :18–22, Oct. 1998.

[21] M. Rajih, P. Comon, and R. Harshman. *SIAM J. Matrix Analysis and Appl.*, to appear.

[22] N. Wiener. *MIT Press, Cambridge, Massachussetts, and Wiley, New-York*, 1958.

[23] M. Zeller and W. Kellerman. *EUSIPCO, Poznan, Poland*, pages 1605–1609, 2007.