

Research Article

Nonlinear Blind Identification with Three-Dimensional Tensor Analysis

I. Cherif, S. Abid, and F. Fnaiech

*Unit Research SICISI, High School of Sciences and Techniques of Tunis (ESSTT),
5 Avenue Taha Hussein, Montfleury, 1008 Tunis, Tunisia*

Correspondence should be addressed to S. Abid, abid2fr@yahoo.fr

Received 15 February 2012; Revised 2 April 2012; Accepted 3 April 2012

Academic Editor: Thomas T. Yang

Copyright © 2012 I. Cherif et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper deals with the analysis of a third-order tensor composed of a fourth-order output cumulants used for blind identification of a second-order Volterra-Hammerstein series. It is demonstrated that this nonlinear identification problem can be converted in a multivariable system with multiequations having the form of $Ax + By = c$. The system may be solved using several methods. Simulation results with the Iterative Alternating Least Squares (IALS) algorithm provide good performances for different signal-to-noise ratio (SNR) levels. Convergence issues using the reversibility analysis of matrices A and B are addressed. Comparison results with other existing algorithms are carried out to show the efficiency of the proposed algorithm.

1. Introduction

Nonlinear system modeling based on real-world input/output measurements is so far used in many applications. The appropriate model and the determination of corresponding parameters using the input/output data are owned to apply a suitable and efficient identification method [1–7].

Hammerstein models are special classes of second-order Volterra systems where the second-order homogenous Volterra kernel is diagonal [8]. These systems have been successfully used to model nonlinear systems in a number of practical applications in several areas such as chemical process, biological process, signal processing, and communications [9–12], where, for example, in digital communication systems, the communication channels are usually impaired by a nonlinear intersymbol interference (ISI). Channel identification allows compensating the ISI effects at the receivers.

In [13], a penalty transformation method is developed. Indeed a penalty function is formed by equations relating the unknown parameters of the model with the autocorrelations

of the signal. This function is then included in the cost function yielding to an augmented Lagrangian function. It has been demonstrated that this approach gives good identification results for a nonlinear systems. However, this approach is still sensitive to additive Gaussian noise because the 2nd-order moment is used as a constraint. Authors, in [7], overcame this sensitivity by using 4th-order cumulants as a constraint instead of 2nd-order moments in order to smooth out the additive Gaussian noise. But the proposed approach which is based on a simplex-genetic algorithm becomes so long and computationally complex.

The main drawback of identification with Volterra series lies on the parametric complexity and the need to estimate a very big number of parameters. In many cases, Volterra series identification problem may be well simplified using the tensor formulation [10–12, 14].

Authors, in [10], used a parallel factor (PARAFAC) decomposition of the kernels to derive Volterra-PARAFAC models yielding an important parametric complexity reduction for Volterra kernels of order higher than two. They proved that these models are equivalent to a set of parallel Wiener models. Consequently, they proposed three adaptive algorithms for identifying these proposed Volterra-PARAFAC models for complex-valued input/output signals, namely, the extended complex Kalman filter, the complex least mean square (CLMS) algorithm, and the normalized CLMS algorithm.

In this paper, the algorithm derived in [14] is extended to be applied to blind identification of a general second-order Volterra-Hammerstein system. The main idea is to develop a general expression for each direction slices of a cubic tensor and then express the tensor slices in an unfolded representation. The three-dimensional tensor elements are formed by the fourth-order output cumulants. This yields to an Iterative Alternating Least Square (IALS) algorithm which has the benefit over the original Volterra filters in terms of implementation and complexity reduction. A convergence analysis based on matrices reversibility study is given showing that the proposed IALS algorithm converges to optimal solutions in the least mean squares sense. Furthermore, some simulation results and comparisons with different existing algorithms are provided.

The present work is organized as follows; in Section 2, a brief study of the three-dimensional tensor is presented. In Section 3, the model under study and the related output cumulants are then proposed, whereas, in Section 4 the decomposition analysis of the cumulant tensor is developed. In Sections 5 to 8, we give, respectively, the proposed blind identification algorithm, the convergence study, some simulation results, and at the end some main conclusions are drawn.

2. Three-Dimensional Tensor and Different Slice Expressions

A three-dimensional tensor $C \in \mathbb{C}^{M \times M \times M}$ can be expressed by

$$C = \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^M C_{ijk} e_i^{(M)} \circ e_j^{(M)} \circ e_k^{(M)}, \quad (2.1)$$

where C_{ijk} is the tensor value in the position (i, j, k) of the cube with dimension M , $e_p^{(M)}$ denotes the p th canonical basis vector with dimension M , and the symbol \circ stands for the outer product (Figure 1).

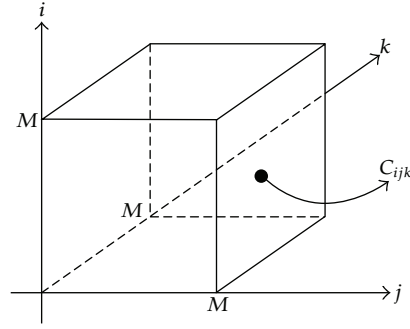


Figure 1: Cubic tensor representation.

A cubic tensor can be always sliced along three possible directions (horizontal, vertical, and frontal) as depicted in Figure 2. This yields, in each case, to $M \times M$ dimensions.

The expression of the i th slice in the horizontal direction is given by

$$C_{i..} = \sum_{j=1}^M \sum_{k=1}^M C_{ijk} e_j^{(M)} \circ e_k^{(M)} = \sum_{j=1}^M \sum_{k=1}^M C_{ijk} e_j^{(M)} e_k^{(M)T}. \quad (2.2)$$

In the same manner, the other matrix expressions along with the vertical and frontal directions are expressed, respectively, by

$$C_{..j} = \sum_{i=1}^M \sum_{k=1}^M C_{ijk} e_i^{(M)} e_k^{(M)T}; \quad C_{..k} = \sum_{i=1}^M \sum_{j=1}^M C_{ijk} e_i^{(M)} e_j^{(M)T}. \quad (2.3)$$

It is important to express the tensor slices in an unfolded representation, obtained by stacking up the 2D matrices. Hence, three unfolded representations of C are obtained. For the horizontal, the vertical, and the frontal directions, we get, respectively,

$$C_{[1]} = \begin{pmatrix} C_{1..} \\ C_{2..} \\ \vdots \\ C_{M..} \end{pmatrix}; \quad C_{[2]} = \begin{pmatrix} C_{.1.} \\ C_{.2.} \\ \vdots \\ C_{.M.} \end{pmatrix}; \quad C_{[3]} = \begin{pmatrix} C_{..1} \\ C_{..2} \\ \vdots \\ C_{..M} \end{pmatrix}. \quad (2.4)$$

We note that each matrix $C_{[p]} : p = 1, 2, 3$ is an $(M \times M, M)$ one.

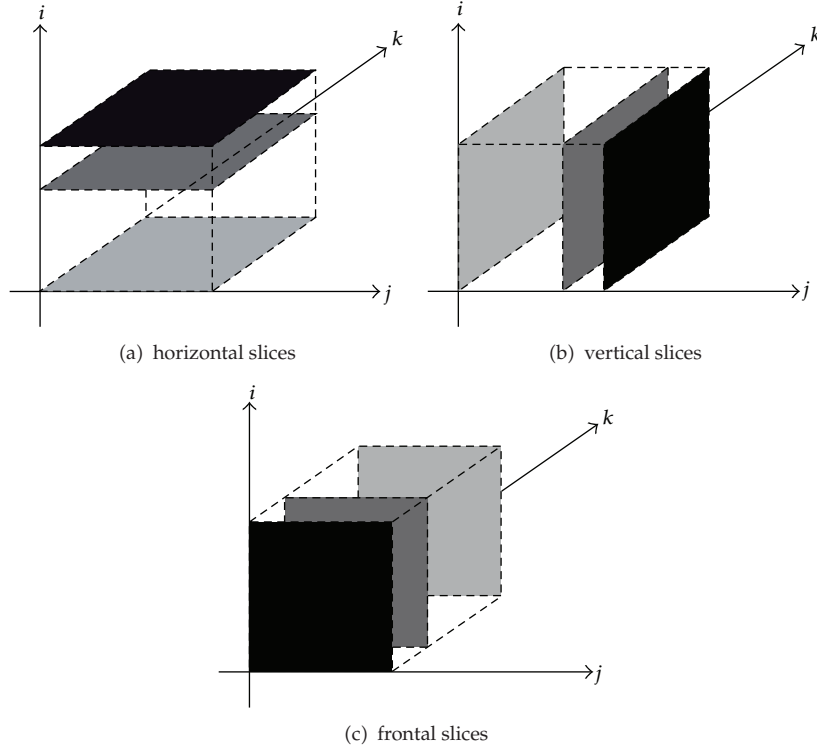


Figure 2: Different direction slices of a cubic tensor.

3. Nonlinear System Model and Output Cumulants Analysis

We focus on the identification of a second-order Volterra-Hammerstein model with finite memory as it is given in [14]:

$$y(n) = \sum_{k=0}^M h_1(k)u(n-k) + \sum_{k=0}^M h_2(k)u^2(n-k); \quad h_1(0) = 1; \quad h_2(0) = 1, \quad h_i(M) \neq 0, \quad (3.1)$$

where $u(n)$ is the input of the system, assumed to be a stationary zero mean Gaussian white random process with $E(u^2(n)) = \gamma_2$. M stands for the model order.

The Hammerstein coefficients vectors h_1 and h_2 are defined by

$$h_p = [h_p(0), h_p(1), \dots, h_p(M)]^T; \quad p = 1; 2. \quad (3.2)$$

As we evoked in the Introduction, identification algorithms based on the computation of 2nd-order output cumulants are sensitive to additive Gaussian noise because 2nd-order cumulants of this latter are in general different to zero. Since the 4th-order cumulants of additive Gaussian noise is null, it will be interesting to use the 4th-order output cumulants to derive identification algorithms. But this will introduce another problem which is the computation complexity. In this paper, we will overcome this shortcoming by using a tensor analysis.

To determine the kernels of this model, we will generate the fourth-order output cumulants. For this purpose, we need to use the standard properties of cumulants and the Leonov-Shiryaev formula for manipulating products of random variables.

The fourth-order output cumulant is given by [15]:

$$\begin{aligned}
 c_{4y}(i_1, i_2, i_3) &= \text{cum}[y(n+i_1), y(n+i_2), y(n+i_3), y(n)] \\
 &= 8\gamma_2^3 \{ \phi_0(i_1, i_2, i_3, 0) + \phi_0(i_1, i_3, i_2, 0) \\
 &\quad + \phi_0(i_2, i_3, i_1, 0) + \phi_0(i_1, 0, i_2, i_3) \\
 &\quad + \phi_0(i_2, 0, i_1, i_3) + \phi_0(i_3, 0, i_1, i_2) \} \\
 &\quad + 48\gamma_2^4 \sum_{l=0}^M h_2(l+i_1)h_2(l+i_2)h_2(l+i_3)h_2(l),
 \end{aligned} \tag{3.3}$$

where

$$\phi_0(i_1, i_2, i_3, i_4) = \sum_{l=0}^M h_1(l+i_1)h_1(l+i_2)h_2(l+i_3)h_2(l+i_4). \tag{3.4}$$

It is easy to verify that $c_{4y}(i_1, i_2, i_3) = 0$ for all $|i_1|, |i_2|, |i_3| > M$.

All the nonzero terms of $c_{4y}(i_1, i_2, i_3)$ are obtained for $(i_1, i_2, i_3) \in [-M, M]^3$. Such a choice allows us to construct a maximal redundant information, in which the fourth-order cumulants are taken for time lags i_1, i_2 , and i_3 within the range $[-M, M]$.

In the sequel we shall present an analysis of a 3rd-order tensor composed of the 4th-order output cumulants.

4. Formulation and Analysis of a Cumulant Cubic Tensor

Let us define the three-dimensional tensor $C^{(4,y)} \in \mathbb{C}^{(2M+1) \times (2M+1) \times (2M+1)}$, in which the element in position (i, j, k) corresponds to $c_{4y}(i_1, i_2, i_3)$, with $i = i_1 + M + 1$; $j = i_2 + M + 1$; $k = i_3 + M + 1$. As $i_1, i_2, i_3 \in [-M, M]$, we get $i, j, k \in [1, 2M + 1]$. Thus,

$$\begin{aligned}
 C_{ijk} &= c_{4y}(i-M-1, j-M-1, k-M-1) \\
 &= 8\gamma_2^3 \{ \phi_0(i-M-1, j-M-1, k-M-1, 0) \\
 &\quad + \phi_0(i-M-1, k-M-1, j-M-1, 0) \\
 &\quad + \phi_0(j-M-1, k-M-1, i-M-1, 0) \\
 &\quad + \phi_0(i-M-1, 0, j-M-1, k-M-1) \\
 &\quad + \phi_0(j-M-1, 0, i-M-1, k-M-1)
 \end{aligned}$$

$$\begin{aligned}
& + \phi_0(k-M-1, 0, i-M-1, j-M-1) \} \\
& + 48\gamma_2^4 \sum_{l=0}^M h_2(l+i-M-1)h_2(l+j-M-1) \\
& \quad \times h_2(l+k-M-1)h_2(l),
\end{aligned} \tag{4.1}$$

$\phi_0(\cdot, \cdot, \cdot, \cdot)$ is given by (3.4). It follows that

$$\begin{aligned}
C_{ijk} = 8\gamma_2^3 \Bigg\{ & \sum_{l=0}^M h_1(l+i-M-1)h_1(l+j-M-1)h_2(l+k-M-1)h_2(l) \\
& + \sum_{l=0}^M h_1(l+i-M-1)h_1(l+k-M-1)h_2(l+j-M-1)h_2(l) \\
& + \sum_{l=0}^M h_1(l+j-M-1)h_1(l+k-M-1)h_2(l+i-M-1)h_2(l) \\
& + \sum_{l=0}^M h_1(l+i-M-1)h_1(l)h_2(l+j-M-1)h_2(l+k-M-1) \\
& + \sum_{l=0}^M h_1(l+j-M-1)h_1(l)h_2(l+i-M-1)h_2(l+k-M-1) \\
& + \sum_{l=0}^M h_1(l+k-M-1)h_1(l)h_2(l+i-M-1)h_2(l+j-M-1) \Bigg\} \\
& + 48\gamma_2^4 \sum_{l=0}^M h_2(l+i-M-1)h_2(l+j-M-1)h_2(l+k-M-1)h_2(l).
\end{aligned} \tag{4.2}$$

Then, expression of the tensor C will be given by

$$\begin{aligned}
C = 8\gamma_2^3 \sum_{i=1}^{2M+1} \sum_{j=1}^{2M+1} \sum_{k=1}^{2M+1} \Bigg\{ & \sum_{l=0}^M h_1(l+i-M-1)h_1(l+j-M-1)h_2(l+k-M-1)h_2(l) \\
& + \sum_{l=0}^M h_1(l+i-M-1)h_1(l+k-M-1)h_2(l+j-M-1)h_2(l) \\
& + \sum_{l=0}^M h_1(l+j-M-1)h_1(l+k-M-1)h_2(l+i-M-1)h_2(l) \\
& + \sum_{l=0}^M h_1(l+i-M-1)h_1(l)h_2(l+j-M-1)h_2(l+k-M-1)
\end{aligned}$$

$$\begin{aligned}
& + \sum_{l=0}^M h_1(l+j-M-1)h_1(l)h_2(l+i-M-1)h_2(l+k-M-1) \\
& + \sum_{l=0}^M h_1(l+k-M-1)h_1(l)h_2(l+i-M-1)h_2(l+j-M-1) \Big\} \\
& \times e_i^{(2M+1)} \circ e_j^{(2M+1)} \circ e_k^{(2M+1)} + 48\gamma_2^4 \\
& \times \sum_{i=1}^{2M+1} \sum_{j=1}^{2M+1} \sum_{k=1}^{2M+1} \sum_{l=0}^M h_2(l+i-M-1) \cdot h_2(l+j-M-1)h_2(l+k-M-1)h_2(l) \\
& \times e_i^{(2M+1)} \circ e_j^{(2M+1)} \circ e_k^{(2M+1)}. \tag{4.3}
\end{aligned}$$

The mathematical development of the expression (4.3) yields to

$$\begin{aligned}
C = 8\gamma_2^3 & \left\{ \sum_{l=0}^M h_2(l)h_{1\bullet l} \circ h_{1\bullet l} \circ h_{2\bullet l} + \sum_{l=0}^M h_2(l)h_{1\bullet l} \circ h_{2\bullet l} \circ h_{1\bullet l} \right. \\
& + \sum_{l=0}^M h_2(l)h_{2\bullet l} \circ h_{1\bullet l} \circ h_{1\bullet l} + \sum_{l=0}^M h_1(l)h_{1\bullet l} \circ h_{2\bullet l} \circ h_{2\bullet l} \\
& \left. + \sum_{l=0}^M h_1(l)h_{2\bullet l} \circ h_{1\bullet l} \circ h_{2\bullet l} + \sum_{l=0}^M h_1(l)h_{2\bullet l} \circ h_{2\bullet l} \circ h_{1\bullet l} \right\} \\
& + 48\gamma_2^4 \sum_{l=0}^M h_2(l)h_{2\bullet l} \circ h_{2\bullet l} \circ h_{2\bullet l}, \tag{4.4}
\end{aligned}$$

where

$$h_{p\bullet l} = \sum_{m=1}^{2M+1} h_p(l+m-M-1)e_m^{(2M+1)}, \quad p = 1; 2. \tag{4.5}$$

This notation leads to define two channel matrices $H_1; H_2 \in \mathbb{C}^{(2M+1) \times (M+1)}$ as follows:

$$H_p \triangleq \mathcal{H}(h_p) = [hp_{\bullet 0}, hp_{\bullet 1}, \dots, hp_{\bullet M}] = \begin{pmatrix} 0 & 0 & \cdots & h_p(0) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & h_p(0) & \cdots & h_p(M-1) \\ h_p(0) & h_p(1) & \cdots & h_p(M) \\ \vdots & \vdots & \ddots & \vdots \\ h_p(M-1) & h_p(M) & \cdots & 0 \\ h_p(M) & 0 & \cdots & 0 \end{pmatrix}, \tag{4.6}$$

with $p = 1; 2$, and $\mathcal{H}(\cdot)$ is the operator that builds a special Hankel matrix from the vector argument as shown above.

Let us compute now the different slices of the proposed tensor.

4.1. Horizontal Slices Expressions

From (2.2) and (4.3), we get

$$\begin{aligned}
 C_{i\bullet\bullet} = & 8\gamma_2^3 \left\{ \sum_{l=0}^M h_2(l) h_1(l+i-M-1) h_{1\bullet l} h_{2\bullet l}^T \right. \\
 & + \sum_{l=0}^M h_2(l) h_1(l+i-M-1) h_{2\bullet l} h_{1\bullet l}^T \\
 & + \sum_{l=0}^M h_2(l) h_2(l+i-M-1) h_{1\bullet l} h_{1\bullet l}^T \\
 & + \sum_{l=0}^M h_1(l) h_1(l+i-M-1) h_{2\bullet l} h_{2\bullet l}^T \\
 & + \sum_{l=0}^M h_1(l) h_2(l+i-M-1) h_{1\bullet l} h_{2\bullet l}^T \\
 & \left. + \sum_{l=0}^M h_1(l) h_2(l+i-M-1) h_{2\bullet l} h_{1\bullet l}^T \right\} \\
 & + 48\gamma_2^4 \sum_{l=0}^M h_2(l) h_2(l+i-M-1) h_{2\bullet l} h_{2\bullet l}^T,
 \end{aligned} \tag{4.7}$$

which can be written as

$$\begin{aligned}
 C_{i\bullet\bullet} = & 8\gamma_2^3 \left\{ H_1 \text{diag}_i(H_1) \Sigma_2^T + \Sigma_2 \text{diag}_i(H_1) H_1 \right. \\
 & + H_1 \text{diag}_i(\Sigma_2) H_1^T + H_2 \text{diag}_i(\Sigma_1) H_2^T \\
 & \left. + \Sigma_1 \text{diag}_i(H_2) H_2^T + H_2 \text{diag}_i(H_2) \Sigma_1^T \right\} \\
 & + 48\gamma_2^4 H_2 \text{diag}_i(\Sigma_2) H_2^T,
 \end{aligned} \tag{4.8}$$

where $\Sigma_1 = H_1 \text{diag}(h_1)$; $\Sigma_2 = H_2 \text{diag}(h_2)$; $\text{diag}_n(\cdot)$ is the diagonal matrix formed by the n th line of its argument.

It can easily be demonstrated that

$$C_{m\bullet\bullet} = C_{\bullet m\bullet} = C_{\bullet\bullet m} = C_{m\bullet} \tag{4.9}$$

It follows that

$$\begin{aligned}
 C_m = 8\gamma_2^3 \big\{ & H_1 \text{diag}_m(H_1) \Sigma_2^T + \Sigma_2 \text{diag}_m(H_1) H_1^T \\
 & + H_1 \text{diag}_m(\Sigma_2) H_1^T + H_2 \text{diag}_m(\Sigma_1) H_2^T \\
 & + \Sigma_1 \text{diag}_m(H_2) H_2^T + H_2 \text{diag}_m(H_2) \Sigma_1^T \big\} \\
 & + 48\gamma_2^4 H_2 \text{diag}_m(\Sigma_2) H_2^T,
 \end{aligned} \tag{4.10}$$

with $m = 1, \dots, 2M + 1$.

The expression of the unfolded tensor representation is given by

$$\begin{aligned}
 C_{[u]} = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_{2M+1} \end{pmatrix} &= 8\gamma_2^3 \underbrace{\left\{ \begin{pmatrix} H_1 \text{diag}_1(H_1) \Sigma_2^T \\ H_1 \text{diag}_2(H_1) \Sigma_2^T \\ \vdots \\ H_1 \text{diag}_{2M+1}(H_1) \Sigma_2^T \end{pmatrix} + \dots \right\}}_{\text{the 6th terms with respect to } C_m \text{ in (4.10)}} \\
 &+ 48\gamma_2^4 \left\{ \begin{pmatrix} H_2 \text{diag}_1(\Sigma_2) H_2^T \\ H_2 \text{diag}_2(\Sigma_2) H_2^T \\ \vdots \\ H_2 \text{diag}_{2M+1}(\Sigma_2) H_2^T \end{pmatrix} \right\}.
 \end{aligned} \tag{4.11}$$

To develop this expression, we need the following property.

Property 1. Let A be the matrix with dimensions (M, N) and B the matrix with dimensions (M', N) , then

$$\begin{pmatrix} \text{Adiag}_1(B) \\ \text{Adiag}_2(B) \\ \vdots \\ \text{Adiag}_{M'}(B) \end{pmatrix} = B \diamond A, \tag{4.12}$$

where \diamond stands for the Khatri-Rao product.

It becomes that

$$C_{[U]} = 8\gamma_2^3 \left\{ \begin{array}{l} (H_1 \diamond H_1) \text{diag}(h_2)H_2^T + (H_1 \diamond H_2) \text{diag}(h_2)H_1^T \\ + (H_2 \diamond H_1) \text{diag}(h_2)H_1^T + (H_1 \diamond H_2) \text{diag}(h_1)H_2^T \\ + (H_2 \diamond H_1) \text{diag}(h_1)H_2^T + (H_2 \diamond H_2) \text{diag}(h_1)H_1^T \end{array} \right\} + 48\gamma_2^4 ((H_2 \diamond H_2) \text{diag}(h_2)H_2^T). \quad (4.13)$$

5. Blind Identification System with Cumulant Tensor

To estimate the Volterra-Hammerstein kernels and to avoid the computation of $H_p : p = 1, 2$, we will use the following Khatri-Rao property to propose an Iterative Alternating Least Square (IALS) procedure.

Property 2. If matrices $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{n \times m}$ and vector $d \in \mathbb{C}^n$ are such that $X = \text{Adiag}(d)B$, then it holds that $\text{vec}(X) = (B^T \diamond A)d$, where $\text{vec}(\cdot)$ stands for the vectorizing operator.

Applying this property to (4.13), it is straightforward to write

$$\begin{aligned} \text{vec}(C_{[U]}) &= 8\gamma_2^3 \left\{ \begin{array}{l} (H_2 \diamond H_1 \diamond H_1)h_2 + (H_1 \diamond H_1 \diamond H_2)h_2 \\ + (H_1 \diamond H_2 \diamond H_1)h_2 + (H_2 \diamond H_1 \diamond H_2)h_1 \\ + (H_2 \diamond H_2 \diamond H_1)h_1 + (H_1 \diamond H_2 \diamond H_2)h_1 \end{array} \right\} \\ &\quad + 48\gamma_2^4 ((H_2 \diamond H_2 \diamond H_2)h_2) \\ &= 8\gamma_2^3 \{ (H_2 \diamond H_1 \diamond H_2) + (H_2 \diamond H_2 \diamond H_1) + (H_1 \diamond H_2 \diamond H_2) \} h_1 \\ &\quad + \left\{ 8\gamma_2^3 ((H_2 \diamond H_1 \diamond H_1) + (H_1 \diamond H_1 \diamond H_2) + (H_1 \diamond H_2 \diamond H_1)) \right. \\ &\quad \left. + 48\gamma_2^4 ((H_2 \diamond H_2 \diamond H_2)) \right\} h_2. \end{aligned} \quad (5.1)$$

Let A and B be

$$\begin{aligned} A &= 8\gamma_2^3 \{ (H_2 \diamond H_1 \diamond H_2) + (H_2 \diamond H_2 \diamond H_1) + (H_1 \diamond H_2 \diamond H_2) \}, \\ B &= 8\gamma_2^3 ((H_2 \diamond H_1 \diamond H_1) + (H_1 \diamond H_1 \diamond H_2) + (H_1 \diamond H_2 \diamond H_1)) + 48\gamma_2^4 ((H_2 \diamond H_2 \diamond H_2)). \end{aligned} \quad (5.2)$$

The problem of the blind nonlinear identification will be expressed as

$$Ah_1 + Bh_2 = \text{vec}(C_U). \quad (5.3)$$

This system can be solved using several methods. We propose to resolve it using the Iterative Alternating Least Square algorithm (IALS).

6. Cost Functions and Iterative Alternating Least Square Algorithm

To apply the IALS algorithm, we suppose alternatively that Ah_1 or Bh_2 is a constant vector. Then, we get two cost functions to be minimized. Assuming that the vector Bh_2 is constant, the first cost function will be expressed by

$$E_1(h_1) = \|\text{vec}(C_U) - Bh_2 - Ah_1\|^2. \quad (6.1)$$

For the second cost function, we assume that Ah_1 is constant; thus

$$E_2(h_2) = \|\text{vec}(C_U) - Ah_1 - Bh_2\|^2. \quad (6.2)$$

The application of the least mean squares algorithm to these two functions leads to the following solutions:

$$\begin{aligned} h_1 &= A^\#(\text{vec}(C_U) - Bh_2), \\ h_2 &= B^\#(\text{vec}(C_U) - Ah_1), \end{aligned} \quad (6.3)$$

where the subscript # denotes the Moore-Penrose pseudoinverse of the corresponding matrix.

Finally, the different steps of the proposed IALS algorithm are summarized in Algorithm 1.

The notation \hat{x} stands for the estimates of the parameter x .

7. Convergence Analysis

Equation (6.3) shows that the ALS algorithm converges to optimal solutions if and only if the Moore-Penrose pseudoinverse matrices $A^\#$ and $B^\#$ exist, which implies that matrices A and B must be full rank [14, 16]. To do this, we start by affirming that, due to the Hankel structure and the assumption that $h_i(M) \neq 0$ (3.1), each of the matrices H_1 and H_2 is full rank. Then

$$\text{rank}(H_1) = \text{rank}(H_2) = M + 1. \quad (7.1)$$

Let us now find out the rank of matrices $H_i \diamond H_j \diamond H_k$; $i, j, k \in \{1, 2\}$ obtained from Khatri-Rao product (5.1). We will make use of the following definition and property defining the k -rank of a matrix and the rank of a Khatri-Rao product of two matrices [17].

Definition 7.1. The rank of a matrix $A \in \mathbb{C}^{E \times F}$ (denoted by k_A) is equal to k if and only if every k columns of A are linearly independent. Note that $k_A \leq \min(E, F)$, for all A .

This means that the rank of the matrix A is the largest integer k for which every set containing k columns of A is independent.

Property 3. Consider the Khatri-Rao product $A \diamond B$, where A is $E \times F$ and B is $G \times F$. If neither A nor B contains a zero column (and hence $k_A \geq 1$, $k_B \geq 1$), then $k_{A \diamond B} \geq \min(k_A + k_B - 1, F)$.

Initialize h_1 and h_2 as random variables (estimates $\hat{h}_1^{(0)}, \hat{h}_2^{(0)}$).

For $n \geq 1$,

(i) build Hankel matrices using (4.5), for $p = 1; 2$,

$$\widehat{H}_1^{(n)} = \mathcal{H}(\hat{h}_1^{(n)}), \quad \widehat{H}_2^{(n)} = \mathcal{H}(\hat{h}_2^{(n)}),$$

(ii) compute matrices estimate \widehat{A} and \widehat{B} as

$$\widehat{A}^{(n)} = 8\gamma_2^3 \{(\widehat{H}_2 \diamond \widehat{H}_1 \diamond \widehat{H}_2) + (\widehat{H}_2 \diamond \widehat{H}_2 \diamond \widehat{H}_1) + (\widehat{H}_1 \diamond \widehat{H}_2 \diamond \widehat{H}_2)\}^{(n)},$$

$$\widehat{B}^{(n)} = \{8\gamma_2^3 ((\widehat{H}_2 \diamond \widehat{H}_1 \diamond \widehat{H}_1) + (\widehat{H}_1 \diamond \widehat{H}_1 \diamond \widehat{H}_2) + (\widehat{H}_1 \diamond \widehat{H}_2 \diamond \widehat{H}_1)) + 48\gamma_2^4 ((\widehat{H}_2 \diamond \widehat{H}_2 \diamond \widehat{H}_2))\}^{(n)},$$

(iii) minimize cost functions (6.1) and (6.2) so that

$$\begin{aligned} \hat{h}_1^{(n+1)} &= \widehat{A}^{(n)\#}(\text{vec}(C_U) - \widehat{B}^{(n)}\hat{h}_2^{(n)}), \\ \hat{h}_2^{(n+1)} &= \widehat{B}^{(n)\#}(\text{vec}(C_U) - \widehat{A}^{(n)}\hat{h}_1^{(n)}), \end{aligned}$$

(iv) reiterate until parametric error convergence

$$\frac{\|\text{vec}((\hat{h}_1^{(n+1)} \quad \hat{h}_2^{(n+1)}) - (\hat{h}_1^{(n)} \quad \hat{h}_2^{(n)}))\|}{\|\text{vec}(\hat{h}_1^{(n+1)} \quad \hat{h}_2^{(n+1)})\|} \leq \varepsilon.$$

Algorithm 1: Different steps of the new blind identification algorithm-based cumulant tensor analysis.

It follows that

$$k_{H_i \diamond H_j} \geq \min(2M + 1, M + 1), \quad \forall i, j \in \{1, 2\}, \quad (7.2)$$

which is equivalent to

$$k_{H_i \diamond H_j} \geq M + 1. \quad (7.3)$$

Due to the definition of the Khatri-Rao product and the structure of the Hankel matrices H_i ; $i \in \{1, 2\}$, we conclude that

$$k_{H_i \diamond H_j} = M + 1 = \text{rank}(H_i \diamond H_j). \quad (7.4)$$

Consequently,

$$k_{(H_i \diamond H_j) \diamond H_k} = M + 1 = \text{rank}(H_i \diamond H_j \diamond H_k), \quad \forall i, j, k \in \{1, 2\}, \quad (7.5)$$

which means that each matrix $H_i \diamond H_j \diamond H_k$ is full rank whatever the values taken by i and j in the set $\{1, 2\}$.

Let us now find out the rank of matrices A and B . For this purpose, we will study the structure of the matrix $H_i \diamond H_j \diamond H_k$. Recall that H_i is a $(2M + 1, M + 1)$ matrix. Let $\Theta_i = [0 \ 0 \ \cdots \ 0]^T$ be the zero column vector of dimension $(2M + 1)$; $i = 1, \dots, M$.

Then, for the matrix $H_i \diamond H_j \diamond H_k$, we will have the following form:

$$H_i \diamond H_j \diamond H_k = \begin{pmatrix} \Theta_1 & \Theta_1 & & \Theta_1 & X_M \\ \Theta_2 & \vdots & & X_{M-1} & \Theta_1 \\ \vdots & \Theta_{M-1} & \cdots & \Theta_2 & \Theta_2 \\ \Theta_M & X_1 & & \vdots & \vdots \\ X_0 & \Theta_M & & \Theta_{M-1} & \Theta_M \end{pmatrix}, \quad (7.6)$$

where X_i stands for the column vector of dimension $(2M+1)^2(M+1)$ which is constituted by products of the kernels model arising from computation of the Khatri-Rao matrix product. We have seen that $H_i \diamond H_j \diamond H_k$ is full rank. The sum of different matrices $H_i \diamond H_j \diamond H_k$ has the same form of $H_i \diamond H_j \diamond H_k$ whatever the system order and the values taken by i , j and k . Consequently, matrices A and B are full rank and then their pseudoinverse exist. We conclude that the IALS converges to an optimal solution in least mean squares sense.

8. Simulation Results

In this section, simulation results will be given to illustrate the performance of the proposed algorithm. Two identification Volterra-Hammerstein systems are considered:

$$\begin{aligned} \text{System 1: } & \begin{cases} z(n) = u(n) - 0.25u(n-1) + 0.9u(n-2) \\ \quad + u^2(n) + 0.5u^2(n-1) - 0.35u^2(n-2), \\ y(n) = z(n) + e(n), \end{cases} \\ \text{System 2: } & \begin{cases} z(n) = u(n) - 0.4u(n-1) + 0.5u(n-2) + 0.95u(n-3) \\ \quad + u^2(n) + 0.2u^2(n-1) - 0.7u^2(n-2) + 0.6u^2(n-3), \\ y(n) = z(n) + e(n). \end{cases} \end{aligned} \quad (8.1)$$

The input sequence $u(n)$ is assumed to be stationary, zero mean, white Gaussian noise with variance $\gamma_2 = 1$. The noise signal $e(n)$ is also assumed to be white Gaussian sequence and independent of the input. The parameter estimation was performed for two different signal-to-noise ratio (SNR) levels: 20 dB and 3 dB.

The SNR is computed with the following expression:

$$\text{SNR} = \frac{E(z^2(n))}{E(e^2(n))}. \quad (8.2)$$

Fourth-order cumulants were estimated from different lengths of output sequences ($N = 4096$ and $N = 16384$) assuming perfect knowledge of the system model. To reduce the realization dependency, parameters were averaged over 500 Monte-Carlo runs. For each

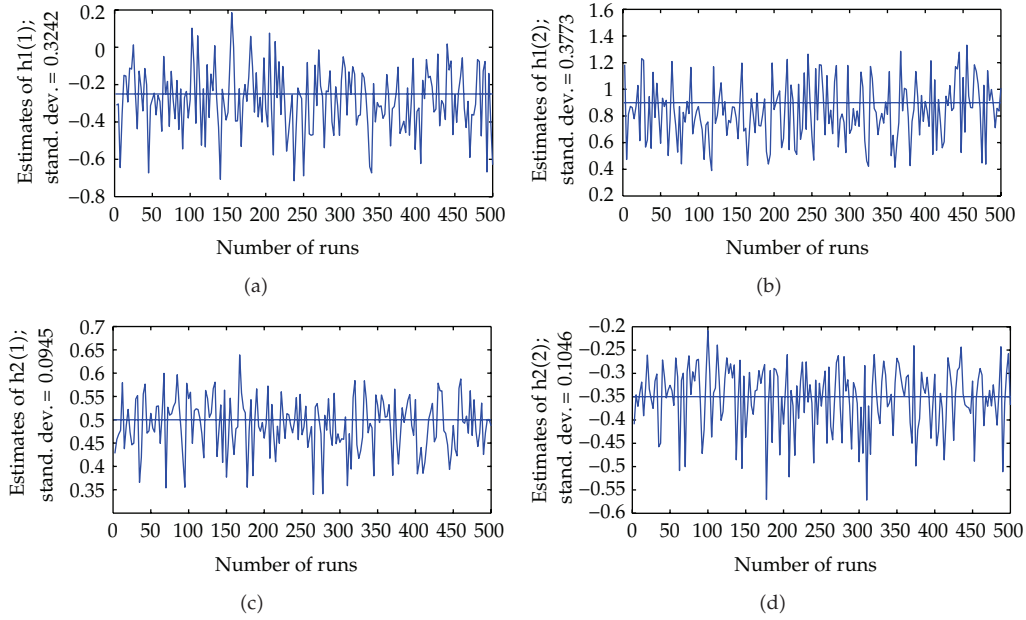


Figure 3: Estimates of the parameters of System 1 with the IALS algorithm for $N = 4096$ and $\text{SNR} = 3$ dB.

simulation, we give the curves representing the variation of the estimates along with the Monte-Carlo runs, and we resume exclusive results in different tables.

System 1

Figures 3 and 4 show the estimates of the different kernels of the proposed model, with the IALS algorithm, for $N = 4096$ and for different SNR levels (3 dB and 20 dB).

The mean and the standard deviation of the estimated kernels against the true ones are shown in Table 1.

Likewise, Figures 5 and 6 show the estimates of the different kernels of System 1 for $N = 16384$ and for SNR levels equal to 3 dB and 20 dB, while, in Table 2, the mean and the standard deviation of the estimated kernels against the true ones are shown.

From these results, we observe that the proposed IALS algorithm performs well generating estimates for a large variation of the SNR (from 20 dB to 3 dB). We also note that the standard deviation is relatively large and decreases with the number of the system observations.

System 2

Figures 7 and 8 show the estimates of the different kernels of the second proposed model for $N = 4096$ and for different SNR (20 dB and 3 dB).

The mean and the standard deviation of the estimated kernels against the true ones are shown in Table 3.

Figures 9 and 10 show the estimates of the different kernels of System 2 for $N = 16384$ and for different SNR (20 dB and 3 dB), while, in Table 4, the mean and the standard deviation of the estimated kernels against the true ones are shown. The mean and the standard

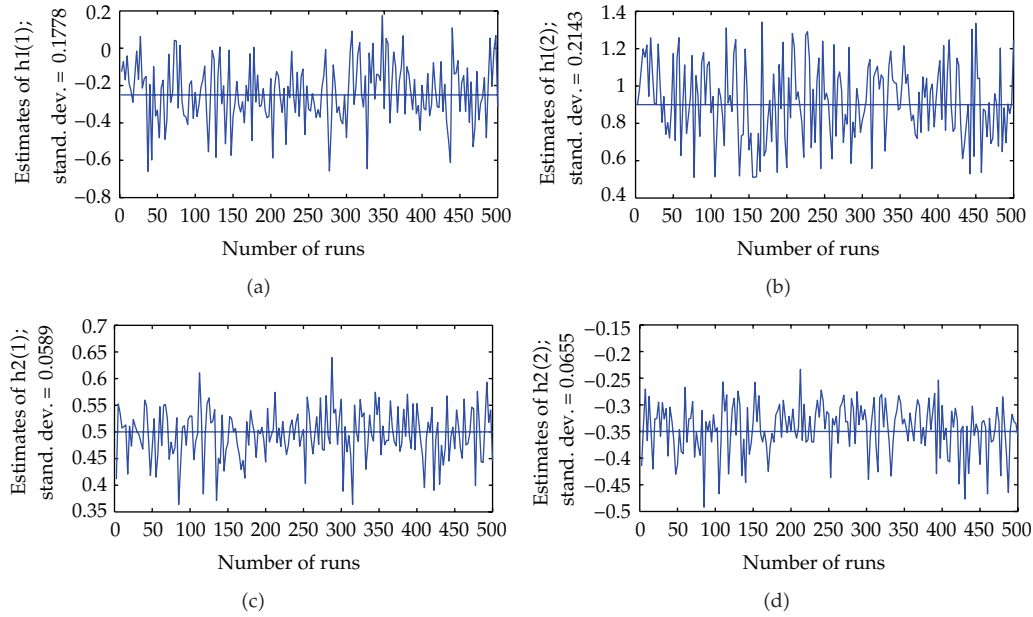


Figure 4: Estimates of the parameters of System 1 with the IALS algorithm for $N = 4096$ and $\text{SNR} = 20$ dB.

Table 1: True and estimated values of the kernels of System 1 for $N = 4096$ (500 Monte-Carlo runs).

True parameters	3 dB		20 dB	
	Mean	St. Dev.	Mean	St. Dev.
-0.25	-0.2401	0.3242	-0.2456	0.1778
0.9	0.9985	0.3773	0.8837	0.2143
0.5	0.5128	0.0945	0.4815	0.0589
-0.35	-0.3411	0.1046	-0.3521	0.0655

Table 2: True and estimated values of the kernels of System 2 for $N = 16384$ (500 Monte-Carlo runs).

True parameters	3 dB		20 dB	
	Mean	St. Dev.	Mean	St. Dev.
-0.25	-0.2464	0.2520	-0.2589	0.1576
0.9	0.9238	0.2947	0.8884	0.1976
0.5	0.4900	0.0679	0.5028	0.0482
-0.35	-0.3665	0.0753	0.3478	0.0508

deviation of the estimated kernels against the true ones, for the second system, are shown in Table 4.

From these results, we note also that the proposed algorithm provides good estimates for the proposed system. The number of observations N affects the range of variation of the standard deviation values. Indeed, for important values of N , this range becomes so small. The method provides good estimates even for low levels of SNR. Furthermore, we note that the larger the Monte-Carlo runs number, the smaller the standard deviations are.

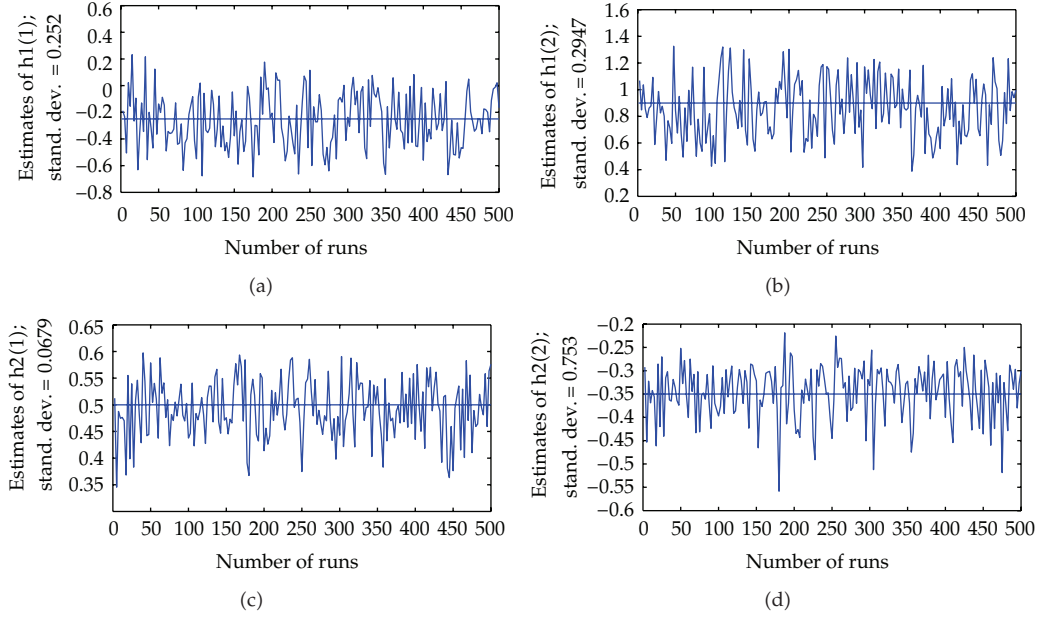


Figure 5: Estimates of the parameters of System 1 with the IALS algorithm for $N = 16384$ and SNR = 3 dB.

Table 3: True and estimated values of the kernels of System 2 for $N = 4096$ (500 Monte-Carlo runs).

True parameters	3 dB		20 dB	
	Mean	St. Dev.	Mean	St. Dev.
-0.4	-0.4512	0.6550	-0.4293	0.1594
0.5	0.5074	0.7632	0.5173	0.1968
0.95	1.1556	0.5329	0.9390	0.1545
0.2	0.2358	0.2525	0.2157	0.0958
-0.7	-0.6794	0.2855	-0.6814	0.1260
0.6	0.5872	0.2268	0.5962	0.0986

8.1. Comparison with Existing Methods

The performance of the previous algorithm was compared with two works: the algorithm proposed in [9] (will be noted as BIL to blind identification with linearization) and the Lagrange Programming Neural Network (LPNN) proposed in [13].

- (i) In [14], the problem of blind identification was converted into a linear multivariable form using Kronecker product of the output cumulants. This can be described by the following equations:

$$\bar{C}_y^k(q, \tau) = b(\tau) \left(\bar{\Gamma}_{kw} \right)_{p \times p} b^T(q), \quad (8.3)$$

where $C_y^k(\tau_1, \tau_2, \dots, \tau_{k-1})$ denotes the output cumulants sequence of order k , Γ_{kw} is the intensity (zero lag cumulant) of order k of the vector W which is formed in its

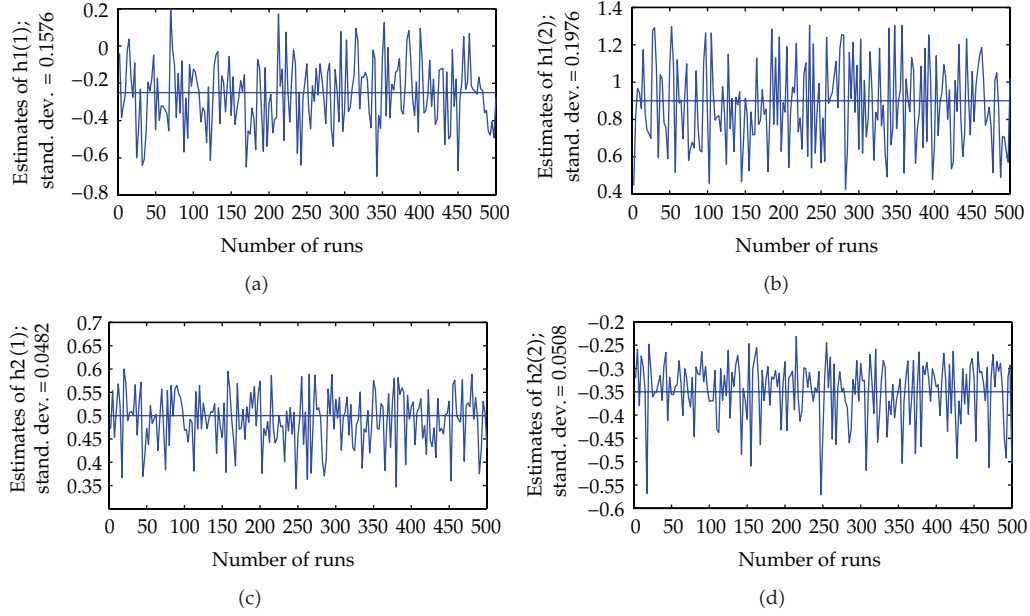


Figure 6: Estimates of the parameters of System 1 with the IALS algorithm for $N = 16384$ and SNR = 20 dB.

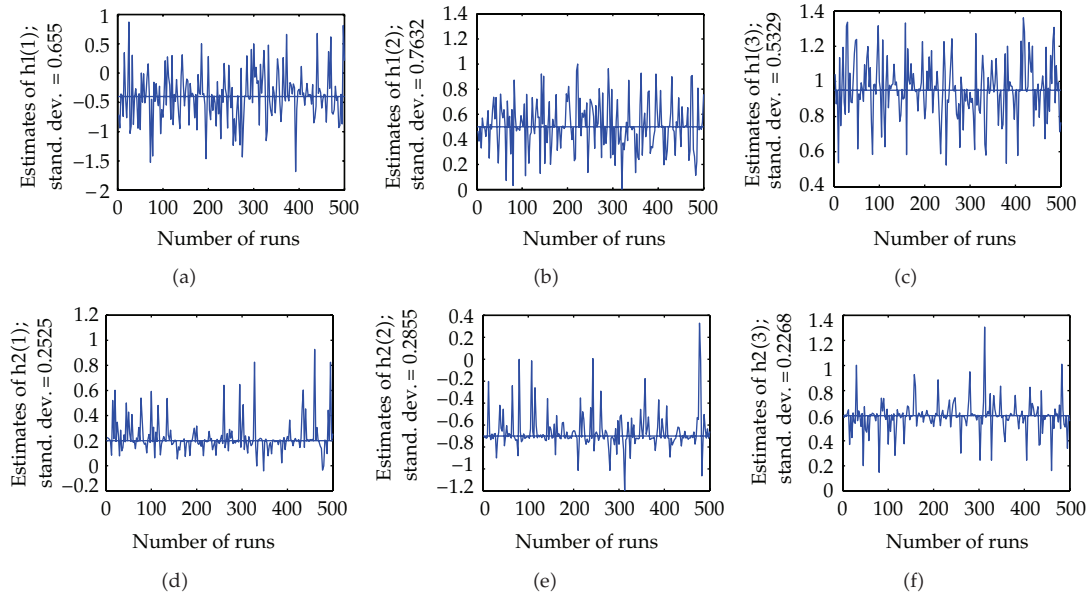


Figure 7: Estimates of the parameters of System 2 with the IALS algorithm for $N = 4096$ and SNR = 3 dB.

turn by the different powers of input, and b is the kernel vector.
For $\tau = 0$, this becomes

$$\bar{C}_y^k(q, 0) = b(0) \left(\bar{\Gamma}_{kw} \right)_{p \times p} b^T(q) = \left(\bar{\Gamma}_{kw} \right)_{p \times p} b^T(q). \quad (8.4)$$

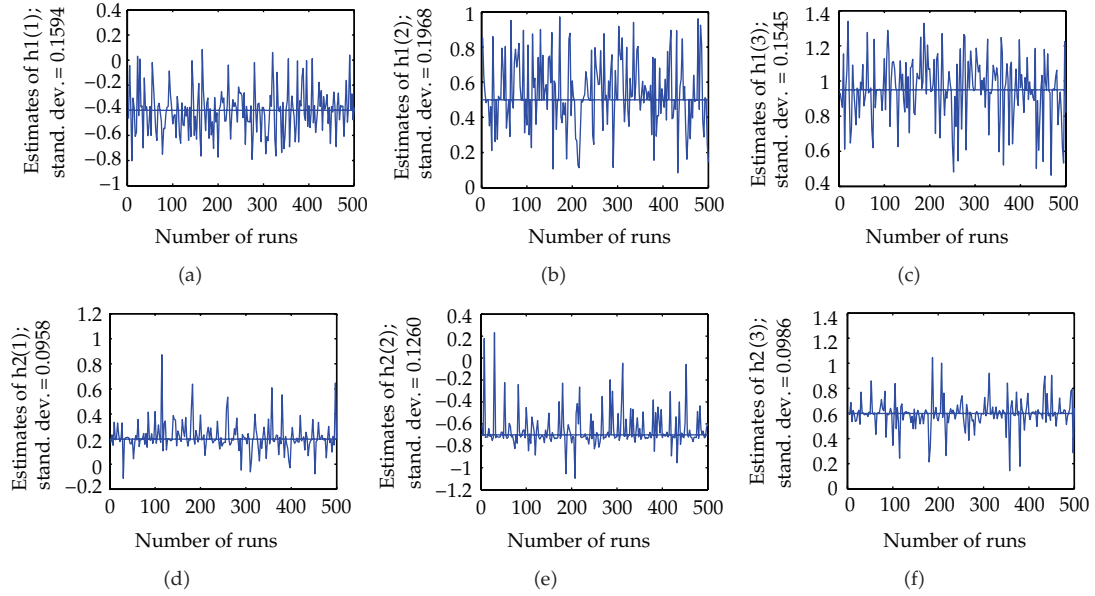


Figure 8: Estimates of the parameters of System 2 with the IALS algorithm for $N = 4096$ and $\text{SNR} = 20$ dB.

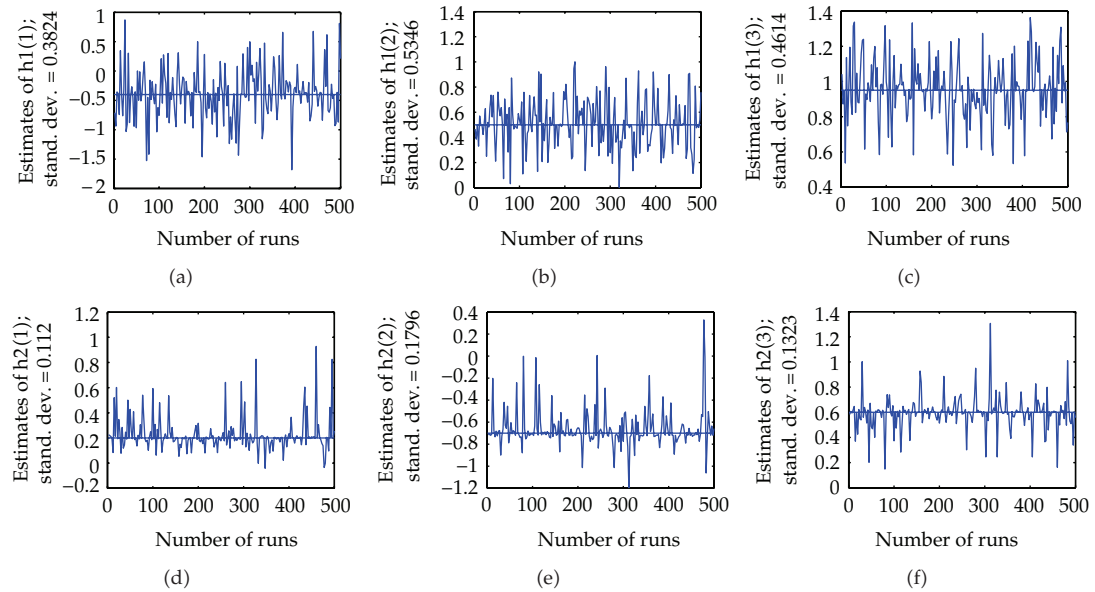


Figure 9: Estimates of the parameters of System 2 with the IALS algorithm for $N = 16384$ and $\text{SNR} = 3$ dB.

Different important scenarios were discussed and successfully resolved. Here, we are interested in the case of Gaussian input when the input statistics are known. Despite the efficiency of the proposed method, the resulting algorithms are in general cumbersome especially for the high series order (As confirmed by authors). For more details, see [14].

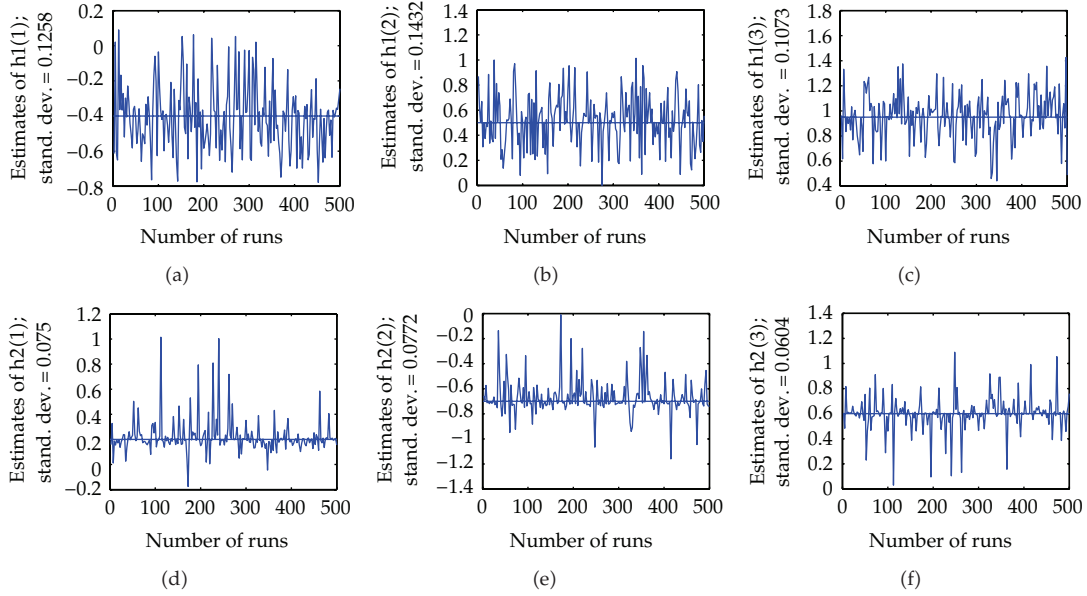


Figure 10: Estimates of the parameters of System 2 with the IALS algorithm for $N = 16384$ and SNR = 20 dB.

Table 4: True and estimated values of the kernels of System 2 for $N = 16384$ (500 Monte-Carlo runs).

True parameters	3 dB		20 dB	
	Mean	St. Dev.	Mean	St. Dev.
-0.4	-0.4378	0.3824	-0.4121	0.1258
0.5	0.5116	0.5346	0.5122	0.1432
0.95	0.9450	0.4614	0.9507	0.1073
0.2	0.1847	0.1120	0.2143	0.0750
-0.7	-0.6877	0.1796	-0.6941	0.0772
0.6	0.6014	0.1323	0.5983	0.0604

- (ii) In their work [13], authors tried to determine the different Volterra kernels and the variance of the input from the autocorrelation estimates $\rho[k]$ and the third-order moments estimates $\mu[k, l]$ of the system output, using the Lagrange Programming Neural Network (LPNN). As the LPNN is essentially designed for general nonlinear programming, they expressed the identification problem as follows:

$$\begin{aligned}
 \text{Minimize: } L(f) &= \sum_i \sum_j (\mu[i, j] - M[i, j, f])^2, \\
 \text{Subject to: } \rho[i] &= R[i, j],
 \end{aligned} \tag{8.5}$$

where $R[i, j]$ is the autocorrelation function of the real process $y[n]$ and $M[i, j]$ is its third order moment sequence. f is the vector formed by the unknown parameters

of the Volterra model and the unknown variance of the driving noise. So the Lagrangian function will be written as

$$L(f, \lambda) = L(f) + \sum_i \lambda_i (\rho[i] - R[i, f]). \quad (8.6)$$

To improve the convergence and the precision of the algorithm, authors extended the preceding function by defining the Augmented Lagrangian Function such as

$$L(f, \lambda) = L(f) + \sum_i \lambda_i (\rho[i] - R[i, f]) + \beta \sum_i (\rho[i] - R[i, f])^2, \quad (8.7)$$

where $\{\beta_k\}$ is a penalty parameter sequence satisfying $0 < \beta_k < \beta_{k+1}$ for all k , $\beta_k \rightarrow \infty$. So the back-propagation algorithm can be established using the Lagrange multiplier.

The performance of the new proposed algorithm was compared with these two algorithms. Each of these algorithms was used to identify the two models presented above (8.1), for the case of Gaussian excitation, $N = 16384$ samples, and for the tow proposed SNR levels 3 dB and 20 dB.

Figures 11 and 12 show a comparison of the standard deviations given by each algorithm. We note that these results may vary considerably depending on the number of the output observations. These results show that the new proposed algorithm performs well. For a small number of unknown parameters, we note that all algorithms give in general the same STD values and these values decrease by increasing the SNR values. We note furthermore that BIL algorithm is so complex for programming in comparison with the LPNN and the new one. For big number of unknown parameters, the BIL algorithm becomes very computational complex and even the LPNN, while the new algorithm keeps its simplicity and provides good parameters with very competitive STD values.

9. Conclusion

In this paper, a new approach for blind nonlinear identification problem of a second-order Hammerstein-Volterra system is developed. Thanks to a matrix analysis of a cubic tensor composed of the fourth-order output cumulants, the nonlinear identification problem is reduced to a system having the following general form: $Ax + By = c$. This system is solved using the Iterative Alternating Least Square. A convergence analysis shows that matrices A and B are full rank which means that the IALS algorithm converges to optimal solutions in the least mean squares sense. Simulation results on two different systems show good performance of the proposed algorithm. It is noted also that the different values of the estimates improve with the number of the system observation even for small values of SNR. Comparison results with two algorithms show that the new proposed algorithm performs well and especially in the case of great number of unknown parameters. Extending the proposed algorithm for more input classes and for more general Volterra-Hammerstein systems remains an open problem, and it is now the subject matter of current works.

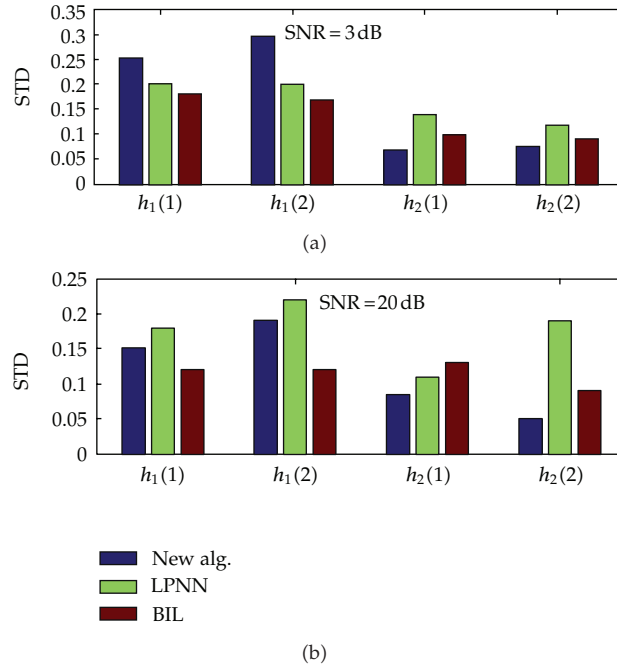


Figure 11: Comparison of the standard deviations (STDs) of the new algorithm against those of the LPNN and the BIL algorithms: System 1.

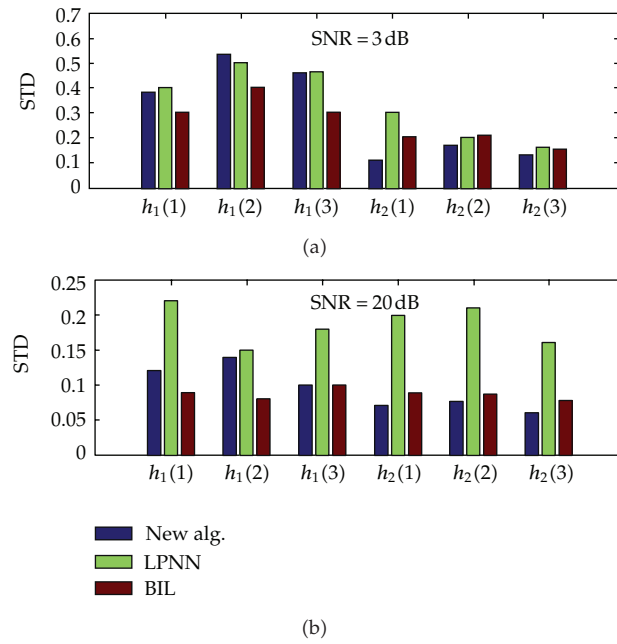


Figure 12: Comparison of the standard deviations (STDs) of the new algorithm against those of the LPNN and the BIL algorithms: System 2.

Acknowledgment

The authors would like to thank Prof. Gerard Favier, Laboratory Director (I3S Laboratory–CNRS, University of Nice Sophia Antipolis, France) for his support, advice, and corrections to carry out this work.

References

- [1] M. Li, C. Cattani, and S. Y. Chen, "Viewing sea level by a one-dimensional random function with long memory," *Mathematical Problems in Engineering*, vol. 2011, Article ID 654284, 2011.
- [2] S. C. Lim, C. H. Eab, K. H. Mak, M. Li, and S. Y. Chen, "Solving linear coupled fractional differential equations by direct operational method and some applications," *Mathematical Problems in Engineering*, vol. 2012, Article ID 653939, 28 pages, 2012.
- [3] S. Y. Chen and Q. Guan, "Parametric shape representation by a deformable NURBS model for cardiac functional measurements," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 3, pp. 480–487, 2011.
- [4] S. Chen, H. Tong, and C. Cattani, "Markov models for image labeling," *Mathematical Problems in Engineering*, vol. 2012, Article ID 814356, 18 pages, 2012.
- [5] E. G. Bakhoun and C. Toma, "Specific mathematical aspects of dynamics generated by coherence functions," *Mathematical Problems in Engineering*, vol. 2011, Article ID 436198, 2011.
- [6] L. Agarossi, S. Bellini, F. Bregoli, and P. Migliorati, "Simplified nonlinear receivers for the optical disc channel," in *Proceedings of the IEEE International Conference on Communications (ICC '99)*, vol. 3, pp. 1945–1950, June 1999.
- [7] K. Watanabe, Y. W. Chen, and K. Yamashita, "Blind nonlinear equalization using 4th -order cumulants," in *Proceedings of the Sice 2002*, pp. 2128–2131, Osaka, Japan, August 2002.
- [8] I. Cherif and F. Fnaiech, "Blind identification of a second order Volterra-Hammerstein series using cumulant cubic tensor analysis," in *Proceedings of the 34th Annual Conference of the IEEE Industrial Electronics Society (IECON '08)*, pp. 1851–1856, Orlando, Fla, USA, November 2008.
- [9] N. Kalouptsidis and P. Koukoulas, "Blind identification of volterra-hammerstein systems," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2777–2787, 2005.
- [10] G. Favier, A. Y. Kibangou, and T. Bouilloc, "Nonlinear system modeling and identification using Volterra-PARAFAC models," *International Journal of Adaptive Control and Signal Processing*, vol. 26, pp. 30–53, 2012.
- [11] T. Bouilloc and G. Favier, "Nonlinear channel modeling and identification using baseband VolterraParafac models," *Signal Processing*, vol. 92, no. 6, pp. 1492–1498, 2012.
- [12] G. Favier, M. N. da Costa, A. L. F. de Almeida, and J. M. T. Romano, "Tensor space-time (TST) coding for MIMO wireless communication systems," *Signal Processing*, vol. 92, pp. 1079–1092, 2012.
- [13] T. Stathaki and A. Scohyers, "Constrained optimization approach to the blind estimation of Volterra kernels," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, vol. 3, pp. 2373–2376, April 1997.
- [14] C. E. R. Fernandes, G. Favier, and J. C. M. Mota, "Tensor-based blind channel identification," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 2728–2732, June 2007.
- [15] P. Koukoulas and N. Kalouptsidis, "Blind identification of second order Hammerstein series," *Signal Processing*, vol. 83, no. 1, pp. 213–234, 2003.
- [16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.
- [17] Y. Hua, A. B. Gershman, and Q. Cheng, *High Resolution and Robust Signal Processing*, Marcell Dekker, New York, NY, USA, 2004.

