

Tensor-based methods for system identification Part 2: Three examples of tensor-based system identification methods

G rard Favier¹ and Alain Y. Kibangou²

¹ Laboratoire I3S, University of Nice Sophia Antipolis, CNRS, Les Algorithmes- B t. Euclide B, 2000 Route des lucioles, B.P. 121 - 06903 Sophia Antipolis Cedex, France
favier@i3s.unice.fr

² LAAS-CNRS, University of Toulouse, 7 avenue Colonel Roche, 31077 Toulouse, France
akibango@laas.fr

Abstract. In this second part of the paper we present three examples for illustrating tensor-based system identification: 1) blind identification of SISO FIR linear systems and instantaneous MIMO linear mixtures; 2) supervised identification of homogeneous cubic Volterra systems; 3) supervised identification of Wiener-Hammerstein systems.

Key words: Tensor models; System identification; Blind identification; Volterra systems; Wiener-Hammerstein systems.

1 Introduction

System modeling and identification play an important role in automatic control and signal processing. The identification can be carried out in a supervised or unsupervised (blind) way depending if a training sequence (i.e. a known input signal) is used or not. For instance, in seismic applications, speech processing or wireless communication systems, the input signal is generally unknown, and blind estimation methods are needed, while in automatic control applications the input signal is always known, which allows a supervised identification from input-output signal measurements. On the other hand, most of physical systems are nonlinear by nature, which explains why nonlinear models like Volterra and block-oriented models are very useful for many applications. Among block-oriented models, the Wiener-Hammerstein one (also called Sandwich or LNL model, formed by a nonlinear static subsystem in sandwich

between two linear dynamic subsystems), is the most general. It has been used to model fluid-structure dynamic interactions [1], human ankle stiffness dynamics [2], biological systems [3], digital communication systems [4], and chemical systems [5], among other applications. In this second part of the paper, we present three examples for illustrating the use of tensor models for system identification. First, we consider SISO FIR linear systems (section 2.1) and instantaneous MIMO linear mixtures (section 2.2). Such systems can be blindly estimated using fourth-order output cumulant tensors that satisfy a PARAFAC model. The second example is concerned with the supervised identification of homogeneous cubic Volterra systems (section 3), the third-order Volterra kernel being modeled by means of a PARAFAC model. Finally, we describe a supervised identification method for Wiener-Hammerstein systems (section 4), the two linear subsystems being estimated from the third-order Volterra kernel associated with the original Wiener-Hammerstein system that satisfies also a PARAFAC model.

We recall the following properties of the $\text{vec}(\cdot)$ operator:

for $\mathbf{v} \in \mathcal{K}^J$, $\mathbf{A} \in \mathcal{K}^{I \times J}$, $\mathbf{B} \in \mathcal{K}^{J \times K}$:

$$\text{vec}(\mathbf{A} \text{diag}(\mathbf{v}) \mathbf{B}) = (\mathbf{B}^T \odot \mathbf{A}) \mathbf{v} \quad (1)$$

$$(\mathbf{A} \text{diag}(\mathbf{v})) \odot \mathbf{B}^T = \mathbf{A} \odot (\mathbf{B}^T \text{diag}(\mathbf{v})) = (\mathbf{A} \odot \mathbf{B}^T) \text{diag}(\mathbf{v}) \quad (2)$$

where the operator $\text{diag}(\cdot)$ forms a diagonal matrix from its vector argument, and \odot denotes the Khatri-Rao product.

We also recall that the unfolded matrices \mathbf{H}_1 , \mathbf{H}_2 , and \mathbf{H}_3 of a third-order tensor \mathbb{H} are linked to the PARAFAC factor matrices \mathbf{A} , \mathbf{B} , and \mathbf{C} by the following equations:

$$\mathbf{H}_1 = (\mathbf{C} \odot \mathbf{A}) \mathbf{B}^T, \quad \mathbf{H}_2 = (\mathbf{A} \odot \mathbf{B}) \mathbf{C}^T, \quad \mathbf{H}_3 = (\mathbf{B} \odot \mathbf{C}) \mathbf{A}^T. \quad (3)$$

2 Blind identification of linear systems [6].

2.1 Single-Input Single-Output (SISO) FIR linear systems

Let us consider a SISO FIR system with input-output relation given by

$$y(n) = \sum_{l=0}^{L-1} h_l x(n-l) + v(n), \quad (4)$$

where $x(\cdot)$, $y(\cdot)$, and $v(\cdot)$ denote respectively the input, the output and the additive noise signals. The additive noise is assumed to be zero-mean, Gaussian with unknown autocorrelation function, and independent from the input signal. The non-measurable input signal $x(\cdot)$ is assumed to be stationary, ergodic, independent and identically distributed (iid) with symmetric distribution, zero-mean, and non-zero kurtosis $\gamma_{4,x}$. The FIR filter with impulse response coefficients h_l is assumed to be causal with memory L and $h_0 = 1$.

Using the system model (4) and taking the multilinearity property of cumulants and the assumptions on the input and noise signals into account, the 4th-order cumulant of the output signal $y(\cdot)$ is given by:

$$\begin{aligned} C_{4,y}(\tau_1, \tau_2, \tau_3) &\triangleq \text{Cum}[y^*(n), y(n + \tau_1), y^*(n + \tau_2), y(n + \tau_3)] \\ &= \gamma_{4,x} \sum_{l=0}^{L-1} h_l^* h_{l+\tau_1} h_{l+\tau_2}^* h_{l+\tau_3}. \end{aligned} \quad (5)$$

Due to the FIR assumption with memory L for the system, we have

$$c_{4,y}(\tau_1, \tau_2, \tau_3) = 0, \quad \forall |\tau_1|, |\tau_2|, |\tau_3| \geq L,$$

which means that the possible nonzero values of the fourth-order cumulants are obtained for $\tau_n = -L + 1, -L + 2, \dots, L - 1$, $n = 1, 2, 3$. Making the coordinate changes $\tau_n = i_n - L$ and $l = m - 1$, we can rewrite (5) as:

$$C_{4,y}(i_1 - L, i_2 - L, i_3 - L) = \gamma_{4,x} \sum_{m=1}^L h_{m-1}^* h_{m+i_1-L-1} h_{m+i_2-L-1}^* h_{m+i_3-L-1}. \quad (6)$$

Defining the third-order tensor \mathbb{X} of fourth-order cumulants, with entries

$$x_{i_1, i_2, i_3} = C_{4,y}(i_1 - L, i_2 - L, i_3 - L), \quad i_n = 1, \dots, 2L - 1, \quad n = 1, 2, 3,$$

we have:

$$x_{i_1, i_2, i_3} = \sum_{m=1}^L a_{i_1, m} b_{i_2, m} c_{i_3, m}, \quad (7)$$

with $a_{i_1, m} = h_{m+i_1-L-1}$, $b_{i_2, m} = h_{m+i_2-L-1}^*$, and $c_{i_3, m} = \gamma_{4,x} h_{m-1}^* h_{m+i_3-L-1}$. We recognize in (7) the scalar writing of the PARAFAC model associated with the third-order tensor \mathbb{X} of dimensions $I_1 \times I_2 \times I_3$, with $I_j = 2L - 1$, $j = 1, 2, 3$. The coefficients $a_{i_1, m}$, $b_{i_2, m}$, and $c_{i_3, m}$ are the respective entries of the three matrix

factors \mathbf{A} , \mathbf{B} and \mathbf{C} . Defining the system coefficient matrix $\mathbf{H} \in \mathcal{C}^{(2L-1) \times L}$ as follows:

$$\mathbf{H} \triangleq \mathcal{H}(\mathbf{h}) = \begin{pmatrix} 0 & 0 & \cdots & h_0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & h_0 & \cdots & h_{L-2} \\ h_0 & h_1 & \cdots & h_{L-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{L-2} & h_{L-1} & \cdots & 0 \\ h_{L-1} & 0 & \cdots & 0 \end{pmatrix}, \quad (8)$$

where $\mathcal{H}(\cdot)$ is an operator that builds a Hankel matrix from its vector argument as shown above and $\mathbf{h} = (h_0 \cdots h_{L-1})^T$ denotes the system impulse response vector, one can note that the matrix factors depend on the system coefficient matrix as follows:

$$\mathbf{A} = \mathbf{H}, \quad \mathbf{B} = \mathbf{H}^*, \quad \mathbf{C} = \gamma_{4,x} \mathbf{H} \text{diag}(\mathbf{h}^*). \quad (9)$$

Using (3) with the correspondences (9) leads to the following unfolded matrix representation of \mathbb{X} along its first dimension:

$$\mathbf{X}_2 = \gamma_{4,x} (\mathbf{H} \odot \mathbf{H}^*) \text{diag}(\mathbf{h}^*) \mathbf{H}^T. \quad (10)$$

Applying property (1) to this equation, we get:

$$\text{vec}(\mathbf{X}_2) = \gamma_{4,x} (\mathbf{H} \odot \mathbf{H} \odot \mathbf{H}^*) \mathbf{h}^*. \quad (11)$$

The system impulse response vector \mathbf{h} can be obtained by iteratively minimizing the following conditional LS cost function:

$$\psi(\mathbf{h}, \hat{\mathbf{h}}^{(k-1)}) \triangleq \left\| \text{vec}(\mathbf{X}_2) - \gamma_{4,x} \left(\mathcal{H}(\hat{\mathbf{h}}^{(k-1)}) \odot \mathcal{H}(\hat{\mathbf{h}}^{(k-1)}) \odot \mathcal{H}(\hat{\mathbf{h}}^{(k-1)*}) \right) \mathbf{h}^* \right\|_2^2 \quad (12)$$

Starting from a random initialization, we get at iteration k :

$$\hat{\mathbf{h}}^{(k)} = \arg \min_{\mathbf{h}} \psi(\mathbf{h}, \hat{\mathbf{h}}^{(k-1)}).$$

In order to take the constraint $h_0 = 1$ into account at each iteration k , the estimate of the system impulse response vector is normalized with respect to its first entry. The so-called Single Step LS PARAFAC-based Blind Channel Identification (SS-LS PBCI) algorithm is summarized as follows, for $k \geq 1$:

1. Normalize $\hat{\mathbf{h}}^{(k-1)}$ and build $\hat{\mathbf{H}}^{(k-1)} = \mathcal{H}(\hat{\mathbf{h}}^{(k-1)})$ as defined in (8).
2. Compute $\hat{\mathbf{G}}^{(k-1)} = \hat{\mathbf{H}}^{(k-1)} \odot \hat{\mathbf{H}}^{(k-1)} \odot \hat{\mathbf{H}}^{(k-1)*}$.
3. Compute $\hat{\mathbf{h}}^{(k)*} = \gamma_{4,x}^{-1} \hat{\mathbf{G}}^{(k-1)\dagger} \text{vec}(\mathbf{X}_2)$, where \dagger denotes the Moore-Penrose pseudo-inverse.
4. Iterate until $\left\| \hat{\mathbf{h}}^{(k)} - \hat{\mathbf{h}}^{(k-1)} \right\| / \left\| \hat{\mathbf{h}}^{(k)} \right\| \leq \varepsilon$, where ε is an arbitrary small positive constant.

2.2 Instantaneous Multi-Input Multi-Output (MIMO) linear mixtures.

Let us now consider an instantaneous MIMO system with Q input signals $s_q(\cdot)$ and M output signals $y_m(\cdot)$ given at the discrete time instant n by:

$$y_m(n) = \sum_{q=1}^Q h_{mq} s_q(n) + v_m(n), \quad m = 1, \dots, M. \quad (13)$$

Defining the complex-valued output vector $\mathbf{y}(n) = (y_1(n) \cdots y_M(n))^T$, the input-output relation of the MIMO system can be written as :

$$\mathbf{y}(n) = \mathbf{H}\mathbf{s}(n) + \mathbf{v}(n), \quad (14)$$

where $\mathbf{s}(n) = (s_1(n) \cdots s_Q(n))^T$ contains the input signals $s_q(\cdot)$ assumed to be stationary, ergodic, mutually independent with symmetric distribution, zero-mean and known non-zero kurtosis γ_{4,s_q} , $\mathbf{v}(n) = (v_1(n) \cdots v_M(n))^T$ contains the additive Gaussian noise at the output, assumed to be independent from the input signals, and the elements of the instantaneous mixing matrix $\mathbf{H} \in \mathbb{C}^{M \times Q}$ are the MIMO system coefficients h_{mq} assumed to be constant, complex valued with real and imaginary parts driven from a continuous Gaussian distribution³.

We now consider the blind MIMO system identification issue using 4th-order output cumulants. It is well known that solutions to this issue only exist up to a column scaling and permutation indeterminacy. First of all, a reference output $y_r(\cdot)$ is arbitrarily selected. Using the system model (13) and taking the multilinearity property of cumulants and the assumptions on the input signals and the noise into account, the 4th-order spatial output cumulants with respect

³ Such a MIMO system matrix represents a Rayleigh flat fading propagation environment.

to the reference output are given by:

$$\begin{aligned} C_{4,y}(r, i_1, i_2, i_3) &\triangleq \text{Cum} [y_r^*(n), y_{i_1}(n), y_{i_2}^*(n), y_{i_3}(n)] \\ &= \sum_{q=1}^Q \gamma_{4,s_q} h_{r,q}^* h_{i_1,q} h_{i_2,q}^* h_{i_3,q}. \end{aligned} \quad (15)$$

Defining the third-order tensor \mathbb{X} with entries $x_{i_1, i_2, i_3} = C_{4,y}(r, i_1, i_2, i_3)$, $i_n = 1, \dots, M$, $n = 1, 2, 3$, we have:

$$x_{i_1, i_2, i_3} = \sum_{q=1}^Q a_{i_1, q} b_{i_2, q} c_{i_3, q}, \quad (16)$$

with $a_{i_1, q} = h_{i_1, q}$, $b_{i_2, q} = h_{i_2, q}^*$, and $c_{i_3, q} = \gamma_{4,s_q} h_{r,q}^* h_{i_3, q}$. We recognize in (16) the scalar writing of the PARAFAC model associated with the third-order tensor \mathbb{X} of dimensions $M \times M \times M$, where $a_{i_1, m}$, $b_{i_2, m}$, and $c_{i_3, m}$ are the respective entries of the three matrix factors \mathbf{A} , \mathbf{B} , and \mathbf{C} linked to the system matrix as follows:

$$\mathbf{A} = \mathbf{H}, \quad \mathbf{B} = \mathbf{H}^*, \quad \mathbf{C} = \mathbf{H} \text{diag}(\mathbf{H}_r^*) \mathbf{\Gamma}_{4,s}, \quad (17)$$

with $\mathbf{\Gamma}_{4,s} = \text{diag}(\gamma_{4,s_1} \cdots \gamma_{4,s_Q})$. Using (3), the unfolded matrix of \mathbb{X} along its first mode is given by:

$$\mathbf{X}_2 = (\mathbf{H} \odot \mathbf{H}^*) \mathbf{\Gamma}_{4,s} \text{diag}(\mathbf{H}_r^*) \mathbf{H}^T.$$

A single step LS algorithm can be derived for estimating the system matrix \mathbf{H} by iteratively minimizing the following conditional LS cost function:

$$\psi(\mathbf{H}, \hat{\mathbf{H}}^{(k-1)}) \triangleq \left\| \mathbf{X}_2 - \left(\hat{\mathbf{H}}^{(k-1)} \odot \hat{\mathbf{H}}^{(k-1)*} \right) \mathbf{\Gamma}_{4,s} \text{diag}(\hat{\mathbf{H}}_r^{(k-1)*}) \mathbf{H}^T \right\|_F^2 \quad (18)$$

where k denotes the iteration number. After convergence, the estimated system matrix equals the actual one up to column permutation and scaling. Although scaling and permutation ambiguities are not explicitly solved, these indeterminacies do not represent a concern in the context of blind mixture identification.

The choice of an arbitrary reference output can influence the obtained results. So, alternatively, we can leave free the first dimension of the 4th-order cumulants, which gives rise to a 4th-order tensor \mathbb{X} with entries

$$x_{i_1, i_2, i_3, i_4} = C_{4,y}(i_1, i_2, i_3, i_4) = \sum_{q=1}^Q \gamma_{4,s_q} h_{i_1, q}^* h_{i_2, q} h_{i_3, q}^* h_{i_4, q}. \quad (19)$$

By considering the equivalent writing

$$x_{i_1, i_2, i_3, i_4} = \sum_{q=1}^Q a_{i_1, q} b_{i_2, q} c_{i_3, q} d_{i_4, q},$$

the matrix factors are given by:

$$\mathbf{A} = \mathbf{H}^*, \quad \mathbf{B} = \mathbf{H}, \quad \mathbf{C} = \mathbf{H}^*, \quad \mathbf{D} = \mathbf{H}\mathbf{\Gamma}_{4,s}. \quad (20)$$

The matricization of the 4th-order tensor \mathbb{X} gives rise to the following $M^3 \times M$ unfolded matrix (see [6] for details):

$$\mathbf{X} = (\mathbf{H}^* \odot \mathbf{H} \odot \mathbf{H}^*) \mathbf{\Gamma}_{4,s} \mathbf{H}^T. \quad (21)$$

So a 4D single step LS algorithm can also be derived for estimating the system matrix \mathbf{H} by iteratively minimizing the following conditional LS cost function:

$$\psi(\mathbf{H}, \hat{\mathbf{H}}^{(k-1)}) \triangleq \left\| \mathbf{X} - \left(\hat{\mathbf{H}}^{(k-1)*} \odot \hat{\mathbf{H}}^{(k-1)} \odot \hat{\mathbf{H}}^{(k-1)*} \right) \mathbf{\Gamma}_{4,s} \mathbf{H}^T \right\|_F^2 \quad (22)$$

where k denotes the iteration number. The iterative minimization of the above cost function yields the LS solution

$$\hat{\mathbf{H}}^{(k)T} \triangleq \arg \min_{\mathbf{H}} \psi(\mathbf{H}, \hat{\mathbf{H}}^{(k-1)}) = \mathbf{\Gamma}_{4,s}^{-1} \left(\hat{\mathbf{H}}^{(k-1)*} \odot \hat{\mathbf{H}}^{(k-1)} \odot \hat{\mathbf{H}}^{(k-1)*} \right)^\dagger \mathbf{X}. \quad (23)$$

3 Supervised identification of homogeneous cubic Volterra systems [7].

Let us consider a third-order homogeneous discrete-time Volterra system, with memory M , the input-output relation of which is given by:

$$y(n) = \sum_{i_1=1}^M \sum_{i_2=1}^M \sum_{i_3=1}^M h_3(i_1-1, i_2-1, i_3-1) u(n-i_1+1) u(n-i_2+1) u(n-i_3+1). \quad (24)$$

One major drawback of the Volterra model for applications concerns the huge number of the parameters required for adequately representing a given system. One way for reducing this parameter number consists in approximating the Volterra kernel, viewed as a tensor, by a PARAFAC model. We get:

$$h_3(i_1-1, i_2-1, i_3-1) = \sum_{r=1}^R a_{i_1, r} b_{i_2, r} c_{i_3, r}, \quad (25)$$

where $a_{i_1,r}$, $b_{i_2,r}$, and $c_{i_3,r}$ are the respective entries of the $M \times R$ matrix factors \mathbf{A} , \mathbf{B} , and \mathbf{C} . So, the PARAFAC-Volterra model output can be written as:

$$\begin{aligned}
 \hat{y}(n) &= \sum_{i_1=1}^M \sum_{i_2=1}^M \sum_{i_3=1}^M \sum_{r=1}^R a_{i_1,r} b_{i_2,r} c_{i_3,r} u(n-i_1+1) u(n-i_2+1) u(n-i_3+1) \\
 &= \sum_{r=1}^R \left(\sum_{i_1=1}^M a_{i_1,r} u(n-i_1+1) \right) \left(\sum_{i_2=1}^M b_{i_2,r} u(n-i_2+1) \right) \\
 &\quad \times \left(\sum_{i_3=1}^M c_{i_3,r} u(n-i_3+1) \right) \\
 &= \sum_{r=1}^R (\mathbf{u}^T(n) \mathbf{A}_{\cdot,r}) (\mathbf{u}^T(n) \mathbf{B}_{\cdot,r}) (\mathbf{u}^T(n) \mathbf{C}_{\cdot,r})
 \end{aligned} \tag{26}$$

with $\mathbf{u}(n) = (u(n) \cdots u(n-M+1))^T$. One can note that provided $R \ll M$, the number of parameters can be significantly reduced from M^3 for the standard Volterra model to $3MR$ for the PARAFAC-Volterra model. However, the linearity in parameters is lost, the resulting model being trilinear in its parameters. Fig. 5 illustrates the parallel computation of the cubic PARAFAC-Volterra model output.

By defining $\phi_r^A(n) = (\mathbf{u}^T(n) \mathbf{B}_{\cdot,r}) (\mathbf{u}^T(n) \mathbf{C}_{\cdot,r})$, we can rewrite (26) as:

$$\begin{aligned}
 \hat{y}(n) &= \sum_{r=1}^R \phi_r^A(n) \mathbf{u}^T(n) \mathbf{A}_{\cdot,r} = (\phi_1^A(n) \mathbf{u}^T(n) \cdots \phi_R^A(n) \mathbf{u}^T(n)) \begin{pmatrix} \mathbf{A}_{\cdot,1} \\ \vdots \\ \mathbf{A}_{\cdot,R} \end{pmatrix} \\
 &= \mathbf{v}_A^T(n) \text{vec}(\mathbf{A}),
 \end{aligned} \tag{27}$$

where $\mathbf{v}_A(n) = \boldsymbol{\phi}^A(n) \otimes \mathbf{u}(n)$ and $\boldsymbol{\phi}^A(n) = (\phi_1^A(n) \cdots \phi_R^A(n))^T$. Similarly, we get:

$$\hat{y}(n) = \mathbf{v}_B^T(n) \text{vec}(\mathbf{B}) = \mathbf{v}_C^T(n) \text{vec}(\mathbf{C}), \tag{28}$$

with

$$\begin{aligned}
 \mathbf{v}_B(n) &= \boldsymbol{\phi}^B(n) \otimes \mathbf{u}(n), \quad \boldsymbol{\phi}^B(n) = (\phi_1^B(n) \cdots \phi_R^B(n))^T, \\
 \phi_r^B(n) &= (\mathbf{u}^T(n) \mathbf{A}_{\cdot,r}) (\mathbf{u}^T(n) \mathbf{C}_{\cdot,r}), \\
 \mathbf{v}_C(n) &= \boldsymbol{\phi}^C(n) \otimes \mathbf{u}(n), \quad \boldsymbol{\phi}^C(n) = (\phi_1^C(n) \cdots \phi_R^C(n))^T, \\
 \phi_r^C(n) &= (\mathbf{u}^T(n) \mathbf{A}_{\cdot,r}) (\mathbf{u}^T(n) \mathbf{B}_{\cdot,r}).
 \end{aligned}$$

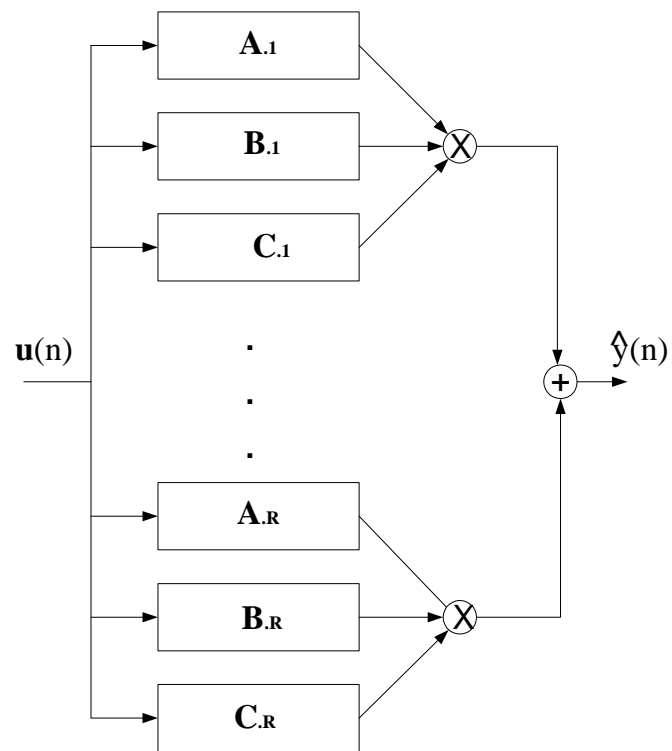


Fig. 1. Parallel realization of a cubic PARAFAC-Volterra model.

Now, we consider a linearization around the current estimates of two matrix components, and we define three conditional LS cost functions:

$$\eta_{\mathbf{X}}(n) = \sum_{t=1}^n \lambda^{n-t} (y(t) - \hat{\mathbf{v}}_X^T(n/n-1) \text{vec}(\hat{\mathbf{X}}(n-1)))^2, \quad \text{with } \mathbf{X} = \mathbf{A}, \mathbf{B}, \mathbf{C}, \quad (29)$$

where λ is a forgetting factor, $\hat{\mathbf{v}}_A(n/n-1)$, $\hat{\mathbf{v}}_B(n/n-1)$, and $\hat{\mathbf{v}}_C(n/n-1)$ depend respectively on $(\hat{\mathbf{B}}(n-1), \hat{\mathbf{C}}(n-1))$, $(\hat{\mathbf{A}}(n), \hat{\mathbf{C}}(n-1))$, and $(\hat{\mathbf{A}}(n), \hat{\mathbf{B}}(n))$. By alternatively minimizing these cost functions with respect to \mathbf{A} , \mathbf{B} , and \mathbf{C} , we get the following alternating recursive least squares (ARLS) algorithm:

$$\text{vec}(\hat{\mathbf{X}}(n)) = \text{vec}(\hat{\mathbf{X}}(n-1)) + \boldsymbol{\varepsilon}_X(n) \mathbf{k}_X(n) \quad (30)$$

with

$$\boldsymbol{\varepsilon}_X(n) = y(n) - \hat{\mathbf{v}}_X^T(n/n-1) \text{vec}(\hat{\mathbf{X}}(n-1)) \quad (31)$$

$$\mathbf{k}_X(n) = \frac{\mathbf{Q}_X(n-1) \hat{\mathbf{v}}_X(n/n-1)}{1 + \hat{\mathbf{v}}_X^T(n/n-1) \mathbf{Q}_X(n-1) \hat{\mathbf{v}}_X(n/n-1)} \quad (32)$$

$$\mathbf{Q}_X(n) = \lambda^{-1} (\mathbf{I} - \mathbf{k}_X(n) \hat{\mathbf{v}}_X^T(n/n-1)) \mathbf{Q}_X(n-1). \quad (33)$$

The ARLS algorithm can be summarized as follows:

1. Initialize $\hat{\mathbf{B}}(0)$ and $\hat{\mathbf{C}}(0)$ with random values, $\hat{\mathbf{A}}(0) = \mathbf{0}$, and

$$\mathbf{Q}_A(0) = \mathbf{Q}_B(0) = \mathbf{Q}_C(0) = \alpha \mathbf{I}_{MR},$$

where α is a large arbitrary constant. Set $n = 0$.

2. Update the PARAFAC matrix factors:
 - (a) Increment n .
 - (b) Update of matrix factor \mathbf{A} :
 - i. Compute $\hat{\mathbf{v}}_A(n/n-1)$.
 - ii. Compute $\hat{\mathbf{A}}(n)$ using (30)-(33).
 - (c) Update of matrix factor \mathbf{B} :
 - i. Compute $\hat{\mathbf{v}}_B(n/n-1)$.
 - ii. Compute $\hat{\mathbf{B}}(n)$ using (30)-(33).
 - (d) Update of matrix factor \mathbf{C} :
 - i. Compute $\hat{\mathbf{v}}_C(n/n-1)$.
 - ii. Compute $\hat{\mathbf{C}}(n)$ using (30)-(33).
3. Go back to step 2 until convergence of the algorithm.
4. Reconstruct the Volterra kernel with the estimated PARAFAC loading factors:

$$\hat{h}_3(i_1-1, i_2-1, i_3-1) = \sum_{r=1}^R \hat{a}_{i_1,r} \hat{b}_{i_2,r} \hat{c}_{i_3,r}.$$

4 Supervised identification of Wiener-Hammerstein systems [8].

Let us consider the Wiener-Hammerstein system depicted in Fig. 2.

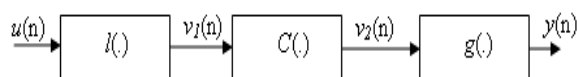


Fig. 2. Wiener-Hammerstein system

Assuming that the nonlinearity is continuous within the considered dynamic range, then, owing to the Weierstrass theorem, it can be approximated to an arbitrary degree of accuracy by a polynomial $C(\cdot)$ of finite degree P , with coefficients c_p . So, the WH model is composed of a polynomial $C(\cdot)$, sandwiched between two linear filters with impulse responses $l(\cdot)$ and $g(\cdot)$, and memories M_l and M_g respectively, i.e.:

$$v_1(n) = \sum_{i=0}^{M_l-1} l(i)u(n-i), \quad v_2(n) = \sum_{p=1}^P c_p v_1^p(n),$$

$$y(n) = \sum_{i=0}^{M_g-1} g(i)v_2(n-i), \quad (34)$$

with $l(0) = g(0) = 1$, a standard constraint for guaranteeing the model uniqueness.

The estimation of the parameters that characterize each block of this model is not simple since this model is not linear in its parameters. This parameter estimation problem can be solved by first identifying an associated Volterra model, linear in its parameters, and then deducing the parameters of linear and non-linear subsystems from the estimated Volterra kernels. Several methods based on this approach were proposed by considering first- and second-order Volterra kernels in time [9] or frequency[10–12] domain.

The output of the Volterra model associated with the Wiener-Hammerstein system represented by means of equations (34) can be written as

$$y(n) = \sum_{p=1}^P \sum_{i_1, \dots, i_p=1}^{M_v} h_p(i_1 - 1, \dots, i_p - 1) \prod_{k=1}^p u(n - i_k + 1) \quad (35)$$

where $h_p(\cdot)$, denoting the p th-order Volterra kernel associated with the WH system, is given by [13]:

$$h_p(i_1 - 1, \dots, i_p - 1) = c_p \sum_{i=1}^{M_g} g(i - 1) \prod_{k=1}^p l(i_k - i), \quad (36)$$

$i = 1, \dots, M_g, i_k = 1, \dots, M_v, k = 1, \dots, p$, with $M_v = M_l + M_g - 1$.

The Wiener-Hammerstein structure can be used for modelling communication channels where nonlinearities are structured as combinations of dynamic linear time-invariant (LTI) blocks with zero-memory nonlinear (ZMNL) ones. It finds application for modelling distortions in nonlinearly amplified digital communication signals (satellite and microwave links), among others.

In the sequel, we consider a nonlinear communication channel represented with a WH model where the nonlinearity is due to the presence of a High Power Amplifier (HPA) at the transmitter of a mobile communication system. The linear filters $l(\cdot)$ and $g(\cdot)$ respectively characterize the transmit filter and the propagation bandpass channel whereas the memoryless nonlinear part $C(\cdot)$ represents the zero-memory HPA. It has been shown that the equivalent baseband representation of the ZMNL-HPA contains only odd-order terms [14, 15]:

$$v_2(n) = C(v_1(n)) = \sum_{p=0}^P \alpha_{2p+1} [v_1(n)]^{p+1} [v_1^*(n)]^p, \quad (37)$$

where $v_1(n) = \sum_{i=0}^{M_l-1} l(i)u(n-i)$, $u(\cdot)$ being the input sequence. The received sequence $y(\cdot)$ is given by

$$y(n) = \sum_{i=0}^{M_g-1} g(i)v_2(n-i).$$

One can note that this representation is not linear with respect to the system parameters. So, parameter estimation of such a system is not a trivial task. However, this system also admits a Volterra model, linear-in-parameters and given

by [16, 17]:

$$y(n) = \sum_{p=0}^P \sum_{i_1, \dots, i_{2p+1}=0}^{M-1} h_{2p+1}(i_1, \dots, i_{2p+1}) \prod_{k=1}^{p+1} u(n-i_k) \prod_{k=p+2}^{2p+1} u^*(n-i_k) \quad (38)$$

where $M = M_g + M_l - 1$ and $h_{2p+1}(\cdot, \dots, \cdot)$ denotes the $(2p+1)$ th-order Volterra kernel. Estimators devoted to linear models can easily be applied to the Volterra model. Since we are interested with the estimation of each subsystem, once the Volterra kernels estimated, we can deduce the parameters of both linear and nonlinear subsystems. Indeed, the Volterra kernels and the parameters of the original Wiener-Hammerstein model are linked as follows [18]:

$$h_{2p+1}(i_1, \dots, i_{2p+1}) = \alpha_{2p+1} \sum_{i=0}^{M_g-1} g(i) \prod_{k=1}^{p+1} l(i_k - i) \prod_{k=p+2}^{2p+1} l^*(i_k - i). \quad (39)$$

In particular, the third-order Volterra kernel is given by:

$$h_3(i_1, i_2, i_3) = \alpha_3 \sum_{i=0}^{M_g-1} g(i) l(i_1 - i) l(i_2 - i) l^*(i_3 - i)$$

or equivalently

$$h_3(i_1 - 1, i_2 - 1, i_3 - 1) = \sum_{i=1}^{M_g} a_{i_1, i} b_{i_2, i} c_{i_3, i}, \quad (40)$$

with $a_{i_1, i} = \alpha_3 g(i-1) l(i_1 - i)$, $b_{i_2, i} = l(i_2 - i)$, and $c_{i_3, i} = l^*(i_3 - i)$, $i = 1, \dots, M_g$, $i_k = 1, \dots, M$, $k = 1, 2, 3$.

We recognize in (40) the scalar writing of the PARAFAC decomposition of the $M \times M \times M$ tensor \mathbb{H}_3 the elements of which are the coefficients of the third-order Volterra kernel $h_3(\cdot)$. The number of factors involved in this decomposition is equal to the memory M_g of the second linear subsystem. The three matrix factors **A**, **B**, and **C**, with the respective entries $a_{i_1, i}$, $b_{i_2, i}$, and $c_{i_3, i}$, are given by:

$$\mathbf{A} = \mathbf{L} \text{diag}(\boldsymbol{\gamma}), \quad \mathbf{B} = \mathbf{L}, \quad \mathbf{C} = \mathbf{L}^*, \quad (41)$$

with $\boldsymbol{\gamma} = \alpha_3 \mathbf{g}$, $\mathbf{g} = (g(0) \dots g(M_g - 1))^T$, $\mathbf{L} = \mathcal{F}(\mathbf{I})$ the $M \times M_g$ Toeplitz matrix whose first column and row are respectively $\mathbf{L}_{\cdot 1} = (\mathbf{I}^T \mathbf{0}_{1 \times (M_g - 1)})^T$ and $\mathbf{L}_{1 \cdot} = (l(0) \mathbf{0}_{1 \times (M_g - 1)})$, and $\mathbf{I} = (l(0) \dots l(M_l - 1))^T$. In the sequel, we assume

that $l(0) = g(0) = 1$.

Assuming that the matrix factors fulfill the Kruskal condition, then the PARAFAC decomposition is essentially unique. Moreover, by taking the Toeplitz structure of \mathbf{L} into account, with $l(0) = 1$, we can remove all column scaling and permutation ambiguities, so that the PARAFAC decomposition of the third-order Volterra kernel associated with a LTI-ZMNL-LTI system be strictly unique.

Applying the formulae (3) with the correspondencies (41) and using the relation (2), we get the following $M^2 \times M$ unfolded matrices:

$$\mathbf{H}_1 = (\mathbf{L}^* \odot \mathbf{L}) \text{diag}(\boldsymbol{\gamma}) \mathbf{L}^T, \quad (42)$$

$$\mathbf{H}_2 = (\mathbf{L} \odot \mathbf{L}) \text{diag}(\boldsymbol{\gamma}) \mathbf{L}^H, \quad (43)$$

$$\mathbf{H}_3 = (\mathbf{L} \odot \mathbf{L}^*) \text{diag}(\boldsymbol{\gamma}) \mathbf{L}^T. \quad (44)$$

The parameters of the linear subsystems can be estimated using a two-step alternating least squares algorithm based on the minimization of the LS criterion

$$\mathcal{J} = \|\mathbf{H}_3 - (\mathbf{L} \odot \mathbf{L}^*) \text{diag}(\boldsymbol{\gamma}) \mathbf{L}^T\|_F^2. \quad (45)$$

Using the relation (1), this criterion can be rewritten as:

$$\mathcal{J} = \|\text{vec}(\mathbf{H}_3) - (\mathbf{L} \odot \mathbf{L} \odot \mathbf{L}^*) \boldsymbol{\gamma}\|_2^2. \quad (46)$$

It follows that the LS update of $\boldsymbol{\gamma}$ conditionally to the previous estimate $\hat{\mathbf{L}}$ is given by

$$\hat{\boldsymbol{\gamma}} = (\hat{\mathbf{L}} \odot \hat{\mathbf{L}} \odot \hat{\mathbf{L}}^*)^\dagger \text{vec}(\mathbf{H}_3). \quad (47)$$

Defining $\mathbf{G} = \text{diag}(\boldsymbol{\gamma}) (\mathbf{L} \odot \mathbf{L}^*)^T$, we have

$$\mathbf{H}_3^T = \mathbf{L} \mathbf{G} = (\mathcal{T}(\mathbf{1}) \mathbf{G}_{.1} \cdots \mathcal{T}(\mathbf{1}) \mathbf{G}_{.M^2}) = (\mathcal{T}(\mathbf{G}_{.1}) \mathbf{1} \cdots \mathcal{T}(\mathbf{G}_{.M^2}) \mathbf{1}). \quad (48)$$

So, we can rewrite the criterion \mathcal{J} as

$$\mathcal{J} = \|\text{vec}(\mathbf{H}_3^T) - \boldsymbol{\Gamma} \mathbf{1}\|_2^2, \quad (49)$$

with $\boldsymbol{\Gamma} = (\mathcal{T}(\mathbf{G}_{.1})^T \cdots \mathcal{T}(\mathbf{G}_{.M^2})^T)^T$. The conditional least squares update of $\mathbf{1}$ given the previous estimate $\hat{\boldsymbol{\Gamma}}$ is then given by

$$\hat{\mathbf{1}} = \hat{\boldsymbol{\Gamma}}^\dagger \text{vec}(\mathbf{H}_3^T). \quad (50)$$

Once the linear subsystems estimated the system model becomes linear in the unknown remaining parameters, i.e. those of the ZMNL subsystem. So, the LS algorithm can be used to complete the system estimation. The overall estimation method is summarized as follows:

1. From input-output measurements, estimate the third-order Volterra kernel associated with the block structured nonlinear system to be estimated and construct the unfolded matrix \mathbf{H}_3 .
2. Estimate the parameter vectors $\boldsymbol{\gamma}$ and \mathbf{l} as follows:
 - (a) $k = 0$, initialize $\hat{\mathbf{l}}_0$ with random values, except $\hat{\mathbf{l}}_0(1) = 1$.
 - (b) $k = k + 1$,
 - (c) Build $\hat{\mathbf{L}}_{k-1} = \mathcal{F}(\hat{\mathbf{l}}_{k-1})$ and compute

$$\hat{\boldsymbol{\gamma}}_k = (\hat{\mathbf{L}}_{k-1} \odot \hat{\mathbf{L}}_{k-1} \odot \hat{\mathbf{L}}_{k-1}^*)^\dagger \text{vec}(\mathbf{H}_3).$$

- (d) Deduce $\hat{\mathbf{G}}^k = \text{diag}(\hat{\boldsymbol{\gamma}}_k) (\hat{\mathbf{L}}_{k-1} \odot \hat{\mathbf{L}}_{k-1}^*)^T$, and then

$$\hat{\mathbf{F}}_k = (\mathcal{F}(\mathbf{G}_{,1}^k)^T \cdots \mathcal{F}(\mathbf{G}_{,M^2}^k)^T)^T.$$
 - (e) Compute $\hat{\mathbf{l}}_k = \hat{\mathbf{F}}_k^\dagger \text{vec}(\mathbf{H}_3)$ and normalize it by dividing its entries by the first one.
 - (f) Return to step 2b until a stop criterion is reached⁴.
 - (g) Deduce the parameters of the second linear subsystem by dividing the entries of $\hat{\boldsymbol{\gamma}}_k$ by the first one, i.e. $\hat{\boldsymbol{\gamma}}_k(1)$.
3. (a) Generate the signals $\bar{u}_{2p+1}(n)$, $n = 1, \dots, N$, $p = 0, \dots, P$, as follows

$$\bar{u}_{2p+1}(n) = \sum_{i_1, \dots, i_{2p+1}=0}^{M-1} \hat{h}_{2p+1}(i_1, \dots, i_{2p+1}) \prod_{k=1}^{p+1} u(n-i_k) \prod_{k=p+2}^{2p+1} u^*(n-i_k),$$

$$\text{with } \hat{h}_{2p+1}(i_1, \dots, i_{2p+1}) = \sum_{i=0}^{M_g-1} \hat{g}(i) \prod_{k=1}^{p+1} \hat{l}(i_k-i) \prod_{k=p+2}^{2p+1} \hat{l}^*(i_k-i), \text{ where}$$

$\hat{g}(\cdot)$ and $\hat{l}(\cdot)$ are the estimated parameters obtained in step 2.

- (b) Calculate the LS estimate of the polynomial coefficient vector $\mathbf{c} = (\alpha_1 \cdots \alpha_{2p+1})^T$ as $\hat{\mathbf{c}} = \bar{\mathbf{U}}^\dagger \mathbf{y}$, with $\mathbf{y} = (y(1) \cdots y(N))^T$, the vector of the measured outputs, $\bar{\mathbf{U}} = (\bar{\mathbf{u}}(1) \cdots \bar{\mathbf{u}}(N))^T$, and

$$\bar{\mathbf{u}}(n) = (\bar{u}_1(n) \cdots \bar{u}_{2p+1}(n))^T.$$

⁴ The stop criterion can be a given number of iterations

The third-order Volterra kernel can be estimated using a closed-form expression like that derived for third and fifth order Volterra channels with i.i.d. QAM and PSK input signals [17]. For more general i.i.d. signals and real-valued Volterra kernels, the closed-form expressions derived in [19] can also be used.

Now, we illustrate the above algorithm by means of simulations where the linear subsystems of the simulated LTI-ZMNL-LTI communication channel are given by $\mathbf{l} = (1, 1 + j, -1 + j)^T$ and $\mathbf{g} = (1, 1.05 + 0.8j)^T$ respectively whereas the parameters of the ZMNL part are $\alpha_1 = 14.9740 + 0.0519j$ and $\alpha_3 = -23.0954 + 4.9680j$, which correspond to the parameters of a class AB power amplifier [15]. The input signal was an i.i.d. 8-QAM sequence, i.e. drawn from the finite alphabet $\Lambda = \{\pm 1 \pm j, \pm(1 + \sqrt{3}), \pm(1 + \sqrt{3})j\}$, with length $N = 16384$. An additive, zero-mean, complex, white Gaussian noise was added to the system output. The simulation results were averaged over 100 independent Monte Carlo runs. The performances were evaluated in terms of Normalized Mean Square Error (NMSE) on the output signal and on the estimated parameters. The third-order Volterra kernel was estimated using the method described in [17].

Figure 3 depicts the convergence behavior of the proposed two-steps ALS algorithm in terms of the estimated parameters of the two linear subsystems and the reconstructed third-order Volterra kernel in the noiseless case. From these simulation results, we can conclude that the two-steps ALS converges fast (about 30 iterations).

In the noisy case, Figure 4 depicts the NMSE for different SNR values. We can note that the output NMSE corresponds approximately to the noise level, implying that the estimated model well approximates the simulated system. We also note that the estimation performances of the two LTI subsystems are very close.

5 Conclusion

In this second part of the paper, some applications of the PARAFAC model have been presented for solving the system identification problem for both SISO and MIMO FIR linear systems and nonlinear Volterra and Wiener-Hammerstein systems. In the cases of FIR linear systems and nonlinear Wiener-Hammerstein systems, the PARAFAC modeling results from the original structure of the systems. In the case of nonlinear Volterra systems, the PARAFAC model is used for reducing the parametric complexity. In the three cases, the computation of

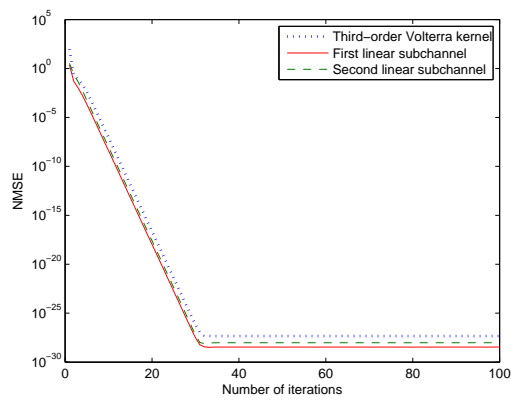


Fig. 3. ALS convergence for the noiseless LTI-ZMNL-LTI communication channel.

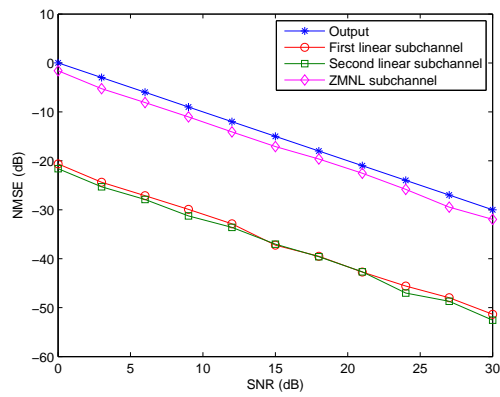


Fig. 4. Output NMSE and parameter NMSE for the estimated LTI-ZMNL-LTI communication channel.

the PARAFAC parameters has been carried out using the ALS algorithm. It is to be noticed that other algorithms could also be applied. For instance, an extended Kalman filter based solution has been derived for PARAFAC-Volterra models [20]. Recently, the authors have proposed a new tensor-based approach for structure identification of block-oriented nonlinear systems [21]. In conclusion, tensors are very well suited for system identification by means of HOS-based methods as well as for nonlinear system representation and their structure and parameter estimation.

References

1. Bendat, J.: Nonlinear system analysis and identification from random data. John Wiley & Sons (1990)
2. Kearney, R., Stein, R., Parameswaran, L.: Identification of intrinsic and reflex contributions to human ankle stiffness dynamics. *IEEE Trans. on Biomedical Engineering* **44**(6) (1997) 493–504
3. Korenberg, M., Hunter, I.: The identification of nonlinear biological systems: LNL cascade models. *Biological Cybernetics* **55** (1986) 125–134
4. Feher, K.: Digital communications- Satellite/Earth station engineering. Englewood Cliffs NJ: Prentice Hall (1993)
5. Kalafatis, A., Wang, L., Cluett, W.: Identification of Wiener type nonlinear systems in a noisy environment. *International Journal of Control* **66**(6) (1997) 923–941
6. Fernandes, C.E.R., Favier, G., Mota, J.C.M.: Blind channel identification algorithms based on the PARAFAC decomposition of cumulant tensors: the single and multiuser cases. *Signal Processing* **88**(6) (2008) 1382–1401
7. Khouaja, A., Favier, G.: Identification of PARAFAC Volterra cubic models using an alternating recursive least squares algorithm. In: Proc. 12th European Signal Processing Conference (EUSIPCO), Vienna, Austria (2004) 1903–1906
8. Kibangou, A., Favier, G.: Matrix and tensor decompositions for identification of block-structured nonlinear channels in digital transmission systems. In: Proc. IEEE Signal Proc. Advances in Wireless Communications (SPAWC), Recife, Brazil (2008)
9. Haber, R.: Structural identification of quadratic block-oriented models. *Int. J. Systems Science* **20**(8) (1989) 1355–1380
10. Weiss, M., Evans, C., Rees, D.: Identification of nonlinear cascade systems using paired multisine signals. *IEEE Trans. on Instrum. and Measurement* **47**(1) (1998) 332–336
11. Vandersteen, G., Schoukens, J.: Measurement and identification of nonlinear systems consisting of linear dynamic blocks and one static nonlinearity. *IEEE Trans. on Automatic Control* **44**(6) (1999) 1266–1271

12. Hui Tan, A., Godfrey, K.: Identification of Wiener-Hammerstein models using linear interpolation in the frequency domain (LIFRED). *IEEE Trans. on Instr. and Measurement* **51**(3) (2002) 509–521
13. Kibangou, A., Favier, G.: Wiener-Hammerstein systems modeling using diagonal Volterra kernels coefficients. *IEEE Signal Proc. Letters* **13**(6) (2006) 381–384
14. Benedetto, S., Biglieri, E., Daffara, E.: Modeling and performance evaluation of nonlinear satellite links- a Volterra series approach. *IEEE Trans. on Aerospace and Electronics Systems* **15**(4) (1979) 494–506
15. Tong Zhou, G., Raich, R.: Spectral analysis of polynomial nonlinearity with applications to RF power amplifiers. *EURASIP Journal on Applied Signal Processing* **12** (2004) 1831–1840
16. Tseng, C.H., Powers, E.: Identification of nonlinear channels in digital transmission systems. In: *Proc. of IEEE Signal Processing Workshop on Higher-order Statistics, South Lake Tahoe, CA (1993)* 42–45
17. Cheng, C.H., Powers, E.: Optimal Volterra kernel estimation algorithms for a nonlinear communication system for PSK and QAM inputs. *IEEE Trans. on Signal Processing* **49**(1) (2001) 147–163
18. Prakriya, S., Hatzinakos, D.: Blind identification of linear subsystems of LTI-ZMNL-LTI models with cyclostationary inputs. *IEEE Trans. on Signal Processing* **45**(8) (1997) 2023–2036
19. Kibangou, A., Favier, G.: Identification of fifth-order Volterra systems using I.I.D. inputs. *IET Signal Processing* (2009) To appear.
20. Favier, G., Bouilloc, T.: Identification de modèles de Volterra basée sur la décomposition PARAFAC. In: *Proc. of GRETSI'09, Dijon, France (2009)*
21. Kibangou, A., Favier, G.: Tensor-based identification of the structure of block-oriented nonlinear systems. In: *Proc. IFAC Symp. on System Identification (SYSID), Saint Malo, France (2009)*

