IDENTIFICATION OF SVD-PARAFAC BASED THIRD-ORDER VOLTERRA MODELS USING AN ARLS ALGORITHM

Anis KHOUAJA and Gérard FAVIER

Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis (I3S), CNRS-UNSA. 2000, route des Lucioles, Les Algorithmes - Euclide B, B.P.121, 06903 Sophia Antipolis, France.
Phone : +33 4 92 94 27 36. Fax : +33 4 92 94 28 96.
`E.mails : {khouaja,favier}@i3s.unice.fr`

Abstract: A broad class of nonlinear systems can be modeled by the Volterra series representation. However, the practical use of such a representation is often limited due to the large number of parameters associated with the Volterra filter structure. This paper is concerned with the problem of identification of third-order Volterra systems. The SVD technique is used to represent the quadratic Volterra kernel and a tensorial decomposition called PARAFAC is used to represent the cubic one. These decompositions allow to significantly reduce the parametric complexity of the Volterra model. Then, a new algorithm called the Alternating Recursive Least Squares (ARLS) algorithm is proposed to estimate the parameters of the linear, quadratic and cubic Volterra kernels. Simulation results show the ability of the proposed solutions to achieve an important complexity reduction and a good identification.

Keywords: Nonlinear system identification, Volterra models, SVD, PARAFAC decomposition.

1. INTRODUCTION

The identification of nonlinear dynamical systems from a given input output data set has attracted considerable interest since many physical systems exhibit nonlinear characteristics. The Volterra model structure can be used to represent a broad class of nonlinearities. The output of a discrete-time, time invariant, cubic Volterra filter is given by :

$$y(n) = \sum_{m=1}^{3}\sum_{n_1=1}^{\infty}\cdots\sum_{n_m=1}^{\infty} h_m(n_1,\ldots,n_m)\prod_{j=1}^{m} u(n-n_j) \quad (1)$$

where u, y and h_m are the input signal, the output signal and the parameters of the m^{th} order kernel, respectively.

In practice, the infinite sum in (1) may be truncated to a finite sum if the system has fading memory (Boyd and Chua, 1985). It has been shown by (Boyd and Chua, 1985) that any time-invariant nonlinear system with fading memory can be well approximated by a finite Volterra series representation to any arbitrary precision. Hence, the class of truncated Volterra models is attractive for nonlinear system modeling.

A very nice property of the Volterra model (1) is that it is linear in its parameters, so standard parameter estimation techniques like Least Squares (LS) method can be applied. However, the large number of parameters associated with the Volterra models limit their practical use to problems involving only small values for the kernels memory and the truncation order. This limitation arises because the estimation of a large number of parameters may be problematic, but also design procedures based upon such models may be cumbersome. To eliminate this drawback, two ways can be followed :

- to arrange the kernels coefficients in matrices that are decomposed in applying a reduced order Singular Value Decomposition (SVD) which leads to a low complexity parallel-cascade realization of the Volterra filter (Panicker and Mathews, 1998),
- to expand the kernels on an orthonormal basis such as the Laguerre functions basis ((Campello *et al.*, 2004),(Dumont and Fu, 1993)) or Generalized Orthonormal Bases (GOB) ((Favier *et al.*, 2003),(Kibangou *et al.*, 2003)).

The purpose of this paper is first to propose third-order Volterra models with a reduced complexity. By considering the quadratic kernel as a matrix and the cubic one as a third-order tensor, we use a Singular Value Decomposition (SVD) and a tensor decomposition called PARAFAC for decomposing these two kernels respectively. The corresponding model called SVD-PARAFAC based Volterra model is presented in section 2. Then in section 3 a new Alternating Recursive Least Squares (ARLS) algorithm is proposed to estimate the parameters of such Volterra models. In section 4, the performance of the proposed approach is evaluated by means of simulations before concluding in section 5.

2. THE SVD-PARAFAC BASED VOLTERRA MODEL

2.1 The Singular Value Decomposition

The SVD has been applied to signal processing problems since the late 1970's and has been described as the most informative general representation of a matrix. The SVD of a $(m \times n)$ matrix X is a factorization of the form :

$$X = U \Lambda V^T \tag{2}$$

where U and V are respectively $(m \times m)$ and $(n \times n)$ matrices the columns of which are respectively the left singular vectors and the right singular vectors of X and satisfying $U^T U = I_m$ and $V^T V = I_n$, Λ is a $(m \times n)$ matrix with all entries zeros except R positive diagonal entries $\{\lambda_r\}$, where $\{\lambda_r\}$ are the singular values of X satisfying $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_R > 0$ with $R = \text{rank}(X) \leq \min(m, n)$. We can also write the SVD of X as :

$$X = \sum_{r=1}^{R} \lambda_r U_{.r} V_{.r}^T \tag{3}$$

where $U_{.r}$ and $V_{.r}$ are the r^{th} column of U and V respectively.

We can rewrite X as :

$$X = \sum_{r=1}^{R} A_{.r} B_{.r}^T \tag{4}$$

where

$$A_{.r} = \lambda_r U_{.r} \tag{5}$$

$$B_{.r} = V_{.r} \tag{6}$$

with the following constraints :

$$A^T A = \Lambda_R^2 \tag{7}$$

$$B^T B = I_R \tag{8}$$

where $\Lambda_R^2 = \text{diag}\ (\lambda_1^2, \ldots, \lambda_R^2)$ and I_R is the identity matrix of order R.

The scalar representation of (4) is :

$$X(n_1, n_2) = \sum_{r=1}^{R} a_{n_1 r} b_{n_2 r} \tag{9}$$

where $a_{n_1 r}$ represents the (n_1, r) element of the matrix A and $b_{n_2 r}$ the (n_2, r) element of the matrix B.

2.2 The PARAFAC Decomposition

The PARAFAC (PARAllel FACtor) decomposition also called CANDECOMP (CANonical DECOMPosition) was introduced by Harshman (1970) (Harshman, 1970) and by Caroll and Chang (1970) (Carroll and Chang, 1970) in order to reduce the complexity of an N^{th} order tensor. This decomposition entirely preserves the information contained in the original tensor.

We define horizontal, vertical and frontal matrices of a third-order $(N_1 \times N_2 \times N_3)$ tensor $\mathbb{H}$.

- $H_{n_1..}$ $(n_1 = 1, \ldots, N_1)$ are $(N_2 \times N_3)$ matrices such as $H_{n_1..}(n_2, n_3) = \mathbb{H}(n_1, n_2, n_3)$.
- $H_{.n_2.}$ $(n_2 = 1, \ldots, N_2)$ are $(N_3 \times N_1)$ matrices such as $H_{.n_2.}(n_3, n_1) = \mathbb{H}(n_1, n_2, n_3)$.
- $H_{..n_3}$ $(n_3 = 1, \ldots, N_3)$ are $(N_1 \times N_2)$ matrices such as $H_{..n_3}(n_1, n_2) = \mathbb{H}(n_1, n_2, n_3)$.

where $\mathbb{H}(n_1, n_2, n_3)$ is the element (n_1, n_2, n_3) of the tensor $\mathbb{H}$. The construction of the matrices $H_{n_1..}$, $H_{.n_2.}$ and $H_{..n_3}$ is described in figure 1.

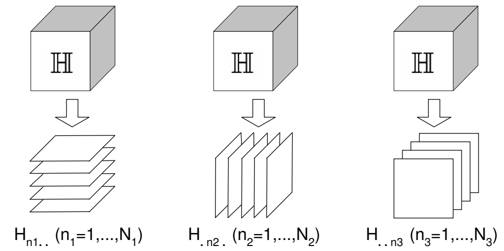

Fig. 1. Horizontal, vertical and frontal matrices $H_{n_1..}$, $H_{.n_2.}$ and $H_{..n_3}$

The scalar representation of the PARAFAC decomposition of a third-order $(N_1 \times N_2 \times N_3)$ tensor $\mathbb{H}$ is written as :

$$\mathbb{H}(n_1, n_2, n_3) = \sum_{p=1}^{P} c_{n_1p} d_{n_2p} e_{n_3p} \qquad (10)$$

where c_{n_1p}, d_{n_2p} and e_{n_3p} constitute the elements of three matrices C, D and E with respective dimensions $(N_1 \times P)$, $(N_2 \times P)$ and $(N_3 \times P)$, and P is the number of PARAFAC model factors. P is the rank of the tensor $\mathbb{H}$ according to Kruskal (Kruskal, 1977).

2.3 The SVD-PARAFAC based Volterra model

The input/output relation of a discrete-time, time invariant third-order Volterra system with M-memory can be written as :

$$\begin{aligned}\widehat{y}(n) &= \sum_{n_1=1}^{M} h_1(n_1)u(n-n_1) \\ &+ \sum_{n_1=1}^{M}\sum_{n_2=1}^{M} h_2(n_1,n_2)u(n-n_1)u(n-n_2) \\ &+ \sum_{n_1=1}^{M}\sum_{n_2=1}^{M}\sum_{n_3=1}^{M} h_3(n_1,n_2,n_3) \\ &\quad \times u(n-n_1)u(n-n_2)u(n-n_3) \qquad (11)\end{aligned}$$

where $\{h_1(n_1)\}$, $\{h_2(n_1, n_2)\}$ and $\{h_3(n_1, n_2, n_3)\}$ represent the coefficients of the linear, quadratic and cubic Volterra kernels respectively. The first kernel is a M-vector, the second one is a $(M \times M)$ matrix with rank R and the third one can be considered as a third-order $(M \times M \times M)$ tensor with rank P.

By using the scalar representation (9) of the SVD and that of PARAFAC (10), equation (11) becomes equation (13) shown in next page.

The input/output relation (13) can be implemented in using a parallel-cascade structure, as shown in figure 2, where $U(n) = \left[\, u(n-1) \;\cdots\; u(n-M) \,\right]^T$, h_1 is the vector containing the coefficients of the linear kernel. $A_{.i}$ and $B_{.j}$ represent respectively the i^{th} column of A and the j^{th} column of B and the boxes

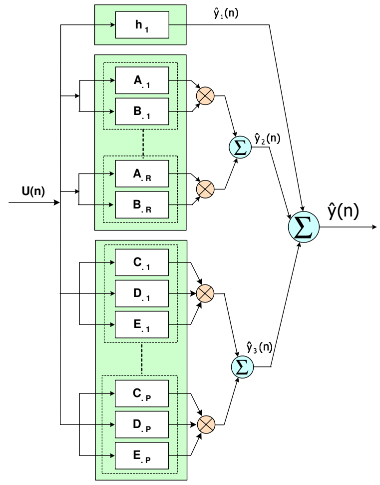

Fig. 2. Parallel-cascade realization of the SVD-PARAFAC based third-order Volterra model

of stage r correspond to the convolution operations $U^T(n)A_{.r}$ and $U^T(n)B_{.r}$. For the third-order kernel, the boxes of stage p correspond to the convolution operations $U^T(n)C_{.p}$, $U^T(n)D_{.p}$ and $U^T(n)E_{.p}$.

The cubic kernel in (11) can be viewed as a third-order $(M \times M \times M)$ tensor with a parameter complexity $C_3 = M^3$ in terms of its coefficients number. The quadratic kernel has a complexity $C_2 = M^2$ and the linear one $C_1 = M$. So, the complexity of a third-order standard Volterra model is $C_{Standard} = M + M^2 + M^3$.

The PARAFAC based third-order kernel complexity is $C_{parafac} = 3MP$ and the SVD-based second-order kernel complexity is $C_{SVD} = 2MR$. So, the complexity of the SVD-PARAFAC based Volterra model is :
$C_{SVD-PARAFAC} = M(1 + 2R + 3P)$.

Using SVD-PARAFAC decomposition, the Ratio of Complexity Reduction (RCR) with respect to the standard third-order Volterra model is $RCR = \frac{1+2R+3P}{1+M+M^2}$. When $P \ll M$ and $R \ll M$, a significant complexity reduction can be achieved.

3. THE ARLS ALGORITHM

Using the scalar representation of the SVD and PARAFAC given by (9) and (10), we have three different writings of the SVD-PARAFAC based third-order Volterra model output. Let us define :

$$\psi_r^A(n) = \sum_{n_2=1}^{M} b_{n_2r} u(n-n_2) \qquad (14)$$

$$\psi_r^B(n) = \sum_{n_1=1}^{M} a_{n_1r} u(n-n_1) \qquad (15)$$

and

$$\widehat{y}(n) = \sum_{n_1=1}^{M} h_1(n_1)u(n-n_1) + \sum_{n_1=1}^{M}\sum_{n_2=1}^{M}\left(\sum_{r=1}^{R} a_{n_1 r} b_{n_2 r}\right) u(n-n_1)u(n-n_2)$$
$$+ \sum_{n_1=1}^{M}\sum_{n_2=1}^{M}\sum_{n_3=1}^{M}\left(\sum_{p=1}^{P} c_{n_1 p} d_{n_2 p} e_{n_3 p}\right) u(n-n_1)u(n-n_2)u(n-n_3) \quad (12)$$
$$= \sum_{n_1=1}^{M} h_1(n_1)u(n-n_1) + \sum_{r=1}^{R}\left(\sum_{n_1=1}^{M} a_{n_1 r}\, u(n-n_1)\right)\left(\sum_{n_2=1}^{M} b_{n_2 r} u(n-n_2)\right)$$
$$+ \sum_{p=1}^{P}\left(\sum_{n_1=1}^{M} c_{n_1 p}\, u(n-n_1)\right)\left(\sum_{n_2=1}^{M} d_{n_2 p} u(n-n_2)\right)\left(\sum_{n_3=1}^{M} e_{n_3 p} u(n-n_3)\right) \quad (13)$$

$$\phi_p^C(n) = \sum_{n_2=1}^{M} d_{n_2 p} u(n-n_2) \sum_{n_3=1}^{M} e_{n_3 p} u(n-n_3) \quad (16)$$
$$\phi_p^D(n) = \sum_{n_1=1}^{M} c_{n_1 p} u(n-n_1) \sum_{n_3=1}^{M} e_{n_3 p} u(n-n_3) \quad (17)$$
$$\phi_p^E(n) = \sum_{n_1=1}^{M} c_{n_1 p} u(n-n_1) \sum_{n_2=1}^{M} d_{n_2 p} u(n-n_2) \quad (18)$$

3.1 The first writing of the model output

Equation (13) can be written as :

$$\widehat{y}(n) = \sum_{n_1=1}^{M}\left[h_1(n_1) + \sum_{r=1}^{R} a_{n_1 r}\psi_r^A(n) + \sum_{p=1}^{P} c_{n_1 p}\phi_p^C(n)\right] \times u(n-n_1) \quad (19)$$
$$= \underbrace{\begin{bmatrix} h_1 \\ \mathrm{vec}(A) \\ \mathrm{vec}(C)\end{bmatrix}^T}_{\Theta_1^T} \underbrace{\left(\begin{bmatrix} 1 \\ \Psi_A(n) \\ \Phi_C(n)\end{bmatrix} \otimes U(n)\right)}_{P_1(n)} \quad (20)$$
$$= \Theta_1^T P_1(n) \quad (21)$$

where

$$\Psi_A(n) = \left[\, \psi_1^A(n)\ \psi_2^A(n)\ \cdots\ \psi_R^A(n)\,\right]^T \quad (22)$$
$$\Phi_C(n) = \left[\, \phi_1^C(n)\ \phi_2^C(n)\ \cdots\ \phi_P^C(n)\,\right]^T \quad (23)$$

3.2 The second writing of the model output

Equation (13) can also be written as :

$$\widehat{y}(n) = y_L(n) + \sum_{n_2=1}^{M}\left[\sum_{r=1}^{R} b_{n_2 r}\psi_r^B(n) + \sum_{p=1}^{P} d_{n_2 p}\phi_p^D(n)\right] \times u(n-n_2) \quad (24)$$
$$= y_L(n) + \underbrace{\begin{bmatrix} \mathrm{vec}(B) \\ \mathrm{vec}(D)\end{bmatrix}^T}_{\Theta_2^T} \underbrace{\left(\begin{bmatrix} \Psi_B(n) \\ \Phi_D(n)\end{bmatrix} \otimes U(n)\right)}_{P_2(n)} \quad (25)$$
$$= y_L(n) + \Theta_2^T P_2(n) \quad (26)$$

where

$$y_L(n) = \sum_{n_1=1}^{M} h_1(n_1)u(n-n_1) \quad (27)$$
$$\Psi_B(n) = \left[\, \psi_1^B(n)\ \psi_2^B(n)\ \cdots\ \psi_R^B(n)\,\right]^T \quad (28)$$
$$\Phi_D(n) = \left[\, \phi_1^D(n)\ \phi_2^D(n)\ \cdots\ \phi_P^D(n)\,\right]^T \quad (29)$$

3.3 The third writing of the model output

The third writing of equation (13) is :

$$\widehat{y}(n) = y_L(n) + y_Q(n) + \sum_{n_3=1}^{M}\left(\sum_{p=1}^{P} e_{n_3 p}\phi_p^E(n)\right) u(n-n_3) \quad (30)$$
$$= y_L(n) + y_Q(n) + \underbrace{[\mathrm{vec}(E)]^T}_{\Theta_3^T} \underbrace{(\Phi_E(n) \otimes U(n))}_{P_3(n)} \quad (31)$$
$$= y_L(n) + y_Q(n) + \Theta_3^T P_3(n) \quad (32)$$

where

$$y_Q(n) = \sum_{r=1}^{R} \psi_r^A(n)\psi_r^B(n) \quad (33)$$
$$\Phi_E(n) = \left[\, \phi_1^E(n)\ \phi_2^E(n)\ \cdots\ \phi_P^E(n)\,\right]^T \quad (34)$$

3.4 The ARLS algorithm

The ARLS algorithm uses the three writings of the SVD-PARAFAC based Volterra model output (21), (26) and (32). It updates the linear kernel h_1, the matrices A and B of the quadratic kernel SVD decomposition and the matrices C, D and E of the cubic kernel PARAFAC decomposition by minimizing the following least squares cost function η in an alternating way :

$$\eta(N) = \sum_{n=1}^{N} (y(n) - \widehat{y}(n))^2 \quad (35)$$

where $y(n)$ denotes the output of the system to be modeled and $\widehat{y}(n)$ denotes the output of the SVD-PARAFAC based Volterra model given by (13).

By substituting the output of the model $\widehat{y}(n)$ by its three writings (21), (26) and (32), the cost function $\eta(N)$ can be rewritten as :

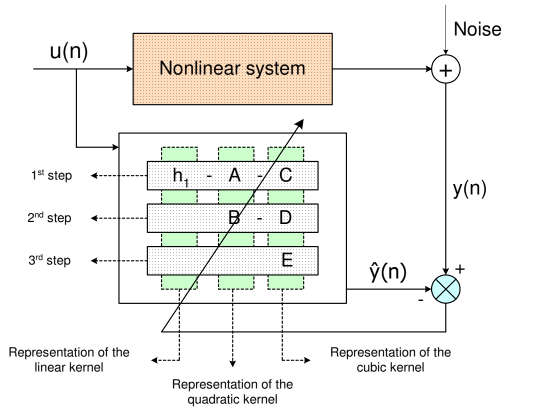

Fig. 3. Update of the SVD-PARAFAC based Volterra model components using the ARLS algorithm

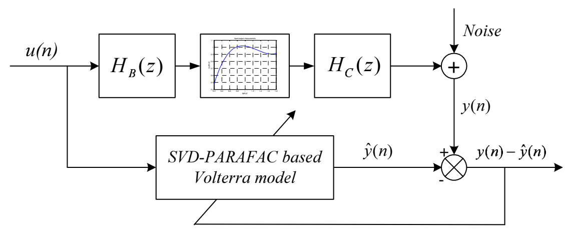

Fig. 4. Use of a SVD-PARAFAC based third-order Volterra model to represent a nonlinear satellite channel

$$\eta_1(N) = \sum_{n=1}^{N} \left(y(n) - \Theta_1^T P_1(n)\right)^2 \quad (36)$$

$$\eta_2(N) = \sum_{n=1}^{N} \left(y(n) - y_L(n) - \Theta_2^T P_2(n)\right)^2 \quad (37)$$

$$\eta_3(N) = \sum_{n=1}^{N} \left(y(n) - y_L(n) - y_Q(n) - \Theta_3^T P_3(n)\right)^2 \quad (38)$$

By alternatively minimizing the cost functions $\eta_1(N)$, $\eta_2(N)$ and $\eta_3(N)$ with respect to Θ_1, Θ_2 and Θ_3 respectively, we update the estimated matrices A, B, C, D and E, and the linear kernel h_1. The ARLS algorithm is illustrated by means of figure 3 and its equations are summarized in table 1.

4. SIMULATION RESULTS

The simulated system is a simplified model of a nonlinear satellite channel (Saleh, 1998) represented in figure 4 and characterized by two fourth-order low-pass linear filters denoted by $H_B(z)$ and $H_C(z)$, and a memoryless nonlinear device defined by its input-output characteristic $A(r)$ represented in figure 4 and modeled by means of a third-order polynomial.

The performance of the proposed identification method is evaluated by using the Normalized Mean Square Error (NMSE$_{output}$) between the system output $y(n)$ and the output $\widehat{\widehat{y}}(n)$ of the SVD-PARAFAC based Volterra model reconstructed with the parameters obtained at the convergence of the ARLS algorithm. It is calculated as :

$$NMSE_{output}(N) = \frac{\sum_{n=1}^{N}\left(y(n) - \widehat{\widehat{y}}(n)\right)^2}{\sum_{n=1}^{N} y^2(n)} \quad (39)$$

Table 1. The ARLS algorithm

(1) **Initialization**

- $\widehat{\Theta}_1(0)$, $\widehat{\Theta}_2(0)$ and $\widehat{\Theta}_3(0)$
- $\begin{cases} Q_1(0) = I_{M(P+R+1)}, \\ Q_2(0) = I_{M(P+R)}, \\ Q_3(0) = I_{MP} \end{cases}$

(2) **Update of the SVD-PARAFAC based Volterra model components**

- $U(n) = \left[\, u(n-1) \ \cdots \ u(n-M) \,\right]^T$
- $$\begin{cases} \text{Calculate } \psi_r^A(n) \text{ and } \phi_p^C(n) \\ \text{Construct } \Psi_A(n) \text{ and } \Phi_C(n) \\ P_1(n) = \left[\, 1 \ \Psi_A^T(n) \ \Phi_C^T(n) \,\right]^T \otimes U(n) \\ \varepsilon_1(n) = y(n) - P_1^T(n)\widehat{\Theta}_1(n-1) \\ K_1(n) = \dfrac{Q_1(n-1)P_1(n)}{1 + P_1^T(n)Q_1(n-1)P_1(n)} \\ Q_1(n) = [I - K_1(n)P_1^T(n)]Q_1(n-1) \\ \widehat{\Theta}_1(n) = \widehat{\Theta}_1(n-1) + K_1(n)\varepsilon_1(n) \\ \widehat{\Theta}_1(n) = \left[\, \widehat{h}_1^T(n) \ \ (\text{vec}\,\widehat{A}(n))^T \ \ (\text{vec}\,\widehat{C}(n))^T \,\right]^T \end{cases}$$
- $$\begin{cases} \text{Calculate } y_L(n)\ ,\ \psi_r^B(n) \text{ and } \phi_p^D(n) \\ \text{Construct } \Psi_B(n) \text{ and } \Phi_D(n) \\ P_2(n) = \left[\, \Psi_B^T(n) \ \Phi_D^T(n) \,\right]^T \otimes U(n) \\ \varepsilon_2(n) = y(n) - y_L(n) - P_2^T(n)\widehat{\Theta}_2(n-1) \\ K_2(n) = \dfrac{Q_2(n-1)P_2(n)}{1 + P_2^T(n)Q_2(n-1)P_2(n)} \\ Q_2(n) = [I - K_2(n)P_2^T(n)]Q_2(n-1) \\ \widehat{\Theta}_2(n) = \widehat{\Theta}_2(n-1) + K_2(n)\varepsilon_2(n) \\ \widehat{\Theta}_2(n) = \left[\, (\text{vec}\,\widehat{B}(n))^T \ \ (\text{vec}\,\widehat{D}(n))^T \,\right]^T \end{cases}$$
- $$\begin{cases} \text{Calculate } y_Q(n) \text{ and } \phi_p^E(n) \\ \text{Construct } \Phi_E(n) \\ P_3(n) = \Phi_E(n) \otimes U(n) \\ \varepsilon_3(n) = y(n) - y_L(n) - y_Q(n) \\ \qquad -P_3^T(n)\widehat{\Theta}_3(n-1) \\ K_3(n) = \dfrac{Q_3(n-1)P_3(n)}{1 + P_3^T(n)Q_3(n-1)P_3(n)} \\ Q_3(n) = [I - K_3(n)P_3^T(n)]Q_3(n-1) \\ \widehat{\Theta}_3(n) = \widehat{\Theta}_3(n-1) + K_3(n)\varepsilon_3(n) \\ \widehat{\Theta}_3(n) = \text{vec}(\widehat{E}(n)) \end{cases}$$

(3) **Reconstruction of the Volterra model kernels**

- linear kernel $\widehat{h}_1$.
- quadratic kernel $\widehat{h}_2$:

$$\widehat{h}_2(n_1, n_2) = \sum_{r=1}^{R} \widehat{a}_{n_1 r}\widehat{b}_{n_2 r}$$

- cubic kernel $\widehat{h}_3$:

$$\widehat{h}_3(n_1, n_2, n_3) = \sum_{p=1}^{P} \widehat{c}_{n_1 p}\widehat{d}_{n_2 p}\widehat{e}_{n_3 p}$$

(4) **Go back to step 2 until convergence of the algorithm**

Table 2. NMSE_{output} obtained with the SVD-PARAFAC based Volterra model for different values of P and R

	R=1	**R=2**	**R=3**	**R=4**	**R=5**	**R=6**
P=1	$4.639\ 10^{-2}$	$3.448\ 10^{-2}$	$2.347\ 10^{-2}$	$2.102\ 10^{-2}$	$1.974\ 10^{-2}$	$1.842\ 10^{-2}$
P=2	$4.638\ 10^{-2}$	$3.202\ 10^{-2}$	$1.947\ 10^{-2}$	$1.871\ 10^{-2}$	$1.779\ 10^{-2}$	$1.681\ 10^{-2}$
P=3	$4.331\ 10^{-2}$	$3.199\ 10^{-2}$	$1.893\ 10^{-2}$	$1.843\ 10^{-2}$	$1.719\ 10^{-2}$	$1.638\ 10^{-2}$
P=4	$4.294\ 10^{-2}$	$3.124\ 10^{-2}$	$1.865\ 10^{-2}$	$1.798\ 10^{-2}$	$1.681\ 10^{-2}$	$1.590\ 10^{-2}$
P=5	$4.243\ 10^{-2}$	$3.079\ 10^{-2}$	$1.802\ 10^{-2}$	$1.771\ 10^{-2}$	$1.653\ 10^{-2}$	$1.545\ 10^{-2}$
P=6	$4.228\ 10^{-2}$	$3.045\ 10^{-2}$	$1.754\ 10^{-2}$	$1.722\ 10^{-2}$	$1.605\ 10^{-2}$	$1.511\ 10^{-2}$

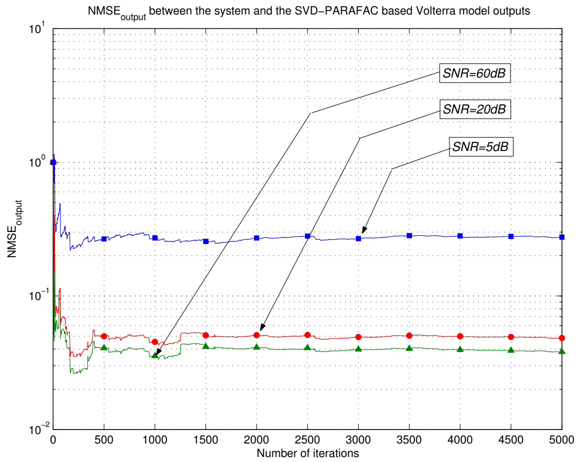

Fig. 5. NMSE_{output} for three different SNRs

The impulse responses of filters H_B and H_C become smaller than 10^{-2} after 20 points. Therefore, we model the simulated system as a cubic Wiener-Hammerstein model i.e. a symmetric third-order Volterra filter with memory M=39. Using a SVD-PARAFAC based Volterra model with $M = 39$, $P = 1$ and $R = 1$ we achieve an important complexity reduction $RCR = 2.04\ 10^{-2}$.

Figure 5 shows the variation of the NMSE_{output} as a function of the iterations number for three different SNRs under the same previous conditions ($P = 1$, $R = 1$ and $M = 39$). The input signal is a gaussian white noise sequence of length $N = 10000$ with zero mean and unit variance. The simulation results were obtained using the Monte Carlo method with 20 different additive noise sequences.

Table 2 contains the NMSE_{output} in function of the PARAFAC factors number P and the SVD factors number R for $SNR = 20\,dB$. The NMSE_{output} slightly decreases as the values of R and P increase, hence the interest in choosing small values of P and R to reduce the parametric complexity.

From these simulation results, we can conclude that the SVD-PARAFAC decomposition approach allows to represent a nonlinear satellite channel with a relatively small modeling error. The choice of numbers P and R of the PARAFAC and SVD factors doesn't influence very much the identification performance.

5. CONCLUSION

In this paper, we have presented a new approach to represent and identify third-order Volterra models using the Singular Value Decomposition and the PARAFAC decomposition, which allows to significantly reduce the parametric complexity of the corresponding kernels. The ARLS algorithm has been proposed to identify such a decomposition. Extension of this work to Volterra models of order higher than three is under study.

REFERENCES

Boyd, S. and L. O. Chua (1985). Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Tr. Circuits and Systems* **32**(11), 1150–1171.

Campello, R. J. G. B., G. Favier and W. C. Amaral (2004). Optimal expansions of discrete-time Volterra models using Laguerre functions. *Automatica* **40**, 815–822.

Carroll, J.D. and J. J. Chang (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika* **35**, 283–319.

Dumont, G. A. and Y. Fu (1993). Nonlinear adaptive control via Laguerre expansion of Volterra kernels. *Int. J. Adaptive Control and Signal Processing* **7**, 367–382.

Favier, G., A. Kibangou and R. J. G. B. Campello (2003). Nonlinear systems modelling by means of generalized orthonormal basis functions. *Invited paper, IEEE Conference on Signals, Systems, Decision and information technology, SSD'03, Sousse, Tunisia.*

Harshman, R. A. (1970). Foundation of the parafac procedure : Models and conditions for an "explanatory" multimodal factor analysis. *UCLA working papers in phonetics* **16**, 1–84.

Kibangou, A., G. Favier and M. M. Hassani (2003). A growing approach for selecting generalized orthonormal basis functions in the context of system modeling. In: *Proc. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, NSIP'03*. Grado, Italy. pp. 1119–1124.

Kruskal, J. B. (1977). Three-way arrays : Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications* **18**, 95–138.

Panicker, T. M. and V. J. Mathews (1998). Parallel-cascade realizations and approximations of truncated volterra systems. *IEEE Trans. Signal Processing* **46**(11), 2829–2832.

Saleh, A. A. M. (1998). Frequency independent and frequency dependent nonlinear models of twt amplifiers. *IEEE Tr. on Communications* **29**(11), 1715–1720.