Tensor Factorization based Estimates of Parallel Wiener Hammerstein Models

David T. Westwick* Mariya Ishteva** Philippe Dreesen** Johan Schoukens**

* Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, 2500 University Drive NW, Calgary Alberta, T2N 1N4 Canada (e-mail: dwestwic@ucalgary.ca)
** Department of Fundamental Electricity and Instrumentation (ELEC), Vrije Universiteit Brussel (VUB), Pleinlaan 2, 1050 Brussels, Belgium (e-mail: mariya.ishteva@vub.ac.be, philippe.dreesen@gmail.com, Johan.Schoukens@vub.ac.be)

Abstract: Factoring the third-order Volterra kernel of a Wiener-Hammerstein model to recover the impulse responses of its two constituent linear systems is a common example in the multilinear algebra literature. Since recent progress in regularization-based system identification has enabled the practical estimation of the third-order Volterra kernel, these tensor factorization based approaches have become attractive. We extend one of these Wiener-Hammerstein factorization methods to the case of the Parallel Wiener-Hammerstein model, since, unlike the WH model, this structure is a universal approximator for Volterra systems. The efficacy of the method is demonstrated using numerical simulations.

Keywords: Block Structured Models, Volterra Kernels, Tensors, CPD

1. INTRODUCTION

System identification is the process of extracting mathematical models of dynamic systems from measurements of their inputs and outputs. The identification of linear systems is a well studied problem, and is documented in textbooks including those by Pintelon and Schoukens (2001) and Ljung (1999). Physical systems are inherently nonlinear. When using linear identification tools, the effects of nonlinear distortion must be minimized, but they are always present (Schoukens et al., 2016).

The Volterra series, which consists of a series of generalized multi-dimensional convolutions between the input and series of multi-dimensional generalized impulse responses, provides a general model capable of approximating any fading memory nonlinear system to within arbitrary accuracy (Boyd and Chua, 1985). Although the model is linear in the parameters, the number of parameters grows combinatorially with the memory length of the system, and with the degree of the nonlinearity (Westwick and Kearney, 2003).

Block-structured models, consisting of one or more branches of alternating dynamic linear and memoryless nonlinear systems are widely used to model nonlinear systems (Giri and Bai, 2010). The simplest of these are the Hammerstein and Wiener models. The Hammerstein model consists of a memoryless nonlinear followed by a dynamic linear system (NL) while in the Wiener system, the order of the elements is reversed (LN). The Wiener-Hammerstein (W-H) system is made up of two dynamic linear systems, separated by a memoryless nonlinearity (LNL).

The third-order Volterra kernel of a W-H system can be factorized using tensor operations to yield the impulse responses of the two linear elements (Kibangou and Favier, 2010), (Favier et al., 2012), (de Goulart et al., 2016). While these works did not consider the difficulty inherent in estimating high-order Volterra kernels, recent progress in the development of regularization-based system identification methods has made this plausible (Birpoutsoukis and Schoukens, 2015).

Unlike the Volterra series, the W-H model is not a universal approximator, and has limited applicability as a result. Boyd and Chua (1985) showed that the parallel Wiener system is a universal approximator, in that it can represent any system that admits a Volterra series representation to within arbitrary accuracy. Palm (1979) suggested that the Parallel Wiener-Hammerstein (PWH) system, consisting of multiple paths each with the LNL structure, would provide a more efficient representation, but was not aware of any methods for its identification.

In this paper, we will extend one of the non-iterative tensor factorizations proposed by de Goulart et al. (2016) to the case of a PWH model. The balance of the paper is organized as follows. This section will conclude with an overview of the notation used throughout the paper. Section 2 will present background material on nonlinear sys-
tem models and tensor decompositions, and will conclude with an overview of the algorithm proposed by de Goulart et al. (2016). Section 3 will propose an extension of the algorithm to the PWH model. Sections 4 and 5 will present the results of some numerical experiments, and conclude the paper, respectively.

1.1 Notation

Lower and upper case letters in a regular type-face, $a, B$, will refer to scalars, bold faced lower case letters, $\mathbf{a}$, refer to vectors, matrices are indicated by bold faced upper case letters, $\mathbf{A}$, and bold faced calligraphic script will be used for tensors, $\mathcal{A}$. The Kronecker product will be denoted by $\otimes$, and $\circ$ will be used for the Khatri-Rao (or column-wise Kronecker) product. The outer, or tensor, product will be denoted $\otimes$. The operator $\text{diag}(\mathbf{a})$ constructs a diagonal matrix with the entries of the vector $\mathbf{a}$ on its diagonal, while $\text{vec}(\mathbf{A})$ stacks the columns of the matrix $\mathbf{A}$ into a vector.

2. BACKGROUND

2.1 Nonlinear System Models

A wide variety of nonlinear systems (Boyd and Chua, 1985) may be represented by a finite Volterra series model:

$$y(t) = \sum_{\ell=0}^{L} \sum_{\tau_1=0}^{M-1} \cdots \sum_{\tau_{\ell}=0}^{M-1} h^{(\ell)}(\tau_1, \ldots, \tau_{\ell}) \prod_{j=1}^{\ell} u(t - \tau_j), \quad (1)$$

where $u(t)$ and $y(t)$ are the input and output at (discrete) time $t$, respectively, and $M$ is the memory length of the system. The output is computed by a series of multidimensional convolutions with the system’s Volterra kernels, $h^{(\ell)}(\tau_1, \ldots, \tau_{\ell})$, where $\ell$ is the degree of nonlinearity of the kernel.

While the Volterra series is a very general model for nonlinear systems, it is difficult to work with. Block structured models (Giri and Bai, 2010), sequences of alternating dynamic linear and static nonlinear elements, have several advantages. They have relatively few parameters, and are hence easy to simulate and analyze. They can also provide insight into the structure of the underlying system.

As shown in Fig. 1, a W-H model comprises two LTI filters separated by a memoryless nonlinearity. In this paper, these will be represented by FIR filters:

$$x(t) = \sum_{\tau=0}^{M_1-1} g(\tau)u(t - \tau)$$
$$y(t) = \sum_{\tau=0}^{M_2-1} h(\tau)w(t - \tau),$$

with memory lengths of $M_1$ and $M_2$ samples respectively. The nonlinearity will be modelled as a polynomial,

$$w(t) = f(x(t)) = \sum_{l=0}^{L} c_l x^l(t),$$

to facilitate computation of the system’s Volterra kernels. Thus, the output of the system is given by:

\begin{tikzpicture}
		node[coordinate] (n1) at (0,0) {$u(t)$};
	
		node[coordinate] (n2) [right of=n1] {$x(t)$};
			node[coordinate] (n3) [right of=n2] {$f(\cdot)$};
			node[coordinate] (n4) [right of=n3] {$w(t)$};
		node[coordinate] (n5) [right of=n4] {$h(\tau)$};
		node[coordinate] (n6) [right of=n5] {$y(t)$};
		node[coordinate] (n7) [below of=n2] {$f(y)$};

		node[draw,shape=circle,inner sep=0.5mm] (e1) [right of=n3] {}; 
		node[draw,shape=circle,inner sep=0.5mm] (e2) [right of=n5] {}; 
		node[draw,shape=circle,inner sep=0.5mm] (e3) [right of=n6] {}; 
		node[draw,shape=circle,inner sep=0.5mm] (e4) [right of=n7] {}; 
	
		node[draw,shape=rectangle, rounded corners=0.5mm] (i1) [right of=n1] {$g(\tau)$};
		node[draw,shape=rectangle, rounded corners=0.5mm] (i2) [right of=n2] {$x(t)$};
		node[draw,shape=rectangle, rounded corners=0.5mm] (i3) [right of=n3] {$f(\cdot)$};
		node[draw,shape=rectangle, rounded corners=0.5mm] (i4) [right of=n4] {$w(t)$};
		node[draw,shape=rectangle, rounded corners=0.5mm] (i5) [right of=n5] {$h(\tau)$};
		node[draw,shape=rectangle, rounded corners=0.5mm] (i6) [right of=n6] {$y(t)$};
		node[draw,shape=rectangle, rounded corners=0.5mm] (i7) [right of=n7] {$f(y)$};

	node[draw,shape=rectangle, rounded corners=0.5mm] (i8) [right of=i1] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i9) [right of=i2] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i10) [right of=i3] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i11) [right of=i4] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i12) [right of=i5] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i13) [right of=i6] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i14) [right of=i7] {}; 

	node[draw,shape=rectangle, rounded corners=0.5mm] (i15) [right of=i8] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i16) [right of=i9] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i17) [right of=i10] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i18) [right of=i11] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i19) [right of=i12] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i20) [right of=i13] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i21) [right of=i14] {}; 

	node[draw,shape=rectangle, rounded corners=0.5mm] (i22) [right of=i15] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i23) [right of=i16] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i24) [right of=i17] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i25) [right of=i18] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i26) [right of=i19] {}; 
	node[draw,shape=rectangle, rounded corners=0.5mm] (i27) [right of=i20] {}; 

Fig. 1. Block diagram of a Wiener-Hammerstein model.

The two linear elements are represented by their impulse responses, $g(\tau)$ and $h(\tau)$.

$$y(t) = \sum_{\tau=0}^{M_2-1} h(\tau) \sum_{\ell=0}^{L} \left( \sum_{a=0}^{M_1-1} g(a)u(t - \tau - a) \right) \ell \quad (2)$$

Comparing (2) and (1), it can be shown that the $\ell$th degree Volterra kernel of the W-H model is given by (Rugh, 1981):

$$h^{(\ell)}(\tau_1, \ldots, \tau_{\ell}) = c_{\ell} \sum_{k=0}^{M_2-1} h(k) \prod_{j=1}^{\ell} g(\tau_j - k). \quad (3)$$

2.2 Canonical Polyadic Decomposition

Let $\mathcal{H}_3$ be a third-degree tensor that contains the third-degree Volterra kernel of a system. It is symmetric with respect to any interchange of its indices, and has dimensions $M \times M \times M$, where $M$ is the memory length of the system. This tensor can be represented as a sum of $R$ rank-1 terms,

$$\mathcal{H}_3 = \sum_{r=1}^{R} (a_r^{(1)} \otimes a_r^{(2)} \otimes a_r^{(3)}), \quad (4)$$

known as a polyadic decomposition (Kolda and Bader, 2009). Such polyadic decompositions can be constructed for tensors of higher degrees. These are generalizations of the SVD of a matrix, although unlike the SVD, the vectors need not be orthogonal to each other.

Defining the factor matrices as:

$$\mathcal{A}^{(k)} = \begin{bmatrix} a_1^{(k)} & a_2^{(k)} & \cdots & a_R^{(k)} \end{bmatrix},$$

the decomposition in (4) will be indicated by the shorthand notation:

$$\mathcal{H}_3 = [\mathcal{A}^{(1)}, \mathcal{A}^{(2)}, \mathcal{A}^{(3)}]. \quad (5)$$

Note that due to the symmetry of the tensor (kernel), all of the factor matrices $\mathcal{A}^{(1)} = \mathcal{A}^{(2)} = \mathcal{A}^{(3)} = \mathcal{A}$, are equal.

The rank of a tensor is defined as the minimal $R$ such that the decomposition (4) holds. If $R$ is minimal, then (4) is called the Canonical Polyadic Decomposition (CPD) of the tensor $\mathcal{H}_3$. The CPD has been independently discovered, and named both the CANDECOMP and the PARAFAC, among others (Kolda and Bader, 2009). For tensors of degree 3 or higher, the CPD has been shown to be unique under mild conditions, modulo changes in the ordering of the rank-one terms in (4), and with respect to the distribution of the scaling between the elements in each of the rank-one terms.

2.3 CPD of a Wiener-Hammerstein Model

The CPD of the third-order Volterra kernel of a W-H system, shown in (3), may be written as (de Goulart et al., 2016):

$$\mathcal{H}_3 = [G_1, G_1, G_1\text{diag}(c_3h_1)], \quad (6)$$
where \( h_1 \) is a vector containing the second impulse response, \( h(\tau) \), and the matrix \( G_1 \in \mathbb{R}^{M \times M_2} \) is Toeplitz structured with entries derived from the impulse response of the first linear element, \( g(t) \), as follows

\[
G_1(i,j) = \begin{cases} 
g(i-j) & i \geq j \\
0 & i < j 
\end{cases} \tag{7}
\]

Note that the last factor matrix in (6) has been post-multiplied by a diagonal matrix, which scales the terms in (4). The scaling could have been applied to any one of the factor matrices. Let the vector \( g_1 \) contain the impulse response \( g(t) \).

The matrix unfolding of the third kernel of a W-H model, given by the CPD in (6), can be written as (de Goultar et al., 2016):

\[
X = G_1 \text{diag}(c_3 h_1) (G_1 \odot G_1)^T \in \mathbb{R}^{M \times M^2}.
\]

Note that due to the symmetry of the kernel, the tensor may be unfolded along any dimension.

**Lemma 2.1.** Provided that the neither \( g(t) \) nor \( c_3 \) are equal to zero, the rank of matrix unfolding of the third-degree kernel of a W-H system is equal to the number of non-zero elements in the second impulse response, \( h(t) \).

**Proof** Since \( g_1^T g_1 \neq 0 \), both \( G_1 \) and \( G_1 \odot G_1 \) will have full column rank \( (M_2) \). Thus each non-zero entry in \( h_1 \) will contribute a linearly independent row and column to the matrix \( X \). \( \square \)

Compute the SVD of the matrix unfolding, \( X \), and partition it:

\[
X = [U_1 \ U_2] \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \tag{8}
\]

such that \( S_1 \in \mathbb{R}^{M_1 \times M_1} \). Since the rank of \( X \) is bounded by \( M_2 \), \( X = U_1 S_1 V_1^T \). Then, there exists an invertible matrix \( N \in \mathbb{R}^{M_2 \times M_2} \) such that (\( \text{Sorensen and Comon}, 2013; \text{de Goultar et al., 2016} \)):

\[
NV_1^T = (G_1 \odot G_1)^T \tag{9}
\]

\[
U_1 S_1 N^{-1} = G_1 \text{diag}(c_3 h_1). \tag{10}
\]

The Toeplitz-structured factor matrix \( G_1 \) can be written as a basis expansion:

\[
G_1 = \sum_{k=0}^{M_1-1} D_k g(k), \tag{11}
\]

where \( D_k \) is a \( M_2 \times M_2 \) matrix whose entries are given by the Kronecker delta: \( D_k(i,j) = \delta(i + j - k) \). Thus

\[
\text{vec}(G_1 \odot G_1) = \sum_{i,j=0}^{M_1-1} \text{vec}(D_i \odot D_j) g(i)g(j) \tag{12}
\]

\[
= D \theta, \tag{13}
\]

where

\[
\theta = \text{vec}(g_1 g_1^T). \tag{14}
\]

Applying the identity

\[
\text{vec}(ABC) = (C^T \boxtimes A) \text{vec}(B)
\]

to \( NT V_1 \) and substituting into (9) results in

\[
(I_{M_2} \boxtimes V_1)\text{vec}(NT) = D \theta.
\]

This must be solved for \( N \) and \( \theta \). Let

\[
M = \begin{bmatrix} -IM_2 \otimes V_1 & D \end{bmatrix}, \tag{15}
\]

and solve

\[
M \begin{bmatrix} \text{vec}(NT) \\ \theta \end{bmatrix} = 0.
\]

Let

\[
M = U_M S_M V_M^T \tag{16}
\]

be the SVD of \( M \). Given the uniqueness results in (Sorensen and Comon, 2013), the last column of \( V_M \) will be proportional to \( \text{vec}(NT)^T \theta)^T \). The last \( M_1^2 \) elements of this singular vector correspond to \( \theta \), which can be reshaped into a \( M_1 \times M_1 \) matrix,

\[
G_0(i,j) = \tilde{\theta}_i (M_1 j + i). \tag{17}
\]

Ideally, \( G_0 \) will be a rank-1 matrix, see (14), from which an estimate of \( g_1 \) can be extracted using a SVD.

**3. PARALLEL WIENER HAMMERSTEIN MODELS**

The Parallel Wiener-Hammerstein model is obtained by summing the outputs of several Wiener-Hammerstein models, as shown in Fig. 2. Similarly the Volterra kernels of a PWH model are obtained by summing the kernels of the individual branches, so that

\[
h^{(i)}(\tau_1, \ldots, \tau_t) = \sum_{p=1}^P \sum_{k=0}^{M_2-1} h_p(k) \prod_{j=1}^t g_p(\tau_j - k), \tag{18}
\]

where the subscript \( p \) indexes the paths in the model.

Let \( G_k \) be a Toeplitz matrix whose columns are shifted copies of the impulse response \( g_k(t) \), as in (7), and define the matrices:

\[
G = [G_1 \ G_2 \ldots G_P] \\
\Gamma = [g_1 \ g_2 \ldots g_P] \\
h = \text{vec} \left( c_3 h_1 c_3 h_2 \ldots c_3 h_P \right).
\]

Then the third-order kernel can be written as the following, not necessarily canonical, polyadic decomposition:

\[
\mathcal{H}_3 = [G \ G \ \text{diag}(h)]. \tag{19}
\]

Using this polyadic decomposition, any matrix unfolding of the symmetric third-order kernel can be written as

\[
X = G \text{diag}(h) (G \odot G)^T.
\]

**Lemma 3.1.** The rank of \( X \), the matrix unfolding of the third-order kernel of a PWH model is less than or equal to the least of: \( M \), \( P \cdot \text{rank}(\Gamma) \), or the number of non-zero entries in \( h \).
Proof From (7), each group of $M_2$ columns in $G$ will have full rank, provided that the corresponding impulse response, $g_k(\tau)$, is not identically zero. Thus, the rank of $G \in \mathbb{R}^{M \times PM_2}$ will be the lesser of $M$, and $P \cdot \text{rank}(\Gamma)$. The rank of $G \odot G$ will be greater or equal to that of $G$. The number of non-zero entries in $h$ gives the rank of the diagonal matrix. The rank of $X$ is less than or equal to the lowest of these ranks. □

Let $R = \min(M, PM_2)$, so that $R \geq \text{rank}(X)$. Then, $X = U_1 S_1 V_1^T$, were $S_1 \in \mathbb{R}^{R \times R}$, and let $N$ be an invertible, $R \times R$ matrix:

$$X = G \, \text{diag}(h) \, NN^{-1} \, (G \odot G)^T = U_1 S_1 V_1^T.$$  

The matrix $N$ is chosen such that

$$V_1 N^T = (G \odot G).$$  

(20)

Since the Khatri-Rao product is the column-wise Kronecker product, we have

$$G \odot G = [G_1 \, G_2 \ldots \, G_P] \odot [G_1 \, G_2 \ldots \, G_P] = \{ [G_1 \odot G_1] \, [G_2 \odot G_2] \ldots \, [G_P \odot G_P] \}.$$  

Let $N_k$, for $k = 1 \ldots P$, denote the $k$'th group of $M_2$ columns in the matrix $N^T$. Thus we have

$$V_1 N_k = G_k \odot G_k.$$  

(21)

Since each of the $G_k$ is a Toeplitz structured matrix, the basis expansion in (11) may be used. Thus

$$\text{vec}(G_k \odot G_k) = D \theta_k,$$  

(22)

where the matrix $D$ was defined in (12) and

$$\theta_k = \text{vec}(g_k h_k^T).$$

Thus, for each submatrix $N_k$, we have

$$[\! - I_{M_2} \otimes V_1 \! ] \, D \, \text{vec}(N_k) = 0.$$  

(23)

Therefore each pathway in the model will generate one dimension in the null-space of the the matrix $M$, as defined in (15). Thus, compute the SVD in (16), and let $\Theta$ comprise the last $M_2$ entries from each of the last $P$ columns of the matrix $V_1\Lambda$. These contain the entries in the null-space corresponding to the parameter vectors $\theta_k$.

Due to the use of the SVD, each column of $\Theta$ will be a linear combination of the $\theta_k$. Thus

$$\Theta = \sum_{k=1}^{P} \theta_k w_k^T,$$

for some weighting vectors $w_k$. Reshape $\Theta$ as a $M_3 \times M_3 \times P$ tensor by creating an $M_3 \times M_3$ matrix from each of the $P$ columns in $\Theta$, and then stacking them:

$$\mathcal{G}(i, j, k) = \Theta(M_3(j - 1) + i, k).$$  

(24)

Since each column in $\Theta$ is a linear combination of the $\theta_k$, the resulting tensor has the CPD

$$\mathcal{G} = [\Gamma, \Gamma, W],$$

where $W = [w_1 \, w_2 \ldots \, w_P]$.

Given an estimate of $\Gamma$, construct the matrix factor $\mathcal{G}$. Then $h$ may be obtained by the linear regression:

$$\text{vec}(H_3) = (G \odot G \odot G) \, h.$$  

(25)

3.1 Algorithm

Given The third-degree Volterra kernel of dimension $M \times M \times M$ of a Parallel Wiener Hammerstein system, and upper bounds on the memory lengths of the two banks of linear filters, $M_1$ and $M_2$, such that $M = M_1 + M_2 - 1$.

(1) Compute and partition the SVD of the matrix unfolding of the kernel as described in (8).

(2) Construct the matrix $M$ in (23), and form the tensor $\mathcal{G}$ from the null-space of $M$.

(3) Calculate the CPD of $\mathcal{G}$ to recover the impulse responses $g_k$.

(4) Solve the linear regression (25) to recover the remaining impulse responses.

4. SIMULATIONS

The algorithm described in Section 3.1 was tested on the third-order Volterra kernel of a PWH model with 3 branches, where the first linear elements, $g_k(\tau)$, had memory lengths of 32 samples, and the output filters, $h_k(\tau)$ each had a memory of 5 samples. Thus, the third-degree kernel was of dimension $36 \times 36 \times 36$ elements. In the first simulation, the algorithm was applied to a noise-free copy of the kernel. This was followed by a series of Monte-Carlo simulations where white Gaussian noise was added to the kernels at Signal-to-Noise Ratios of 40, 30, 20 and 10 dB. 100 trials were performed at each noise level. All computations were done using Tensorlab v3.0 (Vervliet et al., 2016).

Fig. 3 shows the singular values of the matrix unfolding of the third-order kernel. Since all 3 of the $h_k(t)$ had memory lengths of 5 samples, a rank of 15 is expected. In the noise-free case, shown with blue asterisks, there is gap of about 8 orders of magnitude between the 15'th and 16'th singular values, while there is little decay beyond the 16'th singular value. This suggests that these represent the noise floor, and that the matrix has rank 15, corresponding to structure that was simulated.
In each trial of the Monte-Carlo simulations, noise was added to the third-order kernel. Since the kernel is symmetric with respect to any interchanges of arguments, this symmetry was imposed on the noise. Thus, the same noise sample was added to each symmetric point in the kernel. Other than that, the noise samples were independent.

The effects of this noise are evident in Fig. 3, where the noise floor has increased from a level of $10^{-16}$ in the noise-free experiment (blue asterisks) to $10^{-3}$ with 30 dB of noise (red crosses), and $10^{-2}$ with 10 dB (green plusses). As a result, gap between the signal and noise subspaces has vanished, and 7 and 10, respectively, of the smallest singular values from the signal subspace have disappeared below the noise-floor. Thus, the information that was present in the singular vectors associated with these small singular values has been spread into the noise subspace.

As a result, the nullspace in (23) was also poorly characterized (not shown), as its singular vectors were only a factor of 2-3 smaller than the remaining singular values, as opposed to the difference of $10^8$ evident in Fig. 4. The resulting parameter tensor was not well approximated by a rank-3 CPD. Figure 6 shows impulse responses of the first filter bank, estimated from the 30 dB simulations. It compares the simulated impulse responses with the mean plus or minus 2.5 standard deviations of the ensemble of estimates from the Monte-Carlo simulation. While the noise has produced some variability in the impulse response estimates, this is much less significant that the resulting bias error. This bias error appears to be due to the effective truncation of the SVD of the matrix-unfolding of the kernel (see Fig. 3). The results from the 10 dB Monte-Carlo simulation are shown in Fig. 7. Note that the variance is more significant than the bias error.

As a result, the nullspace in (23) was also poorly characterized (not shown), as its singular vectors were only a factor of 2-3 smaller than the remaining singular values, as opposed to the difference of $10^8$ evident in Fig. 4. The resulting parameter tensor was not well approximated by a rank-3 CPD. Figure 6 shows impulse responses of the first filter bank, estimated from the 30 dB simulations. It compares the simulated impulse responses with the mean plus or minus 2.5 standard deviations of the ensemble of estimates from the Monte-Carlo simulation. While the noise has produced some variability in the impulse response estimates, this is much less significant that the resulting bias error. This bias error appears to be due to the effective truncation of the SVD of the matrix-unfolding of the kernel (see Fig. 3). The results from the 10 dB Monte-Carlo simulation are shown in Fig. 7. Note that the variance is more significant than the bias error.

The tensor constructed from the null-space, as described in (24) had a rank of three, as a three element CPD reconstructed the tensor to within machine precision. The rank-3 CPD produced excellent estimates of the $g_k(t)$, to within a rearrangement of the columns, and scaling of the vectors, as illustrated in Fig.5.

Fig. 4. Singular Values of the matrix in Eq. (23). The three singular values that are nearly 8 orders of magnitude smaller than the next smallest SV suggest a 3 dimensional nullspace, corresponding to the 3 paths in the PWH model.

Fig. 4 shows the singular values of the matrix $M$, defined in (15), obtained from the noise-free simulation. Again, there is a gap of approximately 8 orders of magnitude between the last 3 singular values, and the smallest of the remaining singular values. Thus, the span of the last 3 right singular vectors will provide a reliable estimate of the null-space.

Fig. 5. Simulated impulse responses, $g_k(t)$, from the first filter bank and their estimates obtained from the noise-free kernel. Note that the first two impulse responses were interchanged in the estimates. For comparison purposes, all impulse responses have been scaled to unit norm, with the largest peak in the positive direction.

The tensor constructed from the null-space, as described in (24) had a rank of three, as a three element CPD reconstructed the tensor to within machine precision. The rank-3 CPD produced excellent estimates of the $g_k(t)$, to within a rearrangement of the columns, and scaling of the vectors, as illustrated in Fig.5.
Table 1. Ensemble statistics from the Monte-Carlo Simulations. Bias errors are reported as the norm of the difference between the simulated impulse response and the ensemble mean of the estimates. Random errors are reported as the norm of the ensemble standard deviation of the estimates. All impulse responses were scaled to have unit norm.

| SNR (dB) | $\hat{g}_1(\tau)$ bias rand. | $\hat{g}_2(\tau)$ bias rand. | $\hat{g}_3(\tau)$ bias rand. |
|---------|-----------------|-----------------|-----------------|
| 40      | 0.455 0.006     | 0.014 0.007     | 0.307 0.004     |
| 30      | 0.463 0.010     | 0.113 0.015     | 0.279 0.007     |
| 20      | 0.490 0.018     | 0.312 0.021     | 0.182 0.013     |
| 10      | 0.515 0.033     | 0.300 0.025     | 0.128 0.028     |

5. CONCLUSION

This paper describes an algorithm that extracts the impulse responses of the linear elements of a parallel Wiener-Hammerstein model from its third-order Volterra kernel. This is an extension of the subspace based algorithm proposed by de Goulart et al. (2016) that can obtain the linear elements of Wiener-Hammerstein model from its third-order kernel. The algorithm proposed by de Goulart et al. (2016) requires the computation of 2 SVDs, and a single linear regression, and hence provides a closed-form solution without the need for an iterative optimization. In the algorithm developed in this paper, one of the SVDs must be replaced by the computation of the Canonical Polyadic Decomposition of a tensor. Since iterative methods, such as alternating least squares, are used to compute the CPD, our algorithm is technically an iterative algorithm. Nevertheless, the CPD involves a small tensor, and does not impose any structure on the result. As such, it can be expected to function reliably. Thus the proposed algorithm can provide initial estimates for iterative procedures, such as those described by Dreesen et al. (2017).

Fig. 7. Impulse responses, $g_k(t)$ from the first filter bank, and their estimates obtained from kernels corrupted with 10 dB of noise. The plot shows the 99% confidence intervals obtained from the Monte-Carlo simulations. Note that the bias more significant than the variance of the estimates.

REFERENCES

Birpoutsoukis, G. and Schoukens, J. (2015). Nonparametric Volterra kernel estimation using regularization. In IEEE Instrumentation and Measurement Technology Conference, 222–227.

Boyd, S. and Chua, L. (1985). Fading memory and the problem of approximating nonlinear operators with Volterra series. IEEE Transactions on Circuits and Systems, CAS-32(11), 1150–1161.

de Goulart, J., Boizard, M., Boyer, R., Favier, G., and Comon, P. (2016). Tensor CP decomposition with structured factor matrices: Algorithms and performance. IEEE Journal of Selected Topics in Signal Processing, 757–769.

Dreesen, P., Westwick, D., Schoukens, J., and Ishteva, M. (2017). Modeling parallel Wiener-Hammerstein systems using tensor decomposition of Volterra kernels. In 13th International Conference on Latent Variable Analysis and Signal Separation, 16–25.

Favier, G., Kibangou, A., and Bouilloc, T. (2012). Nonlinear system modeling and identification using Volterra-Parafac models. Adaptive Control and Signal Processing, 26(1), 30–53.

Giri, F. and Bai, E.W. (eds.) (2010). Block-Oriented Nonlinear System Identification. Lecture Notes in Control and Information Sciences. Springer, Berlin.

Kibangou, A. and Favier, G. (2010). Tensor analysis-based model structure determination and parameter estimation for block-oriented nonlinear systems. IEEE Journal of Selected Topics in Signal Processing, 4(3), 514–525.

Kolda, T. and Bader, B. (2009). Tensor decompositions and applications. SIAM Review, 51(3), 455–500.

Ljung, L. (1999). System Identification: Theory for the User. Prentice Hall, Upper Saddle River, NJ, 2 edition.

Palm, G. (1979). On representation and approximation of nonlinear sytems. Biological Cybernetics, 34, 49–52.

Pintelon, R. and Schoukens, J. (2001). System Identification: A Frequency Domain Approach. IEEE Press, Piscataway, NJ.

Rugh, W. (1981). Nonlinear system theory, the Volterra/Wiener approach. Johns Hopkins Series in Information Sciences and Systems. The Johns Hopkins University Press, Baltimore.

Schoukens, J., Vaes, M., and Pintelon, R. (2016). Linear system identification in a nonlinear setting. IEEE Control Systems Magazine, 38–69.

Sørensen, M. and Comon, P. (2013). Tensor decompositions with banded matrix factors. Linear Algebra and its Applications, 438(2), 919–941.

Vervliet, N., Debals, O., Sorber, L., Van Barel, M., and De Lathauwer, L. (2016). Tensorlab 3.0. URL http://www.tensorlab.net. Available online.

Westwick, D. and Kearney, R. (2003). Identification of Nonlinear Physiological Systems. IEEE Press Series in Biomedical Engineering. IEEE Press / Wiley, Piscataway, NJ.