KU LEUVEN

STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics

Citation/Reference	Authors: Philippe Dreesen, Mariya Ishteva Title: Parameter Estimation of Parallel Wiener-Hammerstein Systems by Decoupling their Volterra Representations Conference: Proc. of the 19th IFAC Symposium on System Identification (SYSID2021) Pages: 846–851
Dates	**Conference paper** • Date of conference: July 13–16, 2021 • Date of presentation: July 15, 2021
Archived version	☐ Submitted version ☐ Accepted (Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher) ■ Published ☐ Published (Open Access)
Acknowledgements	In this manuscript, the following funding is acknowledged: ■ EU H22020 ■ EU Horizon Europe ■ FWO ■ KULeuven Internal Funds ☐ VLAIO ■ AI ■ Other
Published version	NA
Author contact	philippe.dreesen@gmail.com



Preprints, 19th IFAC Symposium on System Identification
July 13-16, 2021. (Virtual) Padova, Italy

Parameter Estimation of Parallel Wiener-Hammerstein Systems by Decoupling their Volterra Representations ⋆

Philippe Dreesen * **Mariya Ishteva** **

* *KU Leuven, Dept. Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Belgium (e-mail: philippe.dreesen@gmail.com)*
** *KU Leuven, Dept. Computer Science, ADVISE-NUMA, Belgium (e-mail: mariya.ishteva@kuleuven.be)*

Abstract: Nonlinear dynamic systems are often approximated by a Volterra series, which is a generalization of the Taylor series for systems with memory. However, the Volterra series lacks physical interpretation. To take advantage of the Volterra representation while aiming for an interpretable block-oriented model, we establish a link between the Volterra representation and the parallel Wiener-Hammerstein model, based on decoupling of multivariate polynomials. The true link is through a constrained decoupling model with (block-)Toeplitz structure on the factors and sets of identical internal branches. The solution of the modified decoupling problem then reveals directly the parameters of the parallel Wiener-Hammerstein model of the system. However, due to the uniqueness properies of the plain decoupling algorithm, even if the structure is not imposed, the method still leads to the true solution (in the exact case).

Keywords: Block structured system identification, parallel Wiener-Hammerstein systems, Volterra kernels, tensors, canonical polyadic decomposition, decoupling multivariate polynomials

1. INTRODUCTION

System identification is concerned with building mathematical dynamical models from noisy input/output measurements of a system. Although linear system identification (Ljung (1999); Pintelon and Schoukens (2012)) has become a mature research field in the last decades with a myriad of applications in a wide variety of fields, recent years have witnessed a shift to methods that take into account the inherent nonlinear effects in real-life systems.

Modeling nonlinear dynamic systems is a difficult task. Modeling nonlinear static systems is covered extensively in different fields like statistical learning theory and machine learning (Rojas (1996)), but the methods used there are not specifically developed to exploit the dynamics of a system and have limited means for dealing with noise. In the system identification community, nonlinear extensions of well-known methods have been studied (Giannakis and Serpedin (2007)), such as nonlinear ARX and nonlinear ARMAX models, with successful applications.

A good nonlinear dynamical model should strike a proper balance between descriptive power and simplicity to represent a large variety of systems while providing understandable models. With this goal in mind, it is useful to consider block-oriented systems with localized static nonlinearities (Giri and Bai (2010); Haber and Unbehauen (1990); Schoukens and Tiels (2017)). Such systems are composed of interconnected linear dynamic and nonlinear static blocks. They match nicely with the fact that in many real-life systems, nonlinearities are localized at a few places in the system and moreover, some block-oriented structures, such as the parallel Wiener-Hammerstein structures, are universal approximators: they have the ability to represent *any* finite-memory nonlinear dynamical system. However, there are currently no universal identification methods that can identify any given block structure.

From the possible block-oriented models, in this paper we choose the parallel Wiener-Hammerstein (pWH) structure. The parallel Wiener system with a finite number of parallel branches is already a *universal approximator* for any finite-memory nonlinear dynamical system (Korenberg (1991)). However, it has the practical limitation that the number of parallel branches may become large (Westwick and Schoukens (2015)). By considering the parallel Wiener-Hammerstein structure, the universal approximation property is conserved (Palm (1979)), and the presence of linear filters after the static nonlinearity allow for a more compact system representation, i.e., with less parallel branches.

⋆ This research was supported by KU Leuven Research Fund; FWO (EOS Project 30468160 (SeLMA), SBO project S005319N, Infrastructure project I013218N, TBM Project T001919N, G028015N, G090117N, SB/1SA1319N, SB/1S93918, SB/151622); Flemish Government (AI Research Program); European Research Council under the European Union's Horizon 2020 research and innovation programme (ERC AdG grant 885682), KU Leuven start-up-grant STG/19/036 ZKD7924. PD is affiliated to Leuven.AI - KU Leuven institute for AI, Leuven, Belgium. Part of this work was performed while the authors were with Dept. ELEC of Vrije Universiteit Brussel, and PD was with CoSys-lab at Universiteit Antwerpen, Belgium.

© 2021 the authors. Accepted by IFAC for publication under a Creative Commons License CC-BY-NC-ND

As an intermediate step towards finding the parameters of the parallel Wiener-Hammerstein structure, we consider the Volterra series expansion, as it is an approximation for any time-invariant nonlinear dynamic system with fading memory (Boyd and Chua (1985)). It is a non-parametric system description that generalizes the concept of the finite impulse response to the nonlinear case, in much the same way as a Taylor series expansion for function approximation, and that also has the universal approximation property. A major advantage is that estimating the Volterra kernel coefficients is a linear problem. However, due to the rapid increase of the number of coefficients when increasing the memory length (number of lags) and the order of the system, estimating Volterra kernels was until recently limited to weakly nonlinear systems with short memories. Recent advances have shown that Volterra kernels can be estimated in a reliable way, by using a regularization-based framework (Birpoutsoukis et al. (2017); Wachel and Śliwiński (2015); Śliwiński et al. (2017)) or by phrasing the problem into the framework of kernel methods (Franz and Schölkopf (2006)). Thanks to these developments the Volterra approach is becoming relevant again.

In simple terms, a Volterra series can be perceived as a polynomial function of time-shifted input signals and thus can be treated accordingly. In particular, multivariate polynomials can be decoupled (Dreesen et al. (2015)) in a structure resembling the structure of the parallel Wiener-Hammerstein model. The decoupling model has useful uniqueness properties and has been studied in detail in the last years. Computationally, the decomposed polynomial is obtained via the canonical polyadic decomposition of tensors, which in turn has been extensively studied in the tensor community (Kolda and Bader (2009); Sidiropoulos et al. (2017)). In fact, another connection between Volterra kernels and symmetric tensors has been initiated in (Kibangou and Favier (2007); Favier et al. (2012); Kibangou and Favier (2009)) for Wiener, Hammerstein, Wiener-Hammerstein, and parallel Wiener systems. In Usevich et al. (2021), the identification of parallel Wiener-Hammerstein systems from Volterra kernels is viewed as a tensor recovery problem.

To summarize, the main goal of this paper is to develop robust and efficient methods for parameter estimation of parallel Wiener-Hammerstein models from a given non-parametric Volterra kernel description of the system. We assume that the Volterra kernels (up to certain order) have been estimated (e.g. by the method of Birpoutsoukis et al. (2017)) and we decouple the Volterra representation with the recent decoupling methodology.

The rationale for seeking a block-oriented model, rather than using the available Volterra representation, is two-fold. First, a block-oriented representation is typically more compact in terms of the number of required parameters. Second, a block-oriented representation may capture the underlying physical nature of the system, or at least provides a more accessible and interpretable model than the black-box Volterra model. A direct approach to find the block-oriented representation from data would require nonlinear optimization techniques, but here we start from a Volterra representation that can be found from linear relations; the subsequent tensor decomposition is typically computed by an alternating least squares approach.

The remainder of the paper is organized as follows. Section 2 introduces the models that we use in the paper (parallel Wiener-Hammerstein, Volterra and decoupling models). Section 3 links all these models together via a (structured) decoupling model. Section 4 discusses the advantages and disadvantages of imposing the available structure on the basic decoupling model. Section 5 illustrates the proposed parameter estimation procedure. Finally, Section 6 summarizes our findings and discusses future work.

2. PARALLEL WIENER-HAMMERSTEIN, VOLTERRA AND DECOUPLING MODELS

After presenting some background material, this section briefly introduces the three models that we consider in this paper, namely the parallel Wiener-Hammerstein, the Volterra model, and the decoupling models.

2.1 FIR filtering as a matrix operation

In this paper we consider parallel Wiener-Hammerstein representations in which the linear blocks are modeled using finite impulse response (FIR) filters. We denote the input signal at time k by $u(k)$ and the corresponding output signal by $y(k)$.

For an FIR filter of length L (also called order [1]), the output sequence is given as the well-known convolution operation

$$y(k) = h_0 u(k) + h_1 u(k-1) + \cdots + h_L u(k-L), \quad (1)$$

where the filter coefficients h_k are the filter's impulse responses at the k-th instant for $0 \leq k \leq L$ (causality imposes that $h_k = 0$ for $k < 0$). The convolution operation (1) may be viewed as a matrix operation as in

$$\begin{bmatrix} y(k) \\ \vdots \\ y(k-p) \end{bmatrix} = \underbrace{\begin{bmatrix} h_0 & \cdots & h_L & & \\ & \ddots & \ddots & \ddots & \\ & & h_0 & \cdots & h_L \end{bmatrix}}_{(p+1)\times(L+p+1)} \begin{bmatrix} u(k) \\ \vdots \\ u(k-L-p) \end{bmatrix},$$

where the empty spaces in the Toeplitz matrix (containing the filter coefficients h_k) denote zero elements.

2.2 Volterra models

The Volterra system description expresses the output $y(k)$ of a nonlinear dynamic system with memory length L as a polynomial function of the delayed inputs $u(k), u(k-1), \ldots, u(k-L)$ and coefficients given by the Volterra kernels $\mathbf{H}_i$ of order $i = 0, \ldots, d$. The output can hence be written as the finite sum

$$y(k) = \sum_{i=0}^{d} \sum_{\tau_1,\tau_2,\ldots,\tau_i=0}^{L} \mathbf{H}_i(\tau_1, \tau_2, ..., \tau_i). \\ u(k-\tau_1)u(k-\tau_2)...u(k-\tau_i). \quad (2)$$

This can be viewed as a nonlinear extension of the linear convolution operation (1).

[1] We will use the term 'length' because the term 'order' has a different meaning in the context of Volterra models.

© 2021 the authors. Accepted by IFAC for publication
under a Creative Commons License CC-BY-NC-ND

2.3 Parallel Wiener-Hammerstein models

Single-branch Wiener-Hammerstein (WH) systems (Fig. 1) are block-oriented system structures that are composed of a static nonlinearity $g(\cdot)$ that is sandwiched in between two linear dynamic systems. In our setting, the dynamical systems are FIR filters $A(q)$ and $C(q)$, where q denotes the shift operator $qx(k) = x(k+1)$.

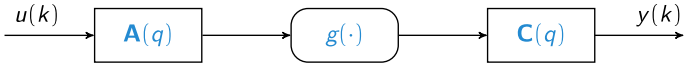

Fig. 1. Single-branch Wiener-Hammerstein model: a static nonlinearity $g(\cdot)$ sandwiched in between two FIR filters $A(q)$ and $C(q)$.

Let the filters have lengths L_1 and L_2, respectively and let their coefficients be a_k, $k = 0, \ldots, L_1$ and c_i, $i = 0, \ldots, L_2$, respectively. The WH system can then be written in a matrix representation as

$$y(k) = \begin{bmatrix} c_0 \; c_1 \; \cdots \; c_{L_2} \end{bmatrix} \begin{bmatrix} w(k) \\ w(k-1) \\ \vdots \\ w(k-L_2) \end{bmatrix},$$

$$w(k) = g(v(k)),$$

$$\begin{bmatrix} v(k) \\ v(k-1) \\ \vdots \\ v(k-L_2) \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & \cdots & a_{L_1} & 0 & \cdots & 0 \\ 0 & a_0 & a_1 & \cdots & a_{L_1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & & & & 0 \\ 0 & \cdots & 0 & a_0 & a_1 & \cdots & a_{L_1} \end{bmatrix} \begin{bmatrix} u(k) \\ u(k-1) \\ \vdots \\ u(k-L_1) \\ u(k-L_1-1) \\ \vdots \\ u(k-L_1-L_2) \end{bmatrix}. \tag{3}$$

In the parallel Wiener-Hammerstein model (Fig. 2), the output $y(k)$ is a sum of the outputs of all the branches.

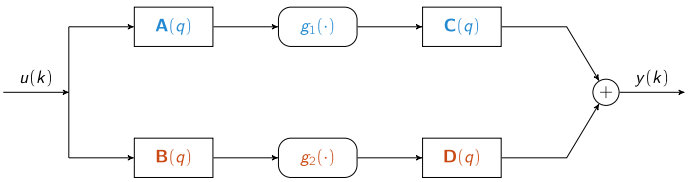

Fig. 2. A parallel Wiener-Hammerstein model with two branches. The output of the system is the sum of the outputs of the branches.

2.4 Decoupling model

The decoupling problem (Fig. 3) aims at unraveling a multivariate vector function $\boldsymbol{f}$ into a structure $\boldsymbol{f}(\boldsymbol{u}) = \mathbf{W}\boldsymbol{g}(\mathbf{V}^\top \boldsymbol{u})$, where the inputs $\boldsymbol{u} = (u_1, \ldots, u_m)$ are mixed by a transformation $\mathbf{V}^\top$, then fed into r univariate polynomials $g_i(\cdot)$, and finally mixed by a transformation $\mathbf{W}$.

Our solution method (Dreesen et al. (2015)) proceeds by collecting the first-order derivatives of $\boldsymbol{f}$ in a set of sampling points. At a given sampling point $\boldsymbol{u}^\star$, this information is represented by the Jacobian,

$$\begin{aligned} \mathbf{J}(\boldsymbol{u}^\star) &= [\partial f_i / \partial u_j]|_{\boldsymbol{u}^\star} \\ &= \mathbf{W} \operatorname{diag}\left(g_1'(\boldsymbol{v}_1^\top \boldsymbol{u}^\star), \ldots, g_r'(\boldsymbol{v}_r^\top \boldsymbol{u}^\star)\right) \mathbf{V}^\top, \end{aligned}$$

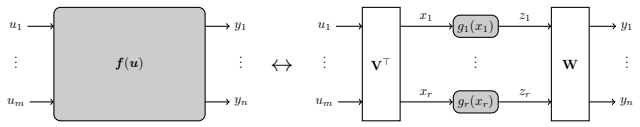

Fig. 3. The decoupling problem: given a multivariate vector function $\boldsymbol{f}(\boldsymbol{u})$, find the matrices $\mathbf{V}$ and $\mathbf{W}$ and (the coefficients of) the univariate polynomials $g_i(\cdot)$, so that $\boldsymbol{f}(\boldsymbol{u}) = \mathbf{W}\boldsymbol{g}(\mathbf{V}^\top \boldsymbol{u})$.

where $\boldsymbol{v}_i$ are the columns of $\mathbf{V}$. Thus the set of Jacobians admits a simultaneous diagonalization with common factors $\mathbf{W}$ and $\mathbf{V}^\top$ (Fig. 4). The factors can be computed from the canonical polyadic decomposition (CPD, Carroll and Chang (1970); Harshman (1970)) of a third-order tensor that is constructed by arranging the Jacobian matrices behind each other.

Note that this procedure operates on the polynomial (Volterra) model of the nonlinear system. Hence, the method is not affected by possible discontinuities or non-differentiability of the nonlinearity that governs the data-generating system itself.

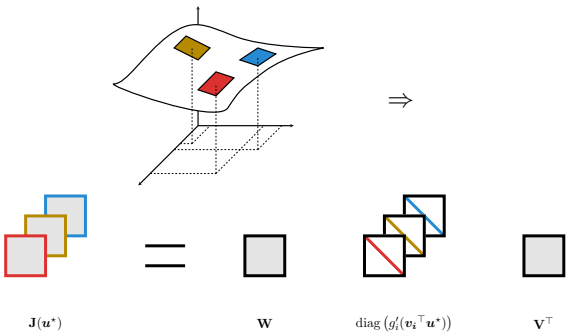

Fig. 4. The matrices $\mathbf{W}$ and $\mathbf{V}$ are two of the factors of the canonical polyadic decomposition of a tensor, which is obtained by stacking the Jacobians of $\boldsymbol{f}$ at different sampling points behind each other.

3. DECOUPLING THE VOLTERRA REPRESENTATION

3.1 The connection between parallel Wiener-Hammerstein, Volterra and the decoupling models

This section explains how decoupling the Volterra system description (2), taking into account the desired parallel Wiener-Hammerstein structure (Fig. 2), leads to a structured decoupling problem (Fig. 5).

The structure in Fig. 5 is obtained by combining the following two observations. On one hand, (the parallel version of) the Wiener-Hammerstein model (3) expresses the output $y(k)$ as a function of delayed inputs $u(k), \ldots, u(k - L_1 - L_2)$, which agrees exactly with the Volterra formulation (2), as $L = L_1 + L_2$. On the other hand, the sandwiched system structure of the Wiener-Hammerstein system (Fig. 2) agrees with the structure assumed by the decoupling model (Fig. 3). The differences are that in the latter we have two combined linear parts and that in the decoupling model we need multiple outputs, in order to use the original implementation. The first difference is only

© 2021 the authors. Accepted by IFAC for publication
under a Creative Commons License CC-BY-NC-ND

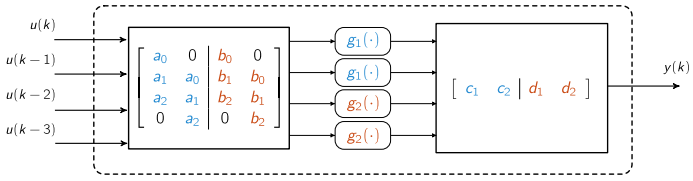

Fig. 5. Decoupling a Volterra model into a parallel Wiener-Hammerstein structure. The Volterra model is of the form $y(k) = f(u(k), u(k-1), \ldots, u(k-L))$ with $f(\cdot)$ a polynomial function. By decoupling $f(\boldsymbol{u})$ as $f(\boldsymbol{u}) = \boldsymbol{w}^\top \boldsymbol{g}(\mathbf{V}^\top \boldsymbol{u})$ while imposing a Toeplitz-like structure on the factor $\mathbf{V}$ and the fact that the functions $g_i(\cdot)$ are repeated internally, one can retrieve the parallel Wiener-Hammerstein block structure (Fig. 2).

artificial, because the matrices in Fig. 3 can be split in parts as in Fig. 5. We discuss the second difference in the next subsection.

3.2 Designing a second output

To overcome the fact that the basic decoupling algorithm (Dreesen et al. (2015)) works for multiple-output functions only, we have two options. We could use more involved optimization techniques (based on second-order derivatives, (Dreesen et al., 2018)) or we could consider pairs of output entries, as in Fig. 6.

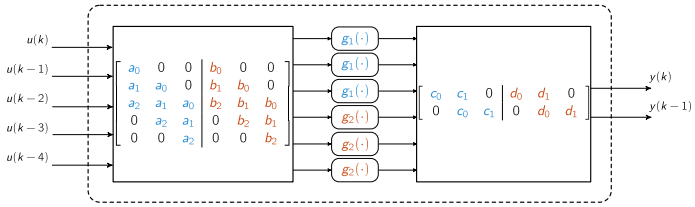

Fig. 6. Decoupling a Volterra model into a parallel Wiener-Hammerstein structure. A second output can be added to the structure of Fig. 5 with only a few adjustments: an additional input entry and some duplications in the matrices and number of branches, so that the represented system remains unchanged.

In this paper we choose the latter option and embed the single-output Volterra model into a two-output function (Fig. 6). We leave the former option for future work. Technically, for the example system from Fig. 5 we need to include one extra input, one extra row in and two extra columns in the first matrix, and one extra row in the second matrix (and some extra zeros), and duplicate some of the branches.

4. COMPUTATION OF THE STRUCTURED DECOUPLED REPRESENTATION

Compared to the basic problem from Dreesen et al. (2015), the decoupling problem in Fig. 6 has some additional structure, which should be taken into account. Namely, the matrices are block-Toeplitz matrices (with nonzero elements equal to the filter coefficients) and moreover, the internal functions are repeated.

We have again two options to deal with these constraints. On one hand, we could impose them in the optimization step and modify the basic decoupling algorithm. This requires changes in the implementation and might also complicate the numerical computation. On the other hand, we could explore the uniqueness properties of the CPD to see if we can obtain any of the required parameters 'for free'. We choose the latter and leave the former as future work.

Fortunately, imposing the structure is not always necessary due to the uniqueness properties of the CPD. Although the basic decoupling algorithm does not provide a solution with a structure exactly as in Fig. 6, we can easily extract the desired parameters from the obtained solution (provided the algorithm has converged to a zero-error solution). What happens is the following: in the second matrix, the first and the fourth column are undistinguishable for the CPD as they are scaled versions of each other (and so are the third and the sixth one). This leads to a mismatch in the corresponding columns of the first matrix as well. However, the second and the fourth columns are correctly recovered in both matrices (again up to scaling) and thus the coefficients for all the filters can be recovered.

Further study on the applicability of this approach is necessary (maximum number of branches, lengths of the filter and other parameters) but the above observation is confirmed by the numerical experiments (Section 5).

5. NUMERICAL EXPERIMENTS

5.1 Data generation and Volterra model estimation

We validate our findings by means of numerical simulations. We consider a two-branch parallel Wiener-Hammerstein system as in Fig. 2, where the filters $A(q)$ and $B(q)$ have length $L_1 = 3$, and the filters $C(q)$ and $D(q)$ have length $L_2 = 1$. The functions $g_1(\cdot)$ and $g_2(\cdot)$ are polynomials of degree $d = 3$. Input and output data is generated as follows: A zero-mean white Gaussian noise input sequence u of 1000 samples is used to generate an undisturbed output signal y_0. The noisy output signal y is obtained by adding to y_0 Gaussian measurement noise with a prescribed SNR.

From the input and (noisy) output data, a Volterra model is constructed by solving for the kernel coefficients. Since the Volterra model is linear-in-the-parameters, this step involves constructing a regression matrix $\mathbf{K}$ and solving a linear least squares problem of the form $\boldsymbol{y} = \mathbf{K}\boldsymbol{\theta}$, where $\boldsymbol{y}$ is a vector containing the output samples $y(k)$, and where the rows of $\mathbf{K}$ contain products of the the shifted inputs $u(k), u(k-1), \ldots, u(k-L_1-L_2)$, raised to all possible exponents of maximal total degree $d = 3$ (cf. all possible monomials of $L+1$ variables with total degree d). The least-squares solution of the parameter vector $\boldsymbol{\theta}$ contains the estimated coefficients of the Volterra model.

5.2 Decoupling the Volterra model

The thus obtained Volterra model is the starting point for our method. The Volterra model can be viewed as a polynomial function that maps the (delayed) inputs to the output as $y(k) = f(u(k), u(k-1), \ldots, u(k-L_1-L_2))$. In order to use the decoupling method of Dreesen et al. (2015), we now embed this single-output function $f(\cdot)$

© 2021 the authors. Accepted by IFAC for publication
under a Creative Commons License CC-BY-NC-ND

into a two-output function $\boldsymbol{f}(\boldsymbol{u}) = [\, f_1(\boldsymbol{u})\ f_2(\boldsymbol{u})\,]^\top$ by considering $f_1(u(k), \ldots, u(k-L_1-L_2-1) = f(u(k), u(k-1), \ldots, u(k-L_1-L_2))$ and $f_2(u(k), \ldots, u(k-L_1-L_2-1)) = f(u(k-1), \ldots, u(k-L_1-L_2-1))$. In other words, we consider two consecutive outputs $y(k)$ and $y(k-1)$ (both of which satisfy the Volterra model), and hence also require to introduce an extra (shifted) input $u(k-L_1-L_2-1)$. In accordance with the structure of Fig. 6, the function $\boldsymbol{f}(\boldsymbol{u})$ that is obtained in this way has $L_1+L_2+1 = 5$ inputs, two outputs and $2 \cdot 3 = 6$ internal branches, containing two univariate polynomial nonlinearities $g_1(\cdot)$ and $g_2(\cdot)$.

The pWH system hence admits an (exact) decoupled representation involving the matrices $\mathbf{V}$ and $\mathbf{W}$ as

$$\mathbf{V} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0.5 & 1 & 0 & 0.8 & 1 & 0 \\ 0.4 & 0.5 & 1 & -0.5 & 0.8 & 1 \\ -0.2 & 0.4 & 0.5 & 0.4 & -0.5 & 0.8 \\ 0 & -0.2 & 0.4 & 0 & 0.4 & -0.5 \\ 0 & 0 & -0.2 & 0 & 0 & 0.4 \end{array}\right]$$

and

$$\mathbf{W} = \left[\begin{array}{ccc|ccc} 1 & -0.7 & 0 & 1 & 0.2 & 0 \\ 0 & 1 & -0.7 & 0 & 1 & 0.2 \end{array}\right].$$

The filter coefficients a_k and b_k can be read off from $\mathbf{V}$ and the coefficients c_k and d_k can be read off from $\mathbf{W}$.

In this representation, the block-Toeplitz structured factor $\mathbf{W}$ exhibits the property that some of the columns are (scalar) multiples of one another. As explained in Section 4, while the associated CPD is not fully unique, it turns out that we may enjoy a 'partial uniqueness' property (i.e., some of the columns of the matrices $\mathbf{V}$ and $\mathbf{W}$ are defined essentially uniquely (up to scaling and permutation), and from these columns, we are able to extract the complete WH parameterization.

For the simulation example in particular, we considered a Volterra model obtained from 1000 pairs of input and (noisy) output datapoints with an SNR of 40 dB. After performing the embedding of the Volterra model into a two-output function, we obtain from the CPD the estimated factor $\hat{\mathbf{W}}$ as

$$\hat{\mathbf{W}} = \left[\begin{array}{ccc|ccc} 0.190 & 0.000 & 0.145 & 0.094 & -0.329 & 0.002 \\ -0.274 & 0.127 & 0.732 & -0.000 & 0.000 & 0.356 \end{array}\right],$$

from which we observe that the columns 1 and 3 are the 'dense' columns. Note that the CPD is never able to recover the factors exactly, explaining 'essential' uniqueness: there are remaining scaling and permutation degrees-of-freedom. For this particular solution, columns 1 and 3 should correspond to columns 2 and 5 in $\mathbf{W}$ (possibly in opposite order). A closer look reveals that columns 1 and 3 exhibit partial uniqueness: when normalizing the columns 1 and 3 of $\hat{\mathbf{W}}$, such that the element at the second row equals one, we get

$$\hat{\mathbf{W}}(:, [1,3]) = \left[\begin{array}{cc} -0.693 & 0.198 \\ 1.000 & 1.000 \end{array}\right],$$

in which we recognize (approximately) the filter coefficients c_k and d_k ($\hat{c}_1 = -0.693$ vs $c_1 = -0.7$ and $\hat{d}_1 = 0.198$ vs $d_1 = 0.2$). When normalizing columns 1 and 3 of $\hat{\mathbf{V}}$ such that the element at the second row equals one, we get

$$\hat{\mathbf{V}}(:, [1,3]) = \left[\begin{array}{cc} -0.001 & -0.000 \\ 1.000 & 1.000 \\ 0.503 & 0.795 \\ 0.401 & -0.500 \\ -0.199 & 0.398 \\ 0.000 & -0.000 \end{array}\right],$$

in which we recognize (approximately) the filter coefficients a_k and b_k. Also from the reconstruction of the (derivatives of the) internal functions $g_1(\cdot)$ and $g_2(\cdot)$, we observe that the same two (out of six) columns, namely columns 1 and 3, accurately contain the derivatives of a second and third degree polynomial, as expected (Fig. 7).

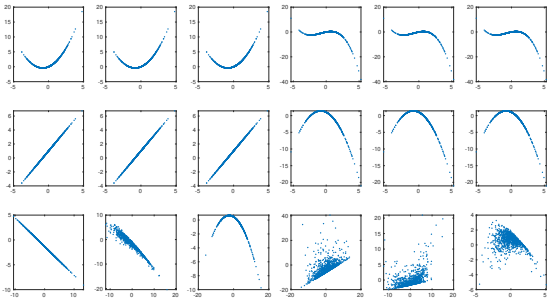

Fig. 7. Although a unique CPD does not exist for our setup, the decoupling procedure is able to reconstruct a partially unique result for some of the columns of the factor matrices (columns 1 and 3 in particular). The top row of plots displays the true nonlinear functions g_1 and g_2; the second row shows their derivatives, and the third row shows the reconstructed derivatives, obtained from the third factor of the CPD, in which columns 1 and 3 exhibit the essentially unique results. In this way, from the (partially) uniquely reconstructed columns, the filter coefficients (from $\hat{\mathbf{V}}$ and $\hat{\mathbf{W}}$) and nonlinear functions (from the third factor of the CPD) can be obtained (essentially) uniquely.

5.3 Remarks

It should be noted that in our problem setup, noise reduction takes place when estimating the Volterra kernels from the (noisy) data. Hence, the SNR level should be considered together with the data length N in order to assess how informative the input/output data is. During extensive simulations, similar results were observed for a wide range of SNR levels, ranging down to 20 dB and time series lengths N, varying between 250 and 1000.

Note that CPD (partial) uniqueness results that are mentioned, are valid only when considering an exact CPD (zero-error). However, our simulations suggest that, when considering low noise, the (partial) uniqueness properties also hold for the noisy problem.

Finally, note that, along the lines of Usevich et al. (2020), the current procedure can be viewed as a variation on the structured CPD approach of Dreesen et al. (2017).

© 2021 the authors. Accepted by IFAC for publication
under a Creative Commons License CC-BY-NC-ND

6. CONCLUSIONS AND FUTURE WORK

We established a connection between parallel Wiener-Hammerstein systems, Volterra models and the decoupling problem. It turns out that under certain conditions, a parallel Wiener-Hammerstein system can be expressed in a decoupled representation, involving block-Toeplitz transformation matrices and repeated polynomial nonlinearities. We explored how this decoupled representation can be obtained from the (approximate) Volterra series representation of the system and observed that the CPD-based decoupling method provides a partially unique decomposition, from which approximations of the pWH system parameters can be read off directly.

Future work should focus on how to handle noisy data. Extra noise reduction can be performed by actually imposing structure on the factors as indicated in Fig. 6. For the computation of a structured decoupling model, a modification of the original CPD algorithm is needed, imposing constraints on the factors $\mathbf{V}$ and $\mathbf{w}$ and the functions $g_i(\cdot)$. Some of these features can readily be implemented with Tensorlab (Vervliet et al. (2016)) and by the methods of Decuyper et al. (2019).

REFERENCES

Birpoutsoukis, G., Marconato, A., Lataire, J., and Schoukens, J. (2017). Regularized nonparametric Volterra kernel estimation. *Automatica*, 82, 324–327.

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

Carroll, J. and Chang, J. (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3), 283–319.

Decuyper, J., Dreesen, P., Schoukens, J., Runacres, M.C., and Tiels, K. (2019). Decoupling multivariate polynomials for nonlinear state-space models. *IEEE Control Systems Letters (L-CSS)*, 3(3), 745–750.

Dreesen, P., De Geeter, J., and Ishteva, M. (2018). Decoupling multivariate functions using second-order information and tensors. In Y. Deville, S. Gannot, R. Mason, M.D. Plumbley, and D. Ward (eds.), *Proc. 14th International Conference on Latent Variable Analysis and Signal Separation*, volume 10891 of *Lecture Notes on Computer Science (LNCS)*, 79–88. Guildford, UK.

Dreesen, P., Ishteva, M., and Schoukens, J. (2015). Decoupling multivariate polynomials using first-order information and tensor decompositions. *SIAM J. Matrix Anal. Appl.*, 36(2), 864–879.

Dreesen, P., Westwick, D., Ishteva, M., and Schoukens, J. (2017). Modeling parallel Wiener-Hammerstein systems using tensor decomposition of Volterra kernels. In P. Tichavsky, M. Babaie-Zadeh, O. Michel, and N. Thirion-Moreau (eds.), *Proc. 13th International Conference on Latent Variable Analysis and Signal Separation*, volume 10169 of *Lecture Notes on Computer Science (LNCS)*, 16–25. Grenoble, France.

Favier, G., Kibangou, A., and Bouilloc, T. (2012). Nonlinear system modeling and identification using Volterra-PARAFAC models. *Int. J. Adapt. Control Signal Process.*, 26(1), 30–53.

Franz, M. and Schölkopf, B. (2006). A unifying view of Wiener and Volterra theory and polynomial kernel regression. *Neural computation*, 18(12), 3097–3118.

Giannakis, G. and Serpedin, E. (2007). A bibliography on nonlinear system identification. *IEEE Transactions on Signal Processing*, 81(3), 533–580.

Giri, F. and Bai, E. (2010). *Block-oriented Nonlinear System Identification*. LNCIS. Springer.

Haber, R. and Unbehauen, H. (1990). Structure identification of nonlinear dynamic systems—a survey on input/output approaches. *Automatica*, 26(4), 651–677.

Harshman, R. (1970). Foundations of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16(1), 1–84.

Kibangou, A. and Favier, G. (2007). Toeplitz–Vandermonde matrix factorization with application to parameter estimation of Wiener–Hammerstein systems. *IEEE Signal Process. Lett.*, 14, 141–144.

Kibangou, A. and Favier, G. (2009). Identification of parallel-cascade Wiener systems using joint diagonalization of third-order Volterra kernel slices. *IEEE Signal Process. Lett.*, 16(3), 188–191.

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

Korenberg, M. (1991). Parallel cascade identification and kernel estimation for nonlinear systems. *Annals of Biomedical Engineering*, 19, 429–455.

Ljung, L. (1999). *System identification: Theory for the user*. Prentice Hall PTR.

Palm, G. (1979). On representation and approximation of nonlinear systems. *Biol. Cybern.*, 34(1), 49–52.

Pintelon, R. and Schoukens, J. (2012). *System Identification: A Frequency Domain Approach*. Wiley-IEEE Press, 2nd edition.

Rojas, R. (1996). *Neural networks*. Springer-Verlag, Berlin.

Schoukens, M. and Tiels, K. (2017). Identification of block-oriented nonlinear systems starting from linear approximations: a survey. *Automatica*, 85, 272–292.

Sidiropoulos, N.D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E.E., and Faloutsos, C. (2017). Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65, 3551–3582.

Śliwiński, P., Marconato, A., Wachel, P., and Birpoutsoukis, G. (2017). Non-linear system modelling based on constrained Volterra series estimates. *IET Control Theory & Applications*, 11(15), 2623–2629.

Usevich, K., Dreesen, P., and Ishteva, M. (2020). Decoupling multivariate polynomials: interconnections between tensorizations. *J. Comp. Appl. Math.*, 363, 22–34.

Usevich, K., Dreesen, P., and Ishteva, M. (2021). Low-rank tensor recovery for Jacobian-based Volterra identification of parallel Wiener-Hammerstein systems. In *19th IFAC Symp. on System Identification (SYSID 2021)*.

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

Wachel, P. and Śliwiński, P. (2015). Aggregative modeling of nonlinear systems. *IEEE Signal Processing Letters*, 22(9), 1482–1486.

Westwick, D. and Schoukens, J. (2015). On the efficiency of parallel Wiener-Hammerstein models. Technical report, Vrije Universiteit Brussel.

© 2021 the authors. Accepted by IFAC for publication under a Creative Commons License CC-BY-NC-ND