From coupled to decoupled polynomial representations in parallel Wiener-Hammerstein models

Koen Tiels and Johan Schoukens

Abstract—A large variety of nonlinear systems can be approximated by parallel Wiener-Hammerstein models. These models consist of a multiple input multiple output (MIMO) nonlinear static block sandwiched between two linear dynamic blocks. One method is available for the identification of a general parallel Wiener-Hammerstein model. It represents the nonlinear block as a multivariate polynomial, which typically contains cross-terms. These make it harder to interpret and to invert the model. We want to eliminate the cross-terms, and thus come to a decoupled polynomial representation. In this paper, the simultaneous decoupling of quadratic and cubic polynomials is formulated as a standard tensor decomposition. A simulation example shows that the simultaneous decoupling can result in a model with less parallel branches than a decoupling of all polynomials separately.

I. INTRODUCTION

Although nonlinear distortions are often present, many dynamical systems can be approximated by a linear model. When the nonlinear distortion level is too high, a linear approximation is insufficient, and a nonlinear model is needed. The nonlinear model structure should be flexible enough to be able to describe a large variety of systems. Furthermore, it should be parameter-parsimonious to avoid a large noise sensitivity.

Volterra series are generally applicable for a large variety of systems [1], but are often parameter-expensive. An alternative is provided by block-oriented models [2], which are built up by linear dynamic and nonlinear static (memoryless) blocks. The simplest block-oriented models are the Wiener model (linear dynamic block followed by a nonlinear static block), and the Hammerstein model (linear dynamic block preceded by a nonlinear static block). They can be generalized to a Wiener-Hammerstein model (nonlinear static block sandwiched between two linear dynamic blocks).

A generalization of the Wiener-Hammerstein model is the parallel Wiener-Hammerstein model (Wiener-Hammerstein model with a multiple input multiple output (MIMO) nonlinear static block). A discrete-time parallel Wiener-Hammerstein model can uniformly approximate any discrete-time system with finite memory and a continuous input-output mapping [3]. Some methods are available that immediately identify a decoupled Wiener-Hammerstein model [4], [5], [2]. However, they consider a somewhat restricted class of parallel Wiener-Hammerstein models, called the class $S_M$. This model has $M$ parallel branches, each containing a simple Wiener-Hammerstein model with an integer power nonlinearity, i.e. branch $m$ contains a nonlinearity $(\cdot)^m$. The restriction is that there is only one branch for each power.

Up to the authors knowledge, only one identification method for a general parallel Wiener-Hammerstein model exists [6]. It represents the nonlinear static block by a polynomial that typically contains cross-terms (a coupled polynomial representation as in Fig. 1). We want to eliminate the cross-terms, and thus come to a decoupled representation as in Fig. 2. This makes the model easier to interpret and to invert. The questions addressed in this paper are: is it possible to decouple the polynomial representation, and if so, how?

Some methods already exist to decouple Volterra models [7] and parallel Wiener models [8]. A disadvantage of the former is that it requires the measurement of the Volterra kernels of different orders, which can be time-consuming. The approach in [8] overcomes this problem. It achieves the same result starting from a polynomial description of the cross-coupled nonlinearity. The polynomial is first split in a sum of homogeneous polynomials. Next, the cross-terms in each homogeneous polynomial are eliminated using tensor decomposition methods. A drawback is the introduction of new input signals for each degree in the polynomial.

Here, we follow a different approach. We first focus on decoupling parallel Wiener-Hammerstein models with quadratic nonlinearities. The square matrices that describe the quadratic forms are decomposed on a common basis, built on a product of vectors. Hence, we decouple simul-
taneously across the outputs of the polynomial. Next, we generalize to higher-degree nonlinearities, and eventually also decouple simultaneously across the degrees of the polynomial. This allows for a small number of input signals to the decoupled polynomial.

This paper is organized as follows. Section II presents the notation and preliminaries. Section III states the problem formally. Section IV presents the simultaneous decoupling of homogeneous polynomials, and eventually of polynomials with mixed degrees of nonlinearity. Section V illustrates the decoupling method on a simulation example. Section VI formulates the concluding remarks.

II. NOTATION AND PRELIMINARIES

The notation in this paper concerning tensors is kept as consistent as possible with the notation used in [9].

A. Basic notation and nomenclature

Loosely speaking, a tensor is a multidimensional array. The order of a tensor is the number of its dimensions.

 Scalars are denoted by lowercase letters, e.g. a. Vectors (first-order tensors) are denoted by boldface lowercase letters, e.g. A. Matrices (second-order tensors) are denoted by boldface capital letters, e.g. A. Higher-order tensors are denoted by boldface calligraphic letters, e.g. X.

The i\textsuperscript{th} entry of a vector a is denoted by a\textsubscript{i}, element (i, j) of a matrix A is denoted by a\textsubscript{ij}, and element (i, j, k) of a third-order tensor X is denoted by x\textsubscript{ijk}. The n\textsuperscript{th} element in a sequence is denoted by a superscript in parentheses, e.g. A\textsuperscript{n} denotes the n\textsuperscript{th} vector in a sequence.

B. The canonical polyadic decomposition

A tensor X ∈ R\textsuperscript{I\times\ldots\times IN} is rank-one if it can be written as the outer product of N vectors, i.e. X = a\textsuperscript{(1)} × ... × a\textsuperscript{(N)}. This means that each element of the tensor is the product of the corresponding vector elements: x\textsubscript{ijk} = \prod\text{N}_{k=1} a\textsubscript{ik}.

The canonical polyadic decomposition (CPD) [10], [11], [9] approximates a tensor with a sum of rank-one tensors. Let X ∈ R\textsuperscript{I\times\ldots\times IN}, then [9]

\[ X ≈ \|A\textsuperscript{(1)}, ..., A\textsuperscript{(N)}\| = ∑_{r=1}^{R} λ_r a\textsubscript{r}\textsuperscript{(1)} × ... × a\textsubscript{r}\textsuperscript{(N)} \] (1)

is a CPD of X. It is useful to assume that the columns of the matrices A\textsuperscript{(n)} ∈ R\textsuperscript{I\times R} for n = 1, ..., N are normalized to length one, with the weights absorbed into the vector λ ∈ R\textsuperscript{R}, so that [9]

\[ X ≈ \|λ; A\textsuperscript{(1)}, ..., A\textsuperscript{(N)}\| = ∑_{r=1}^{R} λ_r a\textsubscript{r}\textsuperscript{(1)} × ... × a\textsubscript{r}\textsuperscript{(N)} \] (2)

The minimal number of rank-one tensors for which the CPD of a tensor is exact, is called the rank of the tensor. For the relationship between tensor decompositions and the matrix singular value decomposition (SVD), the interested reader is referred to [12].

The CPD of a tensor is often calculated via an alternating least-squares (ALS) approach [9]. In each iteration, all matrices are fixed, except one matrix A\textsuperscript{(n)}. The update of the elements of this matrix is then a linear least-squares problem.

Recently, other algorithms to calculate the CPD of a tensor were proposed in [13], [14] that do better overall than ALS.

C. Symmetric tensors and homogeneous polynomials

The n-mode vector product of a tensor X ∈ R\textsuperscript{I\times\ldots\times IN} with a vector a ∈ R\textsuperscript{I} is denoted by X ×\textsubscript{a}. The result is of size I\textsubscript{1} × ... × I\textsubscript{n-1} × I\textsubscript{n+1} × ... × I\textsubscript{N}. Element-wise, (X ×\textsubscript{a})\textsubscript{i1...in-1in+1...in} = \sum\text{N}_{k=1} x\textsubscript{i1...in} a\textsubscript{in}.

A tensor is symmetric if its elements do not change under any permutation of the indices. A tensor can be partially symmetric in two or more modes, which means that the elements of the tensor do not change under any permutation of the indices corresponding to those modes.

Symmetric tensors are bijectively related to homogeneous polynomials [15], [16]. Let X ∈ R\textsuperscript{N×...×N} be a symmetric tensor of order d, and w ∈ R\textsuperscript{N}, then the polynomial

\[ p(w) = ∑_{i_1...i_d=1} x_{i_1...i_d} w_{i_1} ... w_{i_d} \] (3)

\[ = X \times_1 w \times_2 w \ldots \times_d w \]

can be bijectively associated with X. For quadratic polynomials, the notation p(w) = A ×\textsubscript{1} w ×\textsubscript{2} w reduces to the quadratic form p(w) = w\textsuperscript{T} Aw.

III. PROBLEM STATEMENT

We want to eliminate the nonlinear cross-terms in parallel Wiener-Hammerstein models. To keep the notation simple, the focus in this paper is on quadratic and cubic polynomials, but the results extend to higher degrees.

Consider the parallel Wiener-Hammerstein model in Fig. 1, described by

\[ w_k(t) = W(k)(q^{-1}, \theta_{W(k)}) u(t), \quad k = 1, ..., n_W \]

\[ z_k(t) = p(k)(w(t)) = w(t) \textsuperscript{T}(t) A(k) w(t) \]

\[ + X(k) \times_1 w(t) \times_2 w(t) \ldots \times_3 w(t) \]

\[ y(t) = ∑_{k=1}^{n_H} H(k)(q^{-1}, \theta_{H(k)}) z_k(t) \] (4)

where W(k)(q\textsuperscript{-1}, θ\textsubscript{W(k)}) and H(k)(q\textsuperscript{-1}, θ\textsubscript{H(k)}) are linear time-invariant (LTI) discrete-time transfer functions in the delay (backward shift) operator q\textsuperscript{-1}, parametrized by θ\textsubscript{W(k)} and θ\textsubscript{H(k)} respectively. The polynomials p(k)(w(t)) are sums of homogeneous polynomials of degree 2 and 3, described by the symmetric matrices A(k) ∈ R\textsuperscript{N\times N\times N} and the symmetric tensors X(k) ∈ R\textsuperscript{n_W×N\times n_W} respectively.

In general, the multivariate polynomials p(k)(w(t)) contain cross-terms. We want to eliminate these cross-terms, and thus come to a decoupled representation of the Wiener-Hammerstein model (see Fig. 2), described by

\[ \hat{w}_k(t) = \hat{W}(k)(q^{-1}, \theta_{\hat{W}(k)}) u(t), \quad k = 1, ..., n \]

\[ \hat{z}_k(t) = \hat{p}(k)(\hat{w}_k(t)), \quad k = 1, ..., n \]

\[ \hat{y}(t) = ∑_{k=1}^{n} \hat{H}(k)(q^{-1}, \theta_{\hat{H}(k)}) \hat{z}_k(t) \] (5)
where $\hat{W}^{(k)}(q^{-1}, \theta_{\hat{W}(k)})$ and $\hat{H}^{(k)}(q^{-1}, \theta_{\hat{H}(k)})$ are LTI discrete-time transfer functions, parametrized by $\theta_{\hat{W}(k)}$ and $\theta_{\hat{H}(k)}$ respectively. The signal $\hat{y}(t)$ can be exactly equal to the output $y(t)$, or can be an approximation of $y(t)$ (see Remark 1). As the polynomials $\hat{p}^{(k)}(\hat{w}_k(t))$ are univariate instead of multivariate, the model representation is simplified to a sum of $n$ parallel branches, each containing a simple Wiener-Hammerstein model.

The problem addressed in this paper is how to retrieve a decoupled representation (5) of a parallel Wiener-Hammerstein model, given a cross-coupled representation (4). Some approaches exist to achieve this goal for Volterra and parallel Wiener models [7], [8]. A drawback of the latter is the introduction of new parallel branches for each degree in the polynomial. The main contribution in this paper is the simultaneous decoupling of the polynomials $p^{(k)}$, which allows to keep the total number of branches $n$ small.

IV. DECOUPLING THE POLYNOMIAL REPRESENTATION

This section first presents the simultaneous decoupling of quadratic polynomials. Next, a generalization to cubic polynomials is presented, showing that the method extends to higher-degree homogeneous polynomials. Finally, this approach is generalized to the simultaneous decoupling of polynomials with mixed degrees of nonlinearity.

A. Simultaneous decoupling of quadratic polynomials

1) Basic idea: First, quadratic polynomials are considered, i.e. $z_k(t) = p^{(k)}(w(t)) = w^T(t)A^{(k)}w(t)$. The basic idea is to decompose the matrices $A^{(k)}$ on a common basis, built on an outer product of vectors

$$A^{(k)} \approx \sum_{r=1}^{R_2} \lambda_r^{(k)} b_r b_r^T = [\lambda^{(k)}; B; B] .$$  

(6)

Observe that the basis vectors $b_r$ do not depend on $k$; all the matrices $A^{(k)}$ are decomposed using the same basis.

2) Decoupled Wiener-Hammerstein model: The outputs of the polynomials $p^{(k)}(w(t))$ can be rewritten as

$$z_k(t) = w^T(t)A^{(k)}w(t)$$

$$\approx \sum_{r=1}^{R_2} \lambda_r^{(k)} w^T(t)b_r b_r^Tw(t)$$

$$= \sum_{r=1}^{R_2} \lambda_r^{(k)} (b_r^Tw(t))^2$$

$$= \sum_{r=1}^{R_2} \lambda_r^{(k)} \left( \sum_{s=1}^{n_W} b_{sr}w_s(t) \right)^2$$

$$= \sum_{r=1}^{R_2} \lambda_r^{(k)} \left( \sum_{s=1}^{n_W} b_{sr}w_s(q^{-1}, \theta_{\hat{W}(s)})u(t) \right)^2$$

(7)

Defining the LTI systems $\hat{W}^{(r)}(q^{-1}, \theta_{\hat{W}(r)})$ as

$$\hat{W}^{(r)}(q^{-1}, \theta_{\hat{W}(r)}) := \sum_{s=1}^{n_W} b_{sr}w_s(q^{-1}, \theta_{\hat{W}(s)})$$

(8)

allows to write $z_k(t)$ as

$$z_k(t) \approx \sum_{r=1}^{R_2} \lambda_r^{(k)} \left( \hat{W}^{(r)}(q^{-1}, \theta_{\hat{W}(r)})u(t) \right)^2 .$$  

(9)

The output of the parallel Wiener-Hammerstein model is equal to

$$y(t) = \sum_{k=1}^{n_H} H^{(k)}(q^{-1}, \theta_{H^{(k)}})z_k(t)$$

$$\approx \sum_{k=1}^{n_H} H^{(k)}(q^{-1}, \theta_{H^{(k)}}) \sum_{r=1}^{R_2} \lambda_r^{(k)} \hat{w}_r^2(t) .$$  

(10)

Defining the LTI systems $\hat{H}^{(r)}(q^{-1}, \theta_{\hat{H}(r)})$ as

$$\hat{H}^{(r)}(q^{-1}, \theta_{\hat{H}(r)}) := \sum_{k=1}^{n_H} \lambda_r^{(k)} H^{(k)}(q^{-1}, \theta_{H^{(k)}})$$

(11)

allows to write $y(t)$ as

$$y(t) \approx \sum_{r=1}^{R_2} \hat{H}^{(r)}(q^{-1}, \theta_{\hat{H}(r)})\hat{w}_r^2(t) .$$  

(12)

Defining the univariate polynomials $\hat{p}^{(r)}(\hat{w}_r(t))$ as

$$\hat{p}^{(r)}(\hat{w}_r(t)) := \hat{w}_r^2(t)$$

(13)

allows to write

$$y(t) \approx \sum_{r=1}^{R_2} \hat{H}^{(r)}(q^{-1}, \theta_{\hat{H}(r)})\hat{z}_r(t) = \hat{y}(t) ,$$

(14)

with

$$\hat{z}_r(t) = \hat{p}^{(r)}(\hat{w}_r(t)) .$$

(15)

A decoupled representation of the Wiener-Hammerstein model, as in (5), is thus retrieved. The model has $n = R_2$ branches.

Remark 1: Although an equal sign can be used in (6) (see Appendix), we prefer to use an approximation sign, since the entries of the matrices $A^{(k)}$ can be corrupted with noise. An approximation with a far smaller number of terms then outweighs an exact recovery of the noisy matrices $A^{(k)}$.

3) How to calculate the basis vectors? In the foregoing, it was presented that a coupled (quadratic) polynomial representation in a parallel Wiener-Hammerstein model can be replaced by a decoupled representation if a decomposition as in (6) can be made. In this section, it is presented how such a decomposition can be calculated.

The decomposition in (6) is almost a standard CPD (as in (2)), except for the fact that the basis vectors $b_r$ should be common for all $k = 1, \ldots, n_H$, and that the decomposition should be symmetric, i.e. $[\lambda^{(k)}; B; B]$ instead of $[\lambda^{(k)}; (B(1))^{(k)}, (B(2))^{(k)}]$.  

4939
To come to a standard CPD formulation, we first stack the matrices \( A^{(k)} \) in a partially symmetric third-order tensor \( \mathcal{F} \in \mathbb{R}^{n_W \times n_W \times n_H} \). Element-wise,
\[
f_{i_1 i_2 i_3}^{(k)} = a_{i_1 i_2}^{(k)}. \tag{16}
\]
A CPD of \( \mathcal{F} \) is given by
\[
\mathcal{F} \approx \left[ [B, B, \Xi] \right] \equiv \sum_{r=1}^{R_3} b_r \circ b_r \circ \xi_r. \tag{17}
\]
The elements of the matrix \( \Xi \in \mathbb{R}^{n_H \times R_2} \) are given by
\[
\xi_{kr} = \lambda_{kr}^{(k)}. \tag{18}
\]
By stacking the matrices \( A^{(k)} \) in one tensor \( \mathcal{F} \), the basis vectors \( b_r \) are common for each of the \( n_H \) matrices \( A^{(k)} \), which is not guaranteed if these matrices are decomposed separately. To obtain a symmetric decomposition, the symmetry conditions can be imposed in a final iteration, as is done in INDSCAL (Individual differences in scaling) [10], [9], or can be imposed in each iteration, as is done in Tensorlab [17], resulting in a nonlinear least-squares problem. Finally, the vectors \( \lambda^{(k)} \) are obtained from the matrix \( \Xi \) via (18).

4) How to determine the number of branches?: Up to now, it was assumed that the number of rank-one components \( R_2 \) is given. In practice, this number needs to be determined somehow. The strategy used in this paper is to fit CDPs for an increasing number of rank-one components, and to select the first one that fits well. As pointed out in [9], this might not be a good strategy to determine the rank of a tensor, but all we are after is a simple model that approximates the tensor well, but not necessarily recovers the tensor exactly.

B. Generalization to cubic polynomials

The approach is now generalized to cubic polynomials, i.e. \( z_k(t) = p^{(k)}(w(t)) = \mathcal{X}^{(k)} \times_1 w(t) \times_2 w(t) \times_3 w(t) \). Similar as in (6), the tensors \( \mathcal{X}^{(k)} \) are decomposed on a common basis
\[
\mathcal{X}^{(k)} \approx \sum_{r=1}^{R_3} \gamma_{r}^{(k)} c_r \circ c_r \circ c_r = [\gamma^{(k)}; C, C, C]. \tag{19}
\]
Again, the basis vectors \( c_r \) are independent of \( k \).

The outputs of the polynomials \( p^{(k)}(w(t)) \) can now be rewritten as
\[
z_k(t) = \mathcal{X}^{(k)} \times_1 w(t) \times_2 w(t) \times_3 w(t) = \sum_{i_1, i_2, i_3=1}^{n_W} \mathcal{X}^{(k)}_{i_1 i_2 i_3} w_{i_1}(t) w_{i_2}(t) w_{i_3}(t),
\]
\[
\approx \sum_{i_1, i_2, i_3=1}^{n_W} \sum_{r=1}^{R_3} \gamma_{r}^{(k)} c_{i_1} c_{i_2} c_{i_3} w_{i_1}(t) w_{i_2}(t) w_{i_3}(t),
\]
\[
= \sum_{r=1}^{R_3} \sum_{i_1, i_2, i_3=1}^{n_W} \gamma_{r}^{(k)} c_{r}^T w(t)^{i_1} w(t)^{i_2} w(t)^{i_3},
\]
\[
= \sum_{r=1}^{R_3} \sum_{i_1, i_2, i_3=1}^{n_W} \gamma_{r}^{(k)} \tilde{w}_r^{i_1} \tilde{w}_r^{i_2} \tilde{w}_r^{i_3},
\]
\[
= \sum_{r=1}^{R_3} \gamma_{r}^{(k)} \tilde{w}_r^3(t). \tag{20}
\]
The LTI systems \( \tilde{W}^{(r)}(q^{-1}, \theta^{(r)}) \) should now be defined as
\[
\tilde{W}^{(r)}(q^{-1}, \theta^{(r)}) := \sum_{s=1}^{n_W} c_{rs} W^{(s)}(q^{-1}, \theta_{W^{(s)}}). \tag{21}
\]

Defining the LTI systems \( \tilde{H}^{(r)}(q^{-1}, \theta^{(r)}) \) as
\[
\tilde{H}^{(r)}(q^{-1}, \theta^{(r)}) := \sum_{k=1}^{n_H} \gamma_{r}^{(k)} H^{(k)}(q^{-1}, \theta_{H^{(k)}}), \tag{22}
\]
and the univariate polynomials \( \tilde{p}^{(r)}(\tilde{w}_r(t)) \) as
\[
\tilde{p}^{(r)}(\tilde{w}_r(t)) := \tilde{w}_r^3(t) \tag{23}
\]
allows to retrieve a decoupled representation of the Wiener-Hammerstein model with \( n = R_3 \) branches.

Similarly as quadratic polynomials, purely cubic polynomials can be simultaneously decoupled by stacking the tensors \( \mathcal{X}^{(k)} \) in a partially symmetric fourth-order tensor of size \( n_W \times n_W \times n_H \times n_H \).

Note that the approach easily generalizes to higher-degree polynomials.

C. Simultaneous decoupling of polynomials with mixed degrees of nonlinearity

This section considers the simultaneous decoupling of quadratic and cubic polynomials, i.e.
\[
z_k(t) = p^{(k)}(w(t)) = w^T(t) A^{(k)} w(t) + \mathcal{X}^{(k)} \times_1 w(t) \times_2 w(t) \times_3 w(t). \tag{24}
\]
We want to decompose the matrices \( A^{(k)} \) and the tensors \( \mathcal{X}^{(k)} \) on the same common basis of vectors
\[
A^{(k)} \approx \sum_{r=1}^{R} \lambda_{r}^{(k)} b_r b_r^T = [\lambda^{(k)}; B, B, B], \tag{25}
\]
\[
\mathcal{X}^{(k)} \approx \sum_{r=1}^{R} \gamma_{r}^{(k)} b_r \circ b_r \circ b_r = [\gamma^{(k)}; B, B, B],
\]
and cast these decompositions as a standard CPD decomposition. In the future, we consider to use a more advanced common matrix/tensor factorization technique reported in [18].

1) Basic idea: The main idea is to store the \( n_H \) matrices \( A^{(k)} \in \mathbb{R}^{n_W \times n_W} \) and the \( n_H \) tensors \( \mathcal{X}^{(k)} \in \mathbb{R}^{n_W \times n_W \times n_H} \) in a partially symmetric tensor \( \mathcal{G} \in \mathbb{R}^{n_W \times n_W \times (n_W+1) \times 2n_H} \), in such a way that \( \mathcal{G} \) has a CPD
\[
\mathcal{G} \approx \sum_{r=1}^{R} b_r \circ b_r \circ e_r \circ m_r, \tag{26}
\]
where
\[
e_{jr} = b_{jr}, \quad j = 1, \ldots, n_W
\]
\[
e_{jr} = 1, \quad j = n_W + 1
\]
\[
m_{kr} = \gamma_{r}^{(k)}, \quad k = 1, \ldots, n_H
\]
\[
m_{(n_H+k)r} = \lambda_{r}^{(k)}, \quad k = 1, \ldots, n_H. \tag{27}
\]
The entries of $G$ are given by

$$g_{i_1 i_2 i_3 k} = x_{i_1 i_2 i_3}^{(k)}$$  \hspace{1cm} (28)$$

$$g_{i_1 i_2 (n_W+1)(n_H+k)} = a_{i_1 i_2}^{(k)}$$  \hspace{1cm} (29)$$

$$g_{i_1 i_2 i_3 (n_H+k)} \sum_{r=1}^{R} b_{i_1}^{(r)} b_{i_2}^{(r)} b_{i_3} \lambda_r^{(k)}$$  \hspace{1cm} (30)$$

$$g_{i_1 i_2 (n_W+1)k} \sum_{r=1}^{R} b_{i_1}^{(r)} b_{i_2} \gamma_r^{(k)}$$  \hspace{1cm} (31)$$

for $i_1, i_2, i_3 = 1, \ldots, n_W$ and $k = 1, \ldots, n_H$. Note that the entries given by (30) and (31) are unknown. However, this issue can be handled by the $N$-way Toolbox [19] and Tensorlab [17], as they allow missing elements in the tensor to be decomposed. Once the CPD of $G$ has been calculated, the basis vectors $b_r$ are available (see (26)). The determination of $\lambda_r^{(k)}$ and $\gamma_r^{(k)}$ in (25) is then a linear least-squares problem.

2) Decoupled Wiener-Hammerstein model: The resulting decoupled representation of the parallel Wiener-Hammerstein model is shown in Fig. 3. The LTI systems $\tilde{W}^{(r)}(q^{-1}, \theta_{\tilde{W}^{(r)}})$ and $\tilde{H}^{(r)}(q^{-1}, \theta_{\tilde{H}^{(r)}})$, and the univariate polynomials $\hat{p}^{(r)}(\tilde{w}_r(t))$ are defined as

$$\tilde{W}^{(r)}(q^{-1}, \theta_{\tilde{W}^{(r)}}) := \sum_{s=1}^{n_W} b_{i_2} W^{(s)}(q^{-1}, \theta_{W^{(s)}})$$  \hspace{1cm} (32)$$

$$\tilde{H}^{(r)}(q^{-1}, \theta_{\tilde{H}^{(r)}}) := \sum_{k=1}^{n_H} \lambda_r^{(k)} H^{(k)}(q^{-1}, \theta_{H^{(k)}})$$  \hspace{1cm} (33)$$

$$\hat{p}^{(r)}(\tilde{w}_r(t)) := \tilde{w}_r^2(t)$$  \hspace{1cm} (34)$$

for $r = 1, \ldots, R$. The model has $n = 2R$ branches.

V. ILLUSTRATION

In this section, the approach is illustrated on a simulation example.

A. Setup

The example considers the decoupling of $n_H = 2$ quadratic and cubic polynomials with $n_W = 3$ inputs. The entries of the matrices $A^{(k)}$ and the tensors $\chi^{(k)}$ that describe the polynomials are drawn from a normal distribution with zero mean and unit variance. The polynomials are decoupled using three different approaches:

1) Separate decoupling: All polynomials are decoupled separately. Both quadratic polynomials are decoupled via an eigenvalue decomposition [8], while the cubic polynomials are decoupled via their CPD.

2) Simultaneous decoupling of homogeneous polynomials: As described in Section IV-A, the matrices $A^{(k)}$ are stacked in a partially symmetric tensor, which will here be called $A$. The CPD of this tensor allows to decouple both quadratic polynomials on a common basis of vectors $b_r$. Similarly, the tensors $\chi^{(k)}$ are stacked in a partially symmetric tensor $\chi$. Its CPD allows to decouple the cubic polynomials on a common basis of vectors $c_r$ (see Section IV-B).

3) Simultaneous decoupling of all polynomials: As described in Section IV-C, the matrices $A^{(k)}$ and the tensors $\chi^{(k)}$ are stored in a partially symmetric tensor $G$. The CPD of this tensor allows to retrieve a set of basis vectors that is common for both the quadratic and the cubic polynomials.

When a CPD is calculated, we start at one rank-one component, and increase the number of rank-one components, until a good approximation of the tensor is found (see Section IV-A.4). So, what is considered good enough? Let $\tilde{A}$ and $\tilde{\chi}$ denote the obtained approximations of $A$ and $\chi$ respectively, and let $\| \cdot \|_F$ denote the Frobenius norm. Then we consider the approximation good enough if both $\|\tilde{A} - A\|_F$ and $\|\tilde{\chi} - \chi\|_F$ are smaller than $10^{-8}$. Each time, 50 different randomly generated starting values are used to increase the chance of converging to at least a good local minimum. The example was implemented in MATLAB, by making use of the recently released Tensorlab toolbox.

B. Results

The results are summarized in Table I. It can be seen that the separate decoupling approach results in a parallel Wiener-Hammerstein model with 14 branches, whereas both simultaneous decoupling approaches would result in only 12 branches. The advantage of approach 3 to have only 6 input branches is canceled by the fact that each input branch is eventually split in a quadratic and a cubic output branch (see Fig. 3). This is due to the fact that $\lambda_r^{(k)}$ and $\gamma_r^{(k)}$ in (25) are in general different from each other, and thus different output filters $\tilde{H}^{(r)}(q^{-1}, \theta_{\tilde{H}^{(r)}})$ result in (33).

<table>
<thead>
<tr>
<th>approach</th>
<th>$|A - \tilde{A}|_F$</th>
<th>$|\chi - \tilde{\chi}|_F$</th>
<th>number of basis vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$5.96 \times 10^{-12}$</td>
<td>$5.60 \times 10^{-12}$</td>
<td>3 + 3 4 + 4</td>
</tr>
<tr>
<td>2</td>
<td>$5.62 \times 10^{-12}$</td>
<td>$9.67 \times 10^{-12}$</td>
<td>6 6</td>
</tr>
<tr>
<td>3</td>
<td>$5.60 \times 10^{-15}$</td>
<td>$5.35 \times 10^{-12}$</td>
<td>6</td>
</tr>
</tbody>
</table>

TABLE I

Results of the simulation example described in Section V-A.
VI. CONCLUSION

Several approaches were presented to decouple the polynomial representation of the nonlinear behavior in parallel Wiener-Hammerstein models. The simultaneous decoupling approaches were formulated as standard tensor decompositions. A simulation example showed that the simultaneous decoupling approaches can result in a model with less parallel branches than the separate decoupling approach.

APPENDIX

This appendix presents how an exact CPD of a symmetric tensor can be obtained by solving a linear set of equations. The basis vectors can be chosen arbitrarily at the cost of a large number of basis vectors. For a simultaneous decoupling of multiple symmetric tensors, one can decouple the tensors separately, choosing each time the same basis vectors.

A symmetric matrix $A \in \mathbb{R}^{n_W \times n_W}$ has $n_W(n_W+1)/2$ independent entries. Therefore, any symmetric matrix $A \in \mathbb{R}^{n_W \times n_W}$ can be written as a linear combination of $n_W(n_W+1)/2$ linearly independent matrices, and thus

$$A = \sum_{r=1}^{n_W(n_W+1)/2} \lambda_r b_r b_r^T$$

is an exact CPD of $A$, where the basis vectors $b_r$ can be chosen fairly arbitrary, and where the coefficients $\lambda_r$ can be determined by solving a linear set of equations.

More generally, let $p(w)$ be a homogeneous polynomial of degree $d$ in $n_W$ variables, i.e.

$$p(w) = \sum_{i_1, \ldots, i_d=1}^{n_W} x_{i_1 \cdots i_d} w_{i_1} \cdots w_{i_d}$$

where $X \in \mathbb{R}^{n_W \times \cdots \times n_W}$ is a symmetric tensor of order $d$. There are $(d+n_W-1)!/(d! (n_W-1)!)$ independent entries $x_{i_1 \cdots i_d}$. Along the lines of (20), one finds a decoupled representation of $p(w)$, namely

$$p(w) = \sum_{r=1}^{(d+n_W-1)!/(d! (n_W-1)!)} \lambda_r (b_r^T w)^d = \sum_{r=1}^{(d+n_W-1)!/(d! (n_W-1)!)} \lambda_r w_r^d$$

From the multinomial theorem, it follows that

$$w_r^d = (b_{1r} w_1 + \ldots + b_{nWr} w_{nW})^d = \sum_{l_1+\ldots+l_nW=d} \frac{d!}{l_1! \cdots l_{nW}!} \prod_{s=1}^{nW} (b_{sr} w_s)^{l_s}$$

Equating the coefficients of the coupled and the decoupled representation of $p(w)$ results in

$$\sum_{l_1+\ldots+l_{nW}=d} \frac{d!}{l_1! \cdots l_{nW}!} \prod_{s=1}^{nW} b_{sr}^{l_s} = \sum_{r=1}^{(d+nW-1)!/(d! (nW-1)!)} \lambda_r \left( \frac{d!}{l_1! \cdots l_{nW}!} \prod_{s=1}^{nW} b_{sr}^{l_s} \right)$$

and thus in a linear set of $\frac{(d+nW-1)!}{d! (nW-1)!}$ equations in the $\frac{(d+nW-1)!}{d! (nW-1)!}$ unknowns $\lambda_r$.

**Remark 2:** An exact CPD of a tensor $X \in \mathbb{R}^{nW \times \cdots \times nW}$ of order $d$ can be obtained with less than $\frac{(d+nW-1)!}{d! (nW-1)!}$ basis vectors, but then, the basis vectors cannot be chosen arbitrarily.

REFERENCES


