Algorithms for canonical polyadic decomposition with block-circulant factors

Frederik Van Eeghem, Lieven De Lathauwer, *Fellow, IEEE*

***Abstract*—Higher-order tensors and their decompositions are well-known tools in signal processing. More specifically, tensors admitting decompositions with structured factor matrices arise in various applications, such as telecommunications or convolutive independent component analysis. These applications motivate the development of efficient algorithms for structured tensor decompositions. In this paper, we develop a method for canonical polyadic decompositions (CPD) with block-circulant factor matrices by extending a method for circulant factors. Block-circulant factor matrices arise in the identification of homogeneous Wiener–Hammerstein models. By transforming the data tensor to the frequency domain, the available structure is exploited using simple element-wise divisions. This approach reformulates the structured CPD as an unstructured CPD of lower rank. The resulting CPD of a possibly incomplete tensor can then be solved algebraically or using optimization-based methods.**

***Index Terms*—Canonical polyadic decomposition, block-circulant, circulant, tensors, Wiener-Hammerstein, uniqueness**

I. Introduction

Tensor decompositions are well-established tools in signal processing and are used for signal separation, data compression, completion problems and more [3], [13], [27]. The efficient computation of tensor decompositions has been the object of research for decades, and many algorithms have been developed [3], [13]. Broadly speaking, these algorithms can be divided into two classes: optimization-based algorithms and algebraic algorithms. As the name suggests, the former optimize an objective function using traditional techniques to find the decomposition [20], [17], [24], [1]. This approach can handle large tensors (e.g. [26], [15], [18], [10]) and is robust to noise, provided the algorithm is properly initialized. Algebraic algorithms decompose tensors using only algebraic operations, such as (generalized) eigenvalue decompositions or singular value decompositions, and are therefore fast for reasonably-sized tensors. In the noiseless case, algebraic algorithms are guaranteed to find the exact solution. For noisy data, they can provide a good initialization for optimization-based methods.

Frederik Van Eeghem is supported by an Aspirant Grant from the Research Foundation Flanders (FWO). This research is funded by (1) Research Council KU Leuven: C1 project c16/15/059-nD; (2) FWO: projects: G.0830.14N, G.0881.14N; (3) EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Advanced Grant: BIOTENSORS (no. 339804). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

Frederik Van Eeghem and Lieven De Lathauwer are with both the Group of Science, Engineering and Technology, KU Leuven Kulak, E. Sabbelaan 53, B-8500 Kortrijk, Belgium and with the Department of Electrical Engineering (ESAT), KU Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium. (e-mail: frederik.vaneeghem@kuleuven.be, lieven.delathauwer@kuleuven.be)

In various applications additional structure is present in the factors of the tensor decomposition. For instance, Vandermonde structured factors arise when using uniform linear arrays [22] and block-Hankel structured factors can be found in convolutive independent component analysis [8].

For better results, this a priori known structure can be taken into account when computing the decomposition. This explains why methods for various structured tensor decompositions are receiving increasing attention [9], [12], [22], [23], [25], [21]. In this paper, we focus on tensors admitting a canonical polyadic decomposition with (block-)circulant factors. Such type of factors arise in the identification of homogeneous Wiener–Hammerstein systems [11], [9]. Circulant matrices are also used to approximate Toeplitz matrices, e.g. in frequency domain convolutive independent component analysis [16].

An algorithm for CPDs with circulant factors has already appeared in [9]. The algorithm rewrites the constrained CPD as a set of monomial equations, which can be solved to obtain the factors. The method presented in this paper stays in the tensor domain, reducing the circulant constrained CPD to an unstructured CPD with rank equal to the width of the circulant blocks. Both approaches are equivalent in terms of underlying mathematics since any rank-1 CPD can be written as a set of monomial equations. However, staying in the tensor domain allows us to easily generalize the algorithm to block-circulant structured factor matrices. An additional advantage is that existing tensor decomposition algorithms and developed uniqueness properties can be used.

Notations: Scalars are denoted by lowercase letters (e.g., a), vectors by bold lowercase letters (e.g., $\mathbf{a}$), matrices by bold uppercase letters (e.g., $\mathbf{A}$), and tensors by uppercase calligraphic letters (e.g., $\mathcal{A}$). The outer product is denoted by $\circ$. We use $\star$ and $\oslash$ for the element-wise product and division, respectively. The discrete Fourier transform and inverse transform are respectively given by $\mathcal{F}\{\cdot\}$ and $\mathcal{F}^{-1}\{\cdot\}$. The Frobenius norm is given by $||\cdot||_{\mathrm{F}}$. Estimates are denoted by a hat atop, e.g. $\widehat{a}$.

II. Tensors and Problem statement

A. Canonical polyadic decomposition

An Nth-order tensor has rank 1 if and only if it equals the outer product of N nonzero vectors. By extension, the rank of a tensor $\mathcal{A}$ is defined as the minimal number of rank-1 tensors yielding $\mathcal{A}$ in a linear combination. The canonical polyadic

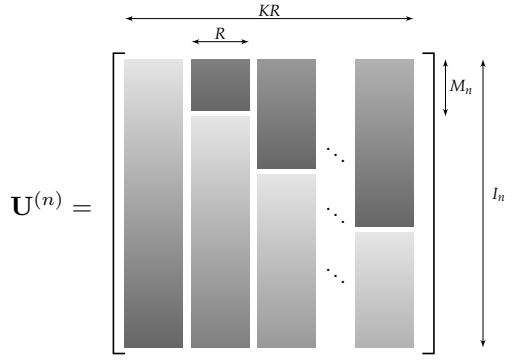


Figure 1. Block-circulant structure of the factor matrix $\mathbf{U}^{(n)}$.

decomposition (CPD) writes a tensor $\mathcal{A}$ as a minimal sum of rank-1 terms:

$$\mathcal{A} = \sum_{q=1}^{Q} \mathbf{u}_q^{(1)} \circ \cdots \circ \mathbf{u}_q^{(N)} = \left[\!\!\left[\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(N)} \right]\!\!\right],$$

in which factor matrix $\mathbf{U}^{(n)}$ has Q columns $\mathbf{u}_q^{(n)}$ [13], [3].

Contrarily to a decomposition of a matrix in rank-1 terms, the CPD of higher-order tensors is essentially unique under fairly mild conditions [14], [19], [4]. Essential uniqueness means that the decomposition is unique up to trivial indeterminacies. More specifically, the different rank-1 terms can be arbitrarily permuted and the vectors within each rank-1 term may be scaled and counterscaled as long as the rank-1 term remains unchanged.

B. Problem statement

Consider an Nth-order tensor $\mathcal{T} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ admitting a CPD of rank KR:

$$\mathcal{T} = \sum_{r=1}^{KR} \mathbf{u}_r^{(1)} \circ \cdots \circ \mathbf{u}_r^{(N)} = \left[\!\!\left[\mathbf{U}^{(1)}, \cdots, \mathbf{U}^{(N)} \right]\!\!\right].$$

Assume all factors $\mathbf{U}^{(n)} \in \mathbb{C}^{I_n \times KR}$ have block-circulant structure with blocks having R columns and circular shifts of size M_n, as depicted in Figure 1. Note that, apart from the block-circulant structure, the columns may be arbitrary. Let $\mathbf{U}_k^{(n)} \in \mathbb{C}^{I_n \times R}$ denote a block-column of $\mathbf{U}^{(n)}$ such that $\mathbf{U}^{(n)} = [\mathbf{U}_1^{(n)}, \mathbf{U}_2^{(n)}, \ldots, \mathbf{U}_K^{(n)}]$. If $\mathbf{U}_1^{(n)}$ is known, the matrix $\mathbf{U}^{(n)}$ is fully determined because of the block-circulant structure. The aim of the presented method is to efficiently compute the generating blocks $\mathbf{U}_1^{(n)}$, for $n \in \{1, \ldots, N\}$.

III. Algorithm and uniqueness

As in [9], we start by computing the multidimensional discrete Fourier transformation (DFT) of $\mathcal{T}$, which we denote by $\mathcal{T}^{(\mathcal{F})}$. Since this transformation is multilinear, the multidimensional DFT can be interpreted as the column-wise DFT of the factor matrix in each mode. Using the shift property of the DFT and the block-circulant structure of $\mathbf{U}^{(n)}$, the DFT of the rth column of the kth block-column of $\mathbf{U}^{(n)}$ can be written as

$$\mathcal{F}\left\{ \left(\mathbf{U}_k^{(n)} \right)_r \right\} = \mathbf{g}_r^{(n)} \star \boldsymbol{\alpha}^{M_n(k-1)/I_n}, \tag{1}$$

in which

$$\boldsymbol{\alpha}^{M_n(k-1)/I_n} = \begin{bmatrix} 1 \\ e^{-2\pi j M_n(k-1)/I_n} \\ e^{-4\pi j M_n(k-1)/I_n} \\ \vdots \\ e^{-2(I_n-1)\pi j M_n(k-1)/I_n} \end{bmatrix} \in \mathbb{C}^{I_n},$$

and $\mathbf{g}_r^{(n)}$ is the rth column of $\mathbf{G}^{(n)}$, given by

$$\mathbf{G}^{(n)} = \mathcal{F}\left\{ \mathbf{U}_1^{(n)} \right\} \in \mathbb{C}^{I_n \times R}.$$

Using expression (1), the CPD of $\mathcal{T}^{(\mathcal{F})}$ can be written as

$$\mathcal{T}^{(\mathcal{F})} = \sum_{r=1}^{R} \sum_{k=1}^{K} \left(\mathbf{g}_r^{(1)} \star \boldsymbol{\alpha}^{M_1(k-1)/I_1} \right) \circ \cdots \circ \left(\mathbf{g}_r^{(N)} \star \boldsymbol{\alpha}^{M_N(k-1)/I_N} \right).$$

A single entry of $\mathcal{T}^{(\mathcal{F})}$ is given in (2) at the top of the next page. Note that the summation over k is fully determined by the known structure of the factor matrices. To simplify notation, rewrite (2) as

$$t_{i_1 i_2 \cdots i_N}^{(\mathcal{F})} = a^{(i_1, i_2, \cdots, i_N)} \sum_{r=1}^{R} g_{i_1 r}^{(1)} g_{i_2 r}^{(2)} \cdots g_{i_N r}^{(N)}, \tag{3}$$

in which $a^{(i_1, i_2, \cdots, i_N)}$ represents the factor containing the summation over k. Because this factor is known in advance, it can be constructed separately for all tensor entries. To do so, construct the matrices $\mathbf{A}^{(n)} \in \mathbb{C}^{I_n \times K}$ given by

$$\mathbf{A}^{(n)} = \begin{bmatrix} \mathbf{1} & \boldsymbol{\alpha}^{M_n/I_n} & \cdots & \boldsymbol{\alpha}^{M_n(K-1)/I_n} \end{bmatrix}.$$

The tensor $\mathcal{A} = \left[\!\!\left[\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)} \right]\!\!\right] \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ then has entries

$$a_{i_1 i_2 \cdots i_N} = a^{(i_1, i_2, \cdots, i_N)}. \tag{4}$$

It follows from (3) and (4) that $\mathcal{T}^{(\mathcal{F})}$ can be written as

$$\mathcal{T}^{(\mathcal{F})} = \mathcal{G} \star \mathcal{A},$$

in which $\mathcal{G} = \left[\!\!\left[\mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \cdots, \mathbf{G}^{(N)} \right]\!\!\right]$ is a rank-R tensor. The rest of the algorithm depends on whether any entries of $\mathcal{A}$ are zero. If all entries are nonzero, we can simply divide $\mathcal{T}^{(\mathcal{F})}$ element-wise by $\mathcal{A}$ to obtain $\mathcal{G}$:

$$\mathcal{G} = \mathcal{T}^{(\mathcal{F})} \oslash \mathcal{A}. \tag{5}$$

By doing so, we have exploited the circulant structure of the factors and have reduced the rank from RK to R. The factors of $\mathcal{G}$ can then be obtained using any algorithm that is able to handle complex data. If the data is noiseless, computing this CPD using a generalized eigenvalue decomposition (GEVD) yields a fully algebraic method [5]. Note that the uniqueness conditions for this unstructured decomposition of $\mathcal{G}$ are equivalent to the uniqueness conditions for the structured decomposition of $\mathcal{T}$, which allows us to verify these conditions easily [4]. Once we have an estimate for the factors $\widehat{\mathbf{G}}^{(n)}$, we can retrieve the generators of the original factors up to scaling and permutation by computing the inverse column-wise DFT

$$\widehat{\mathbf{U}}_1^{(n)} = \mathcal{F}^{-1}\left\{ \widehat{\mathbf{G}}^{(n)} \right\}. \tag{6}$$

$$t^{(\mathcal{F})}_{i_1 i_2 \cdots i_N} = \sum_{r=1}^{R} \sum_{k=1}^{K} g^{(1)}_{i_1 r} g^{(2)}_{i_2 r} \cdots g^{(N)}_{i_N r} e^{-2\pi j i_1 M_1 (k-1)/I_1} e^{-2\pi j i_2 M_2 (k-1)/I_2} \cdots e^{-2\pi j i_N M_N (k-1)/I_N}$$
$$= \left(\sum_{r=1}^{R} g^{(1)}_{i_1 r} g^{(2)}_{i_2 r} \cdots g^{(N)}_{i_N r} \right) \left(\sum_{k=1}^{K} e^{-2\pi j (M_1 (k-1) i_1 / I_1 + M_2 (k-1) i_2 / I_2 + \cdots + M_N (k-1) i_N / I_N)} \right) \quad (2)$$

If the full factor matrices are needed, they can easily be constructed using the generators $\widehat{\mathbf{U}}^{(n)}_1$.

Note that this approach is only possible if none of the entries of $\mathcal{A}$ is zero. In [9], a proposition is derived establishing a necessary and sufficient condition for nonnullity of the entries of $\mathcal{A}$ for circulant factors. For our purposes, it suffices to check the tensor $\mathcal{A}$ for zero values. If $a^{(i_1, i_2, \cdots, i_N)}$ is zero, we will treat the corresponding entry of $\mathcal{G}$ as missing. In this way, we obtain an incomplete tensor, of which the CPD can still be computed provided enough entries are available. Decompositions of incomplete tensors can be computed using optimization-based methods [3], [27], [2]. If there are at least two slices of $\mathcal{G}$ that do not contain any missing values, a GEVD can be computed of the subtensor consisting of these slices to partially initialize the optimization-based method. From the decomposition of $\mathcal{G}$ an estimate $\widehat{\mathbf{G}}^{(n)}$ is obtained, which can be used to find the original circulant factors as described above in (6).

To recap, our main method computes the CPD of $\mathcal{G}$ using a GEVD, unless $\mathcal{G}$ is incomplete, in which case an optimization-based method is used. Note that instead of explicitly dividing in (5) and computing a CPD of a tensor with missing entries, a weighted CP decomposition of $\mathcal{T}^{(\mathcal{F})}$ can be used instead.

The time complexity of the presented method (with a generalized eigenvalue decomposition for the CPD) for a tensor $\mathcal{T} \in \mathbb{C}^{N \times N \times N}$ is in the order of $\mathcal{O}\left(N^3 \log(N)\right)$, mainly due to the dominating multidimensional DFT.

IV. Numerical experiments

To illustrate the performance of the presented methods, numerical experiments are conducted for both circulant and block-circulant factor matrices. In both experiments, the mean relative error (MRE) is used as a measure for accuracy. For an Nth-order tensor with factors $\mathbf{U}^{(n)}$, the MRE is defined as

$$\text{MRE} = \frac{1}{N} \sum_{n=1}^{N} \frac{\left|\left| \mathbf{U}^{(n)} - \widehat{\mathbf{U}}^{(n)} \Pi \Delta_n \right|\right|_{\text{F}}}{\left|\left| \mathbf{U}^{(n)} \right|\right|_{\text{F}}}.$$

In this definition, the matrices Δ_n and Π represent the optimal scaling and permutation matrix to deal with the indeterminacies of the CPD as discussed in Section II.

All tensor computations are implemented using Tensorlab and all optimization-based decompositions are computed using a nonlinear least squares algorithm [20], [28].

A. Circulant factors

Consider a tensor $\mathcal{T} \in \mathbb{C}^{3 \times 3 \times 3}$ with rank 2. It is constructed such that its factor matrices $\mathbf{U}^{(n)} \in \mathbb{C}^{3 \times 2}$ have circulant structure. The generating elements of the different factors have been randomly sampled from a standard normal distribution. After construction of the tensor, additive Gaussian noise was added to obtain a predefined signal-to-noise ratio. The noisy tensor is then decomposed using various methods and the accuracy of the estimated factors is reported.

Table I
OUR METHOD OUTPERFORMS A NAIVE IMPLEMENTATION OF [9] AND IS NEARLY OPTIMAL WHEN COMPARED TO A FULL OPTIMIZATION-BASED SOLUTION. THE TABLE SHOWS THE MEAN ACCURACY OVER 100 EXPERIMENTS EXPRESSED IN DB FOR VARIOUS SNRS.

	SNR			
	0 dB	10 dB	20 dB	30 dB
Method from [9]	−3.84	−6.14	−11.92	−18.24
Method from this paper	−9.72	−20.36	−30.50	−40.73
Optimization initialized with this paper's method	−11.47	−21.92	−31.84	−41.81

The mean results of 100 experiments are shown in Table I. The method presented in [9] seems to perform worst. Note that this experiment is essentially the illustrative example given in [9], where the authors have stated that their way of solving the set of monomial equations is not guaranteed to be optimal in any sense. If more equations are taken into account, the accuracy is expected to improve until it matches the accuracy of our method, which automatically takes all available information into account. In this case, the tensor $\mathcal{A}$ in (4) has no zero entries, which allows us to compute the CPD of $\mathcal{G}$ using a generalized eigenvalue decomposition. This implies that the presented algorithm is fully algebraic in this experiment. If this algebraic result is used to initialize a fully-structured optimization-based method based on nonlinear least squares, the results are similar. This implies that the algebraic method is already quite accurate.

B. Block-circulant factors

Consider a tensor $\mathcal{T} \in \mathbb{C}^{9 \times 9 \times 9}$ with rank 9. Its factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ have block-circulant structure with block width $R = 3$. More specifically, $\mathbf{A}$ and $\mathbf{B}$ are block-diagonal matrices with blocks of size 3×3. The diagonal blocks within each factor are all the same since the matrices have block-circulant structure. Their entries have been randomly sampled from a standard normal distribution. The factor $\mathbf{C}$ has block-circulant structure as depicted in Figure 1 with shifts of size $M_c = 2$. Its generating elements are randomly selected from a standard normal distribution. The noisy tensor is obtained by adding Gaussian noise to the exact tensor.

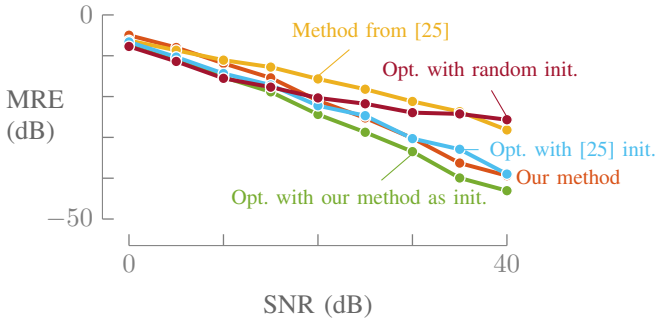


Figure 2. The method presented in this paper outperforms other methods, both as standalone method and as initialization for optimization-based methods. The figure shows the mean accuracy over 500 experiments.

Table II
EXECUTION TIMES OF VARIOUS ALGORITHMS.

	Execution time (s)
Our method	0.093 s
Method from [25]	0.075 s
Optimization-based method, init. with our method	3.54 s
Optimization-based method, init. with [25]	4.45 s
Optimization-based method, random init.	46.62 s

The mean results of 500 experiments are presented in Figure 2. Because of the specific construction of $\mathbf{A}$ and $\mathbf{B}$, the matrix $\mathbf{B} \odot \mathbf{A}$ has block-Toeplitz structure, which allows us to compute the factors using the subspace-based method from [25]. This method exploits the block-Toeplitz structure in two modes to find two factors. The third factor can then be computed using the estimates of the other factors.

In our proposed method, 22% of the entries of $\mathcal{A}$ are zero, which implies that 22% of the entries of $\mathcal{G}$ are missing. However, there are six frontal slices that do not contain any missing entries. We use a GEVD of two of these known slices to partially initialize a nonlinear least squares method that computes the full decomposition of $\mathcal{G}$ [28].

The fully structured optimization-based method constrains the factors to have block-circulant structure. The initialization of this method is either random, is done using our presented method, or is done using the method from [25]. All three initialization strategies are shown in Figure 2. For the random initialization, the algorithm was run three times and the best result was retained.

Figure 2 shows that the subspace-based algorithm from [25] is the least accurate for most SNR values. This is because the structure in only two modes is exploited, whereas the other methods take all structure into account. Our proposed algorithm is especially accurate for higher SNRs, where it outperforms a fully structured decomposition with random initialization and matches the optimization-based method initialized by [25]. The best result is achieved when our proposed method is used to initialize the fully structured decomposition. In the previous experiment, there was hardly any difference between our proposed algorithm and the fully structured one. In this experiment, the difference is larger because the missing elements reduce the available information. Computation times

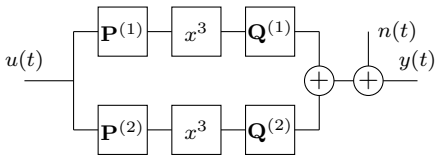


Figure 3. Block scheme of a parallel Wiener–Hammerstein system.

of all methods are shown in Table II and were measured at an SNR of 20 dB on a standard laptop containing 16 GB of RAM and an Intel Core i7-4810MQ processor. The results show that the speed of our method and that of [25] is comparable. However, for tensors with all (block-)circulant factors, our method is more accurate. The presented method is thus a valuable standalone and initialization algorithm.

C. Parallel Wiener–Hammerstein system identification

The identification of a homogeneous Wiener–Hammerstein system can be formulated as a CPD with almost all circulant factors [9], [11], [7]. A Wiener–Hammerstein system is defined as a static nonlinear block sandwiched between two linear time-invariant FIR filters and can be identified by first estimating a Volterra kernel from input/output data and then computing its CPD. This idea was extended to parallel Wiener–Hammerstein systems in [6]. It is shown that the estimated Volterra kernels admit a structured tensor decomposition with almost all block-circulant factors. An optimization-based strategy to solve this problem has been developed in [6]. Here, we focus on the special case depicted in Figure 3. If $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$ are simple moving average filters of length $L+1$, it immediately follows from [6] that the third-order Volterra kernel $\mathcal{K}$ can be written as a CPD with pure block-circulant factors containing the filter coefficients of $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$. Note that for arbitrary filters $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$, one has to resort to optimization-based techniques such as [6]. How the parameters appear in the decomposition is explained in [6] as well.

As an illustration, we simulate the system shown in Figure 3. The input signal $u(t)$ consists of 10^4 random samples from a standard normal distribution. The filters $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$ consist of 6 samples from a standard normal distribution. The filters $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$ are simple moving average filters of length 6. White Gaussian noise $n(t)$ is added to obtain an SNR of 20 dB. After estimating the third-order Volterra kernel as in [6], our method was used to retrieve the coefficients of $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$. Our method reaches an MRE of 0.021 whereas the method from [25] achieves an MRE of 0.049.

V. CONCLUSION

We presented a method to efficiently compute a CPD with block-circulant factors by building on the existing method from [9]. By transforming the data tensor to the frequency domain and computing an element-wise division, the rank of the tensor is reduced to the width of the circulant block-columns. The resulting (possible incomplete) tensor can then be decomposed using standard algebraic or optimization-based methods. This yields an easy-to-compute and accurate estimate that can be used directly or as initialization for optimization-based methods that take the full structure into account.

REFERENCES

[1] E. Acar, T. G. Kolda, and D. M. Dunlavy, "All-at-once optimization for coupled matrix and tensor factorizations," *KDD Workshop on Mining and Learning with Graphs*, 2011.

[2] C. F. Caiafa and A. Cichocki, "Multidimensional compressed sensing and their applications," *WIREs Data Mining Knowl. Discov.*, vol. 3, no. 6, pp. 355–380, 2013.

[3] A. Cichocki, C. Mandic, A. H. Phan, C. Caifa, G. Zhou, Q. Zhao, and L. De Lathauwer, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

[4] I. Domanov and L. De Lathauwer, "On the uniqueness of the canonical polyadic decomposition of third-order tensors — Part II: Overall uniqueness," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 3, pp. 876–903, 2013.

[5] ——, "Canonical polyadic decomposition of third-order tensors: reduction to generalized eigenvalue decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 2, pp. 636–660, 2014.

[6] P. Dreesen, D. Westwick, J. Schoukens, and M. Ishteva, "Modeling parallel Wiener-Hammerstein systems using tensor decomposition of Volterra kernels," in *13th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*. Springer, Feb. 2017, pp. 16–25.

[7] G. Favier and A. Kibangou, "Tensor-based methods for system identification. Part 2: Three examples of tensor-based system identification methods," *International Journal on Sciences and Techniques of Automatic Control (IJ-STA)*, vol. 3, no. 1, pp. 870–889, 2009.

[8] C. E. R. Fernandes, G. Favier, and J. C. M. Mota, "PARAFAC-based blind identification of convolutive MIMO linear systems," vol. 42, no. 10. Elsevier, 2009, pp. 1704–1709.

[9] J. H. M. Goulart and G. Favier, "An algebraic solution for the Candecomp/Parafac decomposition with circulant factors," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 4, pp. 1543–1562, 2014.

[10] U. Kang, E. Papalexakis, A. Harpale, and C. Faloutsos, "Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 316–324.

[11] A. Kibangou and G. Favier, "Wiener-Hammerstein systems modeling using diagonal Volterra kernels coefficients," *IEEE Signal Proc. Letters*, vol. 13, no. 6, p. 381384, Jun. 2006.

[12] A. Y. Kibangou and G. Favier, "Non-Iterative Solution for PARAFAC with a Toeplitz Matrix Factor," in *Proc. EUSIPCO*, Aug. 24-28, Glasgow, Scotland, 2009.

[13] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[14] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with applications to arithmetic complexity and statistics," *Linear Algebra Appl.*, vol. 18, pp. 95–138, 1977.

[15] E. Papalexakis, C. Faloutsos, and N. Sidiropoulos, "Parcube: Sparse parallelizable tensor decompositions," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 521–536.

[16] M. Pedersen, J. Larsen, U. Kjems, and L. Parra, "A survey of convolutive blind source separation methods," *Multichannel Speech Processing Handbook*, pp. 1065–1084, 2007.

[17] A. Phan, P. Tichavsky, and A. Cichocki, "Low complexity damped Gauss–Newton algorithms for CANDECOMP/PARAFAC," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 126–147, 2013.

[18] N. Sidiropoulos, E. Papalexakis, and C. Faloutsos, "A parallel algorithm for big tensor decomposition using randomly compressed cubes (paracomp)," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2014)*. IEEE, 2014, pp. 1–5.

[19] N. D. Sidiropoulos and R. Bro, "On the Uniqueness of Multilinear Decomposition of N-way Arrays," *J. Chemometrics*, vol. 14, pp. 229–239, 2000.

[20] L. Sorber, M. Van Barel, and L. De Lathauwer, "Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 695–720, 2013.

[21] M. Sørensen and P. Comon, "Tensor decompositions with banded matrix factors," *Linear Algebra and its Applications*, vol. 438, no. 2, pp. 919–941, 2013.

[22] M. Sørensen and L. De Lathauwer, "Blind signal separation via tensor decomposition with Vandermonde factor: Canonical polyadic decomposition," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5507–5519, 2013.

[23] M. Sorensen, F. Van Eeghem, and L. De Lathauwer, "Blind multichannel deconvolution and convolutive extensions of canonical polyadic and block term decompositions," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4132–4145, 2017.

[24] G. Tomasi and R. Bro, "A comparison of algorithms for fitting the PARAFAC model," *Computational Statistics & Data Analysis*, vol. 50, no. 7, pp. 1700–1734, 2006.

[25] F. Van Eeghem, M. Sørensen, and L. De Lathauwer, "Tensor decompositions with several block-Hankel factors and application in blind system identification," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4090–4101, 2017.

[26] N. Vervliet and L. De Lathauwer, "A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors," *IEEE J. Sel. Topics Signal Process.*, 2015, (accepted).

[27] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer, "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 71–79, 2014.

[28] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab 3.0," Mar. 2016, available online. [Online]. Available: http://www.tensorlab.net