



**HAL**  
open science

## High-order CPD estimator with dimensionality reduction using a tensor train model

Yassine Zniyed, Remy Boyer, André L. F. de Almeida, Gérard Favier

► **To cite this version:**

Yassine Zniyed, Remy Boyer, André L. F. de Almeida, Gérard Favier. High-order CPD estimator with dimensionality reduction using a tensor train model. 26th European Signal Processing Conference (EUSIPCO 2018), Sep 2018, Rome, Italy. 10.23919/eusipco.2018.8553466 . hal-01815214

**HAL Id: hal-01815214**

**<https://centralesupelec.hal.science/hal-01815214>**

Submitted on 13 Jun 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# HIGH-ORDER CPD ESTIMATION WITH DIMENSIONALITY REDUCTION USING A TENSOR TRAIN MODEL

Yassine Zniyed<sup>\*</sup>   Rémy Boyer<sup>\*</sup>   André L. F. de Almeida<sup>†</sup>   Gérard Favier<sup>‡</sup>

<sup>\*</sup>L2S, Univ. Paris-Sud, CNRS, CentraleSupélec, Gif-sur-Yvette, France

<sup>†</sup>Department of Teleinformatics Engineering, Federal University of Fortaleza, Brazil

<sup>‡</sup>Laboratoire I3S, Université Côte d’Azur, CNRS, Sophia Antipolis, France

## ABSTRACT

The canonical polyadic decomposition (CPD) is one of the most popular tensor-based analysis tools due to its usefulness in numerous fields of application. The  $Q$ -order CPD is parametrized by  $Q$  matrices also called factors which have to be recovered. The factors estimation is usually carried out by means of the alternating least squares (ALS) algorithm. In the context of multi-modal big data analysis, *i.e.*, large order ( $Q$ ) and dimensions, the ALS algorithm has two main drawbacks. Firstly, its convergence is generally slow and may fail, in particular for large values of  $Q$ , and secondly it is highly time consuming. In this paper, it is proved that a  $Q$ -order CPD of rank- $R$  is equivalent to a train of  $Q$  3-order CPD(s) of rank- $R$ . In other words, each tensor train (TT)-core admits a 3-order CPD of rank- $R$ . Based on the structure of the TT-cores, a new dimensionality reduction and factor retrieval scheme is derived. The proposed method has a better robustness to noise with a smaller computational cost than the ALS algorithm.

**Index Terms**— Tensor decompositions, CP decomposition, Tensor train, Big data, Multidimensionnal signal processing, Parameter estimation, Fast algorithms.

## I. INTRODUCTION

In this paper, we focus on high-order CPD models [5]. Such models have a great interest in signal processing for blind equalization [7], [4], blind source separation [16], radar [10], wireless communications [12], [3], among many other fields of application. Most methods of factor estimation are ALS-based techniques [2]. Unfortunately these techniques may require several iterations to converge [11], [6], and convergence is increasingly difficult when the order of the tensor increases [1] and it is not even guaranteed [8]. Hence, algorithms that are stable and scalable [15] with the tensor order and dimensions are needed. In this work, we establish an equivalence between a  $Q$ -order CPD of canonical rank  $R$  and a tensor train (TT) [9] model. Note that the idea of rewriting a CPD into the TT format was briefly mentioned in [9]. In this paper, we exploit this idea to propose a new CPD factor retrieval algorithm. Indeed, we will see

that a  $Q$ -order CPD of canonical rank  $R$  is equivalent to a train of  $Q$  3-order CPD(s) with TT-ranks equal to  $R$ . That leads to a new tensor decomposition that we call a CPD-train model, and which can be viewed as a special case of the TT model whose the TT-cores follow a CPD. The advantages of using the TT model in this case is that its storage cost, just like the CPD, scales linearly with  $Q$  [9]. The estimation of the TT-ranks amounts to estimating the canonical rank of the associated CPD. And finally, the use of the TT-SVD algorithm [9] allows to overcome the drawbacks of iterative algorithms for very high-order tensors. As a consequence of this equivalence, one could apply one trilinear ALS estimator and  $Q - 3$  bilinear ALS estimators to the (much smaller) TT-cores to recover the original CPD factor matrices, instead of applying the ALS algorithm to the very high-order CPD. Hence, a solution with much less computational cost and better convergence is possible thanks to this TT-CPD equivalence.

The rest of this paper is organized as follows. Section II recalls the CPD and TT formulations, and gives the main result in terms of the equivalence between the two models. Section III-B shows how to exploit the known structure of the CPD TT-cores, by applying the ALS algorithm on 3-order tensors instead of the original  $Q$ -order CPD. In section IV, we compare the computational complexities of the TT-SVD and ALS algorithms for estimating the parameters of a  $Q$ -order CPD. Section V presents simulation results that will reinforce the arguments given in the previous sections. Finally, the conclusions are drawn in Section VI.

*Notations:* Scalars, vectors, matrices and tensors are represented by  $x$ ,  $\mathbf{x}$ ,  $\mathbf{X}$  and  $\mathcal{X}$ , respectively. The symbols  $(\cdot)^T$ ,  $\text{rank}(\cdot)$ , and  $\kappa(\cdot)$  denote, respectively, the transpose, the rank and the dominant complexity cost in flops. The Frobenius norm is defined by  $\|\cdot\|_F$ .  $\mathcal{I}_{k,R}$  denotes the  $k$ -order identity tensor of size  $R \times \dots \times R$ , and  $\mathcal{I}_2 = \mathbf{I}_R$ . The matrix  $\text{unfold}_k \mathcal{X}$  of size  $N_k \times N_1 \dots N_{k-1} N_{k+1} \dots N_Q$  refers to the  $k$ -mode unfolding of  $\mathcal{X}$  of size  $N_1 \times \dots \times N_Q$ . The  $n$ -mode product is denoted by  $\times_n$ . The contraction  $\times_q^p$  between two tensors  $\mathcal{A}$  and  $\mathcal{B}$  of size  $N_1 \times \dots \times N_Q$  and  $M_1 \times \dots \times M_P$ , with  $N_q = M_p$  is a tensor of order

$(Q + P - 2)$  such that

$$\begin{aligned} & [\mathcal{A} \times_q^p \mathcal{B}]_{n_1, \dots, n_{q-1}, n_{q+1}, \dots, n_Q, m_1, \dots, m_{p-1}, m_{p+1}, \dots, m_P} \\ &= \sum_{k=1}^{N_q} [\mathcal{A}]_{n_1, \dots, n_{q-1}, k, n_{q+1}, \dots, n_Q} [\mathcal{B}]_{m_1, \dots, m_{p-1}, k, m_{p+1}, \dots, m_P}. \end{aligned}$$

## II. CPD-TRAIN MODEL: EQUIVALENCE BETWEEN THE CPD AND THE TENSOR-TRAIN MODEL

### II-A. Canonical Polyadic Decomposition (CPD)

A  $Q$ -order tensor of size  $N_1 \times \dots \times N_Q$  follows a rank- $R$  CPD if

$$\mathcal{X} = \mathcal{I}_{Q,R} \times_1 \mathbf{P}_1 \times_2 \mathbf{P}_2 \dots \times_Q \mathbf{P}_Q \quad (1)$$

where the  $k$ -mode factor matrix  $\mathbf{P}_k$  is of size  $N_k \times R$ ,  $1 \leq k \leq Q$ .

### II-B. Tensor-Train Model (TTM) and TT-ranks

Let  $\{R_1, \dots, R_{Q-1}\}$  be the  $Q-1$  TT-ranks with bounding conditions  $R_0 = R_Q = 1$ . A  $Q$ -order tensor of size  $N_1 \times \dots \times N_Q$  admits a decomposition into a train of tensors if

$$\mathcal{X} = \mathbf{G}_1 \times_2^1 \mathcal{G}_2 \times_3^1 \mathcal{G}_3 \times_4^1 \dots \times_{Q-1}^1 \mathcal{G}_{Q-1} \times_Q^1 \mathbf{G}_Q \quad (2)$$

in which the TT-cores are defined according to

$$\begin{aligned} \mathbf{G}_1 &: N_1 \times R_1, \quad \mathbf{G}_Q^T : N_Q \times R_{Q-1}, \\ \mathcal{G}_k &: R_{k-1} \times N_k \times R_k, \quad \text{for } 2 \leq k \leq Q-1 \end{aligned}$$

with the TT-ranks given by  $\text{rank}(\mathbf{G}_1) = R_1$ ,  $\text{rank}(\mathbf{G}_Q) = R_{Q-1}$ , and for  $2 \leq k \leq Q-1$ ,  $\text{rank}(\text{unfold}_1 \mathcal{G}_k) = R_{k-1}$ , and  $\text{rank}(\text{unfold}_3 \mathcal{G}_k) = R_k$ .

### II-C. Rewriting of a high order CPD in a TT format

In this section, we present an equivalence between a  $Q$ -order CPD and a train of  $Q$  CPD(s) according to Fig. 1, where the structure of the TT-cores are given in the following Theorem.

**Theorem 1.** Assume that a tensor  $\mathcal{X}$  follows a  $Q$ -order CPD of rank- $R$  given by eq. (1), where the factor matrices  $\mathbf{P}_k$  are full column rank. This tensor admits an equivalent TTM with TT-cores given by:

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{P}_1, \quad \mathbf{G}_Q = \mathbf{P}_Q^T, \\ \mathcal{G}_k &= \mathcal{I}_{3,R} \times_2 \mathbf{P}_k, \quad 2 \leq k \leq Q-1 \end{aligned} \quad (3)$$

Moreover, the TT-ranks are all identical and equal to the canonical rank  $R$ .

*Proof.* The TTM of the  $Q$ -order identity tensor  $\mathcal{I}_{Q,R}$  is given by

$$\mathcal{I}_{Q,R} = \mathbf{I}_R \times_2^1 \mathcal{I}_{3,R} \times_3^1 \dots \times_{Q-1}^1 \mathcal{I}_{3,R} \times_Q^1 \mathbf{I}_R.$$

Substituting the above TTM into the CPD of eq. (1), we get

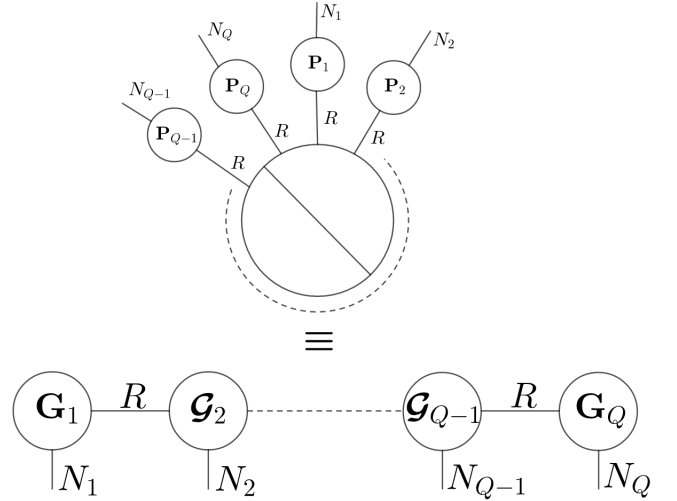
$$\mathcal{X} = (\mathbf{I}_R \times_2^1 \mathcal{I}_{3,R} \times_3^1 \dots \times_{Q-1}^1 \mathbf{I}_R) \times_1 \mathbf{P}_1 \dots \times_Q \mathbf{P}_Q$$

by expressing the entries of  $\mathcal{X}$  and reorganizing them, we can equivalently write

$$\mathcal{X} = (\mathbf{P}_1 \mathbf{I}_R) \times_2^1 (\mathcal{I}_{3,R} \times_2 \mathbf{P}_2) \times_3^1 \dots \times_Q^1 (\mathbf{I}_R \mathbf{P}_Q^T). \quad (4)$$

By identifying the TT-cores in (4) with those in eq. (2), we can deduce the relations (3). Then, it is straightforward to conclude that the TT-ranks are all identical and equal to the canonical rank  $R$ .  $\square$

As a  $Q$ -order CPD of rank- $R$  is equivalent to a train of  $Q$  3-order CPD(s) of rank- $R$ , we refer to Eq. (4) as a CPD-train.



**Fig. 1.** Factor graph of a  $Q$ -order CPD and its TTM. The TT-cores are given in Theorem 1.

## III. DIMENSIONALITY REDUCTION AND CPD FACTORS ESTIMATION BASED ON A CPD-TRAIN

### III-A. Dimensionality reduction: TT-cores recovery

The TT-SVD algorithm presented in [9] computes the TT-cores by extracting in a sequential way the dominant subspaces (thanks to truncated SVDs) of matrix reshapings of one SVD factor calculated at each step of the algorithm. The TT-cores recovery is done up to nonsingular *change-of-basis* matrices. In Fig. 2, we illustrate the TT-SVD algorithm applied to a 4-order tensor  $\mathcal{X}$ . At each step, the original tensor is reshaped into a matrix having one mode in one dimension and a combination of remaining modes in the other dimension, so that each SVD calculation delivers a TT-core. In our example of Fig. 2, the original  $N_1 \times N_2 \times N_3 \times N_4$  tensor  $\mathcal{X}$  is first reshaped into an  $N_1 \times (N_2 N_3 N_4)$  matrix  $\mathbf{X}_{(1)}$  whose the SVD provides the factors  $\mathbf{U}^{(1)}$  that contains the left singular vectors, and  $\mathbf{V}^{(1)}$  that is the product of the diagonal singular values matrix with the matrix containing the right singular vectors. Then, the  $R_1 \times (N_2 N_3 N_4)$  matrix  $\mathbf{V}^{(1)}$  is reshaped into a matrix

$\mathbf{V}_{(2)}^{(1)}$  of size  $(R_1 N_2) \times (N_3 N_4)$ , whose the SVD provides the factors  $(\mathbf{U}^{(2)}, \mathbf{V}^{(2)})$ ,  $\mathbf{V}^{(2)}$  being reshaped into  $\mathbf{V}_{(2)}^{(2)}$ , and so on. The TT-cores  $\mathcal{G}_k$  are sequentially generated from the matrices  $\mathbf{U}^{(k)}$ .

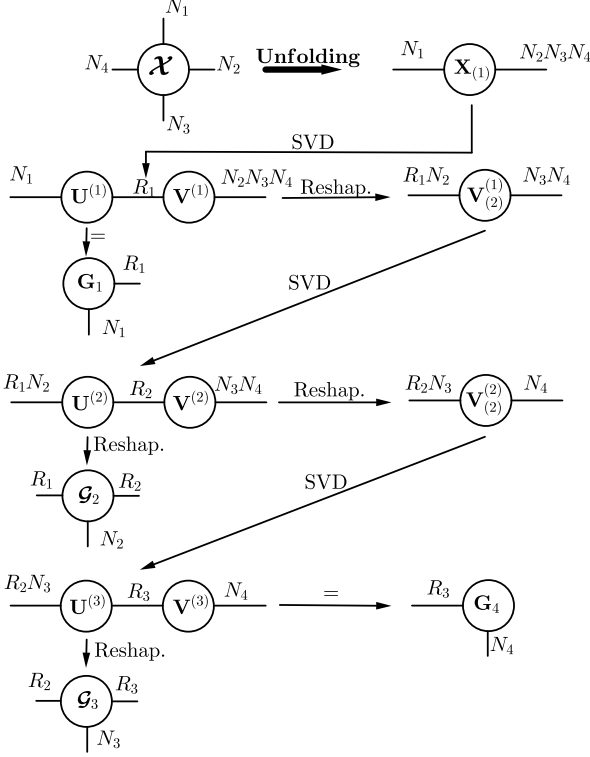


Fig. 2. TT-SVD algorithm applied to a 4-order tensor

**Theorem 2.** Let us consider the CPD-train defined in (4). Applying the TT-SVD algorithm to recover the TT-cores, we obtain the following equations:

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{P}_1 \mathbf{M}_1^{-1}, \\ \mathcal{G}_k &= \mathcal{I}_{3,R} \times_1 \mathbf{M}_{k-1} \times_2 \mathbf{P}_k \times_3 \mathbf{M}_k^{-T}, \quad 2 \leq k \leq Q-1 \\ \mathbf{G}_Q &= \mathbf{M}_{Q-1} \mathbf{P}_Q^T \end{aligned}$$

where  $\mathbf{M}_k \in \mathbb{C}^{R \times R}$  is a nonsingular change-of-basis matrix.

*Proof.* The proof of this theorem is constructive, *i.e.*, it is based on the algebraic structure of the TT-SVD algorithm applied to a  $Q$ -order CPD tensor of a known canonical rank  $R$ . The proof is omitted due to a lack of space.  $\square$

This means that if a  $Q$ -order tensor admits a rank- $R$  CPD with full column rank factors, then its TTM involves a train of  $Q$  3-order CPD(s) having all identical TT-ranks such as  $R_1 = \dots = R_{Q-1} = R$ . The factors can be derived straightforwardly from the TT-cores up to two *change-of-basis* matrices.

**Remark.** Note that each TT-core for  $2 \leq k \leq Q-1$ , follows a CPD linked to its previous TT-core (see Fig. 3). Indeed, the knowledge of the matrix  $\mathbf{M}_{k-1}$  estimated from the TT-core  $\mathcal{G}_{k-1}$  can be exploited for computing the CPD of  $\mathcal{G}_k$ .

The recursion property is exploited in the sequel to propose a new factor retrieval scheme. Note that the involved structure differs from the one of coupled tensors [13].

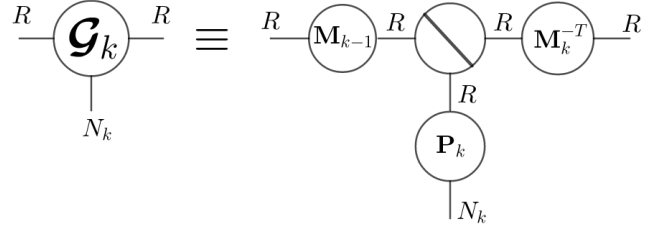


Fig. 3. 3-order CPD of the  $k$ -th TT-core

### III-B. CPD factors retrieval based on a CPD-train

A strategy for CPD factors retrieval based on a CPD-train is derived in this section. Once the dimensionality reduction step is done, the CPD factors can be derived using the model equivalence presented in Theorem 2. More precisely, the results of Theorem 2 allow us to derive the factors retrieval scheme presented in Algorithm 1. In this Algorithm, we

---

**Algorithm 1** Dimensionality reduction and CPD factors Retrieval based on a CPD-train

---

**Input:**  $Q$ -order rank- $R$  tensor  $\mathcal{X}$

**Output:** Estimated CPD factors:  $\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_Q$ .

---

- 1: Dimensionality reduction: (TT-cores estimation)

$$[\mathbf{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{Q-1}, \mathbf{G}_Q] = \text{TT-SVD}(\mathcal{X}, R).$$

- 2: CPD factors retrieval:

$$[\hat{\mathbf{M}}_1, \hat{\mathbf{P}}_2, \hat{\mathbf{M}}_2^{-T}] = \text{Tri-ALS}(\mathcal{G}_2, R).$$

- 3: **for**  $k = 3 \dots Q-1$  **do**
  - 4:  $[\hat{\mathbf{P}}_k, \hat{\mathbf{M}}_k^{-T}] = \text{Bi-ALS}(\mathcal{G}_k, \hat{\mathbf{M}}_{k-1}, R)$
  - 5: **end for**
  - 6:  $\hat{\mathbf{P}}_1 = \mathbf{G}_1 \hat{\mathbf{M}}_1$ , and  $\hat{\mathbf{P}}_Q = \mathbf{G}_Q^T \hat{\mathbf{M}}_{Q-1}^{-T}$
- 

denote by Tri-ALS the ALS algorithm applied to a 3-order tensor, while Bi-ALS denotes the ALS algorithm applied to a 3-order tensor using the knowledge of one factor. The use of the Bi-ALS algorithm is justified relatively to the Remark III-A.

### III-C. Permutation and scaling ambiguities

The factors of the 3-order CPDs are estimated up to trivial ambiguities, *i.e.* up to column permutation and scaling [14].

**Theorem 3.** *The proposed factors retrieval scheme estimates the factors up to*

- 1) *a unique column permutation matrix denoted by  $\mathbf{\Pi}$ ,*
- 2) *diagonal scaling matrices satisfying the following relation:*

$$\mathbf{\Lambda}_1 \mathbf{\Lambda}_2 \mathbf{\Lambda}_3 \cdots \mathbf{\Lambda}_{Q-1} \mathbf{\Lambda}_Q = \mathbf{I}_R \quad (5)$$

where  $\mathbf{\Lambda}_k$  is the scaling ambiguity for the  $k$ -mode factor  $\mathbf{P}_k$ .

*Proof.* Firstly, it is straightforward to check that if a *change-of-basis* matrix  $\mathbf{M}_k^{-T}$  is estimated up to a column permutation matrix  $\mathbf{\Pi}$  and a diagonal scaling matrix  $\mathbf{\Gamma}_k$  then  $\mathbf{M}_k$  is known up to the same column permutation matrix and a diagonal scaling matrix  $\mathbf{\Gamma}_k^{-1}$ . So, we can deduce that the column permutation matrix is unique and common to all the TT-cores due to the recursion property. This proves the first item of the theorem. Secondly, we have

$$\mathbf{G}_1 = \mathbf{P}_1 \mathbf{\Pi} \mathbf{\Lambda}_1 (\mathbf{M}_1 \mathbf{\Pi} \mathbf{\Gamma}_1^{-1})^{-1} \quad (6)$$

$$\mathbf{G}_2 = \mathcal{I}_{3,R} \times_1 \mathbf{M}_1 \mathbf{\Pi} \mathbf{\Gamma}_1^{-1} \times_2 \mathbf{P}_2 \mathbf{\Pi} \mathbf{\Lambda}_2 \times_3 \mathbf{M}_2^{-T} \mathbf{\Pi} \mathbf{\Gamma}_2$$

$$\mathbf{G}_3 = \mathcal{I}_{3,R} \times_1 \mathbf{M}_2 \mathbf{\Pi} \mathbf{\Gamma}_2^{-1} \times_2 \mathbf{P}_3 \mathbf{\Pi} \mathbf{\Lambda}_3 \times_3 \mathbf{M}_3^{-T} \mathbf{\Pi} \mathbf{\Gamma}_3 \quad (7)$$

...

$$\mathbf{G}_{Q-1} = \mathcal{I}_{3,R} \times_1 \mathbf{M}_{Q-2} \mathbf{\Pi} \mathbf{\Gamma}_{Q-2}^{-1} \times_2 \mathbf{P}_{Q-1} \mathbf{\Pi} \mathbf{\Lambda}_{Q-1} \times_3 \mathbf{M}_{Q-1}^{-T} \mathbf{\Pi} \mathbf{\Gamma}_{Q-1}$$

$$\mathbf{G}_Q = \mathbf{M}_{Q-1} \mathbf{\Pi} \mathbf{\Gamma}_{Q-1}^{-1} (\mathbf{P}_Q \mathbf{\Pi} \mathbf{\Lambda}_Q)^T.$$

Based on the above expressions and using the property that the product of scaling matrices is identity [5], we have  $\mathbf{\Gamma}_1^{-1} \mathbf{\Lambda}_2 \mathbf{\Gamma}_2 = \mathbf{I}_R$  from eq. (6), and  $\mathbf{\Gamma}_2 = \mathbf{\Lambda}_3 \mathbf{\Gamma}_3$  from eq. (7). Based on the two above properties, we obtain  $\mathbf{\Gamma}_1^{-1} \mathbf{\Lambda}_2 \mathbf{\Lambda}_3 \mathbf{\Gamma}_3 = \mathbf{I}_R$ . Continuing the same reasoning until the  $k$ -th and  $k+1$ -th TT-cores, we obtain  $\mathbf{\Gamma}_1^{-1} \mathbf{\Lambda}_2 \cdots \mathbf{\Lambda}_k \mathbf{\Gamma}_k = \mathbf{I}_R$ , and  $\mathbf{\Gamma}_k^{-1} \mathbf{\Lambda}_{k+1} \mathbf{\Gamma}_{k+1} = \mathbf{I}_R$ . As  $\mathbf{\Gamma}_1^{-1} = \mathbf{\Lambda}_1$ , and  $\mathbf{\Gamma}_{Q-1} = \mathbf{\Lambda}_Q$ , it is easy to deduce the relation (5).  $\square$

#### IV. COMPUTATIONAL COMPLEXITY

In this section, we compare the computational complexity of the proposed CPD-train algorithm with the one of the ALS algorithm applied to a very large  $Q$ -order tensor. Let  $\mathcal{X}$  be a rank- $R$  CPD tensor of very high order  $Q$  and of size  $N \times \cdots \times N$ . In this comparison, it is assumed that the computational costs of the pseudo-inverse and the SVD are the same. Note that one iteration of the  $Q$ -order ALS algorithm requires the calculation of  $Q$  pseudo-inverses of rank- $R$  matrices of size  $N \times N^{Q-1}$ . Its complexity is  $O(Q \cdot R^2 \cdot N^{Q-1})$ . Meanwhile the application of the TT-SVD algorithm requires the calculation of  $(Q-1)$  truncated SVDs of rank- $R$  matrices of sizes:

- 1st matrix of size  $N \times N^{Q-1}$ , its complexity is  $O(R^2 \cdot N^{Q-1})$

- 2nd matrix of size  $(RN) \times N^{Q-2}$ , its complexity is  $O(R^2 \cdot N^{Q-2})$
- $\vdots$
- $(Q-1)$ th matrix of size  $(RN) \times N$ , its complexity is  $O(R^2 \cdot N)$

For large  $Q$ , the complexity of the TT-SVD algorithm is  $O(R^2 \cdot N^{Q-1})$ , which implies that one iteration of the  $Q$ -order ALS algorithm is  $Q$ -times more complex than the TT-SVD algorithm; consequently the complexity of the ALS algorithm is (number of iterations  $\cdot Q$ )-times that of the TT-SVD algorithm, and we can conclude that:

$$\kappa(Q\text{-order ALS}) \gg \kappa(\text{TT-SVD})$$

On the other hand, one iteration of the 3-order ALS algorithm applied to an  $R \times N \times R$  CPD has a complexity  $O(3 \cdot R^3 \cdot N)$ . As the cost of an iteration of a Bi-ALS algorithm is bounded by the cost of an iteration of a Tri-ALS algorithm, then the total cost of 3-order ALS algorithm is linear *w.r.t.*  $Q$ . From this, we have:

$$\kappa(Q\text{-order ALS}) \gg \kappa(\text{TT-SVD}) \gg Q \cdot \kappa(3\text{-order ALS})$$

From this, it can be seen that the CPD-Train algorithm is much less costly than the ALS one

#### V. SIMULATION RESULTS

In this section, we evaluate the performance of the CPD-train solution through numerical computer simulations. Assuming a noiseless case, we first compare the execution time of the CPD-train algorithm with that of the  $Q$ -order ALS applied to the original tensor. Hypercubic  $Q$ -order tensors of rank  $R = 3$  and dimensions  $N \times \cdots \times N$  following a CPD were generated with random factor matrices whose elements were drawn from a Gaussian distribution with zero mean and unit variance. The rank  $R$  is assumed to be perfectly known. Let  $f(\hat{\mathcal{X}}^{(t)}) = \|\mathcal{X} - \hat{\mathcal{X}}^{(t)}\|_F$ , where  $\hat{\mathcal{X}}^{(t)}$  denotes the estimated tensor at the  $t$ -th iteration. The convergence test was chosen such that  $\frac{|f(\hat{\mathcal{X}}^{(t)}) - f(\hat{\mathcal{X}}^{(t+1)})|}{f(\hat{\mathcal{X}}^{(t)})} < 2 \cdot 10^{-16}$

or when the number of iterations exceeds 1000. The computation time given for the simulations is the average time of 20 realizations. In Table I, we fix  $N = 6$  and vary the tensor order. It is clear that the gain increases with the order. In Table II, we fix  $Q = 8$  and vary the tensor dimensions.

**Table I.** Comparison of the execution time for  $R = 3, N = 6$ .

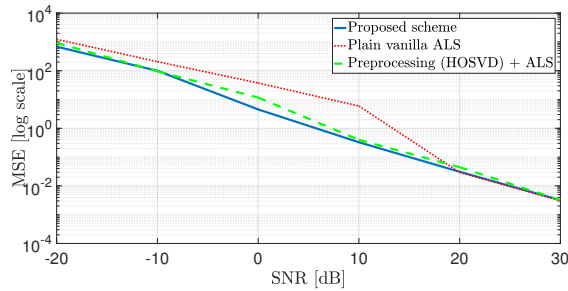
Tensor order	Proposed scheme	ALS	Gain
$Q = 6$	0,63 (s)	7,48 (s)	11,87
$Q = 7$	1,07 (s)	62,57 (s)	58,47
$Q = 8$	1,31 (s)	431,95 (s)	329,73

The conclusion is the same, *i.e.*, the gain in the computation time increases when the dimensions are increased. On the

**Table II.** Comparison of the computation time for  $R = 3, Q = 8$ .

Tensor dimension	Proposed scheme	ALS	Gain
$N = 4$	0, 79 (s)	19, 26 (s)	24, 37
$N = 5$	0, 91 (s)	114, 16 (s)	125, 45
$N = 6$	1, 31 (s)	431, 95 (s)	329, 73

other hand, one can see that the convergence of the ALS is increasingly slow, when the tensor order and dimensions increase. In Fig. 4, the MSE is depicted for the CPD-train algorithm, the ALS algorithm, and the ALS applied after a preprocessing step for noise reduction. The MSEs depicted were obtained by averaging the results over 300 independent Monte Carlo runs, truncated from 5% worst and 5% best MSEs (because the ALS algorithm may not converge in some cases). Note that, due to the use of SVDs, the CPD-



**Fig. 4.** MSE vs SNR in dB with Algo. 1 for  $Q = 8$  with  $N = 3, R = 2, 300$  runs

train algorithm provides a more robust solution than the ALS algorithm for moderate SNR values, with a significant computation time saving.

## VI. CONCLUSION

In this paper, we have shown that a tensor following a  $Q$ -order CPD of rank- $R$  can be expressed in a TT format, and in particular as a train of  $Q$  CPD(s) of rank- $R$ . This result has two main advantages. First, we can exploit the TT model of a  $Q$ -order tensor to perform a dimensionality reduction in the sense that the  $Q$ -order tensor is broken down to  $Q$  at most 3-order tensors. This step allows to replace a difficult alternating optimization in a  $Q$ -dimensional space by  $Q - 1$  alternating optimizations in a 3-dimensional one. In addition, using the recursion property inherent to the TTM, we can go further to consider only a unique alternating optimization into a 3-dimensional space and  $Q - 2$  alternating optimizations into a 2-dimensional one. Finally, the proposed method has a better robustness to noise for a smaller computational cost than the  $Q$ -order ALS.

## VII. REFERENCES

[1] R. Bro. PARAFAC. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38(2):149–171, 1997.

[2] R. Bro, N. D. Sidiropoulos, and G. B. Giannakis. A fast least squares algorithm for separating trilinear mixtures. In *ICA99 - Int. Workshop on Independent Component Analysis and Blind Separation.*, 1999.

[3] A. L. F. de Almeida, G. Favier, and J. C. Mota. PARAFAC-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization. *Signal Processing*, 87(2):337–351, 2007.

[4] C. E.R. Fernandes, G. Favier, and J. C.M. Mota. PARAFAC-based blind identification of convolutive MIMO linear systems. *Proc. of 15th IFAC Symposium on System Identification (SYSID), Saint-Malo, France, July 2009*.

[5] R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

[6] A. Y. Kibangou and G. Favier. Non-iterative solution for PARAFAC with a Toeplitz matrix factor. In *17th European Signal Processing Conference, Glasgow, Scotland, Aug. 2009*.

[7] L. De Lathauwer and J. Castaing. Blind identification of underdetermined mixtures by simultaneous matrix diagonalization. *IEEE Transactions on Signal Processing*, 56:1096–1105, 2008.

[8] N. Li, S. Kindermann, and C. Navasca. Some convergence results on the regularized Alternating Least-Squares method for tensor decomposition. *Linear Algebra and its Applications*, 438(2):796–812, 2013.

[9] I. V. Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, 2011.

[10] N. Petrochilos and P. Comon. Link between the joint diagonalisation of symmetrical cubes and PARAFAC: An application to secondary surveillance radar. In *Fourth IEEE Workshop on Sensor Array and Multichannel Processing.*, 2006.

[11] F. Roemer and M. Haardt. A closed-form solution for Parallel Factor (PARAFAC) analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP.*, 2008.

[12] N.D. Sidiropoulos, G.B. Giannakis, and R. Bro. Blind PARAFAC receivers for DS-CDMA systems. *IEEE Transactions on Signal Processing*, 48(3):810 – 823, 2000.

[13] M. Sørensen and L. De Lathauwer. Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank- $(l_r, n, l_r, n, 1)$  terms—part i: Uniqueness. *SIAM Journal on Matrix Analysis and Applications*, 36(2):496–522, 2015.

[14] M. Sørensen and L. De Lathauwer. New uniqueness conditions for the canonical polyadic decomposition of third-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1381–1403, 2015.

[15] 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 Processing Magazine*, 31:71–79, 2014.

[16] A. Yeredor. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Transactions on Signal Processing*, 50(7):1545–1553, 2002.