



HAL
open science

Learning nonlinearities in the decoupling problem with structured CPD

Yassine Zniyed, Konstantin Usevich, Sebastian Miron, David Brie

► **To cite this version:**

Yassine Zniyed, Konstantin Usevich, Sebastian Miron, David Brie. Learning nonlinearities in the decoupling problem with structured CPD. 19th IFAC Symposium on System Identification, SYSID 2021, Jul 2021, Padova (virtual), Italy. hal-03223831

HAL Id: hal-03223831

<https://hal.science/hal-03223831>

Submitted on 11 May 2021

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.

Learning nonlinearities in the decoupling problem with structured CPD

Yassine Zniyed* Konstantin Usevich* Sebastian Miron*
David Brie*

* CRAN, Université de Lorraine, CNRS, Vandœuvre-lès-Nancy, France
(e-mail: firstname.lastname@univ-lorraine.fr).

Abstract: Decoupling of multivariate functions is an important problem in block-structured system identification. In the literature, different tensor-based solutions have been proposed to solve this problem using a canonical polyadic decomposition (CPD) of a Jacobian tensor. In a recent work, it has been proposed to add polynomial constraint on the model, which leads to a constrained CPD model with two factors depending nonlinearly on each other. In this work, we are interested in the estimation of the static nonlinearities with polynomial constraints. Using the reformulation as a structured CPD, we propose an alternating least squares algorithm with constrained rank-one terms. Numerical simulations show the performance and the usefulness of the proposed method compared to a competing approach.

Keywords: polynomial decoupling, tensor decomposition, canonical polyadic decomposition, structured tensors.

1. INTRODUCTION

In this paper, we focus on the decoupled representation problem [Dreesen et al., 2015b] of multivariate polynomial function. This problem has a great interest in block-structured nonlinear system identification [Schoukens and Tiels, 2017, Schoukens and Rolain, 2012, Schoukens and Ljung, 2019, Karami et al., 2021] and signal processing [Comon et al., 2015, Deville and Duarte, 2015], to mention a few. In this context, the aim is to represent a given multivariate polynomial function as linear transformations of univariate polynomials in linear forms of the inputs. Decoupling such functions is very useful for revealing insights into the mechanisms of polynomial mapping and for reducing the number of parameters.

Several methods use tensor decompositions [Kolda and Bader, 2009, Comon, 2014] to find the decoupled representation. Tensor decompositions are powerful tools from multilinear algebra, which are used in a wide variety of applications, notably in multisensor signal processing [Miron et al., 2020]. The use of tensors presents several advantages over matrices, such as uniqueness [Stegeman and Sidiropoulos, 2007], *i.e.*, the guarantee of identifiability of the retrieved parameters, or even the availability of powerful tools to perform tensor decompositions.

To compute decoupled representation, two classes of tensor-based methods exist. First, the methods [Mulders et al., 2014, Schoukens and Rolain, 2012, Schoukens et al., 2014] that build a tensor from the coefficients of the polynomial map. Second, methods [Dreesen et al., 2015b,a] which consider the first-order derivatives of the coupled nonlinear function. All methods lead to a canonical

polyadic decomposition (CPD) [Harshman, 1970, Hitchcock, 1928] to retrieve the decoupled representation.

While the original methods of [Mulders et al., 2014, Dreesen et al., 2015b] are based on unstructured CPD, in practice we need to take the structure into account for several reasons: first, to reconstruct the static nonlinearities (especially in the noisy case), and second to improve the identifiability of the model. In [Hollander, 2018], the authors propose to parametrize the univariate nonlinear functions of the decoupled model using the polynomial coefficients. This results in introducing a polynomial constraint on the third factor of a 3-order Jacobian tensor. Unfortunately, this may have negative implications since this third factor depends nonlinearly on the second factor of the tensor [Dreesen et al., 2015b].

In this paper, we propose a reformulation of the problem which consists in enforcing the structure on the tensors in the CP decomposition, in the spirit of structured low-rank matrix approximation [Markovsky, 2012, 2008]. For this, we use the correspondence between the polynomial structure and a structural constraint on the rank-one terms, based on results from [Usevich et al., 2020]. Our numerical experiments show that this solution leads to a better convergence of the alternating least squares (ALS) algorithm [Bro et al., 1999] for the CP decomposition.

This work brings the following contributions: (*i*) a novel type of constraints on rank-one tensors resulting in a new algorithm for decoupling multivariate polynomial functions, (*ii*) two new projection methods of rank-one tensors onto sets of rank-one tensors and structure tensors, and (*iii*) an alternating projections estimation scheme to impose the desired constraints. Note that in the structured CP approximation problem we consider, the structure is imposed on tensors and not on the factors as typically done

* This research was supported by the ANR (Agence Nationale de Recherche) grant LeaFleT (ANR-19-CE23-0021).

in the literature. In the context of the decoupling problem, the advantage is that we replace the nonlinear constraint on the factors with a linear constraint on the tensor.

The notations used in this paper are as follows. Scalars, vectors, matrices and tensors are represented by x , \mathbf{x} , \mathbf{X} and \mathcal{X} , respectively. The symbols $(\cdot)^T$, $(\cdot)^\dagger$ and $\text{rank}(\cdot)$ denote, respectively, the transpose, the pseudo-inverse and the rank. The Frobenius norm is defined by $\|\cdot\|_F$. $\mathcal{I}_{3,r}$ denotes the 3-order identity tensor of size $r \times r \times r$. The matrix $\text{unfold}_k \mathcal{X}$ of size $n_k \times n_1 \cdots n_{k-1} n_{k+1} \cdots n_D$ refers to the k -mode unfolding of \mathcal{X} of size $n_1 \times \cdots \times n_D$. The operator $\text{diag}(\cdot)$ forms a diagonal matrix from its vector argument. The operator $\text{vec}(\cdot)$ forms a vector by stacking the columns of its matrix argument. The Kronecker, Khatri-Rao, outer and n -mode products are denoted to by \otimes , \odot , \circ and \times_n .

The rest of the paper is organized as follows. Section 2 formalizes the problem of decoupling multivariate polynomials. In Section 3, we recall the different tensor-based decoupling approaches and the link between them. Section 4 describes the proposed solution. The performance of this latter is evaluated in Section 5. Finally, the conclusions and some perspectives for future research are drawn in Section 6.

2. MULTIVARIATE POLYNOMIAL DECOUPLING

The decoupling problem refers to the decomposition of multivariate real polynomials as linear combinations of univariate polynomials in linear forms of the input variables. In the sequel, the problem is formally expressed. Let $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ be a multivariate polynomial map, with

$$\mathbf{f}(\mathbf{u}) = [f_1(\mathbf{u}) \cdots f_n(\mathbf{u})]^T,$$

and $\mathbf{u} = [u_1 \cdots u_m]^T$.

It is said that \mathbf{f} has a decoupled representation, if we have

$$\mathbf{f}(\mathbf{u}) = \mathbf{W} \mathbf{g}(\mathbf{V}^T \mathbf{u}), \quad (1)$$

or equivalently,

$$\mathbf{f}(\mathbf{u}) = \mathbf{w}_1 \cdot g_1(\mathbf{v}_1^T \mathbf{u}) + \cdots + \mathbf{w}_r \cdot g_r(\mathbf{v}_r^T \mathbf{u}),$$

where $\mathbf{V} \in \mathbb{R}^{m \times r}$, $\mathbf{W} \in \mathbb{R}^{n \times r}$ are transformation matrices, \mathbf{v}_l and \mathbf{w}_l , for $1 \leq l \leq r$, are respectively the columns of \mathbf{V} and \mathbf{W} , and $\mathbf{g} : \mathbb{R}^r \rightarrow \mathbb{R}^n$ follows

$$\mathbf{g}(t_1, \cdots, t_r) = [g_1(t_1) \cdots g_r(t_r)]^T,$$

with $g_l : \mathbb{R} \rightarrow \mathbb{R}$ is a univariate polynomial of degree at most d expressed as:

$$g_l(t) = c_{1,l}t + \cdots + c_{d,l}t^d.$$

Some results on the identifiability of (1) can be found in [Comon et al., 2017], in which the authors show that this decomposition is a special case of the X-rank decomposition [Landsberg, 2012].

3. TENSOR-BASED APPROACHES

There are several tensor-based methods in the literature to find the decoupled representation of a given function. These methods can be classified into two categories. In the first category, there are methods which construct a tensor from the polynomial coefficients [Schoukens and Rolain, 2012, Mulders et al., 2014], while in the second category [Dreesen et al., 2015b], the solution is to build a tensor from the Jacobian matrices of the function evaluated at a set of different sampling points.

3.1 Tensor of unfoldings:

The first approach in [Mulders et al., 2014] proposes to construct a $n \times m \times \delta$ tensor \mathcal{Q} from the coefficients of the polynomial map. Tensor \mathcal{Q} follows a PD and can be expressed as:

$$\mathcal{Q} = \mathcal{I}_{3,r} \times_1 \mathbf{W} \times_2 \mathbf{V} \times_3 \mathbf{Z}, \quad (2)$$

where $\delta = \sum_{k=1}^d m^{k-1}$, and we have

$$\mathbf{z}_l = [c_{1,l} | c_{2,l} \mathbf{v}_l^T | c_{3,l} (\mathbf{v}_l \otimes \mathbf{v}_l)^T | \cdots | c_{d,l} \underbrace{(\mathbf{v}_l \otimes \cdots \otimes \mathbf{v}_l)^T}_{(d-1) \text{ times}}]^T.$$

Based on [Usevich et al., 2020], tensor \mathcal{Q} is a structured tensor with

$$\mathcal{Q}_{i,\dots} = [\psi_i^{(1)} | \psi_i^{(2)} | \text{unfold}_1 \psi_i^{(3)} | \cdots | \text{unfold}_1 \psi_i^{(d)}], \quad (3)$$

for $1 \leq i \leq n$, where $\psi_i^{(k)}$ is a k -order symmetric tensor of size $m \times \cdots \times m$, which can be constructed from the coefficients of the polynomials. It has been proven in [Mulders et al., 2014] that tensor \mathcal{Q} has a CP decomposition from which we can obtain the decoupled representation.

3.2 Tensor of Jacobian matrices:

The second approach was proposed in [Dreesen et al., 2015b]. This approach uses the first-order information of $\mathbf{f}(\mathbf{u})$, and it relies on the observation that

$$\mathbf{J}_{\mathbf{f}}(\mathbf{u}) := \begin{bmatrix} \frac{\partial f_1}{\partial u_1}(\mathbf{u}) & \cdots & \frac{\partial f_1}{\partial u_m}(\mathbf{u}) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1}(\mathbf{u}) & \cdots & \frac{\partial f_n}{\partial u_m}(\mathbf{u}) \end{bmatrix}$$

$$= \mathbf{W} \cdot \text{diag}(g'_1(\mathbf{v}_1^T \mathbf{u}) \cdots g'_r(\mathbf{v}_r^T \mathbf{u})) \cdot \mathbf{V}^T.$$

The $n \times m \times N$ Jacobian tensor \mathcal{J} is then constructed by stacking the Jacobian evaluation at N different sampling points $\mathbf{u}^{(j)} \in \mathbb{R}^m$, for $1 \leq j \leq N$, where

$$\mathcal{J}_{:,j} = \mathbf{J}_{\mathbf{f}}(\mathbf{u}^{(j)}).$$

Therefore, \mathcal{J} admits a CPD expressed as

$$\mathcal{J} = \mathcal{I}_{3,r} \times_1 \mathbf{W} \times_2 \mathbf{V} \times_3 \mathbf{H}, \quad (4)$$

where the columns of $\mathbf{H} \in \mathbb{R}^{N \times r}$ have the following structure

$$\mathbf{h}_l = [g'_l(\mathbf{v}_l^T \mathbf{u}^{(1)}) \cdots g'_l(\mathbf{v}_l^T \mathbf{u}^{(N)})]^T.$$

As with tensor \mathcal{Q} , tensor \mathcal{J} can also be used to extract the decoupled representation of $\mathbf{f}(\mathbf{u})$.

One way to do this is to decompose the Jacobian tensor \mathcal{J} with a polynomial constraint on the columns \mathbf{h}_l as proposed in [Hollander, 2018]. In the latter, the authors have proposed to modify the relaxed ALS algorithm with an intermediate projection of each column \mathbf{h}_l on the column space of a Vandermonde matrix \mathbf{X}_l expressed as

$$\mathbf{X}_l = \begin{bmatrix} 1 & 2\mathbf{v}_l^T \mathbf{u}^{(1)} & 3(\mathbf{v}_l^T \mathbf{u}^{(1)})^2 & \cdots & d(\mathbf{v}_l^T \mathbf{u}^{(1)})^{d-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2\mathbf{v}_l^T \mathbf{u}^{(N)} & 3(\mathbf{v}_l^T \mathbf{u}^{(N)})^2 & \cdots & d(\mathbf{v}_l^T \mathbf{u}^{(N)})^{d-1} \end{bmatrix}$$

The coefficients in [Hollander, 2018] are then estimated as follows.

$$\hat{\mathbf{c}} = \min_{\mathbf{c}} \|\text{vec}(\text{unfold}_3 \mathcal{J}) - \text{vec}(\mathbf{H}(\mathbf{V} \odot \mathbf{W})^T)\|,$$

with $\mathbf{c}_l = [c_{1,l}, \cdots, c_{d,l}]^T$, $\mathbf{H} = [\mathbf{X}_1 \mathbf{c}_1, \cdots, \mathbf{X}_r \mathbf{c}_r]$, and $\mathbf{c} = \text{vec}([\mathbf{c}_1, \cdots, \mathbf{c}_r])$.

3.3 Relation between tensors \mathcal{Q} and \mathcal{J} :

It has been shown in [Usevich et al., 2020] that the CP decompositions in (2) and (4) are related, and we have

$$\mathbf{H} = \mathbf{A}^T \mathbf{Z},$$

or equivalently,

$$\mathcal{J} = \mathcal{Q} \times_3 \mathbf{A}^T,$$

where $\mathbf{A} \in \mathbb{R}^{\delta \times N}$ is a Vandermonde-like matrix whose columns expressed as

$$\mathbf{a}_j = [1 | 2\mathbf{u}^{(j)T} | 3(\mathbf{u}^{(j)} \otimes \mathbf{u}^{(j)})^T | \dots | d \underbrace{(\mathbf{u}^{(j)} \otimes \dots \otimes \mathbf{u}^{(j)})^T}_{(d-1) \text{ times}}]^T.$$

It is worth recalling that if matrix \mathbf{A} has its maximal rank then the three following statements are equivalent [Usevich et al., 2020]:

- (1) the polynomial map $\mathbf{f}(\mathbf{u})$ admits the decomposition (1),
- (2) the tensor $\mathcal{Q}(\mathbf{f})$ admits the structured CPD
$$\mathcal{Q} = \mathcal{Q}_1 + \dots + \mathcal{Q}_r, \quad \text{rank}(\mathcal{Q}_l) = 1, \quad \mathcal{Q}_l \in \mathcal{L}_{\mathcal{Q}},$$
- (3) the tensor $\mathcal{J}(\mathbf{f})$ admits the structured CPD
$$\mathcal{J} = \mathcal{J}_1 + \dots + \mathcal{J}_r, \quad \text{rank}(\mathcal{J}_l) = 1, \quad \mathcal{J}_l \in \mathcal{L}_{\mathcal{J}},$$

where $\mathcal{L}_{\mathcal{Q}} \subset \mathbb{R}^{n \times m \times \delta}$ and $\mathcal{L}_{\mathcal{J}} \subset \mathbb{R}^{n \times m \times N}$ are, respectively, the linear subspaces of tensors with the structure of \mathcal{Q} and \mathcal{J} , $\mathcal{Q}_l = \mathbf{w}_l \circ \mathbf{v}_l \circ \mathbf{z}_l$ and $\mathcal{J}_l = \mathbf{w}_l \circ \mathbf{v}_l \circ \mathbf{h}_l$.

4. PROPOSED SOLUTION

In this section, we rely on the results recalled in Section 3.3 to propose a new algorithm for the Jacobian tensor decomposition, which imposes polynomial constraints on both factors \mathbf{V} and \mathbf{H} simultaneously. In this paper, we propose to decompose the tensor \mathcal{J} while imposing a structural constraint on the rank-one terms of \mathcal{J} , denoted \mathcal{J}_l , instead of imposing it only on the factor \mathbf{H} as in [Hollander, 2018], while two factors are linked. Based on that, the proposed solution tries to solve the following criterion:

$$\min_{\mathbf{W}, \mathbf{V}, \mathbf{H}} \|\mathcal{J} - [\mathbf{W}, \mathbf{V}, \mathbf{H}]\| \quad \text{s.t.} \quad \text{rank}(\mathcal{J}_l) = 1, \quad \mathcal{J}_l \in \mathcal{L}_{\mathcal{J}}. \quad (5)$$

It should be noted that so far the structure of tensors in $\mathcal{L}_{\mathcal{J}}$ has not been defined, contrary to that of tensors in $\mathcal{L}_{\mathcal{Q}}$, which is known and has been given in (3). Consequently, and to facilitate the task, we propose to reformulate the problem (5) into an equivalent problem which is the following

$$\min_{\mathbf{W}, \mathbf{V}, \mathbf{H}} \|\mathcal{J} - [\mathbf{W}, \mathbf{V}, \mathbf{H}]\| \quad \text{s.t.} \quad \text{rank}(\mathcal{J}_l) = 1, \quad \mathcal{Q}_l \in \mathcal{L}_{\mathcal{Q}},$$

while keeping in mind that $\mathcal{Q}_l = \mathcal{J}_l \times_3 (\mathbf{A}^T)^\dagger$.

4.1 Generic solution

To solve the problem (5), we propose an alternating least squares algorithm with constrained rank-one terms, using iterated projections on $\mathcal{L}_{\mathcal{Q}}$ and on the set of rank-one tensors. The idea of this solution is to decompose tensor \mathcal{J} using a relaxed ALS, and then, by iterated projections [Cadzow, 1988] find a rank-one solution that satisfies

$$\mathcal{J}_l = \pi_{\mathcal{L}_{\mathcal{J}} \cap \mathcal{S}_1}(\mathbf{w}_l \circ \mathbf{v}_l \circ \mathbf{h}_l), \quad (6)$$

where $\pi_{\mathcal{V}}$ denotes the projector onto \mathcal{V} , and \mathcal{S}_1 is the set of rank-one tensors. A standard approach to solve (6) is the method of iterated projections, where we first project \mathcal{J}_l onto $\mathcal{L}_{\mathcal{J}}$ and then onto \mathcal{S}_1 , *i.e.*, we compute \mathcal{J}_l such that $\mathcal{J}_l = \pi_{\mathcal{S}_1} \pi_{\mathcal{L}_{\mathcal{J}}}(\mathbf{w}_l \circ \mathbf{v}_l \circ \mathbf{h}_l)$, and re-iterate until convergence. In Algo. 1, we provide the algorithmic description of a generic ALS-based solution to solve the problem (5). This solution is composed of two parts, a first part to find an unconstrained decomposition of the tensor \mathcal{J} , then a second part to ensure that tensors \mathcal{J}_l satisfy (6).

Algorithm 1 Generic Projected ALS (PALS)

Input: Tensor \mathcal{J} of size $n \times m \times N$

Output: Factors \mathbf{W} , \mathbf{V} , \mathbf{H} .

1: Initialize \mathbf{V} , \mathbf{H}

2: **repeat**

3: update \mathbf{W} with $\min_{\mathbf{W}} \left\| \text{unfold}_1 \mathcal{J} - \mathbf{W}(\mathbf{H} \circ \mathbf{V})^T \right\|^2$

4: update \mathbf{V} with $\min_{\mathbf{V}} \left\| \text{unfold}_2 \mathcal{J} - \mathbf{V}(\mathbf{H} \circ \mathbf{W})^T \right\|^2$

5: update \mathbf{H} with $\min_{\mathbf{H}} \left\| \text{unfold}_3 \mathcal{J} - \mathbf{H}(\mathbf{V} \circ \mathbf{W})^T \right\|^2$

6: **for** $l = 1 \dots r$ **do**

7: $\mathcal{J}_l = \pi_{\mathcal{L}_{\mathcal{J}} \cap \mathcal{S}_1}(\mathbf{w}_l \circ \mathbf{v}_l \circ \mathbf{h}_l)$

8: **end for**

9: **until** a convergence test is satisfied or maximum iterations exhausted

Alternatively to Algo. 1, one can avoid updating all components in the same iteration and then project rank-one terms sequentially, by using the Hierarchical ALS (HALS) [Cichocki and Phan, 2009] algorithm. In Algo. 2, we present an HALS-based solution for the problem (5). By using HALS, instead of ALS, one can update only one component per factor matrix and then project the rank-one term before processing the next one. The rank-one terms \mathcal{J}_l used in steps 4-6 of the HALS are computed using $\mathcal{J}_l = \mathcal{J} - \sum_{t \neq l} (\mathbf{w}_t \circ \mathbf{v}_t \circ \mathbf{h}_t)$.

In the next subsections, we will go into more details on the method of projections and on the details of the algorithm PALS as a whole.

4.2 Projection of rank-one tensors

Projection onto $\mathcal{L}_{\mathcal{Q}}$: Before presenting the details of the PALS algorithm which solves the problem (5), we present in Algo. 3 a pseudo-code of a solution which allows to project tensors \mathcal{Q}_l onto subspace $\mathcal{L}_{\mathcal{Q}}$ (or equivalently \mathcal{J}_l onto subspace $\mathcal{L}_{\mathcal{J}}$). Based on the structure presented in (3), the presented solution, referred to as rank-one tensor rectification (R₁R), allows to rectify the horizontal slices of \mathcal{Q}_l , by imposing symmetry on the d blocks of matrices $\mathcal{Q}_l(i, :, :)$ as described in Algo. 3. In this algorithm, and in order to match the structure (3), each block of the horizontal slices is reshaped to a hypercubic tensor, which is transformed to a symmetric tensor. Here, \mathcal{G}_k denotes the group of all possible permutations of $\{1, \dots, k\}$.

Algorithm 2 Generic Projected HALS (PHALS)

Input: Tensor \mathcal{J} of size $n \times m \times N$ **Output:** Factors \mathbf{W} , \mathbf{V} , \mathbf{H} .

- 1: Initialize \mathbf{W} , \mathbf{V} , \mathbf{H}
- 2: **repeat**
- 3: **for** $l = 1 \dots r$ **do**

- 4: update \mathbf{w}_l with $\min_{\mathbf{w}_l} \left\| \text{unfold}_1 \mathcal{J}_l - \mathbf{w}_l (\mathbf{h}_l \odot \mathbf{v}_l)^T \right\|^2$

- 5: update \mathbf{v}_l with $\min_{\mathbf{v}_l} \left\| \text{unfold}_2 \mathcal{J}_l - \mathbf{v}_l (\mathbf{h}_l \odot \mathbf{w}_l)^T \right\|^2$

- 6: update \mathbf{h}_l with $\min_{\mathbf{h}_l} \left\| \text{unfold}_3 \mathcal{J}_l - \mathbf{h}_l (\mathbf{v}_l \odot \mathbf{w}_l)^T \right\|^2$

- 7: $\mathcal{J}_l = \pi_{\mathcal{L}_{\mathcal{J}} \cap \mathcal{S}_1}(\mathbf{w}_l \circ \mathbf{v}_l \circ \mathbf{h}_l)$

- 8: **end for**

- 9: **until** a convergence test is satisfied or maximum iterations exhausted

Algorithm 3 Rank-one tensor rectification ($\mathbf{R}_1\mathbf{R}$)

Input: Unstructured rank-one tensor $\mathcal{A} = \mathbf{w} \circ \mathbf{v} \circ \mathbf{z}$ of size $n \times m \times \delta$ **Output:** $\mathcal{B} \in \mathcal{L}_{\mathcal{Q}}$ of size $n \times m \times \delta$

- 1: **for** $i = 1 \dots n$ **do**

- 2: $\mathbf{F} = \mathcal{A}(i, :, :)$

- 3: $\delta' = 1$

- 4: **for** $k = 1 \dots d - 1$ **do**

- 5: $\mathcal{T} = \text{reshape}(\mathbf{F}(:, \delta' + 1 : \delta' + m^k), \underbrace{[m, \dots, m]}_{(k+1) \text{ times}})$

- 6: $\mathcal{T} = \frac{1}{(k+1)!} \sum_{\sigma \in \mathcal{G}_{k+1}} \sigma(\mathcal{T})$

- 7: $\mathbf{F}(:, \delta' + 1 : \delta' + m^k) = \text{unfold}_1 \mathcal{T}$

- 8: $\delta' \leftarrow \delta' + m^k$

- 9: **end for**

- 10: $\mathcal{B}(i, :, :) = \mathbf{F}$

- 11: **end for**

Remark 1. One should be cautious that, if $\mathcal{B} = \mathbf{R}_1\mathbf{R}(\mathcal{A})$, then

$$\text{rank}(\mathcal{A}) = 1 \not\Rightarrow \text{rank}(\mathcal{B}) = 1. \quad (7)$$

Projection onto \mathcal{S}_1 : As mentioned in Remark 1, projecting rank-one tensors \mathcal{Q}_l onto subspace $\mathcal{L}_{\mathcal{Q}}$ (or equivalently \mathcal{J}_l onto $\mathcal{L}_{\mathcal{J}}$) will cause a change of rank, *i.e.*, the projected tensor will no longer be in \mathcal{S}_1 . Another projection onto \mathcal{S}_1 is therefore necessary to ensure (6). It is worth noting that when $\mathbf{R}_1\mathbf{R}$ is applied, the rectification is applied on the horizontal slices of the tensors, which only rectify the structure of the second and third modes factors. Formally, this means that

$$\pi_{\mathcal{L}_{\mathcal{Q}}}(\mathbf{w}_l \circ \mathbf{v}_l \circ \mathbf{z}_l) = \mathbf{w}_l \circ \pi_{\mathcal{L}_{\mathcal{Q}}}(\mathbf{v}_l \circ \mathbf{z}_l),$$

where $\mathcal{L}_{\mathcal{Q}}$ is the linear subspace of matrices having the same structure as the slice matrices in (3). Thus, the mode-1 factors \mathbf{w}_l remain unchanged after this rectification. Based on that and to find a rank-one decomposition of the projected tensor, it will therefore be sufficient to find a rank-one approximation of matrix $\pi_{\mathcal{L}_{\mathcal{Q}}}(\mathbf{v}_l \circ \mathbf{z}_l)$, using the SVD, instead of finding a rank-one approximation of the whole tensor using the ALS for instance.

4.3 Solution with iterated projections

In this part, we present Algo. 4, which details the solution of iterated projections. This solution is to be used in line 7 in algorithms PALS and PHALS to ensure that $\mathcal{J}_l = \pi_{\mathcal{L}_{\mathcal{J}} \cap \mathcal{S}_1}(\mathbf{w}_l \circ \mathbf{v}_l \circ \mathbf{h}_l)$, or equivalently $\mathcal{Q}_l = \pi_{\mathcal{L}_{\mathcal{Q}} \cap \mathcal{S}_1}(\mathbf{w}_l \circ \mathbf{v}_l \circ \mathbf{z}_l)$. In this algorithm, we project the estimated rank-one tensors onto the desired subspaces, while keeping in mind that if we project \mathcal{Q}_l onto $\mathcal{L}_{\mathcal{Q}}$, this does not mean that the resulting \mathcal{Q}_l is a rank-one tensor as explained in (7), and vice versa, *i.e.*, if \mathcal{Q}_l is a rank-one tensor, this does not mean that $\mathcal{Q}_l \in \mathcal{L}_{\mathcal{Q}}$. To solve this issue several iterations are needed to find a solution \mathcal{Q}_l that satisfies both $\mathcal{Q}_l \in \mathcal{L}_{\mathcal{Q}}$ and $\mathcal{Q}_l \in \mathcal{S}_1$. The idea of this solution is to make an indirect projection of \mathcal{J}_l onto $\mathcal{L}_{\mathcal{J}}$ and \mathcal{S}_1 . To do this, we use the tensor \mathcal{Q}_l , according to the relation $\mathcal{Q}_l = \mathcal{J}_l \times_3 (\mathbf{A}^T)^\dagger$, we project \mathcal{Q}_l onto the desired subspaces and then go back to \mathcal{J}_l . Using this

Algorithm 4 Iterated projections solution

Input: Unstructured factors \mathbf{w}_l , \mathbf{v}_l and \mathbf{h}_l , matrix \mathbf{A} of size $\delta \times N$ as defined in 3.3**Output:** Structured factors \mathbf{v}_l and \mathbf{h}_l .

- 1: $\mathcal{Q}_l = [|\mathbf{w}_l, \mathbf{v}_l, (\mathbf{A}^T)^\dagger \mathbf{h}_l|]$

- 2: **repeat** \triangleright Projection of rank-one tensors \mathcal{Q}_l onto $\mathcal{L}_{\mathcal{Q}}$ and \mathcal{S}_1 recursively

- 3: $\mathcal{Q}_l = \mathbf{R}_1\mathbf{R}(\mathcal{Q}_l)$

- 4: $[\mathbf{v}_l, \mathbf{z}_l] = \text{SVD}(\text{unvec}_{m \times \delta}((\mathbf{w}_l^\dagger \cdot \text{unfold}_1 \mathcal{Q}_l)^T), 1)$

- 5: **until** a convergence test is satisfied or maximum iterations exhausted

- 6: $\mathbf{h}_l = \mathbf{A}^T \mathbf{z}_l$

solution in line 7 of the PALS algorithm, splits it in two parts. The first one consists in computing a relaxed CP decomposition with r components that best approximates \mathcal{J} , and the second part consists in projection steps which perform a rectification of rank-one terms toward a desired structure. A similar solution can be used for an HALS-based algorithm, referred to as PHALS. The interest of this type of approach is to impose the desired constraints jointly and simultaneously on both factors; which has positive implications on the convergence of the algorithm as will be shown in the simulations.

5. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed solution by means of numerical computer simulations. First, we generate a decoupled multivariate polynomial function as in (1). The transformation matrices \mathbf{V} and \mathbf{W} were randomly generated, with elements drawn from a uniform distribution. Let $f(\hat{\mathcal{J}}^{(t)}) = \|\mathcal{J} - \hat{\mathcal{J}}^{(t)}\|_F$, where $\hat{\mathcal{J}}^{(t)}$ denotes the estimated tensor at the t -th iteration. We say that an algorithm has converged when $\frac{|f(\hat{\mathcal{J}}^{(t)}) - f(\hat{\mathcal{J}}^{(t+1)})|}{f(\hat{\mathcal{J}}^{(t)})} < \epsilon$ or when the number of iterations exceeds 1000. In the following experiment, we fix the order $d = 3$, and vary the parameters m , n and r . Table 1 gives the number of ill-converging experiments over 1000 runs. In these experiments, we initialized all algorithms in a

random way, using the same initialization, and we counted the number of ill-converging experiments.

One may remark that the number of ill-converging experiments for the state-of-the-art solution is much higher compared to the proposed PALS algorithm in the various considered configurations. As explained before, this is due to the fact that the algorithm proposed in [Hollander, 2018] does not take into account the nonlinear dependence between \mathbf{V} and \mathbf{H} , meanwhile the proposed one does. In

Table 1. Number of ill-converging experiments with $d = 3$ and $N = 100$.

(m, n, r)	[Hollander, 2018]	PALS	PHALS
(2, 2, 3)	126	10	52
(3, 3, 3)	288	3	6
(4, 4, 3)	151	10	14
(4, 4, 4)	213	15	197
(5, 5, 5)	85	6	102

Figure 1, we plot $f(\mathcal{J}^{(t)})$, which corresponds to the MSE, at each iteration t . For the plotted curves, all algorithms have converged. It is worth noting that, although one iteration of PALS or PHALS can be a little more expensive in terms of computational complexity than one of [Hollander, 2018], PALS still has a faster convergence as shown in this figure. Similar behavior was found in the majority of the experiments that we ran. This means that PALS requires less iterations to converge compared to [Hollander, 2018]. We have also noticed that PHALS has a similar rate of convergence to PALS in the majority of cases, but it fails much more often than PALS as shown in Table 1.

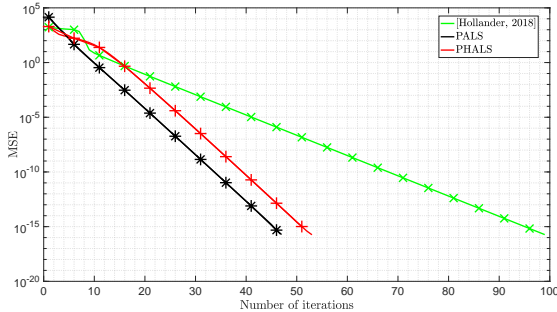


Fig. 1. MSE vs iterations.

In the second experiment, we simulate a multivariate polynomial function which does not follow the representation in (1). The coefficients of the function are generated randomly. The parameters for this function are $d = 3$, $m = 2$ and $n = 2$. The NMSE is depicted in Figure 2 with respect to rank r . The NMSE is defined as $\text{NMSE} = \frac{\|\mathcal{J} - \hat{\mathcal{J}}\|^2}{\|\mathcal{J}\|^2}$. We note that the NMSE is high for low ranks, $r \leq 3$, and then we have a drop in the NMSE for a certain value of the rank, namely $r = 4$ in this case. This shows that the proposed method, from a certain value of the rank, works as well for a polynomial function which follows (1) as for a function which does not necessarily follow it.

In the last experiment, we consider a more realistic case, where we simulate a decoupled multivariate polynomial as in (1) with parameters $d = 3$, $m = 2$, $n = 2$ and $r = 3$, to which we add a 2-D circular sine perturbation. Figure 3 represents the second output of the noisy function and

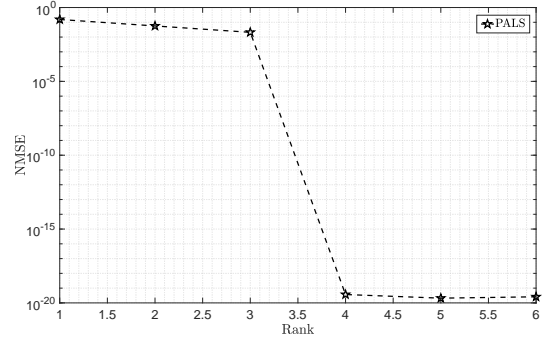


Fig. 2. NMSE vs Rank.

its approximation using the proposed PALS algorithm. The residual of the approximation is given in Figure 4.

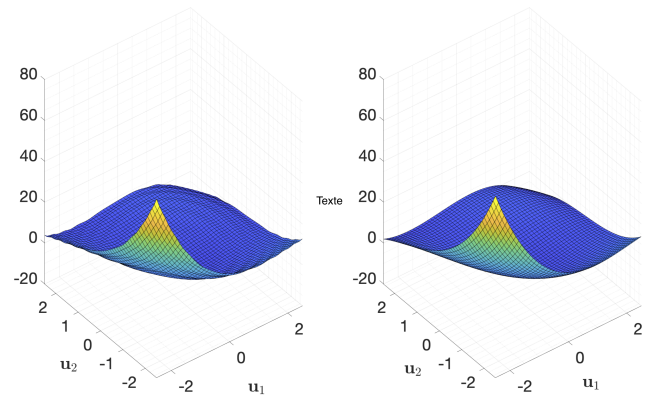


Fig. 3. Left: the original function $f_2(u_1, u_2)$ with noise - Right: the approximation of $f_2(u_1, u_2)$ using PALS with $r = 3$.

In this figure, the residual has a circular sine shape, which corresponds to the simulated perturbation.

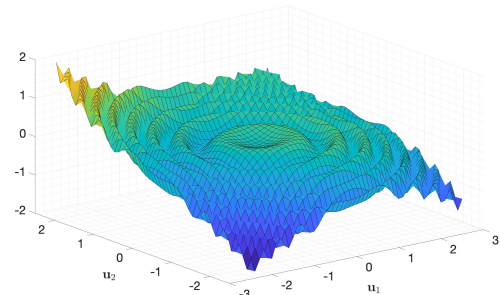


Fig. 4. The residual of the approximation.

We give below the normalized factor matrices of the original noise-free function and those of the approximation of the noisy function with the trivial CPD ambiguities.

$$\mathbf{V} = \begin{bmatrix} 0.8944 & 0.5547 & 0 \\ 0.4472 & -0.8321 & -1 \end{bmatrix}, \hat{\mathbf{V}} = \begin{bmatrix} 0.9017 & -0.0226 & -0.5780 \\ 0.4323 & -0.9997 & 0.8160 \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} 0 & 1 & -0.8944 \\ -1 & 0 & 0.4472 \end{bmatrix}, \hat{\mathbf{W}} = \begin{bmatrix} -0.0119 & -0.8664 & -0.9985 \\ 0.9999 & 0.4993 & -0.0549 \end{bmatrix}$$

From this example, it can be shown that the proposed solution is able to correctly estimate the factors \mathbf{V} and \mathbf{W} , up to scaling and permutation ambiguities, which show

the effectiveness of the proposed solution for both, noisy, as well as for non-noisy functions.

6. CONCLUSION

A new structured CPD based algorithm has been proposed for decoupling multivariate polynomial functions. This approach allows to simultaneously consider a nonlinear dependence between two factors of a structured CPD tensor. The proposed solution constraints the structure of rank-one tensors, instead of constraining the factor matrices. This type of constraint is new in the literature, and allows to impose structures on two factors of the tensor jointly. The effectiveness of the proposed solution is demonstrated by means of numerical simulations, in which we showed that this method exhibits a better convergence compared to the state-of-the-art. Also, we have shown that the proposed solution has a faster convergence and works well for non decoupled multivariate polynomial functions. Possible applications of this work include the estimation of nonlinearities in machine learning problems.

REFERENCES

- R. Bro, N. D. Sidiropoulos, and G. B. Giannakis. A fast least squares algorithm for separating trilinear mixtures. In *Int. Workshop on Independent Component Analysis and Blind Separation*, 1999.
- J. A. Cadzow. Signal enhancement—a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(1):49–62, 1988.
- A. Cichocki and A. H. Phan. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, 92-A(3):708–721, 2009.
- P. Comon. Tensors : A brief introduction. *IEEE Signal Processing Magazine*, 31(3):44–53, May 2014.
- Pierre Comon, Yang Qi, and Konstantin Usevich. A polynomial formulation for joint decomposition of symmetric tensors of different orders. In *A Polynomial Formulation for Joint Decomposition of Symmetric Tensors of Different Orders*, 2015.
- Pierre Comon, Yang Qi, and Konstantin Usevich. Identifiability of an X-rank decomposition of polynomial maps. *SIAM Journal on Applied Algebra and Geometry*, 1:388–414, 2017.
- Yannick Deville and Leonardo Tomazeli Duarte. An overview of blind source separation methods for linear-quadratic and post-nonlinear mixtures. In *International Conference on Latent Variable Analysis and Signal Separation*, 2015.
- P. Dreesen, M. Schoukens, K. Tiels, and J. Schoukens. Decoupling static nonlinearities in a parallel Wiener-Hammerstein system: A first-order approach. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 2015a.
- Philippe Dreesen, Mariya Ishteva, and Johan Schoukens. Decoupling multivariate polynomials using first-order information and tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 36(12):864–879, 2015b.
- R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an explanatory multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.
- F. L. Hitchcock. Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics*, 7(1-4):39–79, 1928.
- Gabriel Hollander. *Multivariate polynomial decoupling in nonlinear system identification*. PhD thesis, Vrije Universiteit Brussel, 2018.
- Kiana Karami, David Westwick, and Johan Schoukens. Applying polynomial decoupling methods to the polynomial narx model. *Mechanical Systems and Signal Processing*, 148:107134, 2021.
- T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51:455–500, 2009.
- J. M. Landsberg. *Tensors: Geomtry and applications*, volume 128. American mathematical soc., 2012.
- I. Markovsky. Structured low-rank approximation and its applications. *Automatica*, 44(4):891–909, 2008.
- I. Markovsky. *Low Rank Approximation: Algorithms, Implementation, Applications*. Communications and Control Engineering. Springer, 2012.
- Sebastian Miron, Yassine Zniyed, Remy Boyer, A. L.F. de Almeida, G. Favier, D. Brie, and P. Comon. Tensor methods for multisensor signal processing. *to appear in IET signal processing*, 2020.
- A. Van Mulders, L. Vanbeylen, and K. Usevich. Identification of a block-structured model with several sources of nonlinearity. In *European Control Conference (ECC)*, Strasbourg, France, 2014.
- J. Schoukens and L. Ljung. Nonlinear system identification: A user-oriented road map. *IEEE Control Systems Magazine*, 39(6):28–99, 2019.
- M. Schoukens and Y. Rolain. Cross-term elimination in parallel Wiener systems using a linear input transformation. *IEEE Transactions on Instrumentation and Measurement*, 61:845–847, 2012.
- M. Schoukens and K. Tiels. Identification of block-oriented nonlinear systems starting from linear approximations: A survey. *Automatica*, 85:272–292, 2017.
- M. Schoukens, K. Tiels, M. Ishteva, and J. Schoukens. Identification of parallel Wiener-Hammerstein systems with a decoupled static nonlinearity. *IFAC Proceedings volumes*, 47:505–510, 2014.
- A. Stegeman and N. D. Sidiropoulos. On Kruskal’s uniqueness condition for the candecomp/parafac decomposition. *Linear Algebra and its Applications*, 420(2):540 – 552, 2007.
- K. Usevich, P. Dreesen, and M. Ishteva. Decoupling multivariate polynomials: interconnections between tensorizations. *Journal of Computational and Applied Mathematics*, 363:22–24, 2020.