Block-Term Tensor Decomposition in Array Signal Processing

Eleftherios Kofidis

Block-Term Tensor Decomposition in Array Signal Processing

Eleftherios Kofidis[a,b]

[a] *Department of Statistics and Insurance Science, University of Piraeus, Piraeus, Greece*
[b] *Computer Technology Institute & Press "Diophantus" (CTI), Patras, Greece*

Abstract

Tensor models and methods have attracted significant interest also in array signal processing (ASP) given the inherently multi-dimensional nature of the problems involved therein. The ability of tensor decomposition (TD)-based methods to recover latent information, including channel parameters and transmitted symbols in the ASP context, in a deterministic manner and with little or no training overhead, fits well with the data efficiency needs of future-generation multi-user massive multiple-input multiple-output systems. Such methods have mostly relied, however, on a few classical TD models, notably the canonical polyadic decomposition (CPD). In this paper, a generalized CPD model, known as block-term decomposition (BTD), which has been increasingly adopted in a wide range of applications, is re-visited in the context of ASP, and shown to be the natural choice in sensor arrays of increased dimensionality that also involve channel multipath. Semi-blind joint channel estimation/data detection (JCD) is addressed in this context via efficient algorithms that wed existing JCD schemes with BTD approximation. Recursive and robust to sensor failures versions are also developed.

Email address: kofidis@unipi.gr (Eleftherios Kofidis)

The special yet important case of the uniform rectangular array (URA) configuration is adopted to illustrate the ideas and results. The signal detection performance of the BTD-inspired semi-blind JCD schemes, at various noise levels, array sizes, and numbers of users and paths, is evaluated with the aid of simulations and seen to be favorably compared with that of the training-only-based solution.

Keywords: array signal processing (ASP), BTD, CPD, joint channel estimation/data detection (JCD), MIMO, tensor, uniform rectangular array (URA)

---

1. Introduction

Though an area with an already long history [1], array signal processing (ASP) has kept growing and extending beyond the classical one-dimensional (1-D) sensor arrays, while targeting additional modern problems, including those involved in future-generation multi-user massive multiple-input multiple-output (MU-mMIMO) systems [2]. One of the principal requirements in the design and operation of the latter is that of resource-efficient channel estimation, which includes the restriction of training information to a limited amount [3]. Limiting the training overhead would facilitate the operation of large-scale MIMO networks, mitigating the pilot contamination problem [4] and relaxing the associated base station (BS)/mobile station (MS) coordination requirements. Moreover, it would save bandwidth for information transmission and more easily accommodate high-mobility scenarios that would otherwise require frequent re-training. It is for these reasons that (semi-)blind methods [5] are re-surging in this context and the problem

of *joint* channel estimation/data detection (JCD) [6] regains interest in the ASP community.

Tensor models and methods [7] provide a suitable means towards this goal [8] given the inherently multi-dimensional (m-D) nature of these systems and the ability of tensor decomposition (TD)-based methods to recover latent information, in a deterministic (and hence data-efficient and free of statistical assumptions) manner and with little or no training overhead. Originated in [9], tensor methods for ASP have developed into a large volume of tools (see [10] for a recent overview), which rely on various TD models for estimating channel parameters (path gains, delays, and angles) and transmitted (Tx) symbols. These include primarily (forms of) the canonical polyadic decomposition (CPD) model, among others, such as the Tucker (TKD) and the tensor train (TT) decomposition [10]. ASP with possibly impaired sensors has also been recently treated as a problem of completing a tensor with missing fibers [11], a problem previously studied for the CPD model in [12] and potentially also useful for sampling large-scale arrays to bring the data complexity down.

The generalized CPD model, known as block-term decomposition (BTD) [13], has seen comparatively fewer applications in ASP, and this is despite its otherwise wide and increasing application range (see [14] for a relatively recent account). BTD decomposes a tensor into a sum of low multilinear rank terms (referred to as blocks) and can be seen as an intermediate between CPD (sum of rank-1 terms) and TKD (only one term of low multilinear rank). This is what gives BTD its flexibility and justifies its wide applicability. Its most commonly appearing variant comprises rank-$(L, L, 1)$ block terms and

is hence referred to as the LL1 model [13]. LL1 can also be viewed as a hierarchical CPD [15].

In [16] and earlier related works including [17, 18, 19], BTD is shown to be the natural modeling choice for the output of an array with signals impinging through multiple paths *and* being subject to oversampling and/or intersymbol interference (ISI)[1]. Rank-$(M, N, \cdot)$ decomposition [13], also known as type-2 BTD (i.e., with Tucker-2 blocks), and coupled LL1 decomposition were used in [20] and [21] to perform target localization in multi-pulse, multi-array MIMO radar, with and without training waveforms, respectively. [21] was recently extended to near-field targets in [22]. The direction-finding (DF) problem in L-shaped electromagnetic vector sensor arrays with time-varying polarization state (i.e., partial or incomplete polarization) was also solved with the aid of type-2 BTD modeling in [23, 24].

The array geometry was not taken into account in [16] and related works and that is why oversampling and/or ISI were necessary to result in an identifiable BTD representation. Instead, [25] develops an LL1-based DF solution by exploiting multipath and the invariances found in a uniform linear array (ULA). With a three-fold tensorization of the spatial mode, DF reduces to a *deterministic* 3rd-order CPD problem per source, whose essential uniqueness allows it to outperform subspace-based direction of arrival (DoA) estimation methods [26] for small array and/or data sizes.

Compared with the ULA, a 2-D array configuration, such as the uni-

[1]That is, to local reflections corresponding to multiple directions of arrival per path and hence to ISI. Oversampling may be there because of a direct sequence-code-division multiple-access (DS-CDMA) transmission.

form rectangular array (URA) [27], reduces the deployment area required in a mMIMO system and offers increased versatility, higher spatial resolution, and better channel estimation performance in both pilot-based and semi-blind transmission scenarios [28]. In this paper, JCD with the (BS) receiver equipped with a URA is studied, to illustrate that an LL1 representation *naturally* results in such a setting, for user signals arriving through multiple paths, not necessarily undergoing ISI or oversampling. Iterative algorithms that wed existing JCD schemes with BTD approximation are then developed. It is demonstrated that considering the channel's low-rankness (which amounts to respecting the underlying LL1 model) pays off in signal copying performance. Recursive and robust to sensor failures versions of the best-performing algorithm are also developed. *Short* training is employed to initialize the iterations, which justifies the semi-blind characterization. This is in contrast to the multiple random [17] or algebraic [29] initialization schemes often employed in the BTD literature, which increase the computational load and/or are restricted to high signal-to-noise ratio (SNR) environments only. The proposed procedures are simple, and computationally- and data-efficient. Their signal detection performance, at various noise levels, array sizes, and numbers of users and paths, is evaluated with the aid of simulations and is seen to be favorably compared with that of the training-only-based solution.

The rest of the paper is organized as follows. The notation employed is defined in the next subsection. Section 2 describes the system model and casts the JCD problem in the considered setting, motivating the LL1 approach. Various possibilities for LL1-based JCD in a URA are developed and discussed in Section 3. The best-performing of these schemes is further

developed in that section to yield recursive and robust to sensor failures versions. Section 4 reports and discusses the simulation results. Conclusions are drawn in Section 5, where pointers to future related research are also given.

1.1. Notation

Vectors and matrices are denoted by bold lower- and upper-case letters, respectively. Bold calligraphic letters are used to designate higher-order tensors. For a tensor $\boldsymbol{\mathcal{A}}$, $\mathbf{A}_{(n)}$ denotes its mode-n unfolding. The superscripts $^{\mathrm{T}}$, *, and $^{\mathrm{H}}$ stand for transposition, complex conjugation, and Hermitian transposition, respectively. The symbols $\circ$, $\diamond_{\mathrm{c}}$, and $\diamond$ respectively denote the outer, the column-wise, and the partition-wise Khatri-Rao products. $\mathrm{Diag}(\mathbf{x})$ is the diagonal matrix with the vector $\mathbf{x}$ on its main diagonal. The block diagonal matrix with diagonal blocks $\mathbf{A}_1, \mathbf{A}_2, \ldots$ is denoted by $\mathrm{blockdiag}(\mathbf{A}_1, \mathbf{A}_2, \ldots)$. $\mathrm{vec}(\mathbf{A})$ stacks the columns of the matrix $\mathbf{A}$ on top of each other in a column vector. The Frobenius norm is denoted by $\|\cdot\|_{\mathrm{F}}$. $\mathbf{I}_N$ is the identity matrix of order N and $\mathbf{1}_N$ stands for the all-ones $N \times 1$ vector. The field of complex numbers is denoted by $\mathbb{C}$ and $\jmath \triangleq \sqrt{-1}$ stands for the imaginary unit.

2. Problem Statement and Solution Approach

Let R far-field sources emit narrowband signals, $s_r(n)$, $r = 1, 2, \ldots, R$, which impinge on a x-y planar URA, of dimensions $N_x \times N_y$. The rth signal arrives through L_r paths that are assumed to be sufficiently closely spaced compared to the symbol duration that they are all of the same delay. The

corresponding $N_x \times N_y$ channel coefficient can then be expressed as [27]

$$\mathbf{H}_r = \sum_{l=0}^{L_r-1} \beta_{r,l} \mathbf{a}_x(\phi_{r,l}, \theta_{r,l}) \mathbf{a}_y(\phi_{r,l}, \theta_{r,l})^{\mathrm{T}}, \tag{1}$$

where $\beta_{r,l}$ is the complex gain of the lth path, incorporating the attenuation coefficient and the initial phase, and, with $\phi_{r,l}$, $\theta_{r,l}$ denoting the corresponding azimuth and zenith angles of arrival (AoA), respectively,

$$\mathbf{a}_x(\phi_{r,l}, \theta_{r,l}) = \left[\begin{array}{cccc} 1 & e^{\jmath \xi_{x,r,l}} & \ldots & e^{\jmath (N_x-1) \xi_{x,r,l}} \end{array} \right]^{\mathrm{T}} \tag{2}$$

is the steering vector in the x-direction, and the y-direction steering vector $\mathbf{a}_y$ is similarly defined. With $k = \frac{2\pi}{\lambda}$ denoting the wave number, with all transmitters being assumed to employ the same carrier frequency (co-channel signals), of wavelength λ, the ξ's are given by

$$\xi_{x,r,l} = k d_x \cos\phi_{r,l} \sin\theta_{r,l}, \tag{3}$$

$$\xi_{y,r,l} = k d_y \sin\phi_{r,l} \sin\theta_{r,l}, \tag{4}$$

where d_x, d_y are the inter-antenna spacings in the x- and y-directions, respectively. Figure 1 illustrates such a scenario, for a single path of a single source signal.

If the impinging signals, $\mathbf{s}_r \in \mathbb{C}^{N_{\mathrm{s}} \times 1}$, $r = 1, 2, \ldots, R$, are of length N_{s}, then the array output can be represented by an $N_x \times N_y \times N_{\mathrm{s}}$ tensor, expressed as[2]

$$\boldsymbol{\mathcal{Y}} = \sum_{r=1}^{R} \mathbf{H}_r \circ \mathbf{s}_r + \boldsymbol{\mathcal{W}}, \tag{5}$$

[2]This LL1 model is (only) alluded to in [30, Eq. (47)], where a spectral analytic DoA estimation approach is taken, based on the use of orthogonal training sequences to split the problem into single-user ones and Fourier transforming each of the corresponding 2-D received signals.

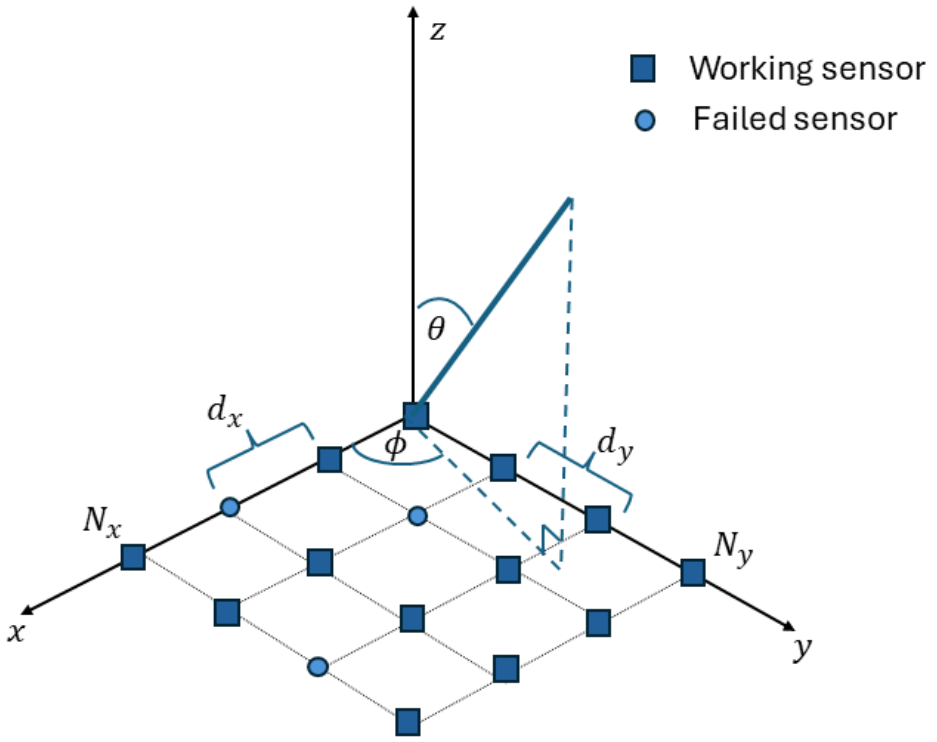


Figure 1: Uniform rectangular array (of size 4×4 in this example). A source signal is depicted, with its corresponding azimuth and zenith angles.

where $\boldsymbol{\mathcal{W}}$ stands for the zero-mean Gaussian noise, assumed spatially and temporally white, and

$$\mathbf{H}_r = \mathbf{A}_{x,r}\mathbf{D}_r\mathbf{A}_{y,r}^{\mathrm{T}} \tag{6}$$

with $\mathbf{A}_{x,r} = \left[\begin{array}{ccc} \mathbf{a}_x(\phi_{r,1},\theta_{r,1}) & \cdots & \mathbf{a}_x(\phi_{r,L_r},\theta_{r,L_r}) \end{array}\right]$ and the similarly defined $\mathbf{A}_{y,r}$ being the $N_x \times L_r$ and $N_y \times L_r$ steering matrices, respectively, for the rth signal. The corresponding path gains, $\boldsymbol{\beta}_r = \left[\begin{array}{cccc} \beta_{r,1} & \beta_{r,2} & \cdots & \beta_{r,L_r} \end{array}\right]^{\mathrm{T}}$, are on the main diagonal of the diagonal matrix $\mathbf{D}_r = \mathrm{Diag}(\boldsymbol{\beta}_r)$. It is clear that, especially for large-scale arrays, (6) is, generically, of full column rank, L_r, which implies that (5) is a noisy decomposition into rank-$(L_r, L_r, 1)$ terms. This in turn implies that one may resort to any of the existing LL1 approximation methods to *blindly* estimate from $\boldsymbol{\mathcal{Y}}$ the channel parameters and user signals. It should be stressed at this point that the LL1 representation above resulted solely from the 2-D array structure and the multipath assumption for the impinging signals.

Collect all the $\mathbf{A}_{x,r}$'s in $\mathbf{A}_x = \left[\begin{array}{ccc} \mathbf{A}_{x,1} & \cdots & \mathbf{A}_{x,R} \end{array}\right] \in \mathbb{C}^{N_x \times \sum_{r=1}^{R} L_r}$ and similarly for the $\mathbf{A}_{y,r}$'s. Let $\mathbf{S} = \left[\begin{array}{ccc} \mathbf{s}_1 & \cdots & \mathbf{s}_R \end{array}\right] \in \mathbb{C}^{N_s \times R}$ be the Tx symbol matrix. The mode-1 unfolding of $\boldsymbol{\mathcal{Y}}$, resulting from stacking its frontal slices next to each other, can then be approximately factorized in its transposed form as

$$\mathbf{Y}_{(1)}^{\mathrm{T}} = (\mathbf{S} \diamond \mathbf{A}_y)\mathrm{Diag}(\boldsymbol{\beta})\mathbf{A}_x^{\mathrm{T}} + \mathbf{W}_{(1)}^{\mathrm{T}}, \tag{7}$$

where $\boldsymbol{\beta} = \left[\begin{array}{cccc} \boldsymbol{\beta}_1^{\mathrm{T}} & \boldsymbol{\beta}_2^{\mathrm{T}} & \cdots & \boldsymbol{\beta}_R^{\mathrm{T}} \end{array}\right]^{\mathrm{T}} \in \mathbb{C}^{\sum_{r=1}^{R} L_r \times 1}$. Stacking the horizontal slices next to each other yields the corresponding relation for the transposed mode-2 unfolding:

$$\mathbf{Y}_{(2)}^{\mathrm{T}} = (\mathbf{A}_x \diamond \mathbf{S})\mathrm{Diag}(\boldsymbol{\beta})\mathbf{A}_y^{\mathrm{T}} + \mathbf{W}_{(2)}^{\mathrm{T}} \tag{8}$$

The factorization relation for the transposed mode-3 unfolding, resulting from stacking lateral slices, is

$$\mathbf{Y}_{(3)}^{\mathrm{T}} = \underbrace{(\mathbf{A}_y \diamond_{\mathrm{c}} \mathbf{A}_x)\mathrm{blockdiag}(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_R)}_{\mathbf{H}} \mathbf{S}^{\mathrm{T}} + \mathbf{W}_{(3)}^{\mathrm{T}}, \tag{9}$$

where $\mathbf{H} \in \mathbb{C}^{N_x N_y \times R}$ represents the combined channel matrix, with columns $\mathrm{vec}(\mathbf{H}_r)$, $r = 1, 2, \ldots, R$. One may employ, for this JCD problem, an *alternating least squares (ALS)* procedure [31], which alternates between (7), (8), and (9) to update the estimates of $\mathbf{A}_x$, $\mathbf{A}_y$, and $\mathbf{S}$, respectively. Such an algorithm is tabulated as Algorithm 1, where the discrete input symbol constellation, say $\mathcal{A}$, is also taken into account to facilitate (and speed up) convergence and most importantly improve the quality of the resulting decomposition by projecting (entry-wise, for the sake of simplicity) each symbol matrix estimate onto the set $\mathcal{F}$ of $N_s \times R$ matrices with entries in $\mathcal{A}$. $\eta > 0$ is a small regularization parameter that helps to stabilize the pseudo-inversion. The path gains can be found from either of the unfoldings above. For example, from (8):

$$\mathrm{vec}(\mathbf{Y}_{(2)}^{\mathrm{T}}) = [\mathbf{A}_y \diamond_{\mathrm{c}} (\mathbf{A}_x \diamond \mathbf{S})]\boldsymbol{\beta} + \mathrm{vec}(\mathbf{W}_{(2)}^{\mathrm{T}}) \tag{10}$$

Given the fact that the KR products involved in the above conditional LS subproblems are (generically) of full column rank [9], Algorithm 1 can be shown to monotonically reduce the cost $\frac{1}{2}\|\boldsymbol{\mathcal{Y}} - \sum_{r=1}^{R}(\mathbf{A}_r\mathbf{B}_r^{\mathrm{T}}) \circ \mathbf{c}_r\|_{\mathrm{F}}^2$ in each iteration, at least when the intermediate symbol projections are omitted. The convergence of such iterations when projections onto $\mathcal{A}$ are intervened will be discussed in more detail in the sequel.

With all $L_r = 1$, i.e., only line-of-sight transmission, the above reduces to a rank-R CPD approximation problem, namely $\boldsymbol{\mathcal{Y}} = [\![\boldsymbol{\beta}; \mathbf{A}_x, \mathbf{A}_y, \mathbf{S}]\!]$, which

Algorithm 1: ALS

Data: $\mathcal{Y}, R, L_r, r = 1, 2, \ldots, R, \eta$

Result: Estimates of $\mathbf{H}$ and $\mathbf{S}$

Initialize $\mathbf{A}_y, \mathbf{S}$;

repeat

  $\mathbf{P} \leftarrow \hat{\mathbf{S}} \diamond \hat{\mathbf{A}}_y$;

  $\hat{\mathbf{A}}_x \leftarrow \mathbf{Y}_{(1)}\mathbf{P}^*(\mathbf{P}^{\mathrm{T}}\mathbf{P}^* + \eta\mathbf{I}_{\sum_r L_r})^{-1}$;

  $\mathbf{Q} \leftarrow \hat{\mathbf{A}}_x \diamond \hat{\mathbf{S}}$;

  $\hat{\mathbf{A}}_y \leftarrow \mathbf{Y}_{(2)}\mathbf{Q}^*(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}^* + \eta\mathbf{I}_{\sum_r L_r})^{-1}$;

  $\hat{\mathbf{H}} \leftarrow (\hat{\mathbf{A}}_y \diamond_{\mathrm{c}} \hat{\mathbf{A}}_x)\mathrm{blockdiag}(\mathbf{1}_{L_1}, \mathbf{1}_{L_2}, \ldots, \mathbf{1}_{L_R})$;

  $\hat{\mathbf{S}} \leftarrow \mathbf{Y}_{(3)}\hat{\mathbf{H}}^*(\hat{\mathbf{H}}^{\mathrm{T}}\hat{\mathbf{H}}^* + \eta\mathbf{I}_R)^{-1}$;

  Scale the columns of $\hat{\mathbf{S}}$;

  Counter-scale the columns of $\hat{\mathbf{A}}_x, \hat{\mathbf{A}}_y$;

  Project $\hat{\mathbf{S}}$ (entry-wise) onto $\mathcal{F}$;

until *convergence*;

$\hat{\mathbf{H}} \leftarrow (\hat{\mathbf{A}}_y \diamond_{\mathrm{c}} \hat{\mathbf{A}}_x)\mathrm{blockdiag}(\mathbf{1}_{L_1}, \mathbf{1}_{L_2}, \ldots, \mathbf{1}_{L_R})$;

can be treated as in, e.g., [32]. The AoAs can then be extracted from the identified $\mathbf{A}_x, \mathbf{A}_y$, given their Vandermonde structure (cf. (2)) and (3), (4). In contrast to CPD, however, the essential uniqueness of BTD involves *matrix-valued* scaling ambiguity per each product (1). [13, Theorem 4.1] says that it is sufficient for this kind of uniqueness to have full column rank $\mathbf{A}_x, \mathbf{A}_y$ (hence sufficiently large array) and no collinear columns in $\mathbf{S}$ (incoherent source signals), conditions that are easily satisfied in practice. Note that this is not a necessary condition, as demonstrated in the simulations section. The non-trivial (matrix-valued) invertible ambiguity per source signal will, in general, prevent the algorithm from identifying the steering matrices and performing DoA estimation. Since the Vandermonde property of a matrix is, in general, lost when undergoing an invertible transformation, the identification of the AoAs could be based on enforcing this property on the estimated steering matrices. Such identifiability issues are beyond the scope of this paper and will not be further considered here. In the algorithms to be presented next, it will be only the discrete-valued nature of $\mathbf{S}$ and the (BTD-induced) low rankness of the columns of $\mathbf{H}$ that will be made use of in solving for the Tx symbols and the channel, respectively. Thus, the focus here will be on the *signal copy* problem, deferring DF to future work.

3. BTD-based JCD

The matrix factorization problem $\mathbf{Y} \triangleq \mathbf{Y}_{(3)}^{\mathrm{T}} \approx \mathbf{H}\mathbf{S}^{\mathrm{T}}$ in (9) also suffers, however, as is, from an intrinsic matrix-valued ambiguity itself. Unless the knowledge available for the factors (in the form of their properties [33]) is appropriately exploited, the problem has infinitely many solutions. Impos-

ing the constraint that $\mathbf{S}$ belongs to $\mathcal{F}$ can remove this ambiguity, as proved in [34] for a sufficiently large number of snapshots, N_s, and sufficiently rich source signals.[3] The factors can be identified via the simple iterative procedure of alternatingly solving for $\mathbf{S}$ and $\mathbf{H}$ while projecting $\mathbf{S}$ onto $\mathcal{F}$ in each iteration [34]. The latter step is non-trivial in general and can be performed in an exhaustive search manner [34] or, more realistically, using sphere decoding [6]. Instead, relying simply on entry-wise projection was demonstrated in [34] to give a computationally efficient and well-performing albeit sub-optimal algorithm, referred to as *iterative least squares with projection (ILSP)*. Its per-iteration complexity in the JCD problem under consideration is dominated by the cost of inverting $R \times R$ matrices, with R being in practice much smaller than the rest of the factor dimensions. Algorithm 1 involves inversions of order $\sum_{r=1}^{R} L_r$, at no gain in detection performance compared to ILSP and the rest of the algorithms developed in the following.

3.1. The ILSP(-SVP) Algorithm

Given an estimate of $\mathbf{H}$, the matrix $\mathbf{S}$ can be computed from

$$\hat{\mathbf{S}} = \mathbf{Y}^{\mathrm{T}}\hat{\mathbf{H}}^{*}(\hat{\mathbf{H}}^{\mathrm{T}}\hat{\mathbf{H}}^{*} + \eta\mathbf{I}_R)^{-1} \tag{11}$$

After projecting each entry of $\hat{\mathbf{S}}$ onto $\mathcal{A}$, $\mathbf{H}$ is updated analogously to keep validating (9):

$$\hat{\mathbf{H}} = \mathbf{Y}\hat{\mathbf{S}}^{*}(\hat{\mathbf{S}}^{\mathrm{T}}\hat{\mathbf{S}}^{*} + \eta\mathbf{I}_R)^{-1} \tag{12}$$

This ILSP iteration can be enhanced with the additional knowledge that the rth column of $\mathbf{H}$, when unfolded into an $N_x \times N_y$ matrix, is of rank (at most)

[3] For more about this identifiability issue, see [34] and the relevant mention in [35].

L_r, $r = 1, 2, \ldots, R$ (cf. (1)). This can be incorporated by replacing the rth column of $\hat{\mathbf{H}}$ by the vectorization of the rank-L_r singular value decomposition (SVD) of its $N_x \times N_y$ matricization. If $\mathcal{L}$ denotes the set of $N_x N_y \times R$ matrices that enjoy this property, the so-called *ILSP-SVP* algorithm summarized in Algorithm 2 is a projected ALS solution to the following constrained ridge-regression problem:

$$\min_{\mathbf{H}\in\mathcal{L},\mathbf{S}\in\mathcal{F}} h(\mathbf{H},\mathbf{S}) \triangleq \frac{1}{2}\|\mathbf{Y} - \mathbf{H}\mathbf{S}^{\mathrm{T}}\|_{\mathrm{F}}^2 + \frac{\eta}{2}(\|\mathbf{H}\|_{\mathrm{F}}^2 + \|\mathbf{S}\|_{\mathrm{F}}^2) \tag{13}$$

The projection onto $\mathcal{L}$ step will be henceforth referred to as *singular value projection (SVP)*, somewhat abusing the use of the acronym SVP that is strictly speaking used in the literature to refer to gradient descent followed by the projection onto $\mathcal{L}$ [36].

Algorithm 2: ILSP(-SVP)

Data: $\mathcal{Y}, R, L_r, r = 1, 2, \ldots, R, \eta$

Result: Estimates of $\mathbf{H}$ and $\mathbf{S}$

Initialize $\mathbf{H}$;

repeat

  Compute $\mathbf{S}$ from (11);

  Project $\mathbf{S}$ entry-wise onto $\mathcal{F}$;

  Compute $\mathbf{H}$ from (12);

  (Project $\mathbf{H}$ onto $\mathcal{L}$ via SVP per column;)

until *convergence*;

3.2. The SIC-ILS(-SVP) Algorithm

Although the entry-wise manner of projecting onto $\mathcal{F}$ renders ILSP a computationally simple procedure, it also removes the guarantee for its monotonic convergence [37].[4] As shown in [37], this is fixed if a successive interference cancellation (SIC)-type scheme is applied in each iteration, with the R signals being recovered and entry-wise projected one by one while considering the remaining parameters as being known. The corresponding iterative LS (*SIC-ILS*) algorithm is provenly monotonically convergent and performs close to optimally while sharing the same order of computational complexity with ILSP. Interestingly, if endowed with an SVP step per iteration, as in Algorithm 3, it turns, in its constellation-constraint-free form, into the Deflationary BTD (DBTD) scheme of [38]. The latter generalizes to LL1 the deflationary CPD (DCPD) algorithm from [39], which first extracts all rank-1 terms in a deflationary manner and then refines these estimates by a rank-1 approximation of the current noise tensor estimate plus the previous corresponding rank-1 term estimate. It is shown in [39] that DCPD converges almost surely for *best* rank-1 approximation steps.[5] DBTD follows the same idea while utilizing a rank-$(L, L, 1)$ approximation step instead.[6] Analogously with what holds for DCPD, it is shown in [38] that with a

[4]No such issues were encountered in the simulations, however.

[5]In practice, these can be sequential rank-one approximation and projection (SeROAP) steps [40]. SeROAP is at least as good as sequential truncated higher-order SVD (ST-HOSVD), which is in turn at least as good as truncated higher-order SVD (T-HOSVD).

[6]This can be done via the LL1 extension of SeROAP, termed sequential low-rank approximation and projection (SeLRAP) [41, 38], which shows some little improvement in accuracy compared to ST-HOSVD and T-HOSVD.

Algorithm 3: SIC-ILS(-SVP)

Data: $\boldsymbol{\mathcal{Y}}, R, L_r, r = 1, 2, \ldots, R, \eta$

Result: Estimates of $\mathbf{H}$ and $\mathbf{S}$

Initialize $\mathbf{H}$;

Compute $\mathbf{S}$ from (11);

Project $\mathbf{S}$ entry-wise onto $\mathcal{F}$;

repeat

  $\hat{\mathbf{Y}} \leftarrow \hat{\mathbf{H}}\hat{\mathbf{S}}^{\mathrm{T}}$;

  $\mathbf{E} \leftarrow \mathbf{Y} - \hat{\mathbf{Y}}$;

  for $r = 1, 2, \ldots, R$ **do**

    $\mathbf{E} \leftarrow \mathbf{E} + \hat{\mathbf{H}}(:, r)\hat{\mathbf{S}}(:, r)^{\mathrm{T}}$;

    $\hat{\mathbf{S}}(:, r) \leftarrow \mathbf{E}^{\mathrm{T}} \frac{\hat{\mathbf{H}}(:,r)^*}{\|\hat{\mathbf{H}}(:,r)\|^2 + \eta}$;

    Project all entries of $\hat{\mathbf{S}}(:, r)$ onto $\mathcal{A}$;

    $\mathbf{E} \leftarrow \mathbf{E} - \hat{\mathbf{H}}(:, r)\hat{\mathbf{S}}(:, r)^{\mathrm{T}}$;

  end

  Compute $\mathbf{H}$ from (12);

  (Project $\mathbf{H}$ onto $\mathcal{L}$ via SVP per column;)

until *convergence*;

best rank-$(L, L, 1)$ approximation in each step, the DBTD algorithm enjoys monotonic convergence.

3.3. The AO-BTD Algorithm

The idea of viewing the LL1 modeling problem as a matrix factorization one with the low-rankness constraint imposed on the columns of one of the factors appeared also in [23] and later on in [42], where an alternating optimization algorithm with (what is called here) SVP projection steps was proposed, in which $\hat{\mathbf{H}}$ is *inexactly* solved for in each iteration via a gradient descent step instead of the more computationally expensive (12). This so-called *alternating optimization (AO)-BTD* procedure is herein adapted in the present context by incorporating in its iterations projections onto the symbol constellation, in analogy with the previous algorithms. This gives rise to Algorithm 4. In each iteration, the step size for the gradient descent step assumes in [42] and in the simulations its largest allowed value, namely the reciprocal of the Lipschitz constant of the gradient with respect to (w.r.t.) $\mathbf{H}$, which is in turn given by the maximum of the singular values of $\mathbf{S}$. It should be recalled at this point that the close connection with LL1 of a blind source separation problem like that of (9) with low-rank mixing vectors was revealed in various ways in [25], [43] (and references therein).

3.4. The ILSP(-SVP)-C Algorithm

As demonstrated in the simulations section, it is the ILSP algorithm, with its SVP addition, that prevails in both detection performance and computational simplicity and will therefore be the focus of the rest of this section, which will be devoted to two challenging practical scenarios: that of missing

Algorithm 4: AO-BTD

Data: $\boldsymbol{\mathcal{Y}}, R, L_r, r = 1, 2, \ldots, R, \eta$

Result: Estimates of $\mathbf{H}$ and $\mathbf{S}$

Initialize $\mathbf{H}$;

repeat

  Compute $\mathbf{S}$ from (11);

  Project $\mathbf{S}$ entry-wise onto $\mathcal{F}$;

  Compute gradient w.r.t. $\mathbf{H}$: $\mathbf{G} \leftarrow -(\mathbf{Y} - \mathbf{H}\mathbf{S})\mathbf{S}^{\mathrm{H}} + \eta\mathbf{H}$;

  Set step size α (e.g., as $1/\lambda_{\max}(\mathbf{S}\mathbf{S}^{\mathrm{H}})$);

  $\mathbf{H} \leftarrow \mathbf{H} - \alpha\mathbf{G}$;

  (Project $\mathbf{H}$ onto $\mathcal{L}$ via SVP per column;)

until *convergence*;

observations, for example, due to failed sensors, and the case of time-varying channel response that needs to be tracked in time.

Having missing/failed sensors in the array (as shown in the example of Fig. 1) implies an Rx signal tensor with missing mode-3 fibers (known as tubes), a scenario that is more challenging for the estimation of the latent parameters than the more common case of (uniformly) random missing entries. In such a scenario, the factorization of $\mathbf{Y}$ takes the form (cf. [12, Eq. (3.4)])

$$\mathbf{Y}_\Omega = \mathbf{H}_\Omega \mathbf{S}^{\mathrm{T}}, \tag{14}$$

where Ω is the set of working sensor indices, and $\mathbf{Y}_\Omega \in \mathbb{C}^{|\Omega| \times N_s}, \mathbf{H}_\Omega \in \mathbb{C}^{|\Omega| \times R}$ stand for the row-subsampled versions of the corresponding matrices. $\mathbf{Y}_\Omega$ can be written as $\boldsymbol{\Omega}\mathbf{Y}$, with $\boldsymbol{\Omega}$ being the $|\Omega| \times N_x N_y$ binary selection matrix that

keeps only the observed rows of $\mathbf{Y}$, and similarly for $\mathbf{H}_\Omega$. The corresponding LS factorization problem, termed singular problem in [44], can be viewed as a weighted LS one with weights on the data matrix entries [45] that are equal here to the squares of the $\mathbf{\Omega}$ entries. The corresponding variant of ILSP-SVP, termed here *ILSP-SVP with completion or ILSP-SVP-C* and summarized in Algorithm 5, iterates between solving for each of the two factors, using only the observed outputs, as in the following variants of (11) and (12), respectively:

$$\hat{\mathbf{S}} = \mathbf{Y}_\Omega^{\mathrm{T}} \hat{\mathbf{H}}_\Omega^* (\hat{\mathbf{H}}_\Omega^{\mathrm{T}} \hat{\mathbf{H}}_\Omega^* + \eta \mathbf{I}_R)^{-1} \quad (15)$$

$$\hat{\mathbf{H}}_\Omega = \mathbf{Y}_\Omega \hat{\mathbf{S}}^* (\hat{\mathbf{S}}^{\mathrm{T}} \hat{\mathbf{S}}^* + \eta \mathbf{I}_R)^{-1} \quad (16)$$

Note that only for the low-rank approximation (SVP step) of the channel

Algorithm 5: ILSP(-SVP)-C

Data: $\mathcal{Y}_\Omega, R, L_r, r = 1, 2, \ldots, R, \mathbf{\Omega}, \eta$

Result: Estimates of $\mathbf{H}$ and $\mathbf{S}$

Initialize $\mathbf{H}_\Omega$;

repeat

  Compute $\mathbf{S}$ from (15);

  Project $\mathbf{S}$ entry-wise onto $\mathcal{F}$;

  Compute $\mathbf{H}_\Omega$ from (16);

  ($\mathbf{H} \leftarrow \mathbf{\Omega}^{\mathrm{T}} \mathbf{H}_\Omega$;

  Project $\mathbf{H}$ onto $\mathcal{L}$ via SVP per column;

  $\mathbf{H}_\Omega \leftarrow \mathbf{\Omega} \mathbf{H}$;)

until *convergence*;

matrix missing rows are filled with zeros, as in the direct method of [44,

Section 5.1]. Nevertheless, in [44] it is the data matrix rows that are filled with zeros, whereas here only the observed sensor outputs are considered with the rest being simply omitted.

3.5. The O-ILSP(-SVP) Algorithm

The array dimensions may be too large to allow the processing of the data tensor in one batch. Additionally, in practice, the path gains may change with time, albeit more slowly than the DoAs [46]. To cope with such problems, an incremental/online version of the ILSP-SVP method should be made available. The exponentially weighted recursive LS (RLS)-type Algorithm 6 describes the so-called *online ILSP-SVP (O-ILSP-SVP)* procedure, which is inspired by our earlier work on online LL1 modeling [47], though here it is the matrix factorization formulation that is adopted instead. Once preliminary estimates of the Tx symbols in the current snapshot are obtained using the current channel matrix estimate, they are used to update the auto- and cross-correlation matrices $\mathbf{V}$ and $\mathbf{G}$, respectively, with the factor $0 < \lambda \leq 1$ serving as a means of down-weighing past snapshots and hence realizing exponential forgetting. To update the channel estimate, the algorithm resorts to the matrix inversion lemma [48] for the inversion of $\mathbf{V}$. It can be noted in passing that the matrix inversion lemma can be fully used, also to invert the channel, in the case that SVP is omitted. The updated channel response is used to refine the symbol estimate and append it to the evolving symbol matrix. Note that the factor $\mathbf{S}$ can be built in this way because of the implicit

Algorithm 6: O-ILSP(-SVP)

Data: $\boldsymbol{\mathcal{Y}}$ in a streaming manner, $R, L_r, r = 1, 2, \ldots, R, \lambda, \eta$

Result: Per-snapshot estimates of $\mathbf{H}$ and $\mathbf{S}$

Initialize $\mathbf{H}_0, \mathbf{S}_0 \leftarrow [\,], \mathbf{V}_0, \mathbf{G}_0$;

for $n = 1, 2, \ldots$ **do**

  $\mathbf{y}_n \leftarrow \mathrm{vec}(\boldsymbol{\mathcal{Y}}(:, :, n))$;

  $\mathbf{F}_{n-1} \leftarrow (\mathbf{H}_{n-1}^{\mathrm{H}}\mathbf{H}_{n-1} + \eta\mathbf{I}_R)^{-1}\mathbf{H}_{n-1}^{\mathrm{H}}$;

  $\mathbf{s}_n \leftarrow \mathbf{F}_{n-1}\mathbf{y}_n$;

  Entry-wise project $\mathbf{s}_n$ onto $\mathcal{A}$;

  $\mathbf{V}_n \leftarrow \lambda\mathbf{V}_{n-1} + \mathbf{s}_n\mathbf{s}_n^{\mathrm{H}}$;

  $\mathbf{G}_n \leftarrow \lambda\mathbf{G}_{n-1} + \mathbf{y}_n\mathbf{s}_n^{\mathrm{H}}$;

  $\mathbf{W} \leftarrow \mathbf{V}_n^{-1}$ (using matrix inversion lemma);

  $\mathbf{H}_n \leftarrow \mathbf{G}_n\mathbf{W}$;

  (Project $\mathbf{H}_n$ onto $\mathcal{L}$ via SVP per column;)

  $\mathbf{F}_n \leftarrow (\mathbf{H}_n^{\mathrm{H}}\mathbf{H}_n + \eta\mathbf{I}_R)^{-1}\mathbf{H}_n^{\mathrm{H}}$;

  $\mathbf{s}_n \leftarrow \mathbf{F}_n\mathbf{y}_n$;

  Entry-wise project $\mathbf{s}_n$ onto $\mathcal{A}$;

  $\mathbf{S}_n^{\mathrm{T}} \leftarrow \begin{bmatrix} \mathbf{S}_{n-1}^{\mathrm{T}} & \mathbf{s}_n^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$;

end

assumption of a slowly-varying channel.[7] Algorithm 6 mainly differs from the recursive algorithm (recursive least squares with enumeration (RLSE)) reported in [34] in that the symbol estimate is therein found via exhaustive enumeration and, of course, in the SVP step per snapshot included here. At this point, one should also refer to the method proposed in [50] (as a follow-up to [30]), which relies on the tracking of the signal subspace (expressed with the aid of principal component analysis (PCA)) to estimate the AoAs. Using orthogonal training, the user signals are then separated to allow easy estimation of the path gains per each of them. Finally, the channel is reconstructed from its angle and path gain parameters. It should be stressed, however, that the algorithm of [50] relies on the rather limiting assumption that all signals come from the same set of DoAs.

Before proceeding to evaluate the previous algorithms, some comments are in order.

1. The projection onto the *non-convex* constellation $\mathcal{A}$ can be relaxed to a projection onto its convex hull as in [51], which is termed PrOX and first scales the signal by a factor strictly larger than unity to "push" the projected values towards the convex hull extremes (i.e., the constellation symbols). An analysis of the convergence of ILSP-SVP, especially with PrOX, can stem from the observation that its steps are projections in the form of closed property mappings, and rely on the results of [33]. This is also in the vein of the lift-and-project approach (see [52] for an

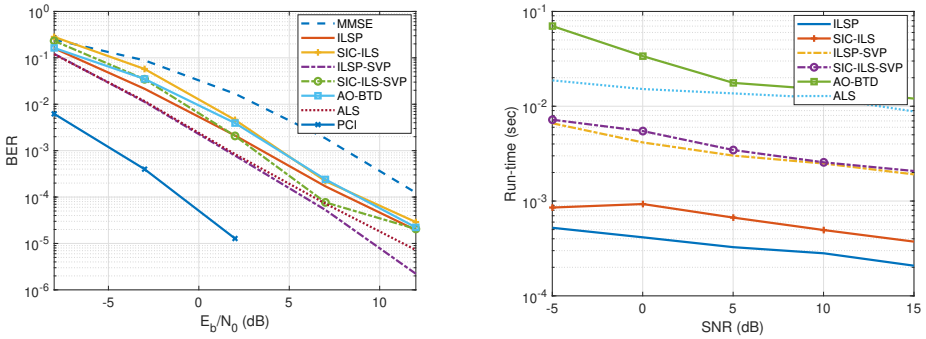
[7] This assumption, of slowly-varying factors in the non-evolving modes of a streamed tensor, underlies all the recursive tensor decomposition methods and has implications in channel tracking that have only recently started to be investigated [49].

application in multi-dimensional harmonic retrieval) and can be placed within the structured low-rank approximation (SLRA) framework [44].

2. In this paper, it is assumed that both R and L are *a-priori* known, an assumption that needs to be relaxed in practice. ILSP-SVP has been seen in our simulations to be robust to overestimation of L (as is the case for LL1 more generally; see, e.g., [53] and references therein). Hence it is primarily the number of sources, R, that needs to be accurately estimated. Besides, this will also dictate the duration of the training sequence (see Section 4). In the most realistic case of $R \leq N_x N_y$, R can be estimated as the rank of (a chunk of) $\mathbf{Y}$, for example with the aid of SVD [44, Section 5.4], possibly facilitated (via compression) or even replaced by a Gram-Schmidt orthogonalization of $\mathbf{Y}^{\mathrm{T}}$, as in, e.g., [44, Section 5.1] or [54]. Alternatively, a rank-revealing version of the previous algorithms is possible, in the spirit of [55] (and of [56] for the case of missing data). Notice that the regularizer in (13) is a tight upper bound of the nuclear norm of $\mathbf{Y}$ and promotes smoothness (hence implicitly low-rankness) of the $\mathbf{H}, \mathbf{S}$ factors. It could be replaced by a (re-)weighted version as in [55] that amounts to the $\ell_{1,2}$ norm of $\begin{bmatrix} \mathbf{H}^{\mathrm{T}} & \mathbf{S}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ and hence more pronouncedly promote joint group sparsity and in turn allow rank recovery by eliminating columns in the two factors that have negligible (very low) magnitude.

4. Simulation Results

In the experimental setup tested, the sources emit QPSK-modulated signals, at the carrier frequency of 1 GHz. All of them arrive through an



(a) (b)

Figure 2: (a) Detection performance and (b) run-time of the algorithms, for a 10×10 array receiving $R = 3$ signals of length $N_s = 30$ through $L = 3$ paths.

equal number of paths, say L, with the azimuth and zenith angles being chosen uniformly at random from $(0, \pi)$ and $(0, \pi/2)$, respectively (i.e., at the "front" of the URA) and the complex path gains drawn from the complex zero-mean, unit-variance Gaussian distribution. In all arrays tested, the inter-sensor spacings take their maximum allowed value for aliasing avoidance, namely $d_x = d_y = \frac{\lambda}{2}$. To initialize the iterative algorithms and account for the permutation ambiguity, information symbols are preceded by orthogonal training preambles of (the smallest) length R, which are used to provide a first estimate of $\mathbf{H}$.

A typical example of the comparative signal copying performance of the algorithms considered, expressed in terms of the bit error rate (BER) for a 10×10 URA with $R = 3$ signals of length $N_s = 30$ arriving through $L = 3$ paths, is given in Fig. 2(a). The BER results of the minimum mean squared error (MMSE) detector as well as those obtained with perfect channel information (PCI) are also included. Though quite apart from the PCI-based

results, the performance of the semi-blind receivers is observed to be superior to that of the training-only-based (MMSE) one. This is because semi-blind JCD extracts channel and hence Tx signal information also from the information part of the Rx signals. The performance of the AO-BTD algorithm lies between those of the simplified (without the low-rank constraint) versions of SIC-ILS and ILSP. ALS performs similarly with ILSP-SVP, especially at lower SNR levels, albeit at a significantly higher computational cost, as shown in Fig. 2(b), where the run-times on a computer employing i7-8550U CPU@1.80 GHz and 8 GB RAM, and using the R2024a release of MATLAB©, are plotted versus the SNR level. Though gradient-descent-based, AO-BTD shows here a relatively high computational cost due to the use of truncated SVD for setting the step size at each iteration. What should be emphasized is that taking the low-rankness of channels into account, i.e., exploiting the underlying LL1 model via the SVP addition, offers improvement, which can be verified to be more significant for larger arrays. ILSP-SVP turns out to be the best-performing choice, and its computational requirements are only out-classed by those of its simplified (ILSP) version. It will therefore be the focus of the rest of the simulations-based study.

Fig. 3 compares the ILSP-SVP and MMSE detectors over a realistic SNR range, and for various values of (a) $N_x = N_y$, (b) R, (c) N_s, and (d) L. The superiority of the semi-blind over the training-only-based approach can again be observed, in all cases. This gain can be seen to improve with the number of snapshots and is already non-negligible with only $N_s = 5$ and $R = 2$ users. As expected, the detection performance also improves with the additional diversity offered by increasing array size and/or number of

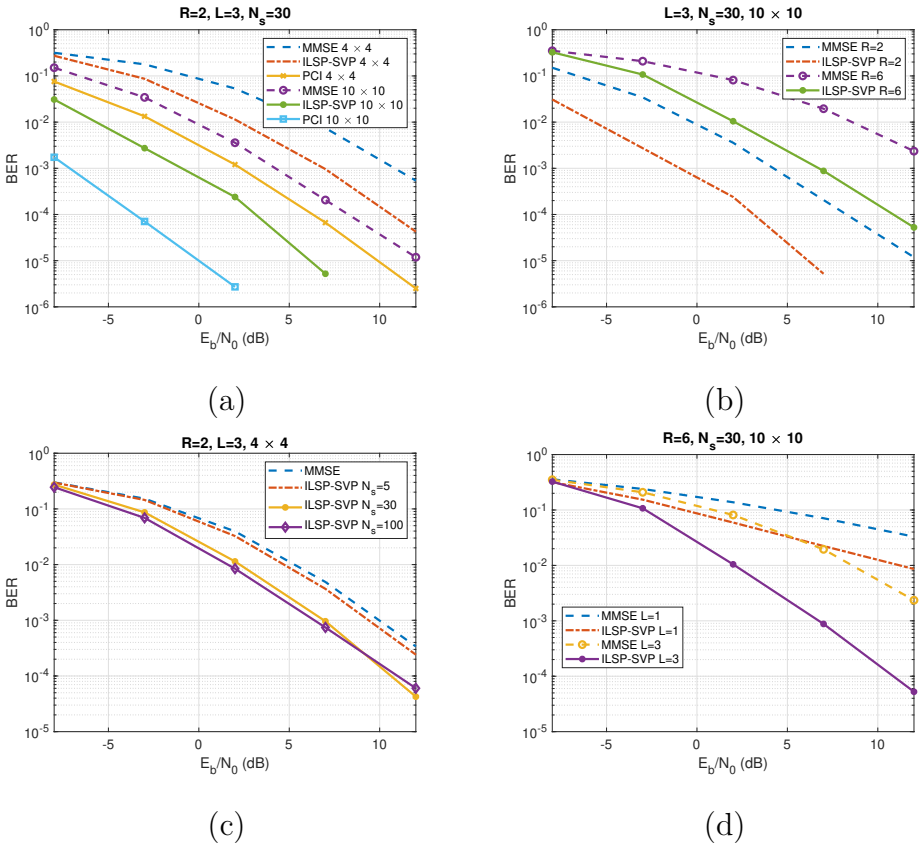
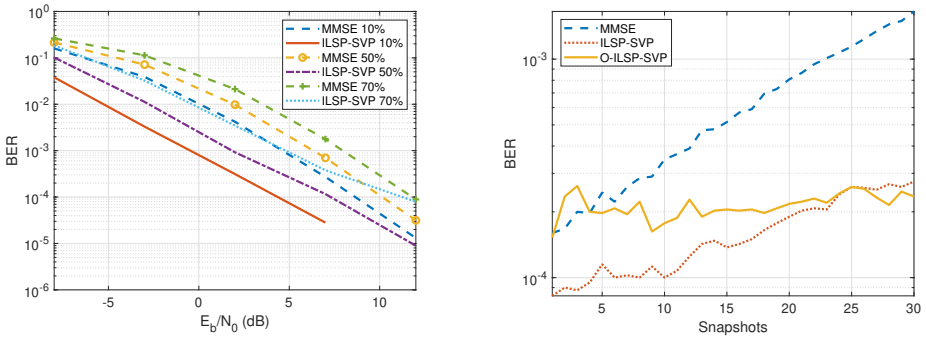


Figure 3: Detection performance of ILSP-SVP for various (a) array sizes, (b) numbers of users, (c) numbers of snapshots, and (d) numbers of paths.



(a) (b)

Figure 4: (a) Detection performance with various missing data ratios, and (b) evolution of BER in time and at $E_b/N_0 = 9$ dB, for a 10×10 array receiving $R = 2$ signals of length $N_s = 30$ through $L = 3$ paths.

paths. On the other hand, the problem becomes more difficult as more signals are to be separated and detected. It should be noted that the semi-blind receiver manages to perform sufficiently well even in scenarios that do *not* satisfy the sufficient uniqueness condition for the LL1 model. The cases of $R = 2, L = 3, N_x = N_y = 4$ and $R = 6, L = 3, N_x = N_y = 10$ are two such examples.

Three cases of missing or failed sensors are tested in Fig. 4(a), where the ILSP-SVP-C algorithm is observed to be able to preserve a good enough detection performance and again significantly outperform the MMSE detector. Both rely on the channel estimate computed with the training-based Rx signal sub-sampled accordingly. When the percentage of missing sensor outputs reaches a quite high level, namely 70% on Fig. 4(a), the ILSP-SVP-C BER curve starts to floor at the high SNR regime.

Consider now the time-varying case, where time variation is realized

through the following Gauss-Markov model for the path gains at snapshot n,

$$\boldsymbol{\beta}^{(n)} = \rho\boldsymbol{\beta}^{(n-1)} + \sqrt{1-\rho^2}\mathbf{x}^{(n)},$$

with $0 \leq \rho \leq 1$ quantifying the rate of variation and $\mathbf{x}^{(n)}$ being a random standard Gaussian vector. In Fig. 4(b), a very slowly-varying system is tested, namely with $\rho = 0.998$, and at a realistically moderate E_b/N_0 level of 9 dB, to allow the application of O-ILSP-SVP, which relies on such an assumption (as all decision-directed-mode recursive schemes derived out of online tensor decomposition). Comparison is made with the batch MMSE and ILSP-SVP methods. Both the iterative and the online ILSP-SVP schemes are initialized with training and a forgetting factor of $\lambda = 0.985$ is employed in the latter. O-ILSP-SVP manages to track this time-varying system in contrast to the MMSE detector and its batch counterpart whose BER diverges with time even in this slowly-varying setting.

5. Conclusions and Future Work

This paper serves to put forward the role of BTD in ASP and showcase its potential in JCD, through a URA-based example. The LL1 BTD model is shown to be the natural choice in sensor arrays of increased dimensionality that also involve channel multipath. Several semi-blind algorithms are considered in this context, that can be viewed as wedding existing JCD schemes with BTD approximation. ILSP-SVP is singled out as the best-performing and computationally efficient of these schemes. Robust to sensor failures and recursive versions of it are then developed. The signal copying performance of the semi-blind JCD algorithms is evaluated via simulation results at var-

ious noise levels, array sizes, and numbers of users and paths and favorably compares with that of the training-only-based solution.

Further work is required to fill in several gaps left open here. These include the question of selecting the model from the data, which has been discarded in the present study and has only been discussed above w.r.t. estimating the number of sources, R, and via a classical SVD-based approach. To recover both R and the numbers of paths, L_r, $r = 1, 2, \ldots, R$, and jointly with the estimation of the LL1 factors, algebraic methods [29, 57] can be adopted. However, these tend to be computationally expensive[8] and only work safely at high enough SNR levels [57]. Rank-revealing methods based on optimization, such as, for example, the $\ell_{1,2}$-regularization-based scheme that was shortly described above for the bilinear factorization problem or its regularized BTD-ALS extension in [14], sound more promising. Bayesian versions, allowing automatic rank determination (e.g., [59]), and with the $\mathbf{S}$ prior properly chosen to meet the discrete-valued nature of the symbols (see, e.g., [60]), also deserve to be studied.[9] One should also extend this work to cover wideband large-scale ASP, by also taking frequency- [16] and spatial-selective [61] effects into account. Near-field effects [26], colored noise [62],

[8]They start, in the present context, with the computation of the null space of an $N_x N_y N_s \times (N_x^2 + N_y^2)$ matrix or equivalently with a monomial factorization formulation of the bilinear factorization problem (13), where the monomial constraints serve to express the requirement of low rank [58].

[9]It is of interest to point out here that the step of entry-wise projecting onto the constellation, that constitutes a basic ingredient of the algorithms considered here, also appears in [60] with the name of *heuristic demapping* and shown therein to be an approximation to a more principled projection step in an expectation-maximization (EM) framework.

dealing with coherent sources [43], and nonuniform arrays [63] (that might also cover missing sensors scenarios) are some more realistic additions. Going from planar to higher-dimensional (e.g., 3-D) arrays [10] seems both feasible and rewarding. These can be 3-D antenna arrays (e.g., [64, 28]) and/or planar arrays configured/viewed as multiple-scale configurations of URAs as in, e.g., [10, Fig. 11]. In those cases, the BTD model can result via segmentation as in [25, Eq. (16)]. Finally, in a fast-varying environment, not only the path gains but also the DoAs and even the number of paths per user and/or the number of user signals may have to be considered as time-varying. To cope with such environments, extensions of the O-ILSP-SVP method presented here should be developed (e.g., in the vein of [47]) that can *recursively* track both the number *and* the values of the system parameters.

References

[1] M. Pesavento, M. Trinh-Hoang, M. Viberg, Three more decades in array signal processing research, IEEE Signal Process. Mag. (2023) 92–106.

[2] E. Björnson, et al., Twenty-five years of signal processing advances for multiantenna communications, IEEE Signal Process. Mag. (2023) 107–117.

[3] O. M. Sleem, et al., Unsupervised learning for reference signals overhead reduction in 3GPP MIMO systems, IEEE Trans. Mach. Learn. Commun. Netw. 2 (2024) 81–94.

[4] O. Elijah, et al., A comprehensive survey of pilot contamination in mas-

sive MIMO—5G system, IEEE Commun. Surveys Tuts. 18 (2) (2016) 905–923.

[5] Z. Luo, C. Li, L. Zhu, A comprehensive survey on blind source separation for wireless adaptive processing: Principles, perspectives, challenges and new research directions, IEEE Access 6 (2018) 66685–66708.

[6] C. Rizogiannis, et al., Semi-blind maximum-likelihood joint channel/data estimation for correlated channels in multiuser MIMO networks, Signal Process. 90 (2010) 1209—-1224.

[7] N. D. Sidiropoulos, et al., Tensor decomposition for signal processing and machine learning, IEEE Trans. Signal Process. 65 (13) (2017) 3551–3582.

[8] H. Chen, et al., Tensor decompositions in wireless communications and MIMO radar, IEEE J. Sel. Topics Signal Process. 15 (3) (2021) 438–453.

[9] N. Sidiropoulos, R. Bro, G. B. Giannakis, Parallel factor analysis in sensor array processing, IEEE Trans. Signal Process. 48 (8) (2000) 2377–2388.

[10] S. Miron, et al., Tensor methods for multisensor signal processing, IET Signal Process. 14 (10) (2020) 693–709.

[11] B. Li, et al., Structural-missing tensor completion for robust DOA estimation with sensor failure, Appl. Sci. 13 (12740) (2023).

[12] M. Sørensen, L. De Lathauwer, Fiber sampling approach to canonical

polyadic decomposition and application to tensor completion, SIAM J. Matrix Anal. Appl. 40 (3) (2019) 888–917.

[13] L. De Lathauwer, Decompositions of a higher-order tensor in block terms — Part II: Definitions and uniqueness, SIAM J. Matrix Anal. Appl. 30 (3) (2008) 1033–1066.

[14] A. A. Rontogiannis, E. Kofidis, P. V. Giampouras, Block-term tensor decomposition: Model selection and computation, IEEE J. Sel. Topics Signal Process. 15 (3) (2021) 464–475.

[15] J. Salmi, A. Richter, V. Koivunen, Sequential unfolding SVD for tensors with applications in array signal processing, IEEE Trans. Signal Process. 57 (12) (2009) 4719–4733.

[16] M. Sørensen, F. Van Eeghem, L. De Lathauwer, Blind multichannel deconvolution and convolutive extensions of canonical polyadic and block term decompositions, IEEE Trans. Signal Process. 65 (15) (2017) 4132–4145.

[17] D. Nion, L. De Lathauwer, A block factor analysis based receiver for blind multi-user access in wireless communications, in: Proc. ICASSP, Toulouse, France, 2006.

[18] M. Sørensen, L. De Lathauwer, Tensor decompositions with block-Toeplitz structure and applications in signal processing, in: Proc. ACSSC, Pacific Grove, CA, 2011.

[19] M. Sørensen, L. De Lathauwer, Coupled tensor decompositions for applications in array signal processing, in: Proc. CAMSAP, Saint Martin, French West Indies, France, 2013.

[20] D. Nion, N. D. Sidiropoulos, Tensor algebra and multidimensional harmonic retrieval in signal processing for MIMO radar, IEEE Trans. Signal Process. 58 (11) (2010) 5693–5705.

[21] J.-X. Yang, et al., Using coupled multilinear rank-$(L, L, 1)$ block term decomposition in multi-static-multi-pulse MIMO radar to localize targets, in: Proc. ISNN, Moscow, Russia, 2019.

[22] L. Khamidullina, M. Haardt, Coupled block-term tensor decomposition for near-field localization in multi-static MIMO radar systems, in: Proc. ICASSP, Seoul, South Korea, 2024.

[23] W. Han, et al., Approach to DOA estimation of incompletely polarized signal based on BTD model, in: Proc. IET Int'l Radar Conf., Hangzhou, China, 2015.

[24] Y.-F. Gao, et al., A type-2 block-component-decomposition based 2D AoA estimation algorithm for an electromagnetic vector sensor array, Sensors 17 (5) (2017).

[25] M. Boussé, O. Debals, L. De Lathauwer, A tensor-based method for large-scale blind source separation using segmentation, IEEE Trans. Signal Process. 65 (2) (2017) 346–358.

[26] P. Ramezani, O. T. Demir, E. Björnson, Localization in massive MIMO

networks: From near-field to far-field, arXiv:2402.07644v1 [eess.SP] (Feb. 2024).

[27] L. Wei, Q. Li, G. Wu, Direction of arrival estimation with uniform planar array, in: Proc. VTC-Fall, Toronto, ON, Canada, 2017.

[28] D. H. Son, T. T. Quynh, Impact analysis of antenna array geometry on the performance of semi-blind structured channel estimation for massive MIMO-OFDM systems, in: Proc. SSP, Hanoi, Vietnam, 2023.

[29] I. Domanov, L. De Lathauwer, On uniqueness and computation of the decomposition of a tensor into multilinear rank-$(1, L_r, L_r)$ terms, SIAM J. Matrix Anal. Appl. 41 (2) (2020) 747–803.

[30] A. Wang, R. Yin, C. Zhong, Channel estimation for uniform rectangular array based massive MIMO systems with low complexity, IEEE Trans. Veh. Technol. 68 (3) (2019) 2545–2556.

[31] L. De Lathauwer, D. Nion, Decompositions of a higher-order tensor in block terms — Part III: Alternating least squares algorithms, SIAM J. Matrix Anal. Appl. 30 (3) (2008) 1067–1083.

[32] M. Sørensen, L. De Lathauwer, Multidimensional harmonic retrieval via coupled canonical polyadic decomposition — Part I: Model and identifiability, IEEE Trans. Signal Process. 65 (2) (2017) 517–527.

[33] J. A. Cadzow, Signal enhancement — A composite property mapping algorithm, IEEE Trans. Acoust., Speech, Signal Process. 36 (1) (1988) 49–62.

[34] S. Talwar, M. Viberg, A. Paulraj, Blind separation of synchronous co-channel digital signals using an antenna array—Part I: Algorithms, IEEE Trans. Signal Process. 44 (5) (1996) 1184–1197.

[35] E. Kofidis, A tensor-based approach to joint channel estimation/data detection in flexible multicarrier MIMO systems, IEEE Trans. Signal Process. 68 (2020) 3179–3193.

[36] P. Jain, R. Meka, I. Dhillon, Guaranteed rank minimization via singular value projection, in: Proc. NIPS, Vancouver, BC, Canada, 2010.

[37] T. Li, N. D. Sidiropoulos, Blind digital signal separation using successive interference cancellation iterative least squares, IEEE Trans. Signal Process. 48 (11) (2000) 3146–3152.

[38] J. H. de Morais Goulart, P. Comon, Non-iterative low-multilinear-rank tensor approximation with application to decomposition in rank-$(1, L, L)$ terms, hal-01516167 (2017).
URL `https://hal.science/hal-01516167/document`

[39] A. P. da Silva, P. Comon, A. L. F. de Almeida, An iterative deflation algorithm for exact CP tensor decomposition, in: Proc. ICASSP, South Brisbane, QLD, Australia, 2015.

[40] A. P. da Silva, P. Comon, A. L. F. de Almeida, A finite algorithm to compute rank-1 tensor approximations, IEEE Signal Process. Lett. 23 (7) (2016) 959–963.

[41] J. H. de Morais Goulart, P. Comon, A novel non-iterative algorithm for

low-multilinear-rank tensor approximation, in: Proc. EUSIPCO, Kos island, Greece, 2017.

[42] X. Fu, K. Huang, Block-term tensor decomposition via constrained matrix factorization, in: Proc. MLSP, Pittsburg, PA, 2019.

[43] N. Govindarajan, E. N. Epperly, L. De Lathauwer, $(L_r, L_r, 1)$-decompositions, sparse component analysis, and the blind separation of sums of exponentials, SIAM J. Matrix Anal. Appl. 43 (2) (2022) 912–938.

[44] I. Markovsky, Low-Rank Approximation: Algorithms, Implementation, Applications, Springer, 2014.

[45] I. Markovsky, Structured low-rank approximation and its applications, Automatica 44 (2008) 891–909.

[46] M. C. Vanderveen, C. B. Papadias, A. Paulraj, Joint angle and delay estimation (JADE) for multipath signals arriving at an antenna array, IEEE Commun. Lett. 1 (1) (1997) 12–14.

[47] A. A. Rontogiannis, E. Kofidis, P. V. Giampouras, Online rank-revealing block-term tensor decomposition, Signal Process. 212 (Nov. 2023).

[48] D. J. Tylavsky, G. R. L. Sohle, Generalization of the matrix inversion lemma, Proc. IEEE 74 (7) (1986) 1050–1052.

[49] E. Kofidis, Adaptive joint channel estimation/data detection in flexible multicarrier MIMO systems — A tensor-based approach, in: Proc. ICASSP-2024, Seoul, South Korea, 2024.

[50] A. Wang, R. Yin, C. Zhong, PCA-based channel estimation and tracking for massive MIMO systems with uniform rectangular arrays, IEEE Trans. Wireless Commun. 19 (10) (2020) 6786–6797.

[51] O. Castañeda, T. Goldstein, C. Studer, VLSI designs for joint channel estimation and data detection in large SIMO wireless systems, IEEE Trans. Circuits Syst. I 65 (3) (2018) 1120–1132.

[52] R. Boyer, P. Comon, Rectified ALS algorithm for multidimensional harmonic retrieval, in: Proc. SAM-2016, Rio de Janeiro, Brazil, 2016.

[53] C. Chatzichristos, et al., Blind fMRI source unmixing via higher-order tensor decompositions, J. Neurosci. Methods (Mar. 2019).

[54] C. Xu, W. Xu, K. Jing, Fast algorithms for singular value decomposition and the inverse of nearly low-rank matrices, Natl Sci. Rev. 10 (2023). doi:https://doi.org/10.1093/nsr/nwad083.

[55] P. V. Giampouras, A. A. Rontogiannis, K. D. Koutroumbas, Alternating iterative reweighted least squares minimization for low-rank matrix factorization, IEEE Trans. Signal Process. 67 (2) (2019) 490–503.

[56] A. A. Rontogiannis, P. V. Giampouras, E. Kofidis, Rank-revealing block-term decomposition for tensor completion, in: Proc. ICASSP, Toronto, ON, Canada, 2021.

[57] E. Evert, et al., Uniqueness result and algebraic algorithm for decomposition into multilinear rank-$(M_r, N_r, \cdot)$ terms and joint block diagonalization, in: Proc. CAMSAP, Los Sueños, Costa Rica, 2023.

[58] M. Sørensen, N. D. Sidiropoulos, L. De Lathauwer, Canonical polyadic decomposition of a tensor that has missing fibers: A monomial factorization approach, in: Proc. ICASSP, Brighton, UK, 2019.

[59] P. V. Giampouras, A. A. Rontogiannis, E. Kofidis, Block-term tensor decomposition model selection and computation: The Bayesian way, IEEE Trans. Signal Process. 70 (2022) 1704–1717.

[60] M. Al-Shoukairi, B. D. Rao, Semi-blind channel estimation in MIMO systems with discrete priors on data symbols, IEEE Signal Process. Lett. 29 (2021) 51–54.

[61] B. Wang, et al., Spatial- and frequency-wideband effects in millimeter-wave massive MIMO systems, IEEE Trans. Signal Process. 66 (13) (2018) 3393–3406.

[62] A. Belouchrani, M. G. Amin, K. Abed-Meraim, Direction finding in correlated noise fields based on joint block-diagonalization of spatio-temporal correlation matrices, IEEE Signal Process. Lett. 4 (9) (1997) 266–268.

[63] M. Sørensen, N. D. Sidiropoulos, Identifiability results for nonuniform linear and rectangular sensor arrays, in: Proc. EUSIPCO, Belgrade, Serbia, 2022.

[64] F. Roemer, H. Becker, M. Haardt, Analytical performance assessment for multi-dimensional tensor-ESPRIT-type parameter estimation algorithms, in: Proc. ICASSP, Dallas, TX, 2010.