

RESEARCH

Open Access

Tensor subspace Tracking via Kronecker structured projections (TeTraKron) for time-varying multidimensional harmonic retrieval

Yao Cheng^{1*}, Florian Roemer², Olaya Khatib¹ and Martin Haardt¹

Abstract

We present a framework for Tensor-based subspace Tracking via Kronecker-structured projections (TeTraKron). TeTraKron allows to extend arbitrary matrix-based subspace tracking schemes to track the tensor-based subspace estimate. The latter can be computed via a structured projection applied to the matrix-based subspace estimate which enforces the multi-dimensional structure in a computationally efficient fashion. This projection is tracked by considering all matrix rearrangements of the signal tensor jointly, which can be efficiently realized via parallel processing. In addition, we incorporate forward-backward-averaging and find a similar link between the real-valued matrix-based and tensor-based subspace estimation. This enables the tracking of the real-valued tensor-based subspace estimate via a similar Kronecker-structured projection applied to the real-valued matrix-based subspace estimate. In time-varying multidimensional harmonic retrieval problems, the TeTraKron-based subspace tracking schemes outperform the original matrix-based subspace tracking algorithms as well as the batch solutions provided by the SVD and the HOSVD. Moreover, incorporating forward-backward-averaging leads to an improved accuracy of the subspace tracking, and only real-valued processing is involved. Furthermore, we evaluate the performances of ESPRIT-type parameter estimation schemes where the subspace estimates obtained by the proposed TeTraKron-based subspace tracking algorithms are used for the tracking of spatial frequencies in time-varying scenarios.

Keywords: Subspace tracking; Tensors; Projection; Forward-backward-averaging

1 Introduction

The design of adaptive algorithms to track the subspace of an instationary random signal has a long standing history in signal processing. The main challenges are achieving a fast adaptation and a good steady-state behavior while keeping the computational complexity low. The first subspace-tracking schemes like [1] still had a complexity of $\mathcal{O}\{M^2 \cdot d\}$ where M is the number of channels (sensors) and d is the rank of the signal subspace. Later, as a singular value decomposition (SVD) technique and a data projection method (DPM) were proposed in [2] and [3], respectively, the complexity was lowered to $\mathcal{O}\{M \cdot d^2\}$.

Approaches that belong to a low complexity class with $\mathcal{O}\{M \cdot d\}$ operations have been developed, such as the scheme in [4] that estimates the eigenvector associated with the largest eigenvalue. Classified into the category of power-based methods similarly as DPM, the fast Rayleigh quotient-based adaptive noise subspace (FRANS) algorithm [5] combines DPM with a fast orthonormalization procedure. Another power-based approach that has the complexity of $\mathcal{O}\{M \cdot d\}$, as well as the fast approximated power iteration (FAPI) algorithm [6], is based on the power iteration method and a novel projection approximation. On the other hand, belonging to the category of projection approximation-based methods, the well-known projection approximation subspace tracking (PAST) algorithm [7] is one of the first low complexity subspace tracking schemes with $\mathcal{O}\{M \cdot d\}$ operations.

*Correspondence: y.cheng@tu-ilmenau.de

¹Communications Research Laboratory, Ilmenau University of Technology, P.O. Box 100565, Ilmenau 98684, Germany

Full list of author information is available at the end of the article

PAST interprets the signal subspace estimate as the solution of a minimization problem which can be simplified via an appropriate projection approximation and then applies a recursive least squares (RLS) procedure to track the signal subspace. A deflation-based version of PAST (PASTd) with an even lower complexity was proposed in [7]. In the literature, variants of some of the aforementioned algorithms have been developed to improve the numerical stability or to track the minor subspace. For a more detailed survey, the reader is referred to [8].

For stationary multidimensional signals, it has been shown that the subspace estimation accuracy can be significantly improved if tensors are used to store and manipulate the signals. A signal subspace estimate based on the Higher-Order Singular Value Decomposition (HOSVD) [9] was introduced in [10]. Therefore, extending this subspace estimation scheme to the tracking of the subspace of a time-varying multidimensional signal is of significant interest.

To this end, we introduce the Tensor-based subspace Tracking via Kronecker structured projections (TeTraKron) framework^a. TeTraKron allows to exploit the rich heritage of matrix-based subspace tracking and flexibly extends arbitrary matrix-based subspace tracking schemes to the tracking of the HOSVD-based subspace estimate defined in [10] by running them on all the unfoldings of the data tensor in parallel. Note that tracking the subspaces of all unfoldings of a tensor has been proposed before, e.g., in [11,12]. For the computer vision application, the incremental tensor subspace learning algorithm developed in [11] tracks the subspace of the unfoldings by using an incremental SVD approach. The unfoldings of the data tensor are projected onto the subspaces. An incremental tensor analysis framework and its variants were introduced in [12] to efficiently compute a compact summary of high-dimensional data and to reveal the hidden correlations. However, these approaches do not consider the recombination of these subspaces to the HOSVD-based subspace estimate from [10]. The computationally efficient recombination is the main focus of the TeTraKron framework. Moreover, [11,12] require to track the core tensor of the HOSVD which TeTraKron does not need at all.

The incorporation of forward-backward-averaging is also investigated. Employed by many parameter estimation algorithms as a preprocessing step, forward-backward-averaging virtually augments the observations and leads to an enhanced estimation accuracy. In addition, complex-valued measurements can be further conveniently and efficiently transformed to real-valued data [10]. Due to the fact that only real-valued computations are involved in the subsequent steps, the complexity is reduced. In this work, we show that after incorporating forward-backward-averaging, the real-valued HOSVD-

based subspace estimate can be obtained by applying a structured projection to the real-valued matrix-based subspace estimate as well. Consequently, the TeTraKron framework enables the extension of matrix-based subspace tracking schemes to realize real-valued tensor-based subspace tracking where forward-backward-averaging is included and provides benefits in terms of both the performance and the complexity. In time-varying multidimensional harmonic retrieval problems, we use the subspace estimates tracked via the TeTraKron-based subspace tracking schemes in ESPRIT-type parameter estimation algorithms [10] and evaluate their resulting performances.

This paper is organized as follows: Section 2 introduces the data model for the matrix-based and the tensor-based subspace estimation. The TeTraKron framework is described in detail in Section 3. The incorporation of forward-backward-averaging and real-valued subspace tracking are also discussed. Section 4 provides examples of how TeTraKron can be employed to develop tensor-based subspace tracking schemes based on PAST, PASTd, and FAPI. Section 5 presents numerical results before the conclusions are drawn in Section 6.

To facilitate the distinction between scalars, vectors, matrices, and tensors, the following notation is used throughout the manuscript: scalars are represented by italic letters, vectors by lowercase bold-faced letters, matrices by uppercase bold-faced letters, and tensors as bold-faced calligraphic letters. The superscripts T , H , $^{-1}$, and $*$ refer to matrix transposition, Hermitian transposition, matrix inversion, and complex conjugation, respectively. The Kronecker product is represented via \otimes and the Khatri-Rao (columnwise Kronecker) product via \diamond . An $M \times M$ identity matrix is symbolized by I_M . A matrix denoted by $I_{M \times d}$ ($M > d$) has the form $I_{M \times d} = [I_d \mathbf{0}_{d \times (M-d)}]^T$. The two-norm of a vector is denoted by $\|\cdot\|$. The operator $\text{Tri}\{\cdot\}$ calculates the upper/lower triangular part of its argument and copies its Hermitian transpose to the other lower/upper triangular part [7].

An R -way tensor with size I_r along mode $r = 1, 2, \dots, R$ is represented as $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_R}$. The r -mode vectors of \mathcal{A} are obtained by varying the r th index from 1 to I_r and keeping all other indices fixed. Aligning all r -mode vectors as the columns of a matrix yields the r -mode unfolding of \mathcal{A} which is denoted by $[\mathcal{A}]_{(r)} \in \mathbb{C}^{I_r \times I_{r+1} \times \dots \times I_{R-1} \times I_{R+1} \times \dots \times I_{R-1}}$. The order of the columns is arbitrary as long as it is chosen consistently. We use the reverse cyclical ordering, as proposed in [9]. The r -mode product between a tensor \mathcal{A} and a matrix \mathbf{U} is written as $\mathcal{A} \times_r \mathbf{U}$. It is computed by multiplying all r -mode vectors of \mathcal{A} with \mathbf{U} . In other words, $[\mathcal{A} \times_r \mathbf{U}]_{(r)} = \mathbf{U} \cdot [\mathcal{A}]_{(r)}$. The r -rank of a tensor \mathcal{A} is the rank of the r -mode unfolding matrix $[\mathcal{A}]_{(r)}$. The tensor $\mathcal{I}_{R,d}$ is an R -dimensional identity tensor of size $d \times d \times \dots \times d$, which is equal to one if all R indices are

equal and zero otherwise. In addition, $[\mathcal{A} \sqcup_r \mathcal{B}]$ symbolizes the concatenation of two tensors \mathcal{A} and \mathcal{B} along the r th mode [10].

2 Data model

In this section, we introduce the data model for both the matrix-based and the tensor-based subspace estimation. To this end, we start with the non-adaptive case where the subspaces are estimated once, based on N observations in a stationary window. We consider a linear mixture of d sources superimposed by additive noise, which can be expressed as

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{S} + \mathbf{W}. \quad (1)$$

Here, $\mathbf{X} \in \mathbb{C}^{M \times N}$ is the matrix of observations from M channels at N subsequent time instants, $\mathbf{A} \in \mathbb{C}^{M \times d}$ is the unknown mixing matrix or the array steering matrix, $\mathbf{S} \in \mathbb{C}^{d \times N}$ contains the unknown source symbols, and \mathbf{W} represents the additive noise samples. Then, the SVD of \mathbf{X} can be expressed as

$$\mathbf{X} = \begin{bmatrix} \hat{\mathbf{u}}_s & \hat{\mathbf{u}}_n \end{bmatrix} \cdot \begin{bmatrix} \hat{\Sigma}_s & \mathbf{0} \\ \mathbf{0} & \hat{\Sigma}_n \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{v}}_s & \hat{\mathbf{v}}_n \end{bmatrix}^H, \quad (2)$$

where the columns of $\hat{\mathbf{u}}_s \in \mathbb{C}^{M \times d}$ represent an orthonormal basis for the estimated signal subspace, i.e., $\text{span}\{\hat{\mathbf{u}}_s\} \approx \text{span}\{\mathbf{A}\}$.

We can arrange the elements of the matrix $\mathbf{X} \in \mathbb{C}^{M \times N}$ into a tensor $\mathcal{X} \in \mathbb{C}^{M_1 \times M_2 \times \dots \times M_R \times N}$, where $M = M_1 \cdot M_2 \dots \cdot M_R$. While such a rearrangement is always possible, it only provides a benefit if the actual underlying signal has a corresponding multidimensional structure, e.g., it resembles a signal sampled on a multidimensional lattice. These dimensions can for instance relate to space (1-D or 2-D arrays at transmitter or receiver), frequency, time, or polarization, depending on the application. The corresponding tensor-valued data model takes the following form [10]

$$\mathcal{X} = \mathcal{A} \times_{R+1} \mathbf{S}^T + \mathcal{W}, \quad (3)$$

where $\mathcal{A} \in \mathbb{C}^{M_1 \times M_2 \times \dots \times M_R \times d}$ and $\mathcal{W} \in \mathbb{C}^{M_1 \times M_2 \times \dots \times M_R \times N}$ represent the mixing tensor and the noise tensor, respectively. Since (3) is a rearranged version of (1), the corresponding quantities are linked via the relations $\mathbf{X} = [\mathcal{X}]_{R+1}^T$, $\mathbf{A} = [\mathcal{A}]_{R+1}^T$, and $\mathbf{W} = [\mathcal{W}]_{R+1}^T$, respectively. As shown in [10], based on (3), we can define a tensor-based subspace estimate by computing a truncated Higher-Order SVD (HOSVD) [9],

$$\mathcal{X} \approx \hat{\mathcal{S}}^{[s]} \times_1 \hat{\mathbf{u}}_1^{[s]} \times_2 \hat{\mathbf{u}}_2^{[s]} \dots \times_{R+1} \hat{\mathbf{u}}_{R+1}^{[s]}, \quad (4)$$

where $\hat{\mathbf{u}}_r^{[s]} \in \mathbb{C}^{M_r \times p_r}$ has unitary columns and denotes the matrix of the estimated r -mode singular vectors. Moreover, p_r is the r -rank of the mixing tensor \mathcal{A} , and $\hat{\mathcal{S}}^{[s]}$

$\in \mathbb{C}^{p_1 \times p_2 \times \dots \times p_{R+1}}$ represents the truncated core tensor that can be computed from \mathcal{X} via

$$\hat{\mathcal{S}}^{[s]} = \mathcal{X} \times_1 \hat{\mathbf{u}}_1^{[s]H} \times_2 \hat{\mathbf{u}}_2^{[s]H} \dots \times_{R+1} \hat{\mathbf{u}}_{R+1}^{[s]H}. \quad (5)$$

Based on the HOSVD, an improved signal subspace estimate is given by $\left[\hat{\mathbf{u}}^{[s]} \right]_{(R+1)}^T \in \mathbb{C}^{M \times d}$, where $\hat{\mathbf{u}}^{[s]}$ is [10]

$$\hat{\mathbf{u}}^{[s]} = \hat{\mathcal{S}}^{[s]} \times_1 \hat{\mathbf{u}}_1^{[s]} \times_2 \hat{\mathbf{u}}_2^{[s]} \dots \times_R \hat{\mathbf{u}}_R^{[s]} \times_{R+1} \hat{\Sigma}_s^{-1}, \quad (6)$$

where $\hat{\Sigma}_s$ has been defined in (2). Compared to the tensor-based subspace estimation in [10], the multiplication with $\hat{\Sigma}_s^{-1}$ represents only a normalization [13].

As discussed in [10], (6) provides a better subspace estimate than $\hat{\mathbf{u}}_s$ if and only if \mathcal{A} is r -rank deficient in at least one mode $r = 1, 2, \dots, R$, i.e., $p_r < M_r$. An example where this assumption is fulfilled is given by R -D harmonic retrieval [10], where we consider a superposition of d harmonics samples on an R -D lattice. This gives rise to a mixing matrix \mathbf{A} and a mixing tensor \mathcal{A} of the following form

$$\mathbf{A} = \mathbf{A}_1 \diamond \mathbf{A}_2 \diamond \dots \diamond \mathbf{A}_R \quad (7)$$

$$\mathcal{A} = \mathcal{I}_{R+1,d} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \dots \times_R \mathbf{A}_R, \quad (8)$$

where $\mathbf{A}_r \in \mathbb{C}^{M_r \times d}$ represents the mixing matrix in the r th mode. In this case, we have $p_r \leq d$ and, therefore, the tensor-based subspace estimate is superior to the matrix-based subspace estimate if $d < M_r$ for at least one $r = 1, 2, \dots, R$ [10]. However, there are applications with r -rank deficiencies where the observed signal obeys (3) but not (7), for instance, the tensor-based blind channel estimation scheme in [14].

3 Tensor subspace Tracking via Kronecker-structured projections (TeTraKron)

In a time-varying scenario, the observation matrix \mathbf{X} is augmented by a new column $\mathbf{x}(n) \in \mathbb{C}^{M \times 1}$ with every new snapshot n

$$\mathbf{x}(n) = \mathbf{A}(n) \cdot \mathbf{s}(n) + \mathbf{w}(n), \quad (9)$$

where $\mathbf{A}(n)$ is the mixing matrix or the array steering matrix for the n th snapshot. By employing a subspace tracking scheme, an estimate of the signal subspace $\hat{\mathbf{u}}_s(n)$ is obtained for each new snapshot n . In the tensor case, $\mathbf{x}(n) \in \mathbb{C}^{M \times 1}$ is rearranged into $\mathcal{X}(n) \in \mathbb{C}^{M_1 \times M_2 \times \dots \times M_R}$. A tensor-based subspace tracking algorithm aims at estimating the HOSVD-based subspace estimate $\left[\hat{\mathbf{u}}^{[s]}(n) \right]_{(R+1)}^T \in \mathbb{C}^{M \times d}$ for each new snapshot.

3.1 Tensor subspace estimation via structured projections

At first sight, (6) suggests that in order to track the signal subspace, we need to track the r -mode singular vectors as well as the core tensor. However, it can be shown that tracking the core tensor is indeed unnecessary, since the tensor-based subspace estimate can be computed from the matrix-based subspace estimate via a structured projection which does not involve the core tensor. This was first pointed out in [15] for the 2-D case. However, it can be generalized to an arbitrary number of dimensions. This claim is summarized in the following theorem:

Theorem 1. *The HOSVD-based subspace estimate can be computed by projecting the unstructured matrix-based subspace estimate obtained via the SVD onto a Kronecker structure in the following manner*

$$\left[\hat{\mathbf{U}}^{[s]}\right]_{(R+1)}^T = \left(\hat{\mathbf{T}}_1 \otimes \hat{\mathbf{T}}_2 \dots \otimes \hat{\mathbf{T}}_R\right) \cdot \hat{\mathbf{U}}_s, \quad (10)$$

where $\hat{\mathbf{T}}_r = \hat{\mathbf{U}}_r^{[s]} \cdot \hat{\mathbf{U}}_r^{[s]H}$ is a projection matrix onto the space spanned by the r -mode vectors.

Proof. [13] □

In the proof, the core tensor in (6) is first eliminated by substituting (5) into (6), and $\left[\hat{\mathbf{U}}^{[s]}\right]_{(R+1)}^T$ is then computed as

$$\left[\hat{\mathbf{U}}^{[s]}\right]_{(R+1)}^T = \left(\hat{\mathbf{T}}_1 \otimes \hat{\mathbf{T}}_2 \dots \otimes \hat{\mathbf{T}}_R\right) \cdot [\mathcal{X}]_{(R+1)}^T \cdot \hat{\mathbf{U}}_{R+1}^{[s]*} \cdot \hat{\Sigma}_s^{-1}.$$

Relying on the observation that $\mathbf{X} = [\mathcal{X}]_{(R+1)}^T$, $\hat{\mathbf{V}}_s = \hat{\mathbf{U}}_{R+1}^{[s]*}$, and

$$\mathbf{X} = \hat{\mathbf{U}}_s \cdot \hat{\Sigma}_s \cdot \hat{\mathbf{V}}_s^H,$$

which follows (2), the identity (10) is proved. Equation (10) provides the central idea behind the TeTraKron framework that we introduce in this paper. It shows that the tensor-based subspace estimate can be understood as a projection of the unstructured matrix-based subspace estimate onto the Kronecker structure inherent in the data. It also shows that for all modes where $p_r = M_r$, we have $\hat{\mathbf{T}}_r = \mathbf{I}_r$, i.e., no projection is performed. Another consequence we can draw from (10) is that there is no need to compute (or track) the core tensor. We can find the tensor-based subspace estimate only based on the r -mode subspaces contained in $\hat{\mathbf{U}}_r^{[s]}$. These are the subspaces obtained from the r -mode unfoldings of \mathcal{X} , which are again matrices. Therefore, any matrix-based subspace tracking scheme can be applied to track these subspaces as well.

Consequently, the main idea can be summarized as follows: in addition to tracking the subspace of the matrix \mathbf{X} (which is the same as tracking the row space of the $(R+1)$ -mode unfolding), we apply the same tracking algorithm to **all** r -mode unfoldings of the tensor which satisfy $p_r < M_r$ for $r = 1, 2, \dots, R$ in parallel. Note that even though this seems to increase the complexity by a factor equal to the number of modes we track, all these trackers can run in parallel which facilitates an efficient implementation. After each step, the tensor-based subspace estimate can be recombined via (10).

However, this recombination requires $\mathcal{O}\{M^2 \cdot d\}$ multiplications, i.e., it is quadratic in M , which is undesirable. To lower the complexity, we rewrite (10) as

$$\left[\hat{\mathbf{U}}^{[s]}\right]_{(R+1)}^T = \mathbf{U}_{\text{Kron}}^{[s]} \cdot \hat{\mathbf{U}}_s, \quad (11)$$

where $\mathbf{U}_{\text{Kron}}^{[s]} = \mathbf{U}_1^{[s]} \otimes \dots \otimes \mathbf{U}_R^{[s]} \in \mathbb{C}^{M \times d^R}$ and $\hat{\mathbf{U}}_s = \mathbf{U}_{\text{Kron}}^{[s]H} \cdot \hat{\mathbf{U}}_s \in \mathbb{C}^{d^R \times d}$ assuming $p_r = d \leq M_r$, for $r = 1, 2, \dots, R$. Note that the matrix product in (11) requires only $\mathcal{O}\{M \cdot d^R\}$ multiplications, i.e., it is linear in M . Moreover, (11) can be used for tensor-based subspace tracking as well: we track $\mathbf{U}_r^{[s]}$ for $r = 1, 2, \dots, R$ by applying matrix-based subspace tracking schemes to all unfoldings, then project our M -dimensional observations into a lower-dimensional space by premultiplying them with $\mathbf{U}_{\text{Kron}}^{[s]H} \in \mathbb{C}^{d^R \times M}$ and finally run a matrix-based subspace tracker on the lower-dimensional data to track the d -dimensional subspace $\hat{\mathbf{U}}_s \in \mathbb{C}^{d^R \times d}$.

TeTraKron allows to readily extend arbitrary matrix-based subspace tracking schemes to tensors which yields an improved estimation accuracy as we demonstrate in Section 5. Therefore, we obtain novel tensor-based subspace trackers by building on known algorithms, which is a particularly attractive feature of the TeTraKron framework. In addition to running these trackers on all unfoldings in parallel and recombining the signal subspace estimate via (10) or (11), the only modification we have to apply to the matrix-based subspace tracking schemes is the following: as introduced at the beginning of this section, it is typically assumed that the observation matrix \mathbf{X} is augmented by a new column $\mathbf{x}(n)$ with each new snapshot n . For the r -mode unfoldings of $\mathcal{X}(n) \in \mathbb{C}^{M_1 \times M_2 \dots \times M_R}$ that is a rearranged version of $\mathbf{x}(n)$, every new snapshot generates not only one but also several new columns. For instance, for the 1-mode unfolding, we obtain $\prod_{r=2}^R M_r$ new columns, each of size M_1 . This new batch of columns can be processed sequentially, or, by modifying the tracking schemes, also in one batch. We demonstrate such a modification using the examples of the PAST algorithm [7] and the FAPI algorithm [6] in Section 4.

3.2 Forward-backward-averaging and real-valued subspace tracking

Forward-backward-averaging and real-valued subspace estimation are introduced for the matrix case in [16] and the tensor case in [10], respectively. By employing forward-backward-averaging, the number of snapshots is virtually doubled. It also results in the decorrelation of two coherent sources. Moreover, the complex spatial covariance matrices can be mapped to the real-valued matrices such that the subsequent computations are real-valued, which contributes to a reduced complexity.

In this section, we revisit the concept of forward-backward-averaging and present real-valued subspace tracking. First, we calculate a forward-backward-averaged version of the measurement tensor $\mathcal{X} \in \mathbb{C}^{M_1 \times M_2 \times \dots \times M_R \times N}$ [10]

$$\mathcal{Z} = [\mathcal{X} \sqcup_{R+1} [\mathcal{X}^* \times_1 \mathbf{\Pi}_{M_1} \times_2 \mathbf{\Pi}_{M_2} \dots \times_{R+1} \mathbf{\Pi}_N]] \in \mathbb{C}^{M_1 \times M_2 \times \dots \times M_R \times 2N}, \quad (12)$$

where $\mathbf{\Pi}_p$ is a $p \times p$ exchange matrix which has ones on its anti-diagonal and zeros elsewhere. We further map this centro-Hermitian tensor to a real-valued tensor $\varphi(\mathcal{Z}) \in \mathbb{R}^{M_1 \times M_2 \times \dots \times M_R \times 2N}$ by using the transformation [10]^b

$$\varphi(\mathcal{Z}) = \mathcal{Z} \times_1 \mathbf{Q}_{M_1}^H \times_2 \mathbf{Q}_{M_2}^H \dots \times_{R+1} \mathbf{Q}_{2N}^T, \quad (13)$$

where \mathbf{Q}_p is a $p \times p$ left- $\mathbf{\Pi}$ real and unitary matrix, i.e., it satisfies $\mathbf{\Pi}_p \cdot \mathbf{Q}_p^* = \mathbf{Q}_p$ [17].

For the matrix-based case, the matrix of observations $\mathbf{X} \in \mathbb{C}^{M \times N}$ is mapped to the following centro-Hermitian matrix

$$\mathbf{Z} = [\mathbf{X} \mathbf{\Pi}_M \cdot \mathbf{X}^* \cdot \mathbf{\Pi}_N] \in \mathbb{C}^{M \times 2N}. \quad (14)$$

Then, \mathbf{Z} is transformed to a real-valued matrix using the transformation [16]

$$\varphi(\mathbf{Z}) = \mathbf{Q}_M^H \cdot \mathbf{Z} \cdot \mathbf{Q}_{2N}, \quad (15)$$

where \mathbf{Q}_M is defined as [10]

$$\mathbf{Q}_M = \mathbf{Q}_{M_1} \otimes \mathbf{Q}_{M_2} \dots \otimes \mathbf{Q}_{M_R}. \quad (16)$$

By computing a truncated HOSVD of $\varphi(\mathcal{Z})$, a real-valued subspace estimate is obtained as $[\hat{\mathcal{E}}^{[s]}]_{(R+1)}^T \in \mathbb{R}^{M \times d}$, where $\hat{\mathcal{E}}^{[s]}$ has the following form [10]

$$\hat{\mathcal{E}}^{[s]} = \hat{\mathcal{S}}_Z^{[s]} \times_1 \hat{\mathbf{E}}_1^{[s]} \times_2 \hat{\mathbf{E}}_2^{[s]} \dots \times_R \hat{\mathbf{E}}_R^{[s]} \times_{R+1} \hat{\mathbf{\Sigma}}_s'^{-1}. \quad (17)$$

The matrices $\hat{\mathbf{E}}_r^{[s]}$, $r = 1, 2, \dots, R + 1$, denote the estimates of the real-valued bases for the r -mode subspaces, and $\hat{\mathbf{\Sigma}}_s'$ represents the diagonal matrix of the $(R + 1)$ -mode singular values. The multiplication of $\hat{\mathbf{\Sigma}}_s'^{-1}$

is only a normalization procedure. By substituting the expression of the core tensor $\hat{\mathcal{S}}_Z^{[s]}$

$$\hat{\mathcal{S}}_Z^{[s]} = \varphi(\mathcal{Z}) \times_1 \hat{\mathbf{E}}_1^{[s]H} \times_2 \hat{\mathbf{E}}_2^{[s]H} \dots \times_{R+1} \hat{\mathbf{E}}_{R+1}^{[s]H} \quad (18)$$

into (17), $[\hat{\mathcal{E}}^{[s]}]_{(R+1)}^T \in \mathbb{R}^{M \times d}$ is further expressed as

$$[\hat{\mathcal{E}}^{[s]}]_{(R+1)}^T = \hat{\mathbf{\Sigma}}_s'^{-1} \cdot \hat{\mathbf{E}}_{R+1}^{[s]H} \cdot [\varphi(\mathcal{Z})]_{(R+1)} \times (\hat{\mathbf{T}}_1' \otimes \hat{\mathbf{T}}_2' \dots \otimes \hat{\mathbf{T}}_R')^T, \quad (19)$$

where $\hat{\mathbf{T}}_r' = \hat{\mathbf{E}}_r^{[s]} \cdot \hat{\mathbf{E}}_r^{[s]H}$. Substituting (12) into (13) gives

$$\begin{aligned} \varphi(\mathcal{Z}) &= [[\mathcal{X} \times_1 \mathbf{Q}_{M_1}^H \dots \times_R \mathbf{Q}_{M_R}^H] \sqcup_{R+1} \\ &[\mathcal{X}^* \times_1 (\mathbf{Q}_{M_1}^H \cdot \mathbf{\Pi}_{M_1}) \dots \times_R (\mathbf{Q}_{M_R}^H \cdot \mathbf{\Pi}_{M_R}) \times_{R+1} \mathbf{\Pi}_N]] \\ &\times_{R+1} \mathbf{Q}_{2N}^T. \end{aligned} \quad (20)$$

Notice that

$$\begin{aligned} &(\mathbf{Q}_{M_1}^H \cdot \mathbf{\Pi}_{M_1}) \otimes (\mathbf{Q}_{M_2}^H \cdot \mathbf{\Pi}_{M_2}) \dots \otimes (\mathbf{Q}_{M_R}^H \cdot \mathbf{\Pi}_{M_R}) \\ &= \mathbf{Q}_M^H \cdot (\mathbf{\Pi}_{M_1} \otimes \mathbf{\Pi}_{M_2} \dots \otimes \mathbf{\Pi}_{M_R}) \\ &= \mathbf{Q}_M^H \cdot \mathbf{\Pi}_M. \end{aligned} \quad (21)$$

Expanding the $(R + 1)$ -mode unfolding of (20) yields

$$[\varphi(\mathcal{Z})]_{(R+1)}^T = \mathbf{Q}_{2N}^T \cdot \begin{bmatrix} [\mathcal{X}]_{(R+1)} \cdot (\mathbf{Q}_M^H)^T \\ \mathbf{\Pi}_N \cdot [\mathcal{X}]_{(R+1)}^* \cdot (\mathbf{Q}_M^H \cdot \mathbf{\Pi}_M)^T \end{bmatrix}. \quad (22)$$

As there exists a link between the matrix-based data model and its tensor-based counterpart such that $\mathbf{X} = [\mathcal{X}]_{(R+1)}^T$, we can express $[\varphi(\mathcal{Z})]_{(R+1)}^T$ as

$$[\varphi(\mathcal{Z})]_{(R+1)}^T = \mathbf{Q}_M^H \cdot [\mathbf{X} \mathbf{\Pi}_M \cdot \mathbf{X}^* \cdot \mathbf{\Pi}_N] \cdot \mathbf{Q}_{2N}. \quad (23)$$

Recall that $\varphi(\mathcal{Z})$ is defined via (14) and (15), leading to the observation that

$$\varphi(\mathcal{Z}) = [\varphi(\mathcal{Z})]_{(R+1)}^T. \quad (24)$$

As a real-valued equivalent to Theorem 1, we have

$$[\hat{\mathcal{E}}^{[s]}]_{(R+1)}^T = (\hat{\mathbf{T}}_1' \otimes \hat{\mathbf{T}}_2' \dots \otimes \hat{\mathbf{T}}_R') \cdot \hat{\mathbf{E}}_s, \quad (25)$$

where the columns of $\hat{\mathbf{E}}_s \in \mathbb{R}^{M \times d}$ represent a real-valued orthonormal basis for the estimated signal subspace in the matrix-based case. The proof proceeds along the lines of the proof of Theorem 1 in [13].

Hence, (25) indicates that the real-valued tensor-based subspace estimate can be computed by applying a Kronecker-structured projection to the real-valued matrix-based subspace estimate similar to the complex-valued case. The calculation of the core tensor is also not required. Note that the obtained real-valued tensor-based subspace estimate can be employed in the unitary tensor

ESPRIT algorithm [10] for parameter estimation in multidimensional harmonic retrieval problems. We show the corresponding numerical results in Section 5.

To summarize, in case that forward-backward-averaging is incorporated, the TeTraKron framework can be employed to extend a matrix-based subspace tracking scheme to tensors such that real-valued tensor-based subspace tracking is realized. For each snapshot, a forward-backward-averaged version of the measurement tensor is mapped to a real-valued tensor similarly as (12) and (13). Then, any matrix-based subspace tracker can be run on all unfoldings of this tensor in parallel where only real-valued computations are involved. Finally, the real-valued subspace estimate is obtained via a recombination procedure such as (25)^c. In Section 4, where the matrix-based algorithms PAST [7] and FAPI [6] are used as examples, we explain explicitly how real-valued tensor-based subspace tracking based on the TeTraKron framework is performed.

4 Examples

In this section, we provide examples of how the TeTraKron framework can be used to devise tensor-based subspace tracking schemes. Since TeTraKron allows us to extend an arbitrary matrix-based subspace tracking scheme to the tensor case, we choose two examples, namely, the simple but widely used PAST algorithm [7] as well as the FAPI scheme [6] that is a fast implementation of the power iteration method for subspace tracking.

4.1 Tensor-based PAST/PASTd

The PAST algorithm for tracking the signal subspace is summarized in Algorithm 1, where $\mathbf{x}(n)$ is the new measurement vector at time n , $\mathbf{P}(n)$ corresponds to the inverse of the correlation matrix of the projected vector $\mathbf{y}(n) = \hat{\mathbf{U}}_s^H(n) \cdot \mathbf{x}(n)$, which is approximated as $\mathbf{y}(n) = \hat{\mathbf{U}}_s^H(n-1) \cdot \mathbf{x}(n)$. Moreover, $\mathbf{g}(n)$ is the gain vector and β the forgetting factor of the underlying RLS procedure. Finally, $\mathbf{e}(n)$ is the approximation error and $\mathbf{I}_{M \times r}$ symbolizes the first r columns of an $M \times M$ identity matrix.

Algorithm 1 Summary of the PAST algorithm [7]

$$\mathbf{P}(0) = \mathbf{I}_d, \hat{\mathbf{U}}_s(0) = \mathbf{I}_{M \times d}$$

FOR $n = 1, 2, \dots$ DO

$$\mathbf{y}(n) = \hat{\mathbf{U}}_s^H(n-1) \cdot \mathbf{x}(n)$$

$$\mathbf{h}(n) = \mathbf{P}(n-1) \cdot \mathbf{y}(n)$$

$$\mathbf{g}(n) = \mathbf{h}(n) / (\beta + \mathbf{y}^H(n) \cdot \mathbf{h}(n))$$

$$\mathbf{P}(n) = \beta^{-1} \cdot \text{Tri}\{\mathbf{P}(n-1) - \mathbf{g}(n) \cdot \mathbf{h}^H(n)\}$$

$$\mathbf{e}(n) = \mathbf{x}(n) - \hat{\mathbf{U}}_s(n-1) \cdot \mathbf{y}(n)$$

$$\hat{\mathbf{U}}_s(n) = \hat{\mathbf{U}}_s(n-1) + \mathbf{e}(n) \cdot \mathbf{g}^H(n)$$

END

Recall that the data tensor observed at each new snapshot $\mathcal{X}(n) \in \mathbb{C}^{M_1 \times M_2 \times \dots \times M_R}$ is a rearranged version of $\mathbf{x}(n) \in \mathbb{C}^{M \times 1}$. In the TeTraKron extension of PAST, we apply the same algorithm to all unfoldings of $\mathcal{X}(n)$, i.e., $[\mathcal{X}(n)]_{(r)} \in \mathbb{C}^{M_r \times \frac{M}{M_r}}$, $r = 1, 2, \dots, R$. For instance, in the $R = 2$ -dimensional case, i.e., our data tensor \mathcal{X} is of size $M_1 \times M_2 \times N$. With each new observation vector $\mathbf{x}(n) \in \mathbb{C}^{M_1 \cdot M_2 \times 1}$, we obtain a new matrix of observations for the one-space and the two-space of \mathcal{X} which is given by $\tilde{\mathbf{X}}(n) \in \mathbb{C}^{M_1 \times M_2}$ for $[\mathcal{X}]_{(1)}$ and $\tilde{\mathbf{X}}^T(n)$ for $[\mathcal{X}]_{(2)}$. Note that $\tilde{\mathbf{X}}(n)$ is a rearranged version of $\mathbf{x}(n)$ which satisfies $\text{vec}\{\tilde{\mathbf{X}}(n)\} = \mathbf{x}(n)$.

Since PAST is based on RLS, it can be modified to process the entire new batch of observations at the same time. The modified update equations for the r -mode unfolding $[\mathcal{X}(n)]_{(r)} \in \mathbb{C}^{M_r \times \frac{M}{M_r}}$ become

$$\mathbf{Y}_r(n) = \hat{\mathbf{U}}_r^{[s]H}(n-1) \cdot [\mathcal{X}(n)]_{(r)} \quad (26)$$

$$\mathbf{H}_r(n) = \mathbf{P}_r(n-1) \cdot \mathbf{Y}_r(n) \quad (27)$$

$$\mathbf{G}_r(n) = \mathbf{H}_r(n) \cdot \left(\beta \cdot \mathbf{I}_{M/M_r} + \mathbf{Y}_r^H(n) \cdot \mathbf{H}_r(n) \right)^{-1} \quad (28)$$

$$\mathbf{P}_r(n) = \beta^{-1} \cdot \text{Tri}\{\mathbf{P}_r(n-1) - \mathbf{G}_r(n) \cdot \mathbf{H}_r^H(n)\} \quad (29)$$

$$\mathbf{F}_r(n) = [\mathcal{X}(n)]_{(r)} - \hat{\mathbf{U}}_r^{[s]}(n-1) \cdot \mathbf{Y}_r(n) \quad (30)$$

$$\hat{\mathbf{U}}_r^{[s]}(n) = \hat{\mathbf{U}}_r^{[s]}(n-1) + \mathbf{F}_r(n) \cdot \mathbf{G}_r^H(n). \quad (31)$$

Notice that to process the batch of $\frac{M}{M_r}$ columns, the inverse of a $\frac{M}{M_r} \times \frac{M}{M_r}$ matrix is involved. Especially for $R > 2$, it is usually true that $d < \frac{M}{M_r}$. In such cases, it becomes computationally more efficient to update the $d \times d$ correlation matrix of $\mathbf{Y}_r(n)$ given by

$$\mathbf{C}_{y_r y_r}(n) = \beta \cdot \mathbf{C}_{y_r y_r}(n-1) + \mathbf{Y}_r(n) \cdot \mathbf{Y}_r^H(n), \quad (32)$$

and directly calculate its inverse $\mathbf{P}_r(n) = \mathbf{C}_{y_r y_r}^{-1}(n)$. Multiplying $(\beta \cdot \mathbf{I}_{M/M_r} + \mathbf{Y}_r^H(n) \cdot \mathbf{H}_r(n))$ to the right of both sides of (28) yields

$$\mathbf{G}_r(n) \cdot (\beta \cdot \mathbf{I}_{M/M_r} + \mathbf{Y}_r^H(n) \cdot \mathbf{H}_r(n)) = \mathbf{H}_r(n). \quad (33)$$

Then, $\mathbf{G}_r(n)$ is expressed as

$$\mathbf{G}_r(n) = \beta^{-1} \cdot (\mathbf{H}_r(n) - \mathbf{G}_r(n) \cdot \mathbf{Y}_r^H(n) \cdot \mathbf{H}_r(n)). \quad (34)$$

Substituting (27) into (34) gives

$$\begin{aligned} \mathbf{G}_r(n) &= \beta^{-1} \cdot (\mathbf{P}_r(n-1) - \mathbf{G}_r(n) \cdot \mathbf{Y}_r^H(n) \cdot \mathbf{P}_r(n-1)) \cdot \mathbf{Y}_r(n) \\ &= \beta^{-1} \cdot (\mathbf{P}_r(n-1) - \mathbf{G}_r(n) \cdot \mathbf{H}_r^H(n)) \cdot \mathbf{Y}_r(n). \end{aligned} \quad (35)$$

Knowing that

$$\mathbf{C}_{y_r y_r}^{-1}(n) = \mathbf{P}_r(n) = \beta^{-1} \cdot (\mathbf{P}_r(n-1) - \mathbf{G}_r(n) \cdot \mathbf{H}_r^H(n)), \quad (36)$$

while the operation $\text{Tri}\{\cdot\}$ in (29) is only employed to preserve the Hermitian symmetry of $\mathbf{P}_r(n)$ in the presence of rounding errors [7], $\mathbf{G}_r(n)$ is alternatively updated as

$$\mathbf{G}_r(n) = \mathbf{C}_{y_r y_r}^{-1}(n) \cdot \mathbf{Y}_r(n). \quad (37)$$

Equations (27), (28), and (29) are replaced by (32) and (37). The deflation-based version of PAST, PASTd [7], is also based on RLS and can hence be modified in the same manner. We summarize the tensor-based PASTd algorithm for updating the r -mode subspace estimates in Algorithm 2, where $\hat{\mathbf{u}}_{r,i}^{[s]}(n)$ denotes the i th column of $\hat{\mathbf{U}}_r^{[s]}(n) \in \mathbb{C}^{M_r \times d}$, and $d_{r,i}(n)$ represents the i th entry of $\mathbf{d}_r(n) \in \mathbb{R}^d$ that contains the eigenvalue estimates. The initial values of the eigenvalue estimates are chosen to be 1 [7], i.e., $\mathbf{d}_r(0)$ is an all-ones vector.

Algorithm 2 Summary of the tensor-based version of the PASTd algorithm [7] for updating the r -mode subspace estimates

$$\bar{\mathbf{X}}_{r,1}(n) = [\mathcal{X}(n)]_{(r)}$$

FOR $i = 1, 2, \dots, d$ DO

$$\mathbf{y}_{r,i}^T(n) = \hat{\mathbf{u}}_{r,i}^{[s]H}(n-1) \cdot \bar{\mathbf{X}}_{r,i}(n)$$

$$d_{r,i}(n) = \beta \cdot d_{r,i}(n-1) + \|\mathbf{y}_{r,i}(n)\|^2$$

$$\mathbf{F}_{r,i}(n) = \bar{\mathbf{X}}_{r,i}(n) - \hat{\mathbf{u}}_{r,i}^{[s]}(n-1) \cdot \mathbf{y}_{r,i}^T(n)$$

$$\hat{\mathbf{u}}_{r,i}^{[s]}(n) = \hat{\mathbf{u}}_{r,i}^{[s]}(n-1) + \mathbf{F}_{r,i}(n) \cdot (\mathbf{y}_{r,i}^*(n)/d_{r,i}(n))$$

$$\bar{\mathbf{X}}_{r,i+1}(n) = \bar{\mathbf{X}}_{r,i}(n) - \hat{\mathbf{u}}_{r,i}^{[s]}(n) \cdot \mathbf{y}_{r,i}^T(n)$$

END

When forward-backward-averaging is incorporated, each new observation vector $\mathbf{x}(n)$ is augmented by a new virtual column. In the matrix-based case, the real-valued subspace estimate $\hat{\mathbf{E}}_s(n)$ is obtained by applying the PAST algorithm to

$$\varphi(\mathbf{Z}(n)) = \mathbf{Q}_M^H \cdot \mathbf{Z}(n) \cdot \mathbf{Q}_2, \quad (38)$$

where $\mathbf{Z}(n)$ is a forward-backward-averaged version of $\mathbf{x}(n)$ given by

$$\mathbf{Z}(n) = [\mathbf{x}(n) \quad \mathbf{\Pi}_M \cdot \mathbf{x}^*(n)]. \quad (39)$$

In the tensor-based case, a real-valued tensor $\varphi(\mathcal{Z}(n)) \in \mathbb{R}^{M_1 \times M_2 \times \dots \times M_R \times 2}$ is computed by using $N = 1$ in (13)

$$\varphi(\mathcal{Z}(n)) = \mathcal{Z}(n) \times_1 \mathbf{Q}_{M_1}^H \times_2 \mathbf{Q}_{M_2}^H \cdots \times_{R+1} \mathbf{Q}_2^T, \quad (40)$$

where $\mathcal{Z}(n)$ is a forward-backward-averaged version of $\mathcal{X}(n)$

$$\begin{aligned} \mathcal{Z}(n) &= [\mathcal{X}(n) \sqcup_{R+1} [\mathcal{X}^*(n) \times_1 \mathbf{\Pi}_{M_1} \times_2 \mathbf{\Pi}_{M_2} \cdots \times_R \mathbf{\Pi}_{M_R}]] \\ &\in \mathbb{C}^{M_1 \times M_2 \times \dots \times M_R \times 2}. \end{aligned} \quad (41)$$

In the new real-valued tensor-based subspace tracking scheme that is an extension of PAST based on the TeTraKron framework, we run the PAST algorithm on all unfoldings of the real-valued tensor $\varphi(\mathcal{Z}(n))$ in parallel. For the r -mode unfolding $[\varphi(\mathcal{Z}(n))]_{(r)} \in \mathbb{R}^{M_r \times 2 \cdot \frac{M}{M_r}}$ where $p_r < M_r$, the updated equations where only real-valued computations are involved are as follows:

$$\mathbf{Y}_r(n) = \hat{\mathbf{E}}_r^{[s]T}(n-1) \cdot [\varphi(\mathcal{Z}(n))]_{(r)} \quad (42)$$

$$\mathbf{C}_{y_r y_r}(n) = \beta \cdot \mathbf{C}_{y_r y_r}(n-1) + \mathbf{Y}_r(n) \cdot \mathbf{Y}_r^T(n) \quad (43)$$

$$\mathbf{G}_r(n) = \mathbf{C}_{y_r y_r}^{-1}(n-1) \cdot \mathbf{Y}_r(n) \quad (44)$$

$$\mathbf{F}_r(n) = [\varphi(\mathcal{Z}(n))]_{(r)} - \hat{\mathbf{E}}_r^{[s]}(n-1) \cdot \mathbf{Y}_r(n) \quad (45)$$

$$\hat{\mathbf{E}}_r^{[s]}(n) = \hat{\mathbf{E}}_r^{[s]}(n-1) + \mathbf{F}_r(n) \cdot \mathbf{G}_r^T(n). \quad (46)$$

Afterwards, the real-valued signal subspace estimate can be recombined via (25).

4.2 Tensor-based FAPI

As an additional example of how to apply the TeTraKron framework to extend a matrix-based subspace tracker to the tensor case, we consider the exponential window FAPI algorithm [6]. As before, to update the r -mode subspace estimate, the matrix-based subspace tracker has to be modified to process a batch of $\frac{M}{M_r}$ observations. This can be carried out at the same time as in the aforementioned example of the extended PAST algorithm. Alternatively, each column of the r -mode unfolding of $\mathcal{X}(n)$ can be treated as a new observation vector, and the columns are processed sequentially^d. The r -mode subspace estimates can be updated via exactly the same procedures as in the matrix-based subspace tracking scheme. However, only when the first column of the r -mode unfolding of $\mathcal{X}(n)$ is used as an equivalent new observation vector, the forgetting factor stays the same as that used in the matrix-based subspace tracking algorithm at each snapshot. Starting from the second column that is treated as the second equivalent observation vector, the forgetting factor is set to one. In the following, we explain in detail how this approach is applied to obtain a tensor-based version of the exponential window FAPI algorithm [6] via the TeTraKron framework. Denote the ℓ th column of the r -mode unfolding $[\mathcal{X}(n)]_{(r)} \in \mathbb{C}^{M_r \times \frac{M}{M_r}}$ as $\mathbf{x}_\ell^{(r)} \in \mathbb{C}^{M_r}$, where $\ell = 1, 2, \dots, \frac{M}{M_r}$. The details on updating $\hat{\mathbf{u}}_r^{[s]}$ are given in Algorithm 3. For each snapshot, after obtaining $\hat{\mathbf{u}}_r^{[s]}$

($r = 1, 2, \dots, R$) via the procedures in Algorithm 3 and the matrix-based subspace estimate $\hat{\mathbf{U}}_s$ by employing exponential window FAPI [6], we recombine them via (10) to obtain the tensor-based subspace estimate $\left[\hat{\mathbf{U}}^{[s]}\right]_{(R+1)}^T$.

Algorithm 3 Summary of the tensor-based version of the exponential window FAPI algorithm [6] for updating the r -mode subspace estimates

$\mathbf{P}'_r(0) = \mathbf{P}_r(n-1)$, $\hat{\mathbf{U}}'_r(0) = \hat{\mathbf{U}}_r^{[s]}(n-1)$

FOR $\ell = 1, 2, \dots, \frac{M}{M_r}$ DO

IF $\ell = 1$

$\beta' = \beta$

ELSE

$\beta' = 1$

exponential window FAPI [6]

$\mathbf{y}_r(\ell) = \hat{\mathbf{U}}_r^H(\ell-1) \cdot \mathbf{x}_\ell^{(r)}$

$\mathbf{h}_r(\ell) = \mathbf{P}'_r(\ell-1) \cdot \mathbf{y}_r(\ell)$

$\mathbf{g}_r(\ell) = \mathbf{h}_r(\ell) / (\beta' + \mathbf{y}_r^H(\ell) \cdot \mathbf{h}_r(\ell))$

$\epsilon_r^2(\ell) = \|\mathbf{x}_\ell^{(r)}\|^2 - \|\mathbf{y}_r(\ell)\|^2$

$\tau_r(\ell) = \frac{\epsilon_r^2(\ell)}{1 + \epsilon_r^2(\ell) \cdot \|\mathbf{g}_r(\ell)\|^2 + \sqrt{1 + \epsilon_r^2(\ell) \cdot \|\mathbf{g}_r(\ell)\|^2}}$

$\eta_r(\ell) = 1 - \tau_r(\ell) \cdot \|\mathbf{g}_r(\ell)\|^2$

$\mathbf{y}'_r(\ell) = \eta_r(\ell) \cdot \mathbf{y}_r(\ell) + \tau_r(\ell) \cdot \mathbf{g}_r(\ell)$

$\mathbf{h}'_r(\ell) = \mathbf{P}'_r(\ell-1) \cdot \mathbf{y}'_r(\ell)$

$\epsilon_r(\ell) = \frac{\tau_r(\ell)}{\eta_r(\ell)} \cdot \left(\mathbf{P}'_r(\ell-1) \cdot \mathbf{g}_r(\ell) - \left(\mathbf{h}'_r(\ell) \cdot \mathbf{g}_r(\ell) \right) \cdot \mathbf{g}_r(\ell) \right)$

$\mathbf{P}'_r(\ell) = \frac{1}{\beta'} \cdot \left(\mathbf{P}'_r(\ell-1) - \mathbf{g}_r(\ell) \cdot \mathbf{h}'_r(\ell) + \epsilon_r(\ell) \cdot \mathbf{g}_r^H(\ell) \right)$

$\mathbf{e}'_r(\ell) = \eta_r(\ell) \cdot \mathbf{x}_\ell^{(r)} - \hat{\mathbf{U}}'_r(\ell-1) \cdot \mathbf{y}'_r(\ell)$

$\hat{\mathbf{U}}'_r(\ell) = \hat{\mathbf{U}}'_r(\ell-1) + \mathbf{e}'_r(\ell) \cdot \mathbf{g}_r^H(\ell)$

END

$\mathbf{P}_r(n) = \mathbf{P}'_r(\frac{M}{M_r})$, $\hat{\mathbf{U}}_r^{[s]}(n) = \hat{\mathbf{U}}'_r(\frac{M}{M_r})$

In case that forward-backward-averaging is incorporated, similar procedures as in Algorithm 3 are employed on the r -mode unfolding of the real-valued tensor, $[\varphi(\mathcal{Z}(n))]_{(r)} \in \mathbb{R}^{M_r \times 2 \cdot \frac{M}{M_r}}$, to update the real-valued estimate of the r -mode subspace $\hat{\mathbf{E}}_r^{[s]}$, i.e., $\hat{\mathbf{U}}_r^{[s]}(n-1)$, $\frac{M}{M_r}$, and $\mathbf{x}_\ell^{(r)}$ in Algorithm 3 are replaced by $\hat{\mathbf{E}}_r^{[s]}(n-1)$, $2 \cdot \frac{M}{M_r}$, and the ℓ th column of $[\varphi(\mathcal{Z}(n))]_{(r)}$, where $\ell = 1, 2, \dots, 2 \cdot \frac{M}{M_r}$, respectively. Then, the real-valued signal subspace estimate can be recombined via (25).

4.3 Summary of the proposed tensor-based subspace tracking schemes

To this end, we summarize the tensor-based subspace tracking algorithms developed via the TeTraKron framework in Algorithms 4 and 5. For the schemes^e that have been employed in the simulations (cf. Section 5), the corresponding equations are specified.

5 Simulation results

In this section, we first demonstrate the performance of the tensor extension of PAST and PASTd achieved via the proposed TeTraKron framework. To this end, we choose a simulation scenario that represents an extension of the one shown in [7] to $R = 2$ dimensions. We consider a uniform rectangular array (URA) with $d = 3$ impinging wavefronts. The first two sources are moved by changing their spatial frequencies (direction cosines) as a function of the time index $n = 1, 2, \dots, N$ according to

$$\begin{aligned} \mu_1^{(1)}[n] &= 0.3 - 0.1 \cdot t[n], \quad \mu_2^{(1)}[n] = 0.2 + 0.1 \cdot t[n], \\ \mu_1^{(2)}[n] &= 0.2 + 0.1 \cdot t[n], \quad \mu_2^{(2)}[n] = 0.2 + 0.1 \cdot t[n], \end{aligned}$$

for $t[n] = \frac{n-1}{N-1}$, whereas the third source remains stationary at $\mu_3^{(1)} = \mu_3^{(2)} = 0.1$. Therefore, for n close to $N/2$ the first and the second sources cross. Note that $\mu_i^{(r)}$ represents the spatial frequency of the i th source in the r th dimension for $i = 1, 2, \dots, d$ and $r = 1, 2, \dots, R$. The total number of snapshots N is set to 1,000 in all examples. The source samples as well as the noise samples are drawn from a zero mean circularly symmetric complex Gaussian distribution with variance one (SNR = 0 dB). We choose the forgetting factor $\beta = 0.97$ for all examples shown in this section. Similar to [7], we compare the algorithms based on the largest principal angle (LPA) between the true and the estimated signal subspace since the LPA provides a measure for the agreement of the subspaces which is invariant to the particular choice of the basis.

Figure 1 shows the LPA for a 9×9 URA. The curve labeled ‘PAST’ refers to the original matrix-based PAST algorithm from [7]. As summarized in Section 4.3, TeTraKron-PAST and TeTraKron-PAST II refer to the tensor extensions of PAST via the proposed TeTraKron framework based on (10) and the reduced-complexity version (11), respectively. For reference, we display two curves labeled ‘SVD’ and ‘HOSVD’ where the entire matrix/tensor of observations^f up to the current snapshot n is used to calculate a subspace estimate via the SVD and the HOSVD, respectively.

In Figure 2, we replace PAST by PASTd. Moreover, we change the array size to a 7×7 URA to demonstrate that the tensor gain is present for different array sizes. Both simulation results show that the tensor-based subspace tracking algorithms outperform the matrix-based algorithms, as expected.

Algorithm 4 Summary of TeTraKron-PAST, TeTraKron-PASTd, TeTraKron-PAST II, TeTraKron-PASTd II, and TeTraKron-FAPI

Initialization:

$$\hat{\mathbf{U}}_s(0) = \mathbf{I}_{M \times d}$$

$$\hat{\mathbf{U}}_r^{[s]}(0) = \mathbf{I}_{M_r \times d} \text{ for } r = 1, 2, \dots, R$$

TeTraKron-PAST / TeTraKron-PAST II: $\mathbf{P}(0) = \mathbf{I}_d, \mathbf{C}_{y_r y_r}(0) = \mathbf{I}_d$ for $r = 1, 2, \dots, R$

TeTraKron-PASTd / TeTraKron-PASTd II: initialize $\mathbf{d}(0)$ and $\mathbf{d}_r(0)$ ($r = 1, 2, \dots, R$) as d -dimensional all-ones vectors

TeTraKron-FAPI: $\mathbf{P}(0) = \mathbf{I}_d, \mathbf{P}_r(0) = \mathbf{I}_d$ for $r = 1, 2, \dots, R$

FOR $n = 1, 2, \dots$ DO

Step 1: Update the r -mode subspace estimates $\hat{\mathbf{U}}_r^{[s]}(n)$ ($r = 1, 2, \dots, R$) in parallel

Step 2: Update the matrix-based subspace estimate $\hat{\mathbf{U}}_s(n)$ or $\hat{\mathbf{U}}_s(n)$ (as defined in (11))

Step 3: Update the tensor-based subspace estimate $\left[\hat{\mathbf{U}}^{[s]}(n)\right]_{(R+1)}^T$

	Step 1	Step 2	Step 3
TeTraKron-PAST	Equations (26), (32), (37), (30), and (31)	Algorithm 1 or [7]	Equation (10)
TeTraKron-PASTd	Algorithm 2	[7]	
TeTraKron-FAPI	Algorithm 3	[6]	
TeTraKron-PAST II	Equations (26), (32), (37), (30), and (31)	Algorithm 1 or [7]	Equation (11)
TeTraKron-PASTd II	Algorithm 2	[7]	

END

Algorithm 5 Summary of TeTraKron-PAST with FBA and TeTraKron-FAPI with FBA

Initialization:

$$\hat{\mathbf{E}}_s(0) = \mathbf{I}_{M \times d}$$

$$\hat{\mathbf{E}}_r^{[s]}(0) = \mathbf{I}_{M_r \times d} \text{ for } r = 1, 2, \dots, R$$

TeTraKron-PAST with FBA: $\mathbf{P}(0) = \mathbf{I}_d, \mathbf{C}_{y_r y_r}(0) = \mathbf{I}_d$ for $r = 1, 2, \dots, R$

TeTraKron-FAPI with FBA: $\mathbf{P}(0) = \mathbf{I}_d, \mathbf{P}_r(0) = \mathbf{I}_d$ for $r = 1, 2, \dots, R$

FOR $n = 1, 2, \dots$ DO

Step 1: Calculate the real-valued tensor $\varphi(\mathcal{Z}(n))$ via (40) and (41)

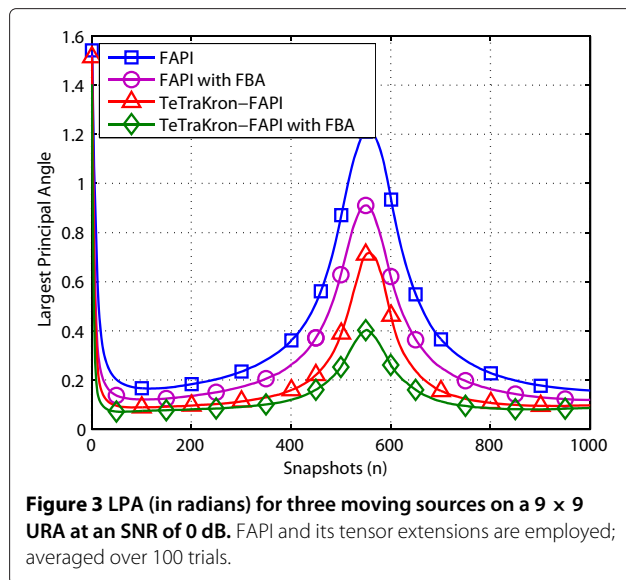
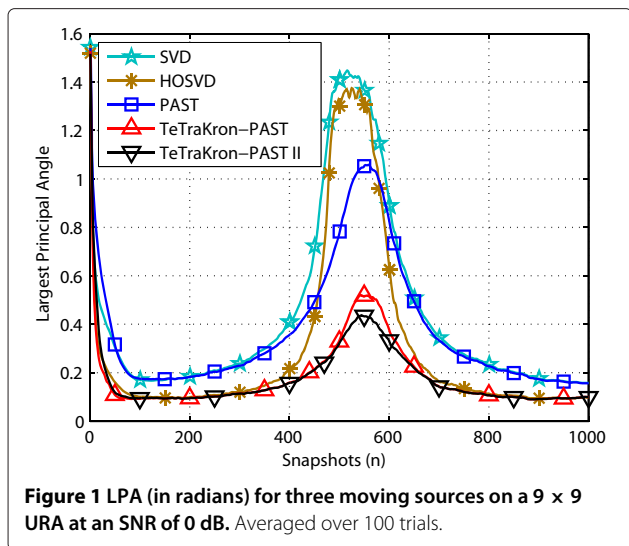
Step 2: Update the real-valued r -mode subspace estimates $\hat{\mathbf{E}}_r^{[s]}(n)$ ($r = 1, 2, \dots, R$) in parallel

Step 3: Update the real-valued matrix-based subspace estimate $\hat{\mathbf{E}}_s(n)$

Step 4: Update the real-valued tensor-based subspace estimate $\left[\hat{\mathcal{E}}^{[s]}(n)\right]_{(R+1)}^T$

	Step 1	Step 2	Step 3	Step 4
TeTraKron-PAST with FBA	Equations (40) and (41)	Equations (42) to (46)	Algorithm 1 or [7]	Equation (25)
TeTraKron-FAPI with FBA		Algorithm 3	[6]	

END



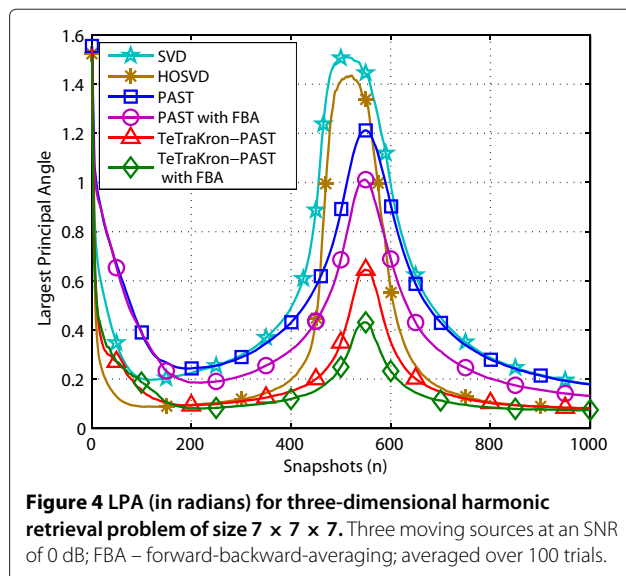
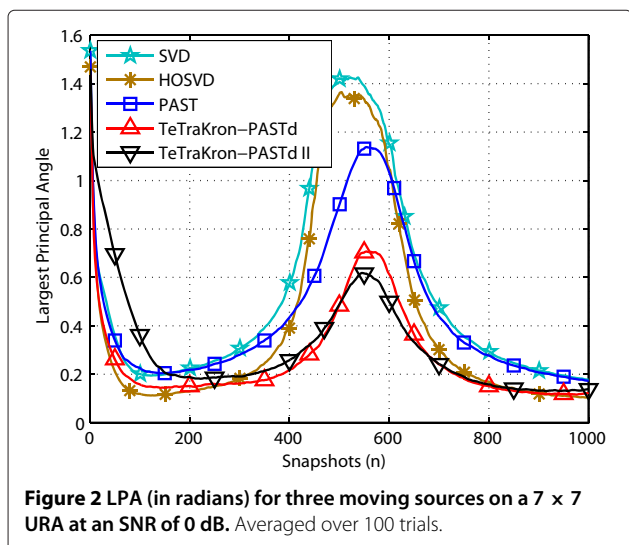
Using the same scenario as for Figure 1, we evaluate the performance of the tensor extensions of the FAPI algorithm [6]. In Figure 3, where the LPA is plotted, the curves labeled ‘TeTraKron-FAPI’ and ‘TeTraKron-FAPI with FBA’ correspond to two tensor-based FAPI schemes that are extensions of the FAPI algorithm via the TeTraKron framework based on (10) and (25), respectively. The curve labeled ‘FAPI with FBA’ corresponds to an extended version of FAPI where forward-backward-averaging is included (cf. (38)). It is observed that the performance of tensor-based FAPI is superior to that of the matrix-based FAPI algorithm. A gain is further obtained by incorporating forward-backward-averaging.

In the fourth example, a three-dimensional harmonic retrieval problem of size $7 \times 7 \times 7$ is simulated. As an extension of the two-dimensional scenarios shown previously, the spatial frequencies (direction cosines) of the first two

sources are represented as a function of the time index $n = 1, 2, \dots, N$

$$\begin{aligned} \mu_1^{(1)}[n] &= 0.3 - 0.1 \cdot t[n], & \mu_2^{(1)}[n] &= 0.2 + 0.1 \cdot t[n], \\ \mu_1^{(2)}[n] &= 0.2 + 0.1 \cdot t[n], & \mu_2^{(2)}[n] &= 0.2 + 0.1 \cdot t[n], \\ \mu_1^{(3)}[n] &= 0.2 + 0.1 \cdot t[n], & \mu_2^{(3)}[n] &= 0.2 + 0.1 \cdot t[n], \end{aligned}$$

for $t[n] = \frac{n-1}{N-1}$, whereas the third source remains stationary at $\mu_3^{(1)} = \mu_3^{(2)} = \mu_3^{(3)} = 0.1$. Similarly, as n approaches $N/2$, the first and the second sources cross. The LPA between the true and the estimated signal subspace is illustrated in Figure 4. The curve labeled ‘TeTraKron-PAST with FBA’ refers to the real-valued subspace tracking scheme as a tensor extension of PAST via the proposed



TeTraKron framework based on (25) where forward-backward-averaging is incorporated. On the other hand, the curve labeled 'PAST with FBA' corresponds to an extended version of the matrix-based PAST algorithm with forward-backward-averaging included (cf. (38)). It can be observed that incorporating forward-backward-averaging contributes to a performance improvement for both the matrix-based and the tensor-based algorithms. Similarly as in the first three examples, the tensor-based subspace tracking schemes provide a better performance compared to the matrix-based schemes due to the fact that the former exploit the multidimensional structure inherent in the data in a better way.

Moreover, we assess the performance of various ESPRIT-type parameter estimation algorithms that use the subspace estimates obtained by the proposed subspace tracking schemes and least squares (LS) to solve the invariance equations [10]. The evaluation criterion is the root mean square estimation error (RMSE). We define the RMSE in the spatial frequency domain as

$$\text{RMSE} = \sqrt{\mathbb{E} \left\{ \frac{1}{d} \frac{1}{R} \sum_{i=1}^d \sum_{r=1}^R \left(\mu_i^{(r)} - \hat{\mu}_i^{(r)} \right)^2 \right\}}, \quad (47)$$

where $\hat{\mu}_i^{(r)}$ represents an estimate of $\mu_i^{(r)}$.

A two-dimensional scenario is considered where the sources are assumed to be correlated. The source samples \mathbf{s} are generated such that $\mathbf{R}_{\text{ss}} = \mathbb{E} \{ \mathbf{s} \cdot \mathbf{s}^H \}$ has the form

$$\mathbf{R}_{\text{ss}} = \begin{bmatrix} 1 & \rho_{1,2} & \rho_{1,3} \\ \rho_{2,1} & 1 & \rho_{2,3} \\ \rho_{3,1} & \rho_{3,2} & 1 \end{bmatrix}, \quad (48)$$

where $\rho_{i,j} = \rho \cdot e^{j\varphi_{ij}}$, and $\varphi_{ij} = -\varphi_{ji}$ are drawn from a uniform distribution in $[0, 2\pi]$ for $i = 1, 2, 3$, $j = 1, 2, 3$, and $i \neq j$. Here, ρ is chosen as 0.7. The other parameters are the same as for Figure 2.

We plot the LPA in Figure 5. Similar observations as in the previous examples can be obtained. In addition, the performance improvement due to the incorporation of forward-backward-averaging is more significant in the presence of source correlation compared to the case of uncorrelated sources. The RMSE of the tracked spatial frequencies is shown in Figure 6. Four parameter estimation techniques, standard ESPRIT (SE) [18], unitary ESPRIT (UE) [16], standard tensor ESPRIT (STE) [10], and unitary tensor-ESPRIT (UTE) [10] are employed where the subspace estimates are tracked by PAST, PAST with forward-backward-averaging, tensor extension of PAST via TeTraKron, and tensor extension of PAST via TeTraKron with forward-backward-averaging incorporated, respectively. It can be observed that the tensor-based parameter estimation algorithms combined with tensor-based subspace tracking schemes outperform the

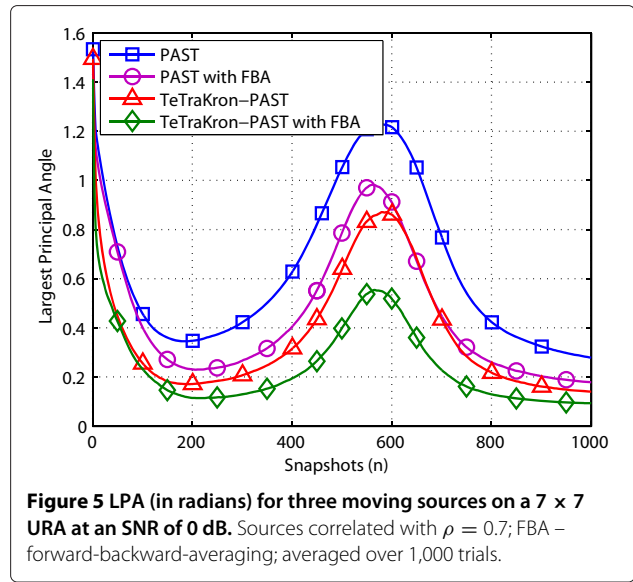


Figure 5 LPA (in radians) for three moving sources on a 7×7 URA at an SNR of 0 dB. Sources correlated with $\rho = 0.7$; FBA – forward-backward-averaging; averaged over 1,000 trials.

combinations of matrix-based parameter estimation and matrix-based subspace tracking algorithms. Moreover, a gain is achieved by incorporating forward-backward-averaging. This corroborates the benefits provided by forward-backward-averaging that it decorrelates coherent sources and contributes to an enhanced accuracy of the parameter estimation.

In Figures 7 and 8, the corresponding spatial frequency estimates of the first mode and the second mode averaged over 1,000 trials are illustrated, respectively. To demonstrate the variation of the spatial frequency estimates around the averaged values, we plot the bars that

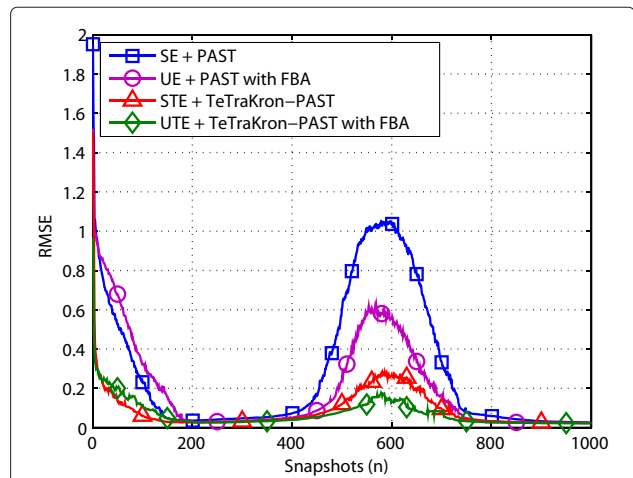
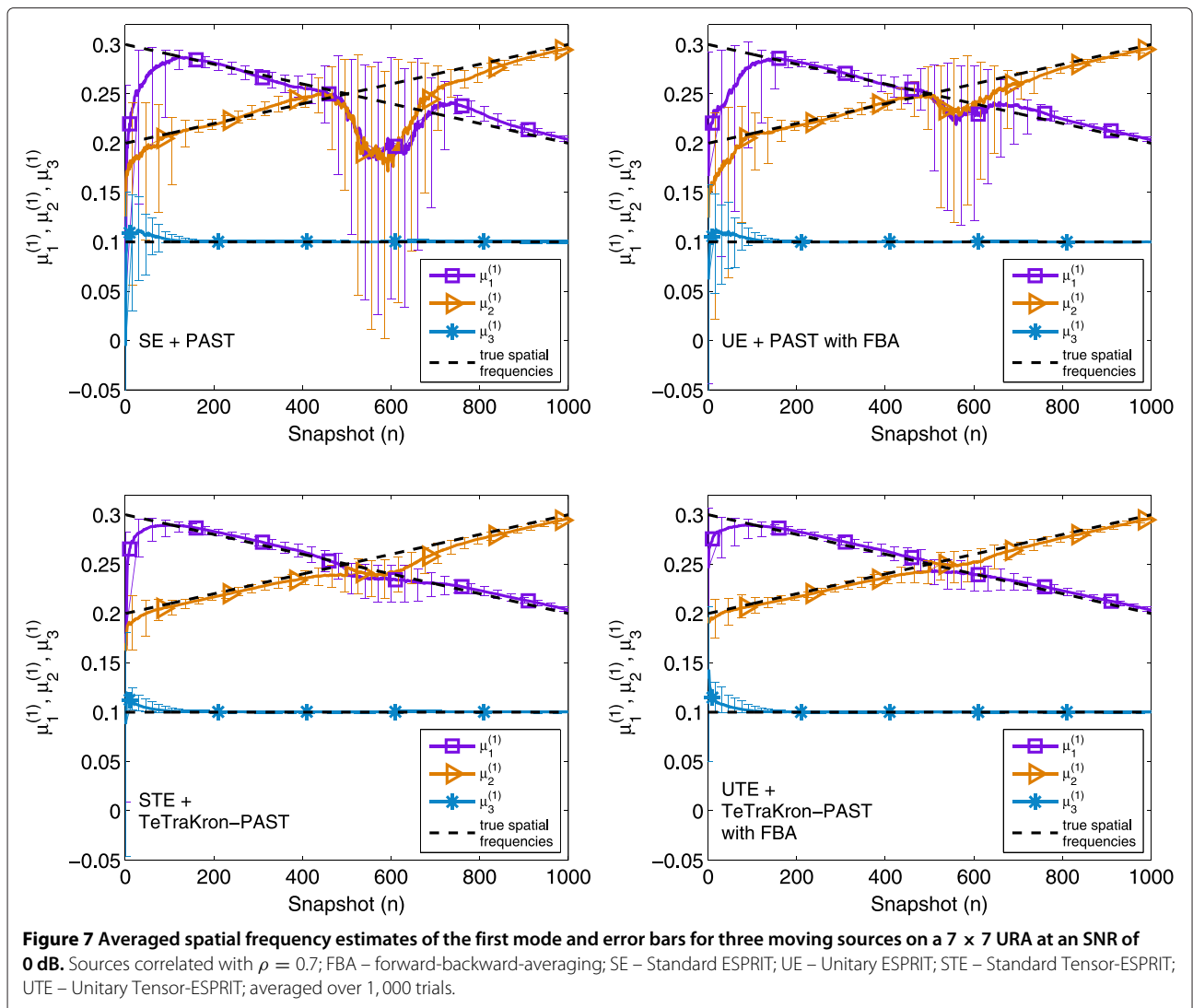


Figure 6 RMSE of spatial frequencies for three moving sources on a 7×7 URA at an SNR of 0 dB. Sources correlated with $\rho = 0.7$; FBA – forward-backward-averaging; SE – Standard ESPRIT; UE – Unitary ESPRIT; STE – Standard Tensor-ESPRIT; UTE – Unitary Tensor-ESPRIT; averaged over 1,000 trials.

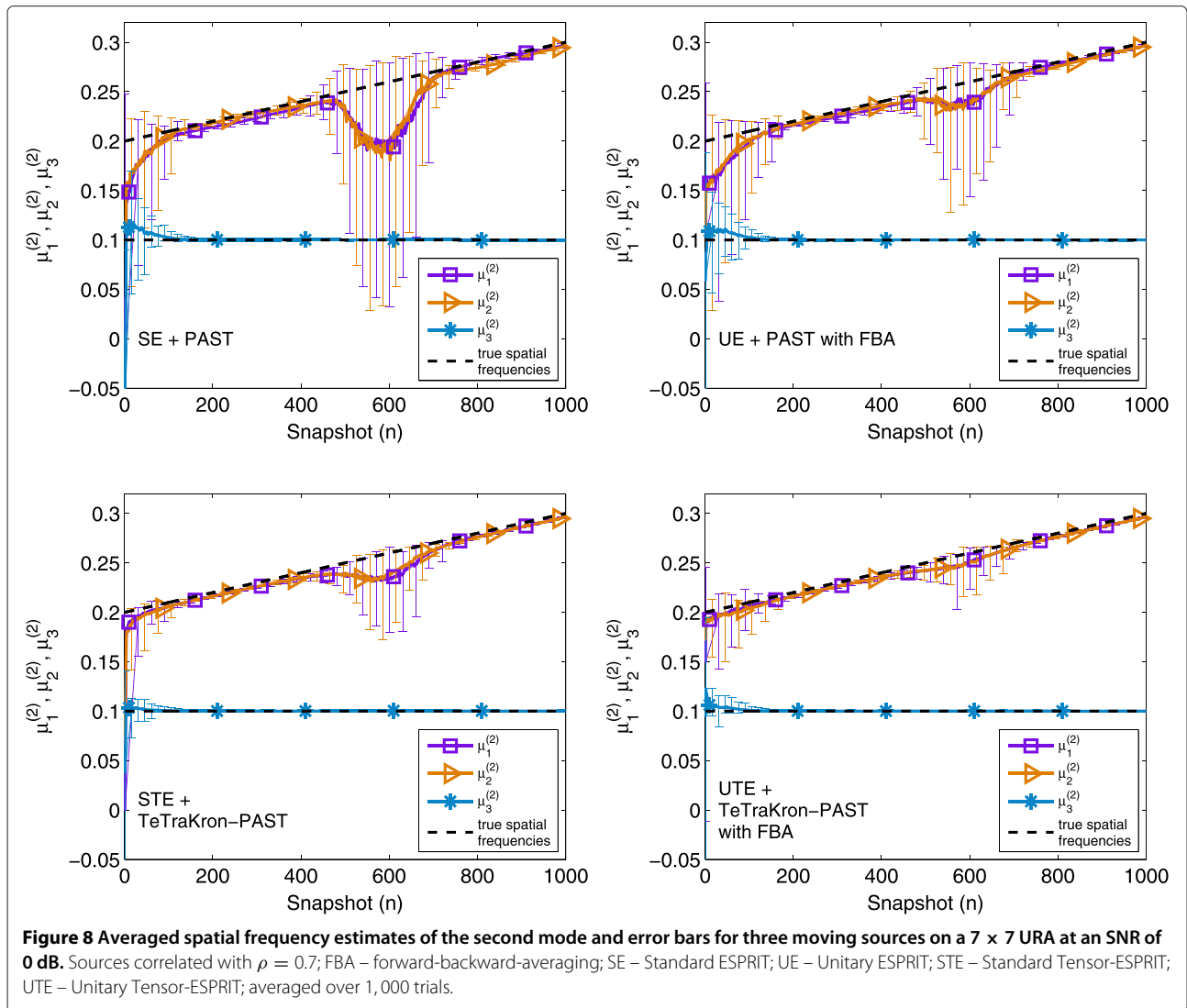


represent the ± 1 standard deviation estimated from the 1,000 trials as well. The two combinations of tensor-based subspace tracking and parameter estimation techniques (corresponding to the two plots labeled ‘STE + TeTraKron-PAST’ and ‘UTE + TeTraKron-PAST with FBA’, respectively, at the bottom of Figures 7 and 8) achieve an accurate estimation of the spatial frequencies. Even when the first two sources cross, their performances do not suffer much, and a fast adaptation is observed. In case of unitary tensor ESPRIT combined with the TeTraKron extension of PAST where forward-backward-averaging is included, its performance is slightly better than that of the combination of standard tensor ESPRIT and the TeTraKron extension of PAST. By contrast, the matrix-based algorithms standard ESPRIT and unitary ESPRIT combined with PAST adapt much slower at the beginning of the tracking as well as at the crossing point of the first two sources and fail to accurately estimate the

spatial frequencies. The deviation of the spatial frequency tracks from the average is also much more severe compared to the deviation in case of the two combinations of tensor-based subspace tracking and parameter estimation techniques.

6 Conclusions

In this paper, we have proposed the Tensor-based subspace Tracking via Kronecker structured projections (TeTraKron) framework. TeTraKron allows to extend arbitrary existing matrix-based subspace tracking schemes to the tracking of the HOSVD-based subspace estimate. Therefore, compared to previous matrix-based subspace tracking schemes, the subspace estimation accuracy is improved. The extension is based on an algebraic link between matrix-based and tensor-based subspace estimates via a Kronecker structured projection. Therefore, matrix-based subspace tracking schemes are applied to



all tensor unfoldings, and there is no need to track the core tensor. We have proposed a low-complexity approach for the recombination of the separate subspaces into one final estimate which is achieved in linear complexity. In addition, we have investigated the incorporation of forward-backward-averaging. To this end, a connection between the real-valued matrix-based and the HOSVD-based subspace estimate via a similar Kronecker structured projection has been revealed. Consequently, the TeTraKron framework can also be employed to devise real-valued tensor-based subspace tracking schemes. As an example, we have used the TeTraKron framework to extend the PAST, the PASTd, and the FAPI algorithms to tensors and demonstrated the enhanced subspace estimation accuracy via numerical simulations. In time-varying multidimensional harmonic retrieval problems, we have investigated the performances of standard tensor ESPRIT and unitary tensor ESPRIT where the tensor-based

subspace estimates tracked by the TeTraKron-based subspace tracking algorithms are used for direction-of-arrival estimation.

Endnotes

^aParts of this paper have been published at the IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2013), Saint Martin, French Antilles, December 2013.

^bHere, \mathbf{Q}_{2N}^T is used instead of \mathbf{Q}_{2N}^H as originally proposed in [10] where this transformation was introduced. This facilitates to build the link between the matrix-based data model and its tensor-based counterpart, i.e., (24).

^cAlternatively, after tracking the r -mode subspaces, a projected lower-dimensional subspace similar to $\hat{\mathbf{U}}_s$ can be tracked, and a recombination procedure similar to (11) can be used, leading to a reduced complexity.

^dBoth approaches, namely, the batch processing and the sequential processing, can be employed in the tensor-based PAST algorithm. We have performed simulations to compare these two methods. It has been observed that when they are used to modify the PAST algorithm to update the r -mode subspace estimates, they lead to the same performance. It should be noted that for some state-of-the-art matrix-based subspace tracking schemes, such as the exponential window FAPI algorithm [6], they have been developed based on the fact that each new observation takes the form of a column vector. Compared to the batch processing, this important feature is better preserved in the sequential processing. Consequently, when extending these schemes to the tensor case via TeTraKron, it is more convenient to use the sequential processing.

^eTeTraKron-PAST, TeTraKron-PASTd, and TeTraKron-FAPI refer to the tensor extensions of PAST [7], PASTd [7], and FAPI [6] via the proposed TeTraKron framework based on (10), respectively. In case of the reduced-complexity version (11), the corresponding tensor extensions of PAST and PASTd are called TeTraKron-PAST II and TeTraKron-PASTd II, respectively. In addition, we use 'TeTraKron-PAST with FBA' and 'TeTraKron-FAPI with FBA' to refer to the real-valued subspace tracking schemes as tensor extensions of PAST and FAPI, respectively. Here the TeTraKron framework based on (25) is applied, and forward-backward-averaging is incorporated.

^fTo render the comparison to the adaptive RLS-based schemes fair, the exponential weighting with a forgetting factor β is also applied here.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Communications Research Laboratory, Ilmenau University of Technology, P.O. Box 100565, Ilmenau 98684, Germany. ²Digital Broadcasting Research Laboratory, Ilmenau University of Technology, P.O. Box 100565, Ilmenau 98684, Germany.

Received: 14 April 2014 Accepted: 14 July 2014

Published: 7 August 2014

References

1. N Owsley, Adaptive data orthogonalization, in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '78)*, vol. 3 (Tulsa, OK, 1978), pp. 109–112
2. I Karasalo, Estimating the covariance matrix by signal subspace averaging. *IEEE Trans. Acoustics, Speech Signal Process.* **34**(1), 8–12 (1986)
3. J-F Yang, M Kaveh, Adaptive eigensubspace algorithms for direction or frequency estimation and tracking. *IEEE Trans. Acoust. Speech Signal Process.* **36**(2), 241–251 (1988)
4. E Oja, A simplified neuron model as a principal components analyzer. *J. Math. Biol.* **15**(3), 267–273 (1982)
5. S Attallah, K Abed-Meraim, Low-cost adaptive algorithm for noise subspace estimation. *Electron. Lett.* **38**(12), 609–611 (2002)
6. R Badeau, B David, G Richard, Fast approximated power iteration subspace tracking. *IEEE Trans. Signal Process.* **53**(8), 2931–2941 (2005)
7. B Yang, Projection approximation subspace tracking. *IEEE Trans. Signal Process.* **43**(1), 95–107 (1995)
8. JP Delmas, Subspace tracking for signal processing, in *Adaptive Signal Processing: Next Generation Solutions*, ed. by T Adali, S Haykin (John Wiley & Sons, Inc. Hoboken, 2010), pp. 211–270
9. L de Lathauwer, B de Moor, J Vanderwalle, A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* **21**(4), 1253–1278 (2000)
10. M Haardt, F Roemer, G Del Galdo, Higher-order SVD based subspace estimation to improve the parameter estimation accuracy in multi-dimensional harmonic retrieval problems. *IEEE Trans. Sig. Proc.* **56**, 3198–3213 (2008)
11. W Hu, X Li, X Zhang, X Shi, S Maybank, Z Zhang, Incremental tensor subspace learning and its applications to foreground segmentation and tracking. *Int. J. Comput. Vis.* **91**(3), 303–327 (2011)
12. J Sun, D Tao, S Papadimitriou, PS Yu, C Faloutsos, Incremental tensor analysis: theory and applications. *ACM Trans. Knowl. Discov. Data (TKDD)*. **2**(3) (2008)
13. F Roemer, M Haardt, G Del Galdo, Analytical performance assessment of multi-dimensional matrix- and tensor-based ESPRIT-type algorithms. *IEEE Trans. Signal Process.* **62**(10), 2611–2625 (2014)
14. B Song, F Roemer, M Haardt, Blind estimation of SIMO channels using a tensor-based subspace method, in *Proc. 44-th Asilomar Conf. on Signals, Systems, and Computers (CA Pacific Grove, 2010)*
15. F Roemer, H Becker, M Haardt, M Weis, Analytical performance evaluation for HOSVD-based parameter estimation schemes, in *Proc. of the IEEE Int. Workshop on Comp. Adv. in Multi-Sensor Adaptive Proc. (CAMSAP 2009)* (Aruba Dutch Antilles, 2009)
16. M Haardt, JA Nosssek, Unitary ESPRIT: How to obtain increased estimation accuracy with a reduced computational burden. *IEEE Trans. Sig. Proc.* **43**(5), 1232–1242 (1995)
17. A Lee, Centrohermitian and skew-centrohermitian matrices. *Linear Algebra Its Appl.* **29**, 205–210 (1980)
18. R Roy, T Kailath, ESPRIT—Estimation of signal parameters via rotational invariance techniques. *IEEE Trans. Acoust. Speech, Signal Process.* **37**, 984–995 (1989)

doi:10.1186/1687-6180-2014-123

Cite this article as: Cheng et al.: Tensor subspace Tracking via Kronecker structured projections (TeTraKron) for time-varying multidimensional harmonic retrieval. *EURASIP Journal on Advances in Signal Processing* 2014 **2014**:123.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com