Tensor subspace tracking via Kronecker structured projections (TeTraKron)

Florian Roemer
Digital Broadcasting Research Laboratory
Ilmenau University of Technology
Ilmenau, Germany
florian.roemer@tu-ilmenau.de

Emin-Koray Kasnakli, *Yao Cheng*, *Martin Haardt*
Communications Research Laboratory
Ilmenau University of Technology
Ilmenau, Germany
koraykasnakli@gmail.com, {y.cheng, martin.haardt}@tu-ilmenau.de

***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 time-varying scenarios, the TeTraKron-based tracking schemes outperform the original algorithms as well as the batch solutions provided by the SVD and the HOSVD.**

I. 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. While the first subspace-tracking schemes like [7] still had a complexity of $\mathcal{O}\left\{M^2 \cdot d\right\}$ where M is the number of channels (sensors) and d is the rank of the signal subspace, this was later lowered to $\mathcal{O}\left\{M \cdot d^2\right\}$ [5] or even $\mathcal{O}\left\{M \cdot d\right\}$ [6]. Overall, a very large number of subspace tracking schemes is known, for a survey the reader is referred to [2].

For stationary multi-dimensional 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 SVD (HOSVD) [1] was introduced in [3]. 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. TeTraKron extends arbitrary matrix-based subspace tracking schemes to the tracking of the HOSVD-based subspace estimate defined in [3] 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 [4, 10]. However, these approaches do not consider the recombination of these subspaces to the HOSVD-based subspace estimate from [3]. The computationally efficient recombination is the main focus of the TeTraKron framework. Moreover, [4, 10] require to track the core tensor of the HOSVD which TeTraKron does not need at all.

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 lower-case bold-faced letters, matrices by upper-case bold-faced letters, and tensors as bold-faced calligraphic letters. The superscripts $^{\mathrm{T}}$, $^{\mathrm{H}}$, $^{-1}$, 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 $\diamond$. The operator $\mathrm{Tri}\{\cdot\}$ calculates the upper/lower triangular part of its argument and copies its Hermitian transpose to the other lower/upper triangular part [11].

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

II. 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

$$\boldsymbol{X} = \boldsymbol{A} \cdot \boldsymbol{S} + \boldsymbol{W}. \tag{1}$$

Here, $\boldsymbol{X} \in \mathbb{C}^{M \times N}$ is the matrix of observations from M channels at N subsequent time instants, $\boldsymbol{A} \in \mathbb{C}^{M \times d}$ is the unknown mixing matrix, $\boldsymbol{S} \in \mathbb{C}^{d \times N}$ contains the unknown source symbols, and $\boldsymbol{W}$ represents the additive noise samples.

This work has been performed while Florian Roemer was with the Communications Research Laboratory, Ilmenau University of Technology.

Then, the SVD of $\boldsymbol{X}$ can be expressed as

$$\boldsymbol{X} = \left[\hat{\boldsymbol{U}}_{\rm s} \;\; \hat{\boldsymbol{U}}_{\rm n}\right] \cdot \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_{\rm s} & \boldsymbol{0} \\ \boldsymbol{0} & \hat{\boldsymbol{\Sigma}}_{\rm n} \end{bmatrix} \cdot \left[\hat{\boldsymbol{V}}_{\rm s} \;\; \hat{\boldsymbol{V}}_{\rm n}\right]^{\rm H}, \quad (2)$$

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

We can arrange the elements of the matrix $\boldsymbol{X} \in \mathbb{C}^{M \times N}$ into a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M_1 \times M_2 \ldots \times M_R \times N}$, where $M = M_1 \cdot M_2 \ldots \cdot M_R$. 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 [3]

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

where $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times M_2 \ldots \times M_R \times d}$ and $\boldsymbol{\mathcal{W}} \in \mathbb{C}^{M_1 \times M_2 \ldots \times M_R \times N}$ 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 $\boldsymbol{X} = [\boldsymbol{\mathcal{X}}]^{\rm T}_{(R+1)}$, $\boldsymbol{A} = [\boldsymbol{\mathcal{A}}]^{\rm T}_{(R+1)}$, and $\boldsymbol{W} = [\boldsymbol{\mathcal{W}}]^{\rm T}_{(R+1)}$, respectively. As shown in [3], based on (3) we can define a tensor-based subspace estimate by computing a truncated Higher-Order SVD (HOSVD) [1],

$$\boldsymbol{\mathcal{X}} \approx \hat{\boldsymbol{\mathcal{S}}}^{[\rm s]} \times_1 \hat{\boldsymbol{U}}_1^{[\rm s]} \times_2 \hat{\boldsymbol{U}}_2^{[\rm s]} \ldots \times_{R+1} \hat{\boldsymbol{U}}_{R+1}^{[\rm s]}, \quad (4)$$

where $\hat{\boldsymbol{U}}_r^{[\rm s]} \in \mathbb{C}^{M_r \times p_r}$ 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 $\boldsymbol{\mathcal{A}}$ and $\hat{\boldsymbol{\mathcal{S}}}^{[\rm s]} \in \mathbb{C}^{p_1 \times p_2 \ldots \times p_{R+1}}$ represents the truncated core tensor. Based on the HOSVD, an improved signal subspace estimate is given by $\left[\hat{\boldsymbol{\mathcal{U}}}^{[\rm s]}\right]^{\rm T}_{(R+1)} \in \mathbb{C}^{M \times d}$, where $\hat{\boldsymbol{\mathcal{U}}}^{[\rm s]}$ is [3]

$$\hat{\boldsymbol{\mathcal{U}}}^{[\rm s]} = \hat{\boldsymbol{\mathcal{S}}}^{[\rm s]} \times_1 \hat{\boldsymbol{U}}_1^{[\rm s]} \times_2 \hat{\boldsymbol{U}}_2^{[\rm s]} \ldots \times_R \hat{\boldsymbol{U}}_R^{[\rm s]} \times_{R+1} \hat{\boldsymbol{\Sigma}}_{\rm s}^{-1}. \quad (5)$$

Compared to the tensor-based subspace estimation in [3], the multiplication with $\hat{\boldsymbol{\Sigma}}_{\rm s}^{-1}$ represents only a normalization which we introduce to simplify notation later on.

As discussed in [3], (5) provides a better subspace estimate than $\hat{\boldsymbol{U}}_{\rm s}$ if and only if $\boldsymbol{\mathcal{A}}$ is r-rank deficient in at least one mode $r = 1, 2, \ldots, R$, i.e., $p_r < M_r$, An example where this assumption is fulfilled is given by R-D harmonic retrieval [3], where we consider a superposition of d harmonics samples on an R-D lattice. This gives rise to a mixing matrix $\boldsymbol{A}$ and a mixing tensor $\boldsymbol{\mathcal{A}}$ of the following form

$$\boldsymbol{A} = \boldsymbol{A}_1 \diamond \boldsymbol{A}_2 \diamond \ldots \diamond \boldsymbol{A}_R \quad (6)$$

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{I}}_{R+1,d} \times_1 \boldsymbol{A}_1 \times_2 \boldsymbol{A}_2 \ldots \times_R \boldsymbol{A}_R, \quad (7)$$

where $\boldsymbol{A}_r \in \mathbb{C}^{M_r \times d}$ 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 $r = 1, 2 \ldots, R$. 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 [9].

III. Tensor subspace estimation via structured projections

At first sight (5) 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 [8] 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*

$$\left[\hat{\boldsymbol{\mathcal{U}}}^{[\rm s]}\right]^{\rm T}_{(R+1)} = \left(\hat{\boldsymbol{T}}_1 \otimes \hat{\boldsymbol{T}}_2 \ldots \otimes \hat{\boldsymbol{T}}_R\right) \cdot \hat{\boldsymbol{U}}_{\rm s}, \quad (8)$$

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

Proof: To show the identity (8) we first need to eliminate the core tensor in (5). This is achieved by observing that $\hat{\boldsymbol{\mathcal{S}}}^{[\rm s]}$ can be computed from $\boldsymbol{\mathcal{X}}$ via

$$\hat{\boldsymbol{\mathcal{S}}}^{[\rm s]} = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{U}_1^{[\rm s]^{\rm H}} \ldots \times_{R+1} \boldsymbol{U}_{R+1}^{[\rm s]^{\rm H}}. \quad (9)$$

Substituting (9) into (5) yields

$$\hat{\boldsymbol{\mathcal{U}}}^{[\rm s]} = \boldsymbol{\mathcal{X}} \times_1 \hat{\boldsymbol{T}}_1 \ldots \times_R \hat{\boldsymbol{T}}_R \times_{R+1} \left(\hat{\boldsymbol{\Sigma}}_{\rm s}^{-1} \cdot \hat{\boldsymbol{U}}_{R+1}^{[\rm s]^{\rm H}}\right). \quad (10)$$

Expanding the $(R+1)$-mode unfolding of (10) gives

$$\left[\hat{\boldsymbol{\mathcal{U}}}^{[\rm s]}\right]_{(R+1)} = \hat{\boldsymbol{\Sigma}}_{\rm s}^{-1} \cdot \hat{\boldsymbol{U}}_{R+1}^{[\rm s]^{\rm H}} \cdot [\boldsymbol{\mathcal{X}}]_{(R+1)} \left(\hat{\boldsymbol{T}}_1 \otimes \hat{\boldsymbol{T}}_2 \ldots \otimes \hat{\boldsymbol{T}}_R\right)^{\rm T}.$$

Therefore, it remains to be shown that $\hat{\boldsymbol{\Sigma}}_{\rm s}^{-1} \cdot \hat{\boldsymbol{U}}_{R+1}^{[\rm s]^{\rm H}} \cdot [\boldsymbol{\mathcal{X}}]_{(R+1)} = \hat{\boldsymbol{U}}_{\rm s}^{\rm T}$. This is achieved by observing that $[\boldsymbol{\mathcal{X}}]_{(R+1)} = \boldsymbol{X}^{\rm T}$ and therefore, $\hat{\boldsymbol{U}}_{R+1}^{[\rm s]} = \hat{\boldsymbol{V}}_{\rm s}^{*}$ (as defined in (2)). Consequently, $[\boldsymbol{\mathcal{X}}]^{\rm T}_{(R+1)} \cdot \hat{\boldsymbol{U}}_{R+1}^{[\rm s]^{*}} \cdot \hat{\boldsymbol{\Sigma}}_{\rm s}^{-1} = \boldsymbol{X} \cdot \hat{\boldsymbol{V}}_{\rm s} \cdot \hat{\boldsymbol{\Sigma}}_{\rm s}^{-1} = \hat{\boldsymbol{U}}_{\rm s}$, where the last step also follows from (2). ∎

Equation (8) provides the central idea behind the TeTraKron framework 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 $\hat{\boldsymbol{T}}_r = \boldsymbol{I}_r$, i.e., no projection is performed. Another consequence we can draw from (8) 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 $\hat{\boldsymbol{U}}_r^{[\rm s]}$. These are the subspaces obtained from the r-mode unfoldings of $\boldsymbol{\mathcal{X}}$, 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 $\boldsymbol{X}$ (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 $r = 1, 2, \ldots, R$ 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 (8).

However, this recombination requires $\mathcal{O}\left\{M^2 \cdot d\right\}$ multiplications, i.e., it is quadratic in M, which is undesirable. To lower the complexity, we rewrite (8) as

$$\left[\hat{\boldsymbol{\mathcal{U}}}^{[\mathrm{s}]}\right]^{\mathrm{T}}_{(R+1)} = \boldsymbol{U}^{[\mathrm{s}]}_{\mathrm{Kron}} \cdot \hat{\bar{\boldsymbol{U}}}_{\mathrm{s}}, \tag{11}$$

where $\boldsymbol{U}^{[\mathrm{s}]}_{\mathrm{Kron}} = \boldsymbol{U}^{[\mathrm{s}]}_1 \otimes \ldots \otimes \boldsymbol{U}^{[\mathrm{s}]}_R \in \mathbb{C}^{M \times d^R}$ and $\hat{\bar{\boldsymbol{U}}}_{\mathrm{s}} = \boldsymbol{U}^{[\mathrm{s}]^{\mathrm{H}}}_{\mathrm{Kron}} \cdot \hat{\boldsymbol{U}}_{\mathrm{s}} \in \mathbb{C}^{d^R \times d}$ assuming $p_r = d \leq M_r$, for $r = 1, 2, \ldots, R$. Note that the matrix product in (11) requires only $\mathcal{O}\left\{M \cdot d^R\right\}$ multiplications, i.e., it is linear in M. Moreover, (11) can be used for tensor-based subspace tracking as well: we track $\boldsymbol{U}^{[\mathrm{s}]}_r$ for $r = 1, 2, \ldots, R$ 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 $\boldsymbol{U}^{[\mathrm{s}]^{\mathrm{H}}}_{\mathrm{Kron}} \in \mathbb{C}^{d^R \times M}$ and finally run a matrix-based subspace tracker on the lower-dimensional data to track the d-dimensional subspace $\hat{\bar{\boldsymbol{U}}}_{\mathrm{s}} \in \mathbb{C}^{d^R \times d}$.

TeTraKron allows to readily extend arbitrary matrix-based subspace tracking schemes to tensors which yields an improved estimation accuracy as we demonstrate in Section V. 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 unfolding in parallel and recombining the signal subspace estimate via (8) or (11), the only modification we have to apply to the matrix-based subspace tracking schemes is the following: Typically, it is assumed that the observation matrix $\boldsymbol{X}$ is augmented by a new column $\boldsymbol{x}(n)$ with every new snapshot n. For the r-mode unfoldings, every new snapshot generates not only one but 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 example of the PAST algorithm in the next section.

IV. Example: Tensor-based PAST/PASTd

In this section we provide one example 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 the simple but widely used projection approximation subspace tracking (PAST) algorithm [11]. For notational simplicity, we focus on the $R = 2$-dimensional case, i.e., our data tensor $\boldsymbol{\mathcal{X}}$ is of size $M_1 \times M_2 \times N$.

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

$\boldsymbol{P}(0) = \boldsymbol{I}_{d \times d}$, $\hat{\boldsymbol{U}}_{\mathrm{s}}(0) = \boldsymbol{I}_{M \times d}$

FOR n = 1, 2, ... DO

$$\begin{aligned}
\boldsymbol{y}(n) &= \hat{\boldsymbol{U}}_{\mathrm{s}}^{\mathrm{H}}(n-1) \cdot \boldsymbol{x}(n) \\
\boldsymbol{h}(n) &= \boldsymbol{P}(n-1) \cdot \boldsymbol{y}(n) \\
\boldsymbol{g}(n) &= \boldsymbol{h}(n) / \left(\beta + \boldsymbol{y}^{\mathrm{H}}(n) \cdot \boldsymbol{h}(n)\right) \\
\boldsymbol{P}(n) &= \beta^{-1} \cdot \mathrm{Tri}\{\boldsymbol{P}(n-1) - \boldsymbol{g}(n) \cdot \boldsymbol{h}^{\mathrm{H}}(n)\} \\
\boldsymbol{e}(n) &= \boldsymbol{x}(n) - \hat{\boldsymbol{U}}_{\mathrm{s}}(n-1) \cdot \boldsymbol{y}(n) \\
\hat{\boldsymbol{U}}_{\mathrm{s}}(n) &= \hat{\boldsymbol{U}}_{\mathrm{s}}(n-1) + \boldsymbol{e}(n) \cdot \boldsymbol{g}^{\mathrm{H}}(n)
\end{aligned}$$

END

TABLE I. Summary of the PAST algorithm [11].

In the TeTraKron extension of PAST we apply the same algorithm to $\boldsymbol{X} = [\boldsymbol{\mathcal{X}}]^{\mathrm{T}}_{(3)}$ and to $[\boldsymbol{\mathcal{X}}]_{(1)}$ as well as $[\boldsymbol{\mathcal{X}}]_{(2)}$ in parallel. With each new observation vector $\boldsymbol{x}(n) \in \mathbb{C}^{M_1 \cdot M_2 \times 1}$ we obtain a new matrix of observations for the 1-space and the 2-space of $\boldsymbol{\mathcal{X}}$ which is given by $\tilde{\boldsymbol{X}}(n) \in \mathbb{C}^{M_1 \times M_2}$ for $[\boldsymbol{\mathcal{X}}]_{(1)}$ and $\tilde{\boldsymbol{X}}^{\mathrm{T}}(n)$ for $[\boldsymbol{\mathcal{X}}]_{(2)}$. Note that $\tilde{\boldsymbol{X}}(n)$ is a rearranged version of $\boldsymbol{x}(n)$ which satisfies $\mathrm{vec}\left\{\tilde{\boldsymbol{X}}(n)\right\} = \boldsymbol{x}(n)$. Since PAST is based on RLS it can be modified to process the entire new batch of observations at the same time. For instance, the modified update equations for the 1-mode unfolding become

$$\boldsymbol{Y}_1(n) = \hat{\boldsymbol{U}}_1^{[\mathrm{s}]^{\mathrm{H}}}(n-1) \cdot \tilde{\boldsymbol{X}}(n) \tag{12}$$

$$\boldsymbol{H}_1(n) = \boldsymbol{P}_1(n-1) \cdot \boldsymbol{Y}_1(n) \tag{13}$$

$$\boldsymbol{G}_1(n) = \boldsymbol{H}_1(n) \cdot \left(\beta \cdot \boldsymbol{I}_{M_2} + \boldsymbol{Y}_1^{\mathrm{H}}(n) \cdot \boldsymbol{H}_1(n)\right)^{-1} \tag{14}$$

$$\boldsymbol{P}_1(n) = \beta^{-1} \cdot \mathrm{Tri}\{\boldsymbol{P}_1(n-1) - \boldsymbol{G}_1(n) \cdot \boldsymbol{H}_1^{\mathrm{H}}(n)\} \tag{15}$$

$$\boldsymbol{E}_1(n) = \tilde{\boldsymbol{X}}(n) - \hat{\boldsymbol{U}}_1^{[\mathrm{s}]}(n-1) \cdot \boldsymbol{Y}_1(n) \tag{16}$$

$$\hat{\boldsymbol{U}}_1^{[\mathrm{s}]}(n) = \hat{\boldsymbol{U}}_1^{[\mathrm{s}]}(n-1) + \boldsymbol{E}_1(n) \cdot \boldsymbol{G}_1^{\mathrm{H}}(n). \tag{17}$$

Note that the inverse of the $d \times d$ correlation matrix of $\boldsymbol{Y}_1(n)$, $\boldsymbol{P}_1(n)$, can be directly calculated

$$\boldsymbol{P}_1(n) = \boldsymbol{C}_{yy_1}^{-1}(n) = (\beta \boldsymbol{C}_{yy_1}(n-1) + \boldsymbol{Y}_1(n) \boldsymbol{Y}_1^{\mathrm{H}}(n))^{-1}, \tag{18}$$

leading to a reduced complexity when $d < M_2$. Then $\boldsymbol{G}_1(n)$ is alternatively updated as

$$\boldsymbol{G}_1(n) = \boldsymbol{P}_1(n) \boldsymbol{Y}_1(n), \tag{19}$$

i.e., equations (13), (14), and (15) are replaced by (18) and (19).

In [11], a deflation-based version of PAST (PASTd) with even lower complexity is proposed. This algorithm is also based on RLS and can hence be modified in the same manner.

V. Simulation Results

In this section we 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 [11] 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 $n = 1, 2, \ldots, N$

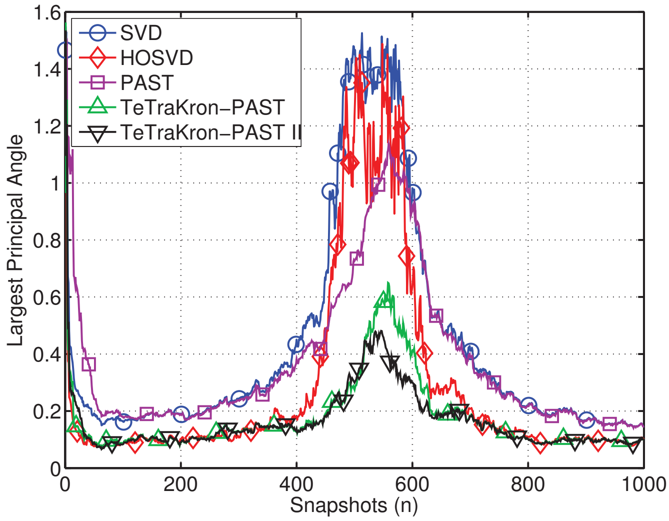

Fig. 1. Three moving sources on a 9×9 URA at an SNR of 0 dB.

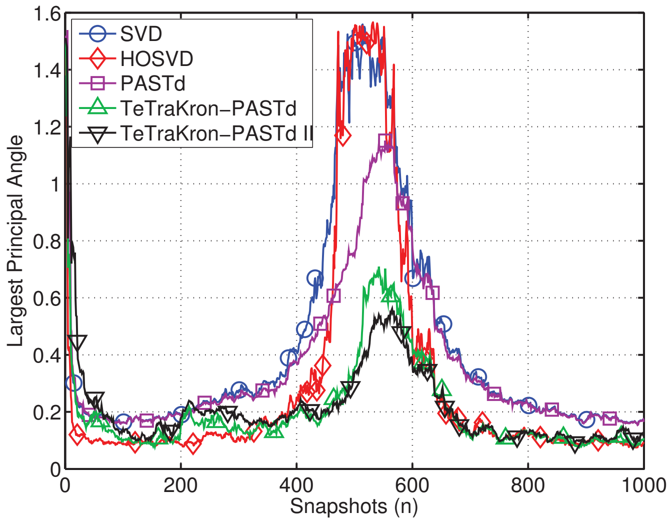

Fig. 2. Three moving sources on a 7×7 URA at an SNR of 0 dB.

according to

$$\mu_1[n] = 0.3 - 0.1 \cdot t[n], \ \mu_2[n] = 0.2 + 0.1 \cdot t[n],$$
$$\nu_1[n] = 0.2 + 0.1 \cdot t[n], \ \nu_2[n] = 0.2 + 0.1 \cdot t[n],$$

for $t[n] = \frac{n-1}{N-1}$, whereas the third source remains stationary at $\mu_3 = \nu_3 = 0.1$. Therefore, for n close to $N/2$ the first and the second source cross. The n-th observed snapshot is given by $\boldsymbol{x}[n] = \boldsymbol{A}[n] \cdot \boldsymbol{s}[n] + \boldsymbol{w}[n]$, where $\boldsymbol{A}[n]$ is the array steering matrix for the current source positions $\mu_i[n], \nu_i[n]$, $i = 1, 2, 3$, and the source samples $\boldsymbol{s}[n]$ as well as the noise samples $\boldsymbol{w}[n]$ are drawn from a zero mean circularly symmetric complex Gaussian distribution with variance one (SNR = 0 dB). We choose the forgetting factor $\beta = 0.97$. Similar to [11], 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 [11] whereas TeTraKron-PAST and TeTraKron-PAST II refer to the tensor extension of PAST via the proposed TeTraKron framework based on (8) and the reduced-complexity version (11), respectively. For reference we display two curves labeled "SVD" and "HOSVD" where the entire matrix/tensor of observations[1] 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. We also observe that the reduced-complexity version based on (11) adapts slightly slower than the one based on (8).

VI. 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 time. As an example, we have used the TeTraKron framework to extend the PAST and the PASTd algorithm to tensors and have demonstrated the enhanced subspace estimation accuracy via numerical simulations.

References

[1] L. de Lathauwer, B. de Moor, and J. Vanderwalle, "A multilinear singular value decomposition", *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[2] J. P. Delmas, "Subspace tracking for signal processing", in *Adaptive Signal Processing: Next Generation Solutions*, T. Adali and S. Haykin, Eds., pp. 211–270, 2010.

[3] M. Haardt, F. Roemer, and G. Del Galdo, "Higher-order SVD based subspace estimation to improve the parameter estimation accuracy in multi-dimensional harmonic retrieval problems", *IEEE Trans. on Sig. Proc.*, vol. 56, pp. 3198–3213, July 2008.

[4] W. Hu, X. Li, X. Zhang, X. Shi, S. Maybank, and Z. Zhang, "Incremental tensor subspace learning and its applications to foreground segmentation and tracking", *Int. J. Computer Vision*, vol. 91, no. 3, pp. 303–327, 2011.

[5] I. Karasalo, "Estimating the covariance matrix by signal subspace averaging", *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 1, pp. 8–12, 1986.

[6] E. Oja, "A simplified neuron model as a principal components analyzer", *Journal of Mathematical Biology*, vol. 15, no. 3, pp. 267–273, 1982.

[7] N. Owsley, "Adaptive data orthogonalization", in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '78)*, vol. 3, pp. 109–112, Tulsa, OK, Apr. 1978.

[8] F. Roemer, H. Becker, M. Haardt, and 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, Dec. 2009.

[9] B. Song, F. Roemer, and M. Haardt, "Blind estimation of SIMO channels using a tensor-based subspace method", in *Proc. 44-th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2010.

[10] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Incremental tensor analysis: Theory and applications", *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 2, no. 3, Oct. 2008.

[11] B. Yang, "Projection approximation subspace tracking", *IEEE Trans. on Signal Processing*, vol. 43, no. 1, pp. 95–107, Jan. 1995.

[1]To render the comparison to the adaptive RLS-based schemes fair, the exponential weighting with a forgetting factor β is also applied here.