STRUCTURED LEAST SQUARES (SLS) BASED ENHANCEMENTS OF TENSOR-BASED CHANNEL ESTIMATION (TENCE) FOR TWO-WAY RELAYING WITH MULTIPLE ANTENNAS

Florian Roemer and Martin Haardt

Ilmenau University of Technology, Communications Research Laboratory
P.O. Box 100565, D-98684 Ilmenau, Germany, http://tu-ilmenau.de/crl
florian.roemer@ieee.org, haardt@ieee.org

***Abstract* — In this paper, we develop a novel tensor-based channel estimation algorithm for two-way relaying with amplify and forward (AF) relays. In two-way relaying two terminals transmit simultaneously to one relay station which then amplifies the received signal and transmits it back to the terminals. With sufficient channel knowledge the terminals can subtract the interference their own transmissions have caused to their received signal and subsequently decode the data from the other terminal. While this relaying scheme uses the radio resources in a particularly efficient way, reliable channel knowledge is crucial to obtain an acceptable link quality.**

It has been shown that we can estimate all relevant channel parameters via the purely algebraic TENCE algorithm. In this paper, we introduce a novel tensor-based approach to enhance these estimates even further via an iterative procedure based on Structured Least Squares (SLS).

The improvement in channel estimation accuracy is demonstrated via computer simulations at the end of the paper. We also show that the number of required iterations is only between one and four, even in critical scenarios.

1. INTRODUCTION

A crucial aspect in designing future mobile communication systems is to enhance transmission rates, reliability, and coverage of the mobile radio access. A promising system concept to meet these demands is the deployment of intermediate relay stations that support the transmissions between communication partners in the radio network.

Relay stations can be divided into two categories: Regenerative or decode-and-forward (DF) relays which decode the received transmissions and reencode them for the second hop and non-regenerative or amplify-and-forward (AF) relays which retransmit an amplified version of their received signal without decoding the individual transmissions. We focus on AF relays since their hardware complexity is significantly lower than for DF relays which facilitates the mass deployment of simple and cheap devices.

A large variety of relaying schemes has been studied in the literature. Two-way relaying is one particular scheme that uses the radio resources very efficiently [7]. The basic two-way relaying scenario which we consider in this paper consists of two user terminals equipped with M_1 and M_2 antennas that want to exchange data and one AF relay equipped with M_R antennas that assists this communication link as depicted in Figure 1. The transmission is performed in two subsequent time slots in a TDD fashion: In the first time slot both terminals transmit their data to the relay which receives the superimposed transmissions. In the second time slot, the relay sends an amplified version of the received signal back to both terminals. If the terminals possess channel knowledge, they can subtract the interference that their own transmission has caused from their received signal and then decode the data from the other communication partner.

Two-way relaying has been studied in many recent publications. However, the channel estimation is often ignored by considering perfect channel state information (CSI) [5], [11]. Moreover, the authors usually assume that the relay station actively suppresses interference. This can be achieved by using decode and forward (DF) relays [6] or by exploiting channel state information at the AF relays [10]. In contrast to this we focus on a two-way relaying scenario where the relay station does not need any channel state information.

In [9] we have introduced the TENCE algorithm which is a tensor-based algebraic channel estimation technique that provides both terminals with knowledge of all relevant channel parameters and is applicable to arbitrary antenna configurations. In this contribution we demonstrate that the initial estimate obtained by TENCE in a closed-form can be further improved via a tensor-based iterative algorithm.

The derivation of the algorithm is inspired by Structured Least Squares (SLS) [3], which is an efficient technique to solve the overdetermined shift invariance equations that appear in ESPRIT-based high-resolution parameter estimation algorithms. Since SLS exploits the structure inherent in the invariance equations, it outperforms alternative schemes such as Total Least Squares (TLS) [2, 12]. The SLS method has been extended for the solution of the tensor shift invariance equations which appear in Tensor-ESPRIT type algorithms [4]. This extended method is known as Tensor-Structure SLS (TS-SLS) [8].

As we show in this paper, we can enhance TENCE by exploiting an additional symmetry in the data model in an iterative fashion inspired by the ideas used in SLS and TS-SLS. Certainly, this iterative procedure results in an increase in the computational complexity as compared to the closed-form TENCE. However, we demonstrate by simulations that between one and four iterations are sufficient since no additional improvement is achieved for more iterations, even in critical scenarios.

2. NOTATION

To facilitate the distinction between scalars, vectors, matrices, and tensors, the following notation is used throughout the paper: Scalars

are denoted as italic letters ($a, b, \ldots, A, B, \ldots$), vectors as lower-case bold-faced letters ($\boldsymbol{a}, \boldsymbol{b}, \ldots$), matrices are written as upper-case bold-faced letters ($\boldsymbol{A}, \boldsymbol{B}$), and tensors as bold-faced calligraphic letters ($\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}$).

The superscripts $^{\mathrm{T}}, ^{\mathrm{H}}, ^{+}, ^{*}$ represent matrix transposition, Hermitian transposition, the Moore-Penrose pseudo inverse, and complex conjugation, respectively. The Kronecker product between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is denoted by $\boldsymbol{A} \otimes \boldsymbol{B}$. Likewise, $\boldsymbol{A} \diamond \boldsymbol{B}$ symbolizes the Khatri-Rao (columnwise Kronecker) product and $\boldsymbol{A} \odot \boldsymbol{B}$ the Schur (elementwise) product.

A three-dimensional tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times M_2 \times M_3}$ is a three-way array with size M_r along mode (or dimension) r for $r = 1, 2, 3$. The n-mode vectors of $\boldsymbol{\mathcal{A}}$ are obtained by stacking the elements of $\boldsymbol{\mathcal{A}}$ into a vector where the n-th index varies in its range and all the other indices are kept fixed. Collecting all the n-mode vectors into a matrix we obtain the n-mode unfolding of $\boldsymbol{\mathcal{A}}$, which is represented by $[\boldsymbol{\mathcal{A}}]_{(n)} \in \mathbb{C}^{M_n \times M_1 \cdot M_2 \cdot M_3 / M_n}$. The order of the columns in $[\boldsymbol{\mathcal{A}}]_{(n)}$ is chosen in accordance with [1].

The n-mode product between a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times M_2 \times M_3}$ and a matrix $\boldsymbol{U} \in \mathbb{C}^{P_n \times M_n}$ is denoted as $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{A}} \times_n \boldsymbol{U}_n$ and computed by multiplying all n-mode vectors of $\boldsymbol{\mathcal{A}}$ from the left-hand side by the matrix $\boldsymbol{U}_n$, i.e., $[\boldsymbol{\mathcal{B}}]_{(n)} = \boldsymbol{U}_n \cdot [\boldsymbol{\mathcal{A}}]_{(n)}$.

To concatenate two tensors along the n-th mode we use the symbol $[\boldsymbol{\mathcal{A}} \sqcup_n \boldsymbol{\mathcal{B}}]$.

The vectorization operators $\mathrm{vec}\{.\}$ aligns all the elements of a matrix or a tensor into a vector, such that the row index is varied first, then the column index, and then the third index. For arbitrary three-dimensional tensors $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I \times J \times K}$, permutation matrices $\boldsymbol{P}^{(n)}_{I,J,K}$ of size $I \cdot J \cdot K \times I \cdot J \cdot K$ are uniquely defined via the following property [8]

$$\boldsymbol{P}^{(n)}_{I,J,K} \cdot \mathrm{vec}\left\{ [\boldsymbol{\mathcal{X}}]_{(n)} \right\} = \mathrm{vec}\left\{ \boldsymbol{\mathcal{X}} \right\}, \; n = 1, 2, 3, \forall \, \boldsymbol{\mathcal{X}}. \quad (1)$$

The higher-order norm of a tensor $\boldsymbol{\mathcal{A}}$ is written as $\|\boldsymbol{\mathcal{A}}\|_{\mathrm{H}}$. It represents a generalization of the Frobenius norm of matrices expressed as $\|\boldsymbol{A}\|_{\mathrm{F}}$ in the sense that both are computed via the square root of the sum of the squared magnitudes of all elements. It is easy to see that for a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{I \times J \times K}$ and for a matrix $\boldsymbol{A} \in \mathbb{C}^{I \times J}$, the norms satisfy the following identities

$$\|\boldsymbol{\mathcal{A}}\|^2_{\mathrm{H}} = \|\mathrm{vec}\{\boldsymbol{\mathcal{A}}\}\|^2_2 = \left\| [\boldsymbol{\mathcal{A}}]_{(n)} \right\|^2_{\mathrm{F}}, \; n = 1, 2, 3 \quad (2)$$

$$\|\boldsymbol{A}\|^2_{\mathrm{F}} = \|\mathrm{vec}\{\boldsymbol{A}\}\|^2_2, \quad (3)$$

where $\|\boldsymbol{x}\|_2$ is the Euclidean norm (2-norm) of the vector $\boldsymbol{x}$.

Finally, $\boldsymbol{0}_{p \times q}$ indicates a zero matrix of size $p \times q$, $\boldsymbol{I}_p$ is the identity matrix of size $p \times p$, and $\boldsymbol{\mathcal{I}}_{3,r}$ is the $r \times r \times r$ identity tensor, which is equal to one if all three indices are equal and zero otherwise.

3. SYSTEM DESCRIPTION

The system under investigation is depicted in Figure 1. Two terminals equipped with M_1 and M_2 antennas, respectively, exchange data. This transmission is supported by an intermediate relay station having M_{R} antennas. In the first transmission phase the terminals transmit simultaneously to the relay. The relay receives the superimposed transmissions, amplifies the received signal vector by a complex matrix $\boldsymbol{G} \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$, and transmits the amplified signal back to the terminals in the second time slot in a TDD fashion.

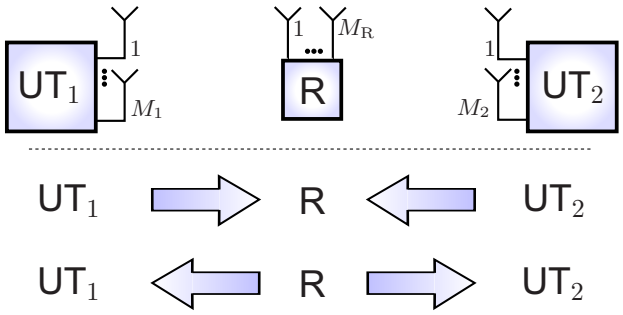

Fig. 1. Two-way relaying system model.

Assuming frequency-flat fading, the signal received at the relay can be expressed as

$$\boldsymbol{r} = \boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \boldsymbol{n}_{\mathrm{R}}, \quad (4)$$

where $\boldsymbol{x}_1 \in \mathbb{C}^{M_1}$ and $\boldsymbol{x}_2 \in \mathbb{C}^{M_2}$ are the transmitted vectors from terminal 1 and terminal 2, the matrices $\boldsymbol{H}_1 \in \mathbb{C}^{M_{\mathrm{R}} \times M_1}$ and $\boldsymbol{H}_2 \in \mathbb{C}^{M_{\mathrm{R}} \times M_2}$ represent the MIMO channels between the relay and the two terminals, and the vector $\boldsymbol{n}_{\mathrm{R}}$ is the additive noise vector at the relay station.

In the second time slot, the relay amplifies the vector $\boldsymbol{r}$ by a complex matrix $\boldsymbol{G} \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$ and transmits the amplified signal back to the two terminals. The received signals therefore obey the following model

$$\boldsymbol{y}_1 = \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G} \cdot (\boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \boldsymbol{n}_{\mathrm{R}}) + \boldsymbol{n}_1 \quad (5)$$

$$\boldsymbol{y}_2 = \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G} \cdot (\boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \boldsymbol{n}_{\mathrm{R}}) + \boldsymbol{n}_2, \quad (6)$$

where we have assumed that reciprocity holds and the channels have not changed between the two transmission phases. The vectors $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ represent the additive noise at the first and second terminal, respectively. Note that (5) and (6) can also be expressed in the following fashion

$$\boldsymbol{y}_1 = \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \tilde{\boldsymbol{n}}_1 \quad (7)$$

$$\boldsymbol{y}_2 = \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \tilde{\boldsymbol{n}}_2. \quad (8)$$

Here $\tilde{\boldsymbol{n}}_1$ and $\tilde{\boldsymbol{n}}_2$ represent the effective noise terms which are given by

$$\tilde{\boldsymbol{n}}_1 = \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{n}_{\mathrm{R}} + \boldsymbol{n}_1 \quad (9)$$

$$\tilde{\boldsymbol{n}}_2 = \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{n}_{\mathrm{R}} + \boldsymbol{n}_2. \quad (10)$$

We conclude from (7) and (8) that each terminal receives the data from the other terminal via the effective channels $\boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_2$ and $\boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_1$, respectively. Moreover, these transmissions are superimposed by interference caused via the terminals' own transmitted data. However, since this data is known, the terminals can subtract this self-interference term if they have knowledge of the channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$. Therefore we now focus on the acquisition of reliable channel state information at both terminals.

In order to facilitate the channel estimation we require a training phase which consists of M_{R} frames. In each frame the terminals transmit known pilot sequences of length N_{P} described by $\boldsymbol{X}_1 = [\boldsymbol{x}_{1,1}, \ldots, \boldsymbol{x}_{1,N_{\mathrm{P}}}] \in \mathbb{C}^{M_1 \times N_{\mathrm{P}}}$ and $\boldsymbol{X}_2 = [\boldsymbol{x}_{2,1}, \ldots, \boldsymbol{x}_{2,N_{\mathrm{P}}}] \in \mathbb{C}^{M_2 \times N_{\mathrm{P}}}$. Moreover, for the i-th frame, the relay amplification matrix $\boldsymbol{G}^{(i)} \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$ is used. Since the relay is assumed to have no

see the program

channel state information, all matrices $\boldsymbol{G}^{(i)}$ can be designed beforehand and are therefore known to both terminals.

Consequently, the signal received by the user terminals in the i-th frame from the transmission of the j-th pilot can be expressed as

$$\begin{aligned}\boldsymbol{y}_{1,i,j} &= \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}^{(i)} \cdot \boldsymbol{H}_1 \cdot \boldsymbol{x}_{1,j} + \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}^{(i)} \cdot \boldsymbol{H}_2 \cdot \boldsymbol{x}_{2,j} + \tilde{\boldsymbol{n}}_{1,i,j} \\ \boldsymbol{y}_{2,i,j} &= \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G}^{(i)} \cdot \boldsymbol{H}_1 \cdot \boldsymbol{x}_{1,j} + \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G}^{(i)} \cdot \boldsymbol{H}_2 \cdot \boldsymbol{x}_{2,j} + \tilde{\boldsymbol{n}}_{2,i,j}.\end{aligned} \tag{11}$$

Let us introduce the following definitions

$$\boldsymbol{H} = [\boldsymbol{H}_1,\ \boldsymbol{H}_2] \in \mathbb{C}^{M_{\mathrm{R}} \times (M_1 + M_2)} \tag{12}$$

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{bmatrix} \in \mathbb{C}^{(M_1+M_2) \times N_{\mathrm{P}}} \tag{13}$$

$$\boldsymbol{\mathcal{G}} = \left[\boldsymbol{G}^{(1)} \sqcup_3 \ldots \sqcup_3 \boldsymbol{G}^{(M_{\mathrm{R}})}\right] \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}} \times M_{\mathrm{R}}}, \tag{14}$$

where $\sqcup_n$ represents the concatenation operator along the n-th dimension. Using these definitions we can express (11) in tensor form

$$\boldsymbol{\mathcal{Y}}_1 = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{H}_1^{\mathrm{T}} \times_2 (\boldsymbol{H} \cdot \boldsymbol{X})^{\mathrm{T}} + \boldsymbol{\mathcal{N}}_1 \in \mathbb{C}^{M_1 \times N_{\mathrm{P}} \times M_{\mathrm{R}}} \tag{15}$$

$$\boldsymbol{\mathcal{Y}}_2 = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{H}_2^{\mathrm{T}} \times_2 (\boldsymbol{H} \cdot \boldsymbol{X})^{\mathrm{T}} + \boldsymbol{\mathcal{N}}_2 \in \mathbb{C}^{M_2 \times N_{\mathrm{P}} \times M_{\mathrm{R}}}, \tag{16}$$

where $\times_n$ represents the n-mode product between a tensor and a matrix [1].

4. SLS-BASED CHANNEL ESTIMATION

In [9] we have shown that each terminal can estimate both channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ in closed-form from its own received training data with the help of the TENCE algorithm. Via TENCE we are able to resolve all scaling ambiguities except for one sign ambiguity, i.e., instead of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ we might estimate $-\boldsymbol{H}_1$ and $-\boldsymbol{H}_2$. However, since the transmissions take place via the effective channels $\boldsymbol{H}_i^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_j$, $i, j = 1, 2$, this sign ambiguity is canceled and therefore does not have any impact on the transmission.

In the presented SLS-based improvement of TENCE we can additionally exploit the fact that in (15) and (16) the channel matrix that appears in the first factor is also present in the second factor and therefore improve the TENCE solution. Here, we show the solution for terminal 1. Due to the symmetry of the data model in (15) and (16) the solution for terminal 2 is similar.

Let $\hat{\boldsymbol{H}}_1$ and $\hat{\boldsymbol{H}}_2$ be the initial channel estimates for $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$, respectively, and $\hat{\boldsymbol{H}} = [\hat{\boldsymbol{H}}_1,\ \hat{\boldsymbol{H}}_2]$. Moreover, define $\tilde{\boldsymbol{\mathcal{Y}}}_1 = \boldsymbol{\mathcal{Y}}_1 \times_2 \left(\boldsymbol{X}^{\mathrm{T}}\right)^{+}$, where the superscript $^{+}$ represents the Moore-Penrose pseudo inverse. In the absence of noise, $\tilde{\boldsymbol{\mathcal{Y}}}_1 = \boldsymbol{\mathcal{G}} \times_1 \hat{\boldsymbol{H}}_1^{\mathrm{T}} \times_2 \hat{\boldsymbol{H}}^{\mathrm{T}}$. Therefore, an indicator for the channel estimation accuracy is the norm of the tensor $\tilde{\boldsymbol{\mathcal{Y}}}_1 - (\boldsymbol{\mathcal{G}} \times_1 \hat{\boldsymbol{H}}_1^{\mathrm{T}} \times_2 \hat{\boldsymbol{H}}^{\mathrm{T}})$.

Finding the channel matrices $\hat{\boldsymbol{H}}_1$ and $\hat{\boldsymbol{H}}_2$ that minimize this norm requires solving a non-linear optimization problem which is not guaranteed to have a unique solution. However, we can take advantage of the fact that we already have an initial estimate for the channel matrices which is obtained via TENCE. We therefore introduce update terms $\Delta\boldsymbol{H}_1$ and $\Delta\boldsymbol{H}_2$ for the estimates $\hat{\boldsymbol{H}}_1$ and $\hat{\boldsymbol{H}}_2$, similarly to the SLS algorithm [3]. By including penalty terms for the norm of the updates we confine the final solution to be close to the original estimate obtained via TENCE. This can be seen as a form of regularization which enhances the numerical stability.

The cost function that we minimize in an iterative fashion can be expressed in the following way

$$\begin{aligned}J(\Delta\boldsymbol{H}_k) = &\left\| \tilde{\boldsymbol{\mathcal{Y}}}_1 - \boldsymbol{\mathcal{G}} \times_1 \left(\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k}\right)^{\mathrm{T}} \times_2 \left(\hat{\boldsymbol{H}} + \Delta\boldsymbol{H}_k\right)^{\mathrm{T}} \right\|_{\mathrm{H}}^2 \\ &+ \kappa_1^2 \left\|\Delta\boldsymbol{H}_{1,k}\right\|_{\mathrm{F}}^2 + \kappa_2^2 \left\|\Delta\boldsymbol{H}_{2,k}\right\|_{\mathrm{F}}^2,\end{aligned} \tag{17}$$

where $\Delta\boldsymbol{H}_{1,k}$ and $\Delta\boldsymbol{H}_{2,k}$ represent the update terms for $\hat{\boldsymbol{H}}_1$ and $\hat{\boldsymbol{H}}_2$ after the k-th iteration and $\Delta\boldsymbol{H}_k = [\Delta\boldsymbol{H}_{1,k},\ \Delta\boldsymbol{H}_{2,k}]$. Moreover, $\kappa_1^2 = M_1/\alpha$ and $\kappa_2^2 = M_2/\alpha$. Here $\alpha \in \mathbb{R}, \alpha > 0$ is a design parameter that allows to controls the amount of regularization used: The larger α is chosen, the less weight is given to the penalty terms and therefore less regularization is included.

The solution of the non-linear least squares problem (17) is established in an iterative fashion by local linearization inspired by the derivations of SLS in [3] and TS-SLS in [8]. To this end, introduce the residual tensor $\boldsymbol{\mathcal{R}}_k$ after the k-th iteration as

$$\boldsymbol{\mathcal{R}}_k = \tilde{\boldsymbol{\mathcal{Y}}}_1 - \boldsymbol{\mathcal{G}} \times_1 \left(\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k}\right)^{\mathrm{T}} \times_2 \left(\hat{\boldsymbol{H}} + \Delta\boldsymbol{H}_k\right)^{\mathrm{T}}. \tag{18}$$

Using this definition and the identities (2) and (3), we can express (17) in the following form

$$J(\Delta\boldsymbol{H}_k) = \left\| \begin{bmatrix} \operatorname{vec}\{\boldsymbol{\mathcal{R}}_k\} \\ \kappa_1 \cdot \operatorname{vec}\{\Delta\boldsymbol{H}_{1,k}\} \\ \kappa_2 \cdot \operatorname{vec}\{\Delta\boldsymbol{H}_{2,k}\} \end{bmatrix} \right\|_2^2. \tag{19}$$

The matrices $\Delta\boldsymbol{H}_{1,k}$ and $\Delta\boldsymbol{H}_{2,k}$ are updated in each iteration according to

$$\Delta\boldsymbol{H}_{1,k+1} = \Delta\boldsymbol{H}_{1,k} + \Delta\Delta\boldsymbol{H}_{1,k} \tag{20}$$

$$\Delta\boldsymbol{H}_{2,k+1} = \Delta\boldsymbol{H}_{2,k} + \Delta\Delta\boldsymbol{H}_{2,k}, \tag{21}$$

starting from the initial values

$$\Delta\boldsymbol{H}_{1,k} = \boldsymbol{0}_{M_{\mathrm{R}} \times M_1} \tag{22}$$

$$\Delta\boldsymbol{H}_{2,k} = \boldsymbol{0}_{M_{\mathrm{R}} \times M_2}. \tag{23}$$

Therefore, our goal is to find $\Delta\Delta\boldsymbol{H}_{1,k}$ and $\Delta\Delta\boldsymbol{H}_{2,k}$ that minimize (19) in the k-th iteration, which requires a series of algebraic manipulations. As a first step, we use (20) and (21) in (18) in order to express the residual tensor in the $(k+1)$-th iteration in the following form

$$\begin{aligned}\boldsymbol{\mathcal{R}}_{k+1} = \boldsymbol{\mathcal{R}}_k &- \boldsymbol{\mathcal{G}} \times_1 \Delta\Delta\boldsymbol{H}_{1,k}^{\mathrm{T}} \times_2 \left(\hat{\boldsymbol{H}} + \Delta\boldsymbol{H}_k\right)^{\mathrm{T}} \\ &- \boldsymbol{\mathcal{G}} \times_1 \left(\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k}\right)^{\mathrm{T}} \times_2 \Delta\Delta\boldsymbol{H}_k^{\mathrm{T}} \\ &- \boldsymbol{\mathcal{G}} \times_1 \Delta\Delta\boldsymbol{H}_{1,k}^{\mathrm{T}} \times_2 \Delta\Delta\boldsymbol{H}_k^{\mathrm{T}}.\end{aligned} \tag{24}$$

In order to linearize (24), the last term is neglected since it contains the product of two update terms which we assumed to have a small Frobenius norm. The next step is to vectorize (24) and to rearrange the terms applying the permutation matrices defined in (1). We obtain

$$\begin{aligned}\operatorname{vec}\{\boldsymbol{\mathcal{R}}_{k+1}\} \approx\ &\operatorname{vec}\{\boldsymbol{\mathcal{R}}_k\} \\ &- \boldsymbol{P}^{(3)}_{M_1, M_1+M_2, M_{\mathrm{R}}} \cdot \left(\boldsymbol{I}_{M_1} \otimes \left[\boldsymbol{\mathcal{G}} \times_2 \left(\hat{\boldsymbol{H}} + \Delta\boldsymbol{H}_k\right)^{\mathrm{T}}\right]_{(1)}^{\mathrm{T}}\right) \\ &\quad \cdot \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{1,k}\} \\ &- \boldsymbol{P}^{(1)}_{M_1, M_1+M_2, M_{\mathrm{R}}} \cdot \left(\boldsymbol{I}_{M_1+M_2} \otimes \left[\boldsymbol{\mathcal{G}} \times_1 \left(\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k}\right)^{\mathrm{T}}\right]_{(2)}^{\mathrm{T}}\right) \\ &\quad \cdot \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_k\}.\end{aligned} \tag{25}$$

To separate the terms depending on $\Delta\Delta\boldsymbol{H}_{1,k}$ and $\Delta\Delta\boldsymbol{H}_{2,k}$ we observe that

$$\operatorname{vec}\{\Delta\Delta\boldsymbol{H}_k\} = \operatorname{vec}\left\{\left[\Delta\Delta\boldsymbol{H}_{1,k},\ \Delta\Delta\boldsymbol{H}_{2,k}\right]\right\} \tag{26}$$

$$= \begin{bmatrix} \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{1,k}\} \\ \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{2,k}\} \end{bmatrix}, \tag{27}$$

which follows from the definition of the vec-operator. Inserting (27) into (25) we get

$$\begin{aligned}\operatorname{vec}\{\boldsymbol{\mathcal{R}}_{k+1}\} \approx \operatorname{vec}\{\boldsymbol{\mathcal{R}}_k\} &- \tilde{\boldsymbol{F}}_k^{(1)} \cdot \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{1,k}\} \\ &- \tilde{\boldsymbol{F}}_k^{(2)} \cdot \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{2,k}\}, \end{aligned} \quad (28)$$

where the matrices $\tilde{\boldsymbol{F}}_k^{(1)}$ and $\tilde{\boldsymbol{F}}_k^{(2)}$ are given by

$$\begin{aligned}\tilde{\boldsymbol{F}}_k^{(1)} =& \boldsymbol{P}^{(3)}_{M_1,M_1+M_2,M_\mathrm{R}} \cdot \left(\boldsymbol{I}_{M_1} \otimes \left[\boldsymbol{\mathcal{G}} \times_2 \left(\hat{\boldsymbol{H}} + \Delta\boldsymbol{H}_k\right)^{\mathrm{T}}\right]^{\mathrm{T}}_{(1)}\right) \\ &+ \boldsymbol{P}^{(1)}_{M_1,M_1+M_2,M_\mathrm{R}} \cdot \left(\boldsymbol{I}_{M_1+M_2} \otimes \left[\boldsymbol{\mathcal{G}} \times_1 \left(\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k}\right)^{\mathrm{T}}\right]^{\mathrm{T}}_{(2)}\right) \cdot \boldsymbol{J}_1 \\ \tilde{\boldsymbol{F}}_k^{(2)} =& \boldsymbol{P}^{(1)}_{M_1,M_1+M_2,M_\mathrm{R}} \cdot \left(\boldsymbol{I}_{M_1+M_2} \otimes \left[\boldsymbol{\mathcal{G}} \times_1 \left(\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k}\right)^{\mathrm{T}}\right]^{\mathrm{T}}_{(2)}\right) \cdot \boldsymbol{J}_2 \\ \boldsymbol{J}_1 = \begin{bmatrix} \boldsymbol{I}_{M_1\cdot M_\mathrm{R}} \\ \boldsymbol{0}_{M_2\cdot M_\mathrm{R} \times M_1 \cdot M_\mathrm{R}} \end{bmatrix} & \quad \boldsymbol{J}_2 = \begin{bmatrix} \boldsymbol{0}_{M_1\cdot M_\mathrm{R} \times M_2 \cdot M_\mathrm{R}} \\ \boldsymbol{I}_{M_2\cdot M_\mathrm{R}} \end{bmatrix}.\end{aligned}$$

We can rewrite the cost function in (19) by inserting (20), (21), and (28). As a result we have

$$J(\Delta\boldsymbol{H}_{k+1}) = \left\| \begin{bmatrix} \operatorname{vec}\{\boldsymbol{\mathcal{R}}_k\} \\ \kappa_1 \cdot \operatorname{vec}\{\Delta\boldsymbol{H}_{1,k}\} \\ \kappa_2 \cdot \operatorname{vec}\{\Delta\boldsymbol{H}_{2,k}\} \end{bmatrix} + \boldsymbol{F}_k \cdot \begin{bmatrix} \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{1,k}\} \\ \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{2,k}\} \end{bmatrix} \right\|_2^2,$$

where the matrix $\boldsymbol{F}_k$ is given by

$$\boldsymbol{F}_k = \begin{bmatrix} -\tilde{\boldsymbol{F}}_k^{(1)} & -\tilde{\boldsymbol{F}}_k^{(2)} \\ \kappa_1 \cdot \boldsymbol{I}_{M_1\cdot M_\mathrm{R}} & \boldsymbol{0}_{M_1\cdot M_\mathrm{R} \times M_2 \cdot M_\mathrm{R}} \\ \boldsymbol{0}_{M_2\cdot M_\mathrm{R} \times M_1 \cdot M_\mathrm{R}} & \kappa_2 \cdot \boldsymbol{I}_{M_2\cdot M_\mathrm{R}} \end{bmatrix}. \quad (29)$$

Note that our goal is to find the update terms $\Delta\Delta\boldsymbol{H}_{1,k}$ and $\Delta\Delta\boldsymbol{H}_{2,k}$ that minimize the cost function in the k-th iteration. We have reduced this task to a linear least squares problem, for which the solution is well-known and given by

$$\begin{bmatrix} \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{1,k}\} \\ \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{2,k}\} \end{bmatrix} = -\boldsymbol{F}_k^{+} \cdot \begin{bmatrix} \operatorname{vec}\{\boldsymbol{\mathcal{R}}_k\} \\ \kappa_1 \cdot \operatorname{vec}\{\Delta\boldsymbol{H}_{1,k}\} \\ \kappa_2 \cdot \operatorname{vec}\{\Delta\boldsymbol{H}_{2,k}\} \end{bmatrix}. \quad (30)$$

The iteration (30) is repeated until convergence is achieved. We propose the following method to detect convergence: After the k-th iteration is complete, let us define $\rho_k = \|\boldsymbol{\mathcal{R}}_{k-1}\|_\mathrm{H} - \|\boldsymbol{\mathcal{R}}_k\|_\mathrm{H}$ as a measure for how much the fit of the estimates has improved. Then we terminate the iteration if $\rho_k < \delta$ for a predefined threshold δ. Moreover, if ρ_k should become negative, the last iteration is ignored and the estimates from the $(k-1)$-th iteration are used as final solutions instead.

Our simulations have shown that the algorithm is not very sensitive to the choice of the regularization parameter α as long as it is not chosen too small. For $\alpha < 1$ the amount of regularization is too large and the algorithm fails to converge to the desired minimum. Instead, the algorithm is terminated too early very close to the initial solution. For larger values of α the impact of α on the performance is negligible.

In critical scenarios, we found that a value of α in the range of 100 is beneficial to enhance the numerical stability. On the other hand, for less critical scenarios, regularization is not needed and α can be chosen bigger to reduce the number of required iterations. If the SNR is very high we can even set $\alpha = \infty$, i.e., skip the regularization completely. In this case, (30) is replaced by

$$\begin{bmatrix} \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{1,k}\} \\ \operatorname{vec}\{\Delta\Delta\boldsymbol{H}_{2,k}\} \end{bmatrix} = \left[\tilde{\boldsymbol{F}}_k^{(1)}, \tilde{\boldsymbol{F}}_k^{(2)}\right]^{+} \cdot \operatorname{vec}\{\boldsymbol{\mathcal{R}}_k\}. \quad (31)$$

The threshold parameter δ represents a trade off between computational complexity and estimation accuracy. For very small values of δ, the number of iterations is increased. The experience from our simulations has shown that $\delta = 10^{-3}$ is a reasonable value. Smaller values lead to more iterations, however these do not result in a significant improvement in accuracy. Larger values of δ terminate the algorithm too early.

5. SIMULATION RESULTS

In this section we evaluate the achievable channel estimation accuracy of the proposed algorithm. The channel matrices are generated according to the following model

$$\boldsymbol{H}_i = \sqrt{\frac{K_i}{K_i+1}} \cdot \boldsymbol{H}_{\mathrm{LOS},i} + \sqrt{\frac{1}{K_i+1}} \cdot \boldsymbol{H}_{\mathrm{NLOS},i}, \; i = 1, 2, \quad (32)$$

where $\boldsymbol{H}_{\mathrm{LOS},i} \in \mathbb{C}^{M_\mathrm{R} \times M_i}$ is a rank-1 matrix capturing the line of sight (LOS) component, $\boldsymbol{H}_{\mathrm{NLOS},i} \in \mathbb{C}^{M_\mathrm{R} \times M_i}$ contains the non line of sight (NLOS) component, and K_i is the Rician K-factor. The matrix $\boldsymbol{H}_{\mathrm{NLOS},i}$ consists of zero mean circularly symmetric complex Gaussian random variables. Moreover, a Kronecker spatial correlation model is assumed with factors given by

$$\mathrm{E}\{\boldsymbol{H}_{\mathrm{NLOS},i} \cdot \boldsymbol{H}^{\mathrm{H}}_{\mathrm{NLOS},i}\} = \boldsymbol{R}_\mathrm{R} \in \mathbb{C}^{M_\mathrm{R} \times M_\mathrm{R}}, \; i = 1, 2 \quad (33)$$

$$\mathrm{E}\{\boldsymbol{H}^{\mathrm{H}}_{\mathrm{NLOS},i} \cdot \boldsymbol{H}_{\mathrm{NLOS},i}\} = \boldsymbol{R}_i \in \mathbb{C}^{M_i \times M_i}, \; i = 1, 2, \quad (34)$$

where $\boldsymbol{R}_i$ is the spatial correlation matrix at the user terminal i and $\boldsymbol{R}_\mathrm{R}$ represents the spatial correlation matrix at the relay. To model correlated channels we construct $\boldsymbol{R}_i$ and $\boldsymbol{R}_\mathrm{R}$ such that their main diagonal elements are equal to one and all off-diagonal elements have magnitude ρ_i and ρ_R, respectively. Consequently, $K_1 = K_2 = 0$ and $\rho_1 = \rho_2 = \rho_\mathrm{R} = 0$ corresponds to uncorrelated Rayleigh fading.

To estimate the channel we use training data which is constructed following the rules derived in [9]: The pilot matrix $\boldsymbol{X} \in \mathbb{C}^{(M_1+M_2)\times N_\mathrm{P}}$ in (13) is obtained from the first $M_1 + M_2$ rows of a DFT matrix of size $N_\mathrm{P} \times N_\mathrm{P}$, scaled in such a way that the transmit power constraint is satisfied. If not stated otherwise, we use the smallest possible value for N_P which is $N_\mathrm{P} = M_1 + M_2$. Moreover, the tensor $\boldsymbol{\mathcal{G}}$ in (14) is constructed via

$$\boldsymbol{\mathcal{G}} = \gamma \cdot \boldsymbol{\mathcal{I}}_{3,M_\mathrm{R}} \times_2 \boldsymbol{D}_{M_\mathrm{R}} \times_3 \left(\boldsymbol{D}_{M_\mathrm{R}} \odot \boldsymbol{S}_{M_\mathrm{R},\min\{M_1,M_2\}}\right), \quad (35)$$

where $\gamma \in \mathbb{R}$ ensures the relay transmit power constraint and $\boldsymbol{D}_{M_\mathrm{R}}$ is a DFT matrix of size $M_\mathrm{R} \times M_\mathrm{R}$. The (i,j) element of the matrix $\boldsymbol{S}_{p,q}$ is equal to one if $\mod(p-i+j,p) < q$ and zero otherwise for $i, j = 1, 2, \ldots, p$. Consequently, for $\min\{M_1, M_2\} \geq M_\mathrm{R}$ the matrix $\boldsymbol{S}$ is equal a matrix of ones and the Schur product $\odot$ in (35) vanishes.

To evaluate the channel estimation accuracy we compute the relative mean square error, which is defined as

$$\mathrm{rMSE} = \mathrm{E}\left\{ \min_{s=1,-1} \frac{\left\|\hat{\boldsymbol{H}} + s \cdot \boldsymbol{H}\right\|_\mathrm{F}^2}{\|\boldsymbol{H}\|_\mathrm{F}^2} \right\}, \quad (36)$$

where $\hat{\boldsymbol{H}}$ is an estimate of $\boldsymbol{H}$ and s accounts for the sign ambiguity in the estimation. The estimation error curves are labeled as `He11, He12, He21, He22`, where the first number indicates the terminal which estimates the channel referenced by the second number. For example, `He12` represents the estimate of $\boldsymbol{H}_2$ at user terminal 1. If not stated otherwise, we set $\alpha = 100$ and $\delta = 10^{-3}$.

see the program

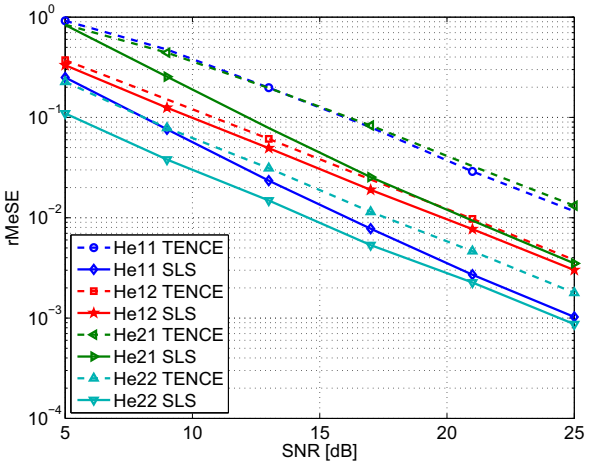

Fig. 2. Median of the channel estimation error vs. SNR in a mixed LOS/NLOS scenario where $M_1 = M_2 = M_{\mathrm{R}} = 2$, $\rho_1 = \rho_2 = \rho_{\mathrm{R}} = 0$, $K_1 = 20$, and $K_2 = 0$.

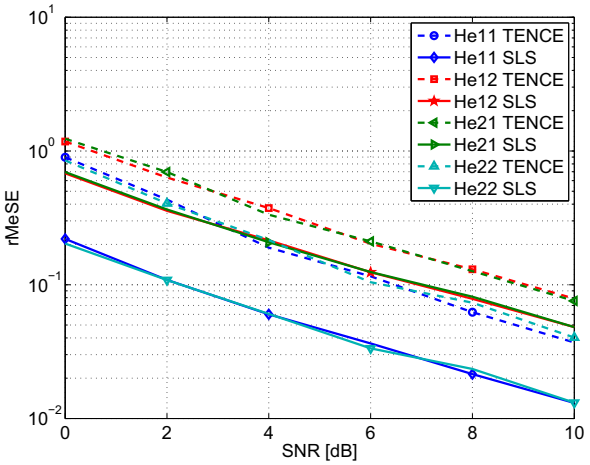

Fig. 3. Median of the channel estimation error vs. SNR in a pure NLOS scenario where $M_1 = M_2 = M_{\mathrm{R}} = 4$, $\rho_1 = \rho_2 = \rho_{\mathrm{R}} = 0$, $K_1 = 0$, and $K_2 = 0$.

The first simulation result in Figure 2 shows a scenario where $M_1 = M_2 = M_{\mathrm{R}} = 2$, $\rho_1 = \rho_2 = \rho_{\mathrm{R}} = 0$, $K_1 = 20$, and $K_2 = 0$. Consequently, the link from terminal one to the relay is a line of sight connection, whereas the link between terminal two and the relay is non line of sight. We observe that the improvements from our SLS-based technique are very large for the estimate of $\boldsymbol{H}_1$ at terminal 1. This is due to the fact that the high K-factor leads to a strong correlation which deteriorates the performance of TENCE. Via the iterative algorithm we can significantly improve this initial estimate by exploiting the structure of the training tensor.

In the second simulation shown in Figure 3 we study a pure NLOS scenario, i.e., $K_1 = K_2 = 0$. We consider $M_1 = M_2 = M_{\mathrm{R}} = 4$ antennas at the terminals and the relay and no spatial correlation, i.e., $\rho_1 = \rho_2 = \rho_{\mathrm{R}} = 0$. We observe that even in low SNR regimes, which are typical for mobile communication systems, the improvements from the SLS-based algorithm are significant.

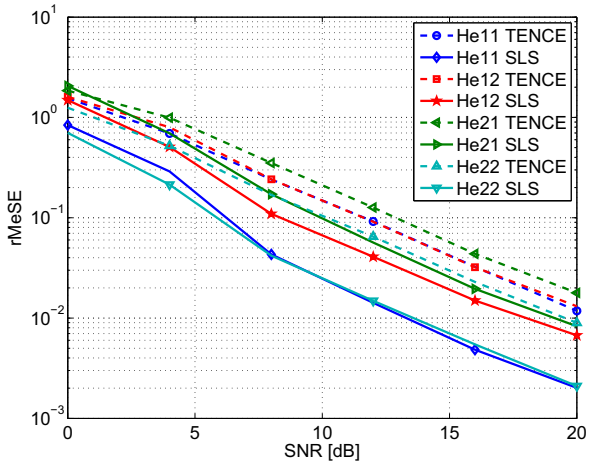

Fig. 4. Median of the channel estimation error vs. SNR in a pure NLOS scenario with spatial correlation where $M_1 = 4$, $M_2 = 5$, $M_{\mathrm{R}} = 3$, $\rho_1 = \rho_2 = 0$, $\rho_{\mathrm{R}} = 0.9$, $K_1 = 0$, and $K_2 = 0$.

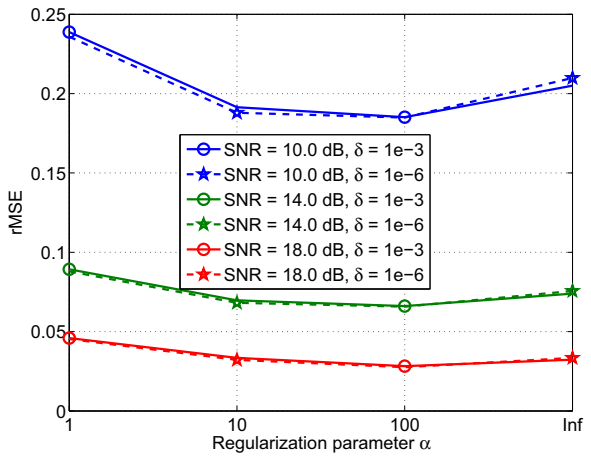

Fig. 5. Mean channel estimation error of the SLS-based channel estimation algorithm for $\boldsymbol{H}_1$ at terminal 1 vs. the regularization parameter α for different values of the SNR and two choices for the threshold parameter δ. Again, an NLOS scenario with $K_1 = K_2 = 0$ and with no spatial correlation ($\rho_1 = \rho_2 = \rho_{\mathrm{R}} = 0$) was considered. The number of antennas was chosen according to $M_1 = M_2 = 2$ and $M_{\mathrm{R}} = 4$.

Figure 4 depicts a scenario where $M_1 = 4$, $M_2 = 5$, $M_{\mathrm{R}} = 3$ and spatial correlation is introduced by setting $\rho_1 = \rho_2 = 0$, $\rho_{\mathrm{R}} = 0.9$. As before we consider the NLOS case by choosing $K_1 = K_2 = 0$. The strong correlation in both channel matrices deteriorate the performance of the TENCE algorithm. Therefore there is a lot of room for improvement via the SLS-based technique which leads to a significant gain especially in the estimation of $\boldsymbol{H}_1$ at terminal 1 and $\boldsymbol{H}_2$ at terminal 2. This is a reasonable observation that confirms the original goal to exploit the fact that the "own" channel matrix is present in the training tensor in the first as well as the second mode.

To show the effect of the regularization parameter α and the threshold parameter δ we study a scenario where each terminal is equipped with 4 antennas, there is no line of sight and also no spatial

see the program

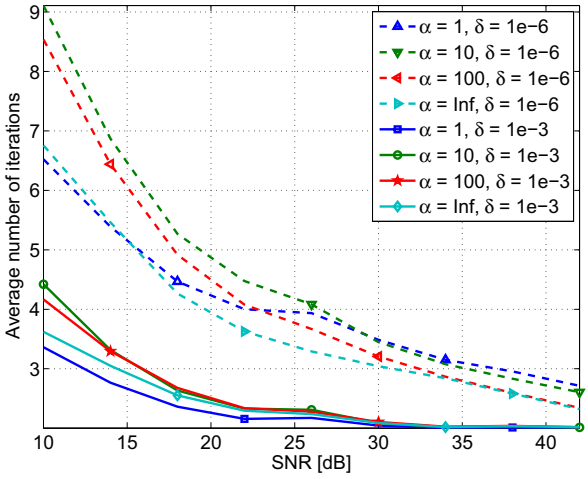

Fig. 6. Average number of iterations for the SLS-based channel estimation algorithm vs. the SNR for the same scenario as in Fig. 5.

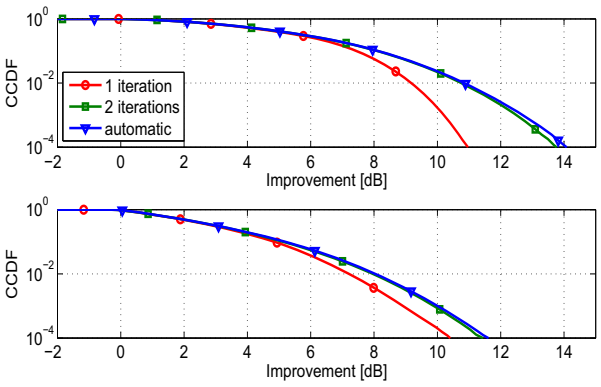

Fig. 8. CCDF of the improvement through the SLS-based refinement in a scenario where $M_1 = M_2 = 6$, $M_\mathrm{R} = 3$, SNR = 30 dB, $\rho_1 = \rho_2 = \rho_\mathrm{R} = 0.9$, and $K_1 = K_2 = 0$. The top graph shows $\boldsymbol{H}_1$, the bottom graph $\boldsymbol{H}_2$, both estimated at terminal 1.

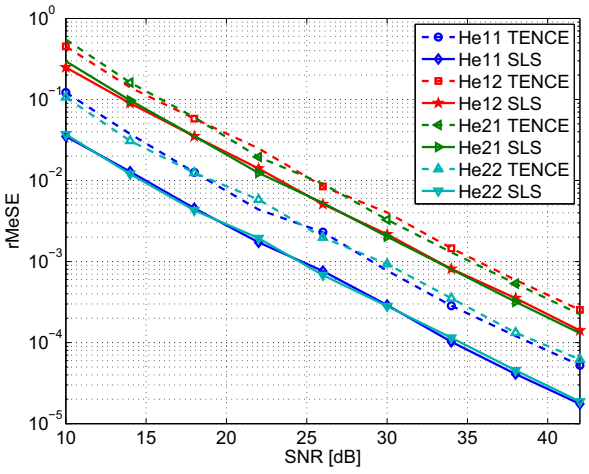

Fig. 7. Median of the channel estimation error vs. SNR for the same scenario as in Fig. 5. Here α was chosen to 100 and δ to 10^{-3}. The curves for $\delta = 10^{-6}$ are indistinguishable from these results (no improvement in accuracy). Also the curves for different values of α are exactly on the ones displayed here (except for $\alpha = 1$ which would be slightly worse).

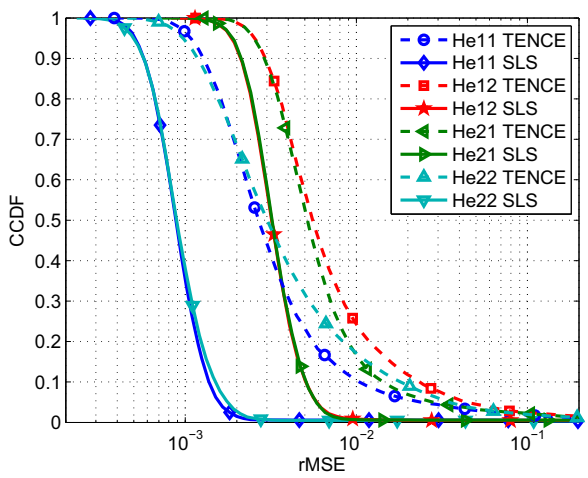

Fig. 9. CCDF of the estimation error for TENCE and the SLS-based technique in a scenario where $M_1 = M_2 = M_\mathrm{R} = 5$, SNR = 20 dB, $\rho_1 = \rho_2 = \rho_\mathrm{R} = 0.0$, and $K_1 = K_2 = 0$.

see the program

correlation. In Figure 5 we display the channel estimation error for $\boldsymbol{H}_1$ vs. the regularization parameter α. The last value for α corresponds to the case where no regularization is used. We observe that for a low SNR, α has an impact on the performance. If α is chosen too small, the correct minimum in the cost function is not achieved and if α is chosen too large the missing regularization has a negative effect on the performance. For an increasing SNR the impact of α diminishes. Consequently, for a very high SNR, no regularization is needed. We also display two values of the threshold parameter δ. As the difference between $\delta = 10^{-3}$ and $\delta = 10^{-6}$ is negligible we can conclude that $\delta = 10^{-3}$ is sufficient, since it requires less iterations. To confirm this in Figure 6 we demonstrate the average number of iterations for the same scenario versus the SNR. We observe that for $\delta = 10^{-3}$, two iterations are sufficient for high SNRs[1]. For low SNRs the number of iteration increases up to four. If δ is set to 10^{-6} the number of iterations increases significantly, however, as confirmed in Figure 5 this does not lead to a visible improvement in channel estimation accuracy. For comparison to the previous results, in Figure 7 we depict the estimation error vs. the SNR for the same scenario as in Figure 5 and Figure 6. Again, the accuracy of all channel estimates is enhanced and the improvement for the "own" channels is more pronounced.

In Figure 8 we show a simulation result in a scenario where both terminals are equipped with six antennas and the relay station with three antennas. We consider a spatial correlation of $\rho_1 = \rho_2 = \rho_{\mathrm{R}} = 0.9$ and no line of sight. Moreover, we set the SNR to 30 dB. In the previous figures we have only seen statistical moments of the estimation error but not its distribution. Therefore, in this simulation we estimate the complementary cumulative density function (CCDF) of the ratio of the channel estimation error before and after the SLS-based refinement, i.e., the probability that this ratio exceeds the abscissa. The top graph shows the estimation of the channel $\boldsymbol{H}_1$ and the bottom graph displays $\boldsymbol{H}_2$, both estimated at terminal 1. We observe that already after a single iteration (depicted in red), significant gains are achievable. The second iteration (green) is already almost indistinguishable from the blue line where up to 10 iterations are allowed and the SLS algorithm is terminated after convergence is detected. Consequently, two iterations are sufficient and the improvement ranges up to 14 dB.

Finally, Figure 9 displays the CCDF of the relative mean square estimation error for TENCE and for the SLS-based technique. The SNR is fixed to 20 dB, the number of antennas is set to 5 for both terminals and the relay. Also, NLOS with no spatial correlation ($\rho_1 = \rho_2 = \rho_{\mathrm{R}} = 0.0$) is considered. We observe that the improvement through the SLS-based refinement is particularly pronounced for the points where the estimation error obtained via TENCE is large. Moreover, the slope of the CCDF for the SLS-based techniques is significantly larger, which demonstrates that the estimates are more robust.

6. CONCLUSIONS

In this contribution an iterative channel estimation algorithm for two-way relaying based on Structured Least Squares is developed. We demonstrate that based on an initial channel estimate that can be obtained by the closed-form TENCE algorithm, the accuracy is improved further by exploiting the structure inherent in the received tensor during the training phase. Furthermore we show that only between one and four iterations are required, even for critical scenarios. Numerical computer simulations depict the achievable gain in terms of the channel estimation accuracy.

[1]Note that the algorithm is terminated if the change from the k-th to the previous iteration drops below the threshold. Consequently, in the case where two iterations are needed, only one iteration changed the result significantly and therefore one could limit the number of iterations to one without losing any performance in the high SNR regime.

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, 2000.

[2] G. Golub and C. F. Van Loan, "An analysis of the total least squares problem", *SIAM Journal Numer. Analysis*, vol. 17, pp. 883 – 893, Dec. 1980.

[3] M. Haardt, "Structured least squares to improve the performance of ESPRIT-type algorithms", *IEEE Transactions on Signal Processing*, vol. 45(3), pp. 792–799, Mar. 1997.

[4] 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 Transactions on Signal Processing*, vol. 56, pp. 3198–3213, July 2008.

[5] I. Hammerstrom, M. Kuhn, C. Esli, J. Zhao, A. Wittneben, and G. Bauch, "MIMO two-way relaying with transmit CSI at the relay", in *Proc. IEEE 8th Workshop on Sig. Proc. Adv. in Wireless Comm. (SPAWC 2007)*, Helsinki, Finland, June 2007.

[6] T. J. Oechtering and H. Boche, "Bidirectional relaying using interference cancellation", in *Proc. ITG/IEEE Workshop on Smart Antennas (WSA '07)*, Vienna, Austria, Feb. 2007.

[7] B. Rankov and A. Wittneben, "Spectral efficient signaling for half-duplex relay channels", in *Proc. 39th Asilomar Conference on Signals, Systems and Computers*, pp. 1066–1071, Pacific Grove, CA, USA, Oct. 2005.

[8] F. Roemer and M. Haardt, "Tensor-structure structured least squares (TS-SLS) to improve the performance of multi-dimensional ESPRIT-type algorithms", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, vol. II, pp. 893–896, Honolulu, HI, Apr. 2007.

[9] F. Roemer and M. Haardt, "Tensor-based channel estimation (TENCE) for two-way relaying with multiple antennas and spatial reuse", in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP 2009)*, Apr. 2009.

[10] T. Unger and A. Klein, "Duplex schemes in multiple antenna two-hop relaying", *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 2008, doi: 10.1155/2008/128592.

[11] T. Unger and A. Klein, "Maximum sum rate of non-regenerative two-way relaying in systems with different complexities", in *Proc. IEEE 9th Intl. Symp. on Personal, Indoor and Mobile Radio Comm. (PIMRC 2008)*, Cannes, France, Sept. 2008.

[12] S. van Huffel and J. Vanderwalle, *The total least squares problem: Computational Aspects and analysis*, Philadelphia: Society for Industrial and Applied Mathematics, 1991.

see the program