Tensor-Based Channel Estimation and Iterative Refinements for Two-Way Relaying With Multiple Antennas and Spatial Reuse

Florian Roemer, *Student Member, IEEE*, and Martin Haardt, *Senior Member, IEEE*

***Abstract*—Relaying is one of the key technologies to satisfy the demands of future mobile communication systems. In particular, two-way relaying is known to exploit the radio resources in a very efficient manner. In this contribution, we consider two-way relaying with amplify-and-forward (AF) MIMO relays. Since AF relays do not decode the signals, the separation of the data streams has to be performed by the terminals themselves. For this task both nodes require reliable channel knowledge of all relevant channel parameters. Therefore, we examine channel estimation schemes for two-way relaying with AF MIMO relays. We investigate a simple Least Squares (LS) based scheme for the estimation of the compound channels as well as a tensor-based channel estimation (TENCE) scheme which takes advantage of the special structure in the compound channel matrices to further improve the estimation accuracy. Note that TENCE is purely algebraic (i.e., it does not require any iterative procedures) and applicable to arbitrary antenna configurations. Then we demonstrate that the solution obtained by TENCE can be improved by an iterative refinement which is based on the structured least squares (SLS) technique. In this application, between one and four iterations are sufficient and consequently the increase in computational complexity is moderate. The iterative refinement is optional and targeted for cases where the channel estimation accuracy is critical. Moreover, we propose design rules for the training symbols as well as the relay amplification matrices during the training phase to facilitate the estimation procedures. Finally, we evaluate the achievable channel estimation accuracy of the LS-based compound channel estimation scheme as well as the tensor-based approach and its iterative refinement via numerical computer simulations.**

***Index Terms*—Amplify and forward, channel estimation, structured least squares, two-way relaying.**

I. Introduction

ONE of the major goals in the development of future mobile communication systems is the ubiquitous provision of a reliable radio access supporting very high data rates. This is a challenging task since the network faces different propagation conditions within its coverage area. Due to the fact that large distances as well as obstacles such as tall buildings severely attenuate the signal, a large density of network nodes is required. However, this density is limited by installation and maintenance costs of the network nodes. Consequently, lowering this cost is a key aspect in the design of mobile communication systems.

A promising technique to achieve this goal is the deployment of relays. These intermediate network nodes require less space and less power than base stations and hence have a significantly lower installation and maintenance cost. They can assist the transmission between any two communication partners in the mobile network, i.e., between two users as well as between a user and a base station. The concept of relaying has sparked a significant research interest in recent years. An overview of relaying techniques and their impact on mobile communication systems is presented in [19].

A significant part of the existing literature on relaying is dedicated to one-way relaying. Here one-way means that the transmission is directed in one direction, i.e., from a specific source node via one or several relays to a specific destination node. The one-way relaying channel is quite well understood. Performance limits, achievable rates, and efficient signaling schemes in the single hop case are, for example, examined in [16], a treatment of the multi-hop case is found in [1].

In contrast to one-way relaying, the transmission in both directions is considered by the two-way relaying scheme. In the first phase both terminals transmit their data simultaneously to the relay which receives the superposition of these transmissions. In a subsequent second phase, the relay transmits to both terminals simultaneously. The advantage of this scheme is that radio resources are used in a particularly efficient manner. The two-way communication channel was already studied by Shannon [28] and has been rediscovered as a means to compensate the spectral efficiency loss in one-way relaying due to the half duplex constraint of the relay [21], [22].

Relay are usually further divided into two types: regenerative or decode-and-forward (DF) relays and nonregenerative or amplify-and-forward (AF) relays. The difference is that DF relays decode the received transmissions and reencode them for the second hop, whereas AF relays amplify the received signal and retransmit it without any decoding step. We focus on AF relays since they are simpler to implement, do not need to support all modulation and coding schemes in the network, and do not cause additional decoding delays present for DF relays. For a thorough treatment of two-way relaying with DF relays, the

Manuscript received October 01, 2009; accepted July 12, 2010. Date of publication July 29, 2010; date of current version October 13, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiqi Gao. Parts of this paper have been published at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan, April 2009, and at the IEEE/ITG Workshop on Smart Antennas (WSA), Berlin, Germany, February 2009.

The authors are with Ilmenau University of Technology, Communications Research Laboratory, D-98684 Ilmenau, Germany (e-mail: florian.roemer@tu-ilmenau.de; martin.haardt@tu-ilmenau.de; website: www: http://www.tu-ilmenau.de/crl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2010.2062179

reader is referred to [13], [17], and [18]. Note that besides AF and DF other types of relaying schemes exist, e.g., space-time coding is discussed in [3], XOR and superposition coding are discussed in [10], estimate-and-forward (EF) as well as compress-and-forward (CF) in the context of one-way relaying are found in [14].

Most previous publications on two-way AF relaying have assumed that channel knowledge is available at the terminals. While the impact of imperfect channel state information on the performance of relay networks has been investigated in [31], no particular channel estimation schemes suitable for two-way relaying have been proposed. A least-squares-based estimation scheme for one-way relaying can be found in [15]. Maximum likelihood channel estimation schemes for two-way relaying with AF relays are proposed in [5] and [6]; however, these techniques are limited to the single-antenna case and a MIMO extension is not straightforward. Channel estimation in two-way relaying systems with multiple antennas is limited to relays employing DF [32] or space-time coding [30]. The very recent manuscript [20] considers channel estimation in MIMO two-way relaying systems based on OFDM and relays using "purely analog AF," i.e., the received signal at each antenna is multiplied by one scalar real-valued amplification α and then retransmitted. Note that [20] cannot be compared to the channel estimation schemes proposed in this manuscript since a) we consider another form of AF where the relay may multiply the received signal vector with one complex relay amplification matrix, b) in [20] the OFDM system and the resulting circulant structure of the channels is explicitly exploited, and c) in [20] only the compound channels are estimated whereas we focus on decoupling the compound channels into the separate channels between the terminals and the relay.

We examine channel estimation schemes for MIMO two-way relaying systems with amplify-and-forward relays in this paper. First we discuss a simple least squares (LS) based scheme for the estimation of the compound channel matrices. Next, we propose the purely algebraic tensor-based channel estimation (TENCE) algorithm and an iterative scheme based on structured least squares (SLS) [8] to refine the initial solution obtained via TENCE. Moreover, we develop design rules and recommendations for the training sequences as well as the relay amplification matrices during the training phase to facilitate the channel estimation.

We compare the LS-based compound channel estimator with the tensor-based approach in terms of the required training overhead as well as the achievable estimation accuracy. Due to the fact that the tensor-based approach solves a nonlinear least squares problem and exploits the structure of the channels, it can yield a more accurate channel estimate in the case where the number of antennas at the relay is smaller than the number of antennas at the user terminals.

The main extensions compared to the conference versions of the channel estimation schemes [26], [27] are the following: a) The detailed development of the design rules and recommendations for the pilot symbol matrix and the relay amplification tensor, highlighting the remaining flexibility in their design; b) a more elaborate discussion of the ambiguities in the channel estimates showing how the ambiguities have been reduced to a single sign and why this is irrelevant; c) a more detailed and modular presentation of the required procedures for TENCE, e.g., via the separated algorithms 1–3; d) the complete proof for the required algebraic manipulations along with some Lemmas that might be used in other applications; e) the LS-based compound channel estimation scheme and its comparison to the tensor-based approach; and f) the discussion chapter elaborating on the complexity and the single-antenna case.

The remainder of this paper is organized as follows. In Section II, we introduce the notation used in the paper and define the necessary operators to handle matrices and tensors. Section III describes the two-way relaying system and explains the data model. In Section IV, the LS-based compound channel estimator is introduced. Then, in Section V, we derive the TENCE algorithm and propose design rules for the training data as well as the relay amplification matrices. The iterative refinement of TENCE is derived in Section VI. A discussion of all schemes in terms of complexity and the special case of a single antenna at the terminals follows in Section VII. Finally, simulation results are presented in Section VIII before the conclusions are drawn in Section IX. To enhance the readability of the paper, some of the proofs on properties of matrices, tensors, and norms are moved into the Appendix.

II. 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, A, B), vectors as lower-case bold-faced letters $(\boldsymbol{a}, \boldsymbol{b})$, matrices are represented by upper-case bold-faced letters $(\boldsymbol{A}, \boldsymbol{B})$, and tensors are written as bold-faced calligraphic letters $(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}})$. To retrieve the element (i, j) from a matrix $\boldsymbol{A}$ we use the notation $[\boldsymbol{A}]_{i,j}$. Similarly the ith column and the jth row of $\boldsymbol{A}$ are represented by $[\boldsymbol{A}]_{:,i}$ and $[\boldsymbol{A}]_{j,:}$, respectively.

The superscripts $^{\mathrm{T}}, ^{\mathrm{H}}, ^{-1}, ^{+}$ represent matrix transposition, Hermitian transposition, matrix inverse, and the Moore–Penrose pseudo inverse, respectively. Moreover, * denotes the complex conjugation operator. The Kronecker product between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is symbolized by $\boldsymbol{A} \otimes \boldsymbol{B}$ and the Khatri–Rao (columnwise Kronecker) product by $\boldsymbol{A} \diamond \boldsymbol{B}$. Moreover, the Schur product $\boldsymbol{A} \odot \boldsymbol{B}$ and the inverse Schur product $\boldsymbol{A} \oslash \boldsymbol{B}$ represent the elementwise multiplication and division of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively.

A 3-dimensional tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times M_2 \times M_3}$ is a three-way array of size M_r along mode r. The r-mode vectors of $\boldsymbol{\mathcal{A}}$ are obtained by varying the rth index and keeping all other indexes fixed. Collecting all r-mode vectors into a matrix we obtain the r-mode unfolding of $\boldsymbol{\mathcal{A}}$ which is represented by $[\boldsymbol{\mathcal{A}}]_{(r)} \in \mathbb{C}^{M_r \times (M_1 \cdot M_2 \cdot M_3)/M_r}$. The ordering of the columns in $[\boldsymbol{\mathcal{A}}]_{(r)}$ is chosen in accordance with [4]. The r-rank of $\boldsymbol{\mathcal{A}}$ is defined as the (matrix) rank of $[\boldsymbol{\mathcal{A}}]_{(r)}$. Note that, in general, all the r-ranks of one tensor can be different.

The r-mode product between a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times M_2 \times M_3}$ and a matrix $\boldsymbol{U}_r \in \mathbb{C}^{P_r \times M_r}$ is symbolized by $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{A}} \times_r \boldsymbol{U}_r$. It is computed by multiplying all r-mode vectors from the left-hand side by the matrix $\boldsymbol{U}_r$, i.e., $[\boldsymbol{\mathcal{B}}]_{(r)} = \boldsymbol{U}_r \cdot [\boldsymbol{\mathcal{A}}]_{(r)}$. To represent the concatenation of two tensor $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ along the rth mode

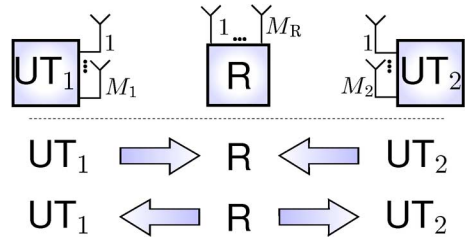

Fig. 1. Two-way relaying system model: two user terminals equipped with M_1 and M_2 antennas communicate with a relay station that has M_{R} antennas. There are two transmission phases: first both terminals transmit to the relay then the relay sends the amplified signal back to both terminals.

we introduce the operator $[\boldsymbol{\mathcal{A}} \sqcup_r \boldsymbol{\mathcal{B}}]$ [9]. Note that this operation requires $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ to have the same size in all modes except for the rth mode.

The rank of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{M_1 \times M_2 \times M_3}$ can be defined as the smallest integer number r such that there exist matrices $\boldsymbol{F}_1 \in \mathbb{C}^{M_1 \times r}$, $\boldsymbol{F}_2 \in \mathbb{C}^{M_2 \times r}$, and $\boldsymbol{F}_3 \in \mathbb{C}^{M_3 \times r}$ which satisfy $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{I}}_{3,r} \times_1 \boldsymbol{F}_1 \times_2 \boldsymbol{F}_2 \times_3 \boldsymbol{F}_3$. This is known as the Parallel Factor (PARAFAC) decomposition of $\boldsymbol{\mathcal{A}}$ [12]. Note that the tensor rank satisfies $\mathrm{rank}\{\boldsymbol{\mathcal{A}}\} \geq \mathrm{rank}\{[\boldsymbol{\mathcal{A}}]_{(n)}\}$ for $n = 1, 2, 3$.

The matrices $\boldsymbol{0}_{p\times q}$, $\boldsymbol{1}_{p\times q}$, and $\boldsymbol{I}_p$ symbolize the zero matrix of size $p \times q$, a $p \times q$ matrix of ones, and the $p \times p$ identity matrix, respectively. The tensor $\boldsymbol{\mathcal{I}}_{3,p}$ is the 3-dimensional identity tensor of size $p \times p \times p$ which is one if all three indexes are equal and zero otherwise.

The vectorization operator $\mathrm{vec}\{.\}$ aligns all the elements of a matrix or a tensor into a vector. For a tensor, the order of the elements is chosen consistent with the matrix, i.e., first the first (row) index is varied, then the second (column) index, and then the third index. For a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I \times J \times K}$, 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 [23]:

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

III. System Description

A. Two-Way AF Relaying

The two-way AF relaying scenario under investigation is depicted in Fig. 1. We consider the communication between two user terminals UT_1 and UT_2 with the help of an intermediate relay station RS. The terminals UT_1 and UT_2 are equipped with M_1 and M_2 antennas, respectively. The number of antennas at the relay station is denoted by M_{R}. The terminals and the relay station are assumed to operate in a half-duplex mode, i.e., they cannot transmit and receive at the same time.

To save the rare time and frequency resources, only two transmission phases are used in two-way relaying. In the first phase, both user terminals transmit their data to the relay, where the transmissions interfere. The AF relay amplifies the received signal and sends it back to the user terminals in the second phase. We assume time-division duplex (TDD), i.e., the same frequencies are used for the two transmission phases in subsequent time slots.

Both terminals receive a superposition of the transmission from the other terminal and interference caused by their own transmissions. However, since each terminal has knowledge of the data it has transmitted, with additional channel knowledge this "self-interference" can be canceled. This technique is often referred to as analogue network coding (ANC) [11].

B. Data Model

In the first transmission phase, the terminals transmit data to the relay station. Assuming frequency-flat fading, the signal received at the relay is given by

$$\boldsymbol{r} = \boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \boldsymbol{n}_{\mathrm{R}} \in \mathbb{C}^{M_{\mathrm{R}}} \tag{2}$$

where $\boldsymbol{x}_1 \in \mathbb{C}^{M_2}$ and $\boldsymbol{x}_2 \in \mathbb{C}^{M_2}$ are the transmitted vectors from UT_1 and UT_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 quasi-static block fading MIMO channel between the relay and UT_1 and UT_2. Moreover, the vector $\boldsymbol{n}_{\mathrm{R}}$ represents the additive noise vector at the relay station.

The amplified signal the relay station transmits in the second time slot is expressed as

$$\overline{\boldsymbol{r}} = \underbrace{\gamma \cdot \boldsymbol{G}}_{\boldsymbol{G}_\gamma} \cdot \boldsymbol{r} = \boldsymbol{G}_\gamma \cdot \boldsymbol{r} \in \mathbb{C}^{M_{\mathrm{R}}}. \tag{3}$$

Here, $\boldsymbol{G}_\gamma \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$ denotes the relay amplification matrix, which consists of an amplification matrix $\boldsymbol{G}$ normalized such that $\|\boldsymbol{G}\|_{\mathrm{F}} = 1$ and a scalar parameter $\gamma \in \mathbb{R}^+$. The task of γ is to compensate the path loss in the transmissions from the terminals to the relay such that the relay transmit power constraint is not violated. An instantaneous estimate of γ is given by

$$\gamma^2 = \frac{P_{\mathrm{T,R}}}{\|\boldsymbol{r}\|_2^2}. \tag{4}$$

Since a rapid adaptation of γ renders the ANC step infeasible, this instantaneous estimate is typically replaced by a longer-term average of the received power levels.[1]

The signals received by UT_1 and UT_2 are denoted by $\boldsymbol{y}_1 \in \mathbb{C}^{M_1}$ and $\boldsymbol{y}_2 \in \mathbb{C}^{M_2}$, respectively. Since the system operates in TDD mode, the received signals can be expressed as

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{H}_1^{\mathrm{T}} \cdot \overline{\boldsymbol{r}} + \boldsymbol{n}_1 \\ &= \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}_\gamma \cdot (\boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \boldsymbol{n}_{\mathrm{R}}) + \boldsymbol{n}_1 \\ \boldsymbol{y}_2 &= \boldsymbol{H}_2^{\mathrm{T}} \cdot \overline{\boldsymbol{r}} + \boldsymbol{n}_2 \\ &= \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G}_\gamma \cdot (\boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \boldsymbol{n}_{\mathrm{R}}) + \boldsymbol{n}_2 \end{aligned} \tag{5}$$

where we have assumed that reciprocity holds and that the channels have not changed between the two transmission phases. Note that (5) can be rewritten in the following form:

$$\begin{aligned} \boldsymbol{y}_1 &= \gamma \cdot \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \gamma \cdot \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \tilde{\boldsymbol{n}}_1 \\ \boldsymbol{y}_2 &= \gamma \cdot \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \gamma \cdot \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \tilde{\boldsymbol{n}}_2 \end{aligned} \tag{6}$$

where $\tilde{\boldsymbol{n}}_i = \boldsymbol{H}_i^{\mathrm{T}} \cdot \boldsymbol{G}_\gamma \cdot \boldsymbol{n}_{\mathrm{R}} + \boldsymbol{n}_i$ represents the effective noise contribution for $i = 1, 2$. If the user terminals possess knowledge

[1] In practice, γ should be chosen a bit smaller than the average to accommodate instantaneous signal fluctuations within the safe transmit power range.

of the channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ they can cancel the interference they have received from their own transmissions and then decode the transmissions of the other user terminal. Therefore, we focus on the acquisition of channel state information at the terminals. For simplicity, we drop the scaling parameter γ by considering $\gamma = 1$ and focus on the design of the normalized relay amplification matrix $\boldsymbol{G}$. Since the terminals do not know γ, they estimate it as part of their channels. For most schemes, such a scaling is irrelevant. If the power levels are important, the value of γ used during the training phase has to be signaled by the relay to obtain this unknown parameter.

Introducing the short-hand notation $\boldsymbol{H}_{i,j}^{(\mathrm{e})} = \gamma \cdot \boldsymbol{H}_i^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_j$ for the effective channel between UT_i and UT_j, (6) simplifies to

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{H}_{1,1}^{(\mathrm{e})} \cdot \boldsymbol{x}_1 + \boldsymbol{H}_{1,2}^{(\mathrm{e})} \cdot \boldsymbol{x}_2 + \tilde{\boldsymbol{n}}_1 \\ \boldsymbol{y}_2 &= \boldsymbol{H}_{2,1}^{(\mathrm{e})} \cdot \boldsymbol{x}_1 + \boldsymbol{H}_{2,2}^{(\mathrm{e})} \cdot \boldsymbol{x}_2 + \tilde{\boldsymbol{n}}_2 \end{aligned} \tag{7}$$

where $\boldsymbol{H}_{i,i}^{(\mathrm{e})}$ conveys the self-interference terms for $i = 1, 2$ and $\boldsymbol{H}_{i,j}^{(\mathrm{e})}$ conveys the desired signals for $i, j = 1, 2$, $i \neq j$. Consequently, UT_1 requires knowledge of a) $\boldsymbol{H}_{1,1}^{(\mathrm{e})}$ in order to subtract the self-interference caused by its own transmitted signal $\boldsymbol{x}_1$, b) $\boldsymbol{H}_{1,2}^{(\mathrm{e})}$ in order to decode the transmission from UT_2, and c) $\boldsymbol{H}_{2,1}^{(\mathrm{e})}$ in order to precode its own transmission for UT_2. For instance, UT_1 may choose the r dominant right singular vectors of $\boldsymbol{H}_{2,1}^{(\mathrm{e})}$ for precoding and the Hermitian transpose of the r dominant left singular vectors of $\boldsymbol{H}_{1,2}^{(\mathrm{e})}$ for decoding the transmissions, where r is the number of data streams that are spatially multiplexed.

We will discuss two channel estimation schemes in the sequel. In Section IV we introduce a LS-based channel estimation scheme that finds estimates for the effective channels $\boldsymbol{H}_{i,j}^{(\mathrm{e})}$ at UT_i directly without taking advantage of their special structure. In Section V we show a tensor-based channel estimation scheme that exploits the structure of the compound channels by estimating $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ separately.

IV. Least-Squares Based Channel Estimation

In this section we show a LS-based scheme for estimating the compound channel matrices $\boldsymbol{H}_{i,j}^{(\mathrm{e})}$ at UT_i for $i, j = 1, 2$. While this scheme is simple and robust, it is not necessarily optimal, since it ignores the special structure of the compound channel matrices. It also fails to provide UT_i with an estimate of $\boldsymbol{H}_{j,i}^{(\mathrm{e})}$ which it needs to compute a proper precoding matrix. Note that $\boldsymbol{H}_{j,i}^{(\mathrm{e})} = \boldsymbol{H}_{i,j}^{(\mathrm{e})^{\mathrm{T}}}$ only if $\boldsymbol{G} = \boldsymbol{G}^{\mathrm{T}}$. We have shown in [25] that ANOMAX with unequal weighting should be chosen in near-far scenarios. In this case, $\boldsymbol{G} \neq \boldsymbol{G}^{\mathrm{T}}$.

In order to estimate the channels, both terminals transmit a sequence of N_{P} pilot symbols $\boldsymbol{x}_{1,j}, \boldsymbol{x}_{2,j}$ for $j = 1, 2, \ldots, N_{\mathrm{P}}$. The overall training data received by the relay can be expressed as

$$\boldsymbol{R} = \boldsymbol{H}_1 \cdot \boldsymbol{X}_1 + \boldsymbol{H}_2 \cdot \boldsymbol{X}_2 + \boldsymbol{N}_{\mathrm{R}} \in \mathbb{C}^{M_{\mathrm{R}} \times N_{\mathrm{P}}} \tag{8}$$

where the pilot symbol matrices $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are defined as

$$\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,N_{\mathrm{P}}}] \in \mathbb{C}^{M_i \times N_{\mathrm{P}}}. \tag{9}$$

Let $\boldsymbol{X} = [\boldsymbol{X}_1^{\mathrm{T}}, \boldsymbol{X}_2^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{(M_1+M_2) \times N_{\mathrm{P}}}$. Then, a least-squares estimate of the channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ at the relay station is obtained via

$$[\hat{\boldsymbol{H}}_1, \hat{\boldsymbol{H}}_2] = \boldsymbol{R} \cdot \boldsymbol{X}^{+}. \tag{10}$$

Note that (10) requires $N_{\mathrm{P}} \geq M_1 + M_2$. Based on these estimates, the relay can compute a suitable relay amplification matrix $\boldsymbol{G}$, e.g., via the Algebraic Norm-Maximizing (ANOMAX) transmit strategy [24]. The received training data $\boldsymbol{R}$ is then multiplied with $\boldsymbol{G}$ and transmitted back to the terminals. The signal received at UT_i, $i = 1, 2$ can be expressed as

$$\boldsymbol{Y}_i = \boldsymbol{H}_{i,i}^{(\mathrm{e})} \cdot \boldsymbol{X}_i + \boldsymbol{H}_{i,j}^{(\mathrm{e})} \cdot \boldsymbol{X}_j + \tilde{\boldsymbol{N}}_i \tag{11}$$

Consequently, the LS estimates of the effective channels are given by

$$\begin{aligned} \left[\hat{\boldsymbol{H}}_{1,1}^{(\mathrm{e})}, \hat{\boldsymbol{H}}_{1,2}^{(\mathrm{e})}\right] &= \boldsymbol{Y}_1 \cdot \boldsymbol{X}^{+} \quad \text{for } \mathsf{UT}_1 \quad \text{and} \\ \left[\hat{\boldsymbol{H}}_{2,1}^{(\mathrm{e})}, \hat{\boldsymbol{H}}_{2,2}^{(\mathrm{e})}\right] &= \boldsymbol{Y}_2 \cdot \boldsymbol{X}^{+} \quad \text{for } \mathsf{UT}_2 \end{aligned} \tag{12}$$

where we again require that $N_{\mathrm{P}} \geq M_1 + M_2$. Consequently, with $M_1 + M_2$ pilots we have estimated the channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ at the relay, the effective channel matrices $\boldsymbol{H}_{1,1}^{(\mathrm{e})}$ and $\boldsymbol{H}_{1,2}^{(\mathrm{e})}$ at UT_1, and the effective channel matrices $\boldsymbol{H}_{2,1}^{(\mathrm{e})}$ and $\boldsymbol{H}_{2,2}^{(\mathrm{e})}$ at UT_2. However, to compute proper precoding matrices, UT_1 requires an estimate of $\boldsymbol{H}_{2,1}^{(\mathrm{e})}$ and UT_2 needs an estimate of $\boldsymbol{H}_{1,2}^{(\mathrm{e})}$. In the case where the relay chooses its amplification matrix $\boldsymbol{G}$ such that $\boldsymbol{G} = \boldsymbol{G}^{\mathrm{T}}$, UT_1 can obtain an estimate of $\boldsymbol{H}_{2,1}^{(\mathrm{e})}$ via $\hat{\boldsymbol{H}}_{2,1}^{(\mathrm{e})} = \hat{\boldsymbol{H}}_{1,2}^{(\mathrm{e})^{\mathrm{T}}}$. Otherwise, additional pilots are needed to estimate $\boldsymbol{H}_{2,1}^{(\mathrm{e})}$ at UT_1 and $\boldsymbol{H}_{1,2}^{(\mathrm{e})}$ at UT_2. Alternatively, open loop techniques such as Orthogonal Space-Time Codes can be used to convey the desired information without transmit channel state information. Another drawback of the simple LS-based channel estimation procedure is that the structure of the compound channels is completely ignored. We show in the next section how the estimation accuracy can be improved by exploiting this special structure and estimating the channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ directly.

V. Algebraic Channel Estimation Algorithm: TENCE

The LS-based scheme for the estimation of the effective (compound) channel ignores their structure completely. For instance, $\boldsymbol{H}_{i,i}^{(\mathrm{e})} = \boldsymbol{H}_i^{\mathrm{T}} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_i$, i.e., the M_i^2 elements of $\boldsymbol{H}_{i,i}^{(\mathrm{e})}$ are second-order polynomials in the $M_i \cdot M_{\mathrm{R}}$ coefficients in $\boldsymbol{H}_i$. Consequently, if $M_{\mathrm{R}} < M_i$ it may be more efficient to estimate $\boldsymbol{H}_i$ by solving a quadratic LS problem and exploiting the special structure of $\boldsymbol{H}_{i,i}^{(\mathrm{e})}$. This is the motivation behind the tensor-based channel estimation (TENCE) scheme presented in this section. TENCE itself is an algebraic (i.e., noniterative) solution to the nonlinear least squares problem, which is very simple to compute. If a more accurate solution is required, TENCE can be refined by a few iterations of an iterative channel estimation scheme described in Section VI.

A. Training

In order to acquire channel knowledge of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ at the user terminals we require a special training phase in which known pilot symbols are transmitted for known relay amplification matrices. We therefore divide the training phase into N_{R} frames. For each frame, we choose a particular relay amplification matrix $\boldsymbol{G}^{(i)} \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$, $i = 1, 2, \ldots, N_{\mathrm{R}}$. For this fixed $\boldsymbol{G}^{(i)}$, pilot sequences $\boldsymbol{x}_{1,j} \in \mathbb{C}^{M_1}$ and $\boldsymbol{x}_{2,j} \in \mathbb{C}^{M_2}$ for $j = 1, 2, \ldots, N_{\mathrm{P}}$ are transmitted from UT_1 and UT_2, respectively. The number of pilot symbols N_{P} that are transmitted for each $\boldsymbol{G}^{(i)}$ and the number of frames N_{R} will be specified later. Note that the total number of training time slots is given by $N_{\mathrm{R}} \cdot N_{\mathrm{P}}$. The received signal from the jth pilot symbol within the ith training block is given by

$$\begin{aligned} \boldsymbol{y}_{1,j,i} &= \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}^{(i)} \cdot \boldsymbol{H}_1 \cdot \boldsymbol{x}_{1,j} + \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}^{(i)} \cdot \boldsymbol{H}_2 \cdot \boldsymbol{x}_{2,j} + \tilde{\boldsymbol{n}}_{1,j,i} \\ \boldsymbol{y}_{2,j,i} &= \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G}^{(i)} \cdot \boldsymbol{H}_1 \cdot \boldsymbol{x}_{1,j} + \boldsymbol{H}_2^{\mathrm{T}} \cdot \boldsymbol{G}^{(i)} \cdot \boldsymbol{H}_2 \cdot \boldsymbol{x}_{2,j} + \tilde{\boldsymbol{n}}_{2,j,i}. \end{aligned} \tag{13}$$

The data model in (13) can be expressed in a more compact form using tensor notation. To this end, let us introduce the following definitions:

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

$$\begin{aligned} \boldsymbol{X}_1 &\doteq [\boldsymbol{x}_{1,1}, \ldots, \boldsymbol{x}_{1,N_{\mathrm{P}}}] \in \mathbb{C}^{M_1 \times N_{\mathrm{P}}} \\ \boldsymbol{X}_2 &\doteq [\boldsymbol{x}_{2,1}, \ldots, \boldsymbol{x}_{2,N_{\mathrm{P}}}] \in \mathbb{C}^{M_2 \times N_{\mathrm{P}}} \end{aligned} \tag{14}$$

$$\boldsymbol{X} \doteq \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{bmatrix} \in \mathbb{C}^{(M_1 + M_2) \times N_{\mathrm{P}}} \tag{15}$$

$$\boldsymbol{\mathcal{G}} \doteq \left[\boldsymbol{G}^{(1)} \sqcup_3 \boldsymbol{G}^{(2)} \ldots \sqcup_3 \boldsymbol{G}^{(N_{\mathrm{R}})} \right] \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}} \times N_{\mathrm{R}}}. \tag{16}$$

Using these definitions, the received training data can be rewritten as

$$\begin{aligned} \boldsymbol{\mathcal{Y}}_1 &= \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{H}_1^{\mathrm{T}} \times_2 (\boldsymbol{H} \cdot \boldsymbol{X})^{\mathrm{T}} + \boldsymbol{\mathcal{N}}_1 \in \mathbb{C}^{M_1 \times N_{\mathrm{P}} \times N_{\mathrm{R}}} \\ \boldsymbol{\mathcal{Y}}_2 &= \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{H}_2^{\mathrm{T}} \times_2 (\boldsymbol{H} \cdot \boldsymbol{X})^{\mathrm{T}} + \boldsymbol{\mathcal{N}}_2 \in \mathbb{C}^{M_2 \times N_{\mathrm{P}} \times N_{\mathrm{R}}} \end{aligned} \tag{17}$$

where the tensors $\boldsymbol{\mathcal{Y}}_1$ and $\boldsymbol{\mathcal{Y}}_2$ contain the vectors $\boldsymbol{y}_{1,j,i}$ and $\boldsymbol{y}_{2,j,i}$ in such a way that the second index in the tensor represents $j = 1, 2, \ldots, N_{\mathrm{P}}$ and the third index represents $i = 1, 2, \ldots, N_{\mathrm{R}}$. The tensors $\boldsymbol{\mathcal{N}}_1$ and $\boldsymbol{\mathcal{N}}_2$ collect of the noise vectors $\tilde{\boldsymbol{n}}_{1,j,i}$ and $\tilde{\boldsymbol{n}}_{2,j,i}$ in a similar fashion.

It should be noted that the structure of (17) is similar to a Tucker-2 decomposition [12]. However, the difference to Tucker-2 is that the core tensor is known (and can even be designed). Also, a certain symmetry in the factors is present since the two-mode factor includes $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ which are also present in the one-mode factor. Finally, the decomposition involves the pilot matrix $\boldsymbol{X}$ which is also known and can be designed. These particular properties can be exploited to derive efficient solutions to the channel estimation problem. Moreover, we obtain design rules and recommendations on how to choose the pilot matrix $\boldsymbol{X}$ and the training tensor $\boldsymbol{\mathcal{G}}$ in order to facilitate the implementation of these channel estimation algorithms.[2]

[2]We use the term "design rules" for properties that $\boldsymbol{X}$ and $\boldsymbol{\mathcal{G}}$ *must* fulfill for TENCE to be applicable and "design recommendations" for additional properties that $\boldsymbol{X}$ and $\boldsymbol{\mathcal{G}}$ *may* satisfy to improve the estimation accuracy.

B. Derivation of TENCE

Based on this training data we show the derivation of TENCE in this section. For notational convenience, we ignore the contribution of the noise and write equalities. In the presence of noise, the following identities will only hold approximately. Also, we derive the solution for UT_1 only. Due to the symmetry of the problem, the solution for UT_2 is very similar.

First of all, consider the training tensor $\boldsymbol{\mathcal{G}}$. Let r_G be the rank of the tensor $\boldsymbol{\mathcal{G}}$. Then $\boldsymbol{\mathcal{G}}$ can be expressed in terms of its PARAFAC decomposition [12]

$$\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{I}}_{3,r_G} \times_1 \boldsymbol{G}_1 \times_2 \boldsymbol{G}_2 \times_3 \boldsymbol{G}_3 \tag{18}$$

where $\boldsymbol{\mathcal{I}}_{3,r_G}$ is the identity tensor of size $r_G \times r_G \times r_G$ and the matrices $\boldsymbol{G}_1 \in \mathbb{C}^{M_{\mathrm{R}} \times r_G}$, $\boldsymbol{G}_2 \in \mathbb{C}^{M_{\mathrm{R}} \times r_G}$, and $\boldsymbol{G}_3 \in \mathbb{C}^{N_{\mathrm{R}} \times r_G}$ represent the factor matrices of the decomposition. Instead of designing the tensor $\boldsymbol{\mathcal{G}}$ directly, we propose design rules for the matrices $\boldsymbol{G}_1$, $\boldsymbol{G}_2$, and $\boldsymbol{G}_3$ individually from the steps in the derivation where they appear.

Inserting (18) into (17) yields

$$\boldsymbol{\mathcal{Y}}_1 = \boldsymbol{\mathcal{I}}_{r_G} \times_1 \left(\boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}_1 \right) \times_2 (\boldsymbol{X}^{\mathrm{T}} \cdot \boldsymbol{H}^{\mathrm{T}} \cdot \boldsymbol{G}_2) \times_3 \boldsymbol{G}_3. \tag{19}$$

Using the elementary properties of n-mode products shown in (58) in the Appendix, it is easy to verify that the three-mode unfolding of (19) satisfies

$$[\boldsymbol{\mathcal{Y}}_1]_{(3)} = \boldsymbol{G}_3 \cdot \left[\left(\boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}_1 \right) \diamond (\boldsymbol{X}^{\mathrm{T}} \cdot \boldsymbol{H}^{\mathrm{T}} \cdot \boldsymbol{G}_2) \right]^{\mathrm{T}}. \tag{20}$$

In order to isolate the Khatri–Rao product, the multiplication by $\boldsymbol{G}_3$ must be inverted. To guarantee that this inversion is unique, we require that $N_{\mathrm{R}} \geq r_G$ and $\boldsymbol{G}_3$ to be a full rank matrix. This leads to the first design rule for $\boldsymbol{\mathcal{G}}$.

Design Rule 1: The number of training blocks N_{R} must satisfy $N_{\mathrm{R}} \geq r_G$ and $\boldsymbol{G}_3$ must have full column rank (r_G).

Since we can design $\boldsymbol{G}_3$ we can choose this matrix such that it has orthogonal columns, i.e., $\boldsymbol{G}_3^{\mathrm{H}} \boldsymbol{G}_3$ is a scaled identity. This guarantees that the inversion step is well conditioned, which is favorable from a numerical standpoint and avoids explicit matrix inversion.

Design Recommendation 1: The three-mode factor matrix $\boldsymbol{G}_3$ should have orthogonal columns.

We can now isolate the Khatri–Rao product in (20) in the following way:

$$\left(\boldsymbol{G}_3^{+} \cdot [\boldsymbol{\mathcal{Y}}_1]_{(3)} \right)^{\mathrm{T}} = \left(\boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}_1 \right) \diamond (\boldsymbol{X}^{\mathrm{T}} \cdot \boldsymbol{H}^{\mathrm{T}} \cdot \boldsymbol{G}_2) \tag{21}$$

where $\boldsymbol{G}_3^{+}$ is the pseudo-inverse of $\boldsymbol{G}_3$ (which is a scaled version of $\boldsymbol{G}_3^{\mathrm{H}}$ if $\boldsymbol{G}_3$ is chosen to have orthogonal columns).

The Khatri–Rao product in (21) can be inverted up to one scaling ambiguity per column. That means we can find matrices $\boldsymbol{F}_1 \in \mathbb{C}^{M_1 \times r_G}$ and $\boldsymbol{F}_2 \in \mathbb{C}^{N_{\mathrm{P}} \times r_G}$ such that

$$\boldsymbol{F}_1 = \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}_1 \cdot \boldsymbol{\Lambda} \tag{22}$$

$$\boldsymbol{F}_2 = \boldsymbol{X}^{\mathrm{T}} \cdot \boldsymbol{H}^{\mathrm{T}} \cdot \boldsymbol{G}_2 \cdot \boldsymbol{\Lambda}^{-1} \tag{23}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}\{[\lambda_1, \lambda_2, \ldots, \lambda_{r_G}]\}$ and λ_n represent arbitrary complex numbers. Since in the presence of noise (21) is only approximately a Khatri–Rao product, the factors represent an

estimate. The algorithm to obtain these estimates is summarized below.

Algorithm 1: Least-Squares Factorization of a Khatri–Rao Product

- Consider a matrix $\boldsymbol{C} \in \mathbb{C}^{M\cdot N\times P}$ which is an approximation of the Khatri-Rao product between a matrix $\boldsymbol{A} \in \mathbb{C}^{M\times P}$ and a matrix $\boldsymbol{B} \in \mathbb{C}^{N\times P}$, i.e., $\boldsymbol{C} \approx \boldsymbol{A} \diamond \boldsymbol{B}$.
- Set $p = 1$.
  1) Let $\boldsymbol{c}_p$, $\boldsymbol{a}_p$, and $\boldsymbol{b}_p$ be the pth columns of the matrices $\boldsymbol{C}$, $\boldsymbol{A}$, and $\boldsymbol{B}$, respectively. We know that $\boldsymbol{c}_p \approx \boldsymbol{a}_p \otimes \boldsymbol{b}_p$.
  2) Reshape the vector $\boldsymbol{c}_p$ into a matrix $\tilde{\boldsymbol{C}}_p \in \mathbb{C}^{N\times M}$, such that $\mathrm{vec}\{\tilde{\boldsymbol{C}}_p\} = \boldsymbol{c}_p$. It is easy to see that this matrix satisfies $\tilde{\boldsymbol{C}}_p \approx \boldsymbol{b}_p \cdot \boldsymbol{a}_p^{\mathrm{T}}$.
  3) Compute the singular value decomposition of $\tilde{\boldsymbol{C}}_p$ as $\tilde{\boldsymbol{C}}_p = \boldsymbol{U}_p\boldsymbol{\Sigma}_p\boldsymbol{V}_p^{\mathrm{H}}$. Now the best rank-one approximation of $\tilde{\boldsymbol{C}}_p$ is given by truncating the SVD, i.e., $\hat{\boldsymbol{a}}_p = \sqrt{\sigma_1}\cdot\boldsymbol{v}_1^*$ and $\hat{\boldsymbol{b}}_p = \sqrt{\sigma_1}\cdot\boldsymbol{u}_1$, where $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ represent the first column vectors of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively, and σ_1 is the largest singular value.
  4) If $p < P$, set $p = p + 1$ and go to 1).

Note that from the Eckart–Young theorem it follows that this algorithm provides the best approximation of the Khatri-Rao product in the least squares sense. Also note that for every p there is one scaling ambiguity in inverting the outer product since $\boldsymbol{b}_p \cdot \boldsymbol{a}_p^{\mathrm{T}} = (\lambda_p \cdot \boldsymbol{b}_p) \cdot (\boldsymbol{a}_p/\lambda_p)^{\mathrm{T}}, \forall \lambda_p \in \mathbb{C} \setminus \{0\}$. A similar idea was used to solve a channel estimation problem for a one-way relaying scenario in [15].

In order to resolve the unknown parameters λ_p we need to eliminate the unknown channels in (22) and (23). First of all, $\boldsymbol{H}_2$ can easily be eliminated in (23) if we restrict the pilot matrix $\boldsymbol{X}$ to have orthogonal rows. Again, this choice is also desirable from a numerical point of view because then the pilot matrix does not affect the conditioning of the problem. Note that the rows can only be orthogonal if the matrix is square or "flat" which yields the necessary condition $N_{\mathrm{P}} \geq M_1 + M_2$.

Design Rule 2: The number of pilot symbols per training block N_{P} must satisfy $N_{\mathrm{P}} \geq M_1 + M_2$.

Design Rule 3: The pilot symbol matrix $\boldsymbol{X} \in \mathbb{C}^{(M_1+M_2)\times N_{\mathrm{P}}}$ must have orthogonal rows.

From these design rules it also follows that the pilot transmissions of the two users are mutually orthogonal. Therefore,

$$\left(\boldsymbol{X}_1^{\mathrm{T}}\right)^+ \cdot \boldsymbol{X}^{\mathrm{T}} = [\boldsymbol{I}_{M_1}, \boldsymbol{0}_{M_1\times M_2}]$$
$$\text{and } \left(\boldsymbol{X}_2^{\mathrm{T}}\right)^+ \cdot \boldsymbol{X}^{\mathrm{T}} = [\boldsymbol{0}_{M_2\times M_1}, \boldsymbol{I}_{M_2}]. \tag{24}$$

Due to the orthogonality constraint, $(\boldsymbol{X}_1^{\mathrm{T}})^+$ and $(\boldsymbol{X}_2^{\mathrm{T}})^+$ are scaled versions of $\boldsymbol{X}_1^*$ and $\boldsymbol{X}_2^*$, respectively. Using (24) in (23) we can eliminate $\boldsymbol{H}_2$ in the following fashion:

$$\begin{aligned}\tilde{\boldsymbol{F}}_2 \doteq \left(\boldsymbol{X}_1^{\mathrm{T}}\right)^+ \cdot \boldsymbol{F}_2 &= \left(\boldsymbol{X}_1^{\mathrm{T}}\right)^+ \cdot \boldsymbol{X}^{\mathrm{T}} \cdot \boldsymbol{H}^{\mathrm{T}} \cdot \boldsymbol{G}_2 \cdot \boldsymbol{\Lambda}^{-1} \\ &= \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}_2 \cdot \boldsymbol{\Lambda}^{-1} \\ \Rightarrow \tilde{\boldsymbol{F}}_2 \cdot \boldsymbol{\Lambda} &= \boldsymbol{H}_1^{\mathrm{T}} \cdot \boldsymbol{G}_2. \end{aligned} \tag{25}$$

In order to remove the unknown $\boldsymbol{H}_1^{\mathrm{T}}$ we need to solve (25) for $\boldsymbol{H}_1^{\mathrm{T}}$. This solution is only unique if $\boldsymbol{G}_2$ is a square or a flat matrix, i.e., $r_G \geq M_{\mathrm{R}}$. Also, to render this inversion numerically stable, $\boldsymbol{G}_2$ should have orthogonal rows.

Design Rule 4: The rank of the tensor $\boldsymbol{\mathcal{G}}$ must satisfy $r_G \geq M_{\mathrm{R}}$. Also, from design rule 1, the number of training blocks N_{R} must be greater or equal to r_G. Therefore, to reduce the pilot overhead, r_G should be as small as possible. Consequently, we choose $r_G = M_{\mathrm{R}}$. Note that it follows that $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are square matrices.

Design Rule 5: The two-mode factor matrix $\boldsymbol{G}_2$ must have full rank.

Design Recommendation 2: The two-mode factor matrix $\boldsymbol{G}_2$ should be an orthogonal matrix.

Now we can solve (25) for $\boldsymbol{H}_1^{\mathrm{T}}$ and insert this solution into (22). We obtain

$$\begin{aligned}\boldsymbol{H}_1^{\mathrm{T}} &= \tilde{\boldsymbol{F}}_2 \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{G}_2^+ \\ \Rightarrow \boldsymbol{F}_1 &= \tilde{\boldsymbol{F}}_2 \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{G}_2^+ \cdot \boldsymbol{G}_1 \cdot \boldsymbol{\Lambda} &(26)\\ \boldsymbol{F}_1 &= \tilde{\boldsymbol{F}}_2 \cdot \left[\left(\boldsymbol{G}_2^+ \cdot \boldsymbol{G}_1\right) \odot (\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}})\right] &(27)\end{aligned}$$

where in the last step we have used the fact that $\boldsymbol{\Lambda} = \mathrm{diag}\{\boldsymbol{\lambda}\}$ and property (52) proven in the Appendix. In order to solve (27) for the unknown vector $\boldsymbol{\lambda}$, we have to isolate $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$ on one side of the equation. However, to achieve this, we need to move $\tilde{\boldsymbol{F}}_2$ to the other side. Since $\tilde{\boldsymbol{F}}_2$ is of size $M_1 \times r_G$ this step requires $M_1 \geq r_G$. For the smallest possible r_G, which was chosen in design rule 4, this condition reduces to $M_1 \geq M_{\mathrm{R}}$. From the equivalent equation at the other user terminal, we also get the condition $M_2 \geq M_{\mathrm{R}}$. As a consequence, we now consider two cases separately. First of all, we solve the case where both conditions are met, i.e., $\min\{M_1, M_2\} \geq M_{\mathrm{R}}$. Then we consider the case where this condition is not true. Note that TENCE is only expected to outperform the LS-based compound channel estimator in case 1, as pointed out in the beginning of this section. The second case is only shown for completeness to demonstrate that the tensor-based approach can be used for arbitrary antenna configurations.

Case 1: $\min\{M_1, M_2\} \geq M_{\mathrm{R}}$*:* In this case, we can solve (27) directly for $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$ in the following fashion

$$\begin{aligned}\tilde{\boldsymbol{F}}_2^+ \cdot \boldsymbol{F}_1 &= \left(\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1\right) \odot (\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}) \\ \left(\tilde{\boldsymbol{F}}_2^+ \cdot \boldsymbol{F}_1\right) \oslash \left(\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1\right) &= \boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}. \end{aligned} \tag{28}$$

Note that since we assume $r_G = M_{\mathrm{R}}$, the matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are square and hence the pseudo-inverse is replaced by the matrix inverse. Here we apply the inverse Schur product $\oslash$ (i.e.,

element-wise division), which requires that the matrix $\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1$ does not contain any zero entries. This leads to another design rule

Design Rule 6: The factor matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ must be chosen such that the matrix $\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1$ does not contain any entries that are equal to zero or very close to zero.

In the presence of noise, (28) holds only approximately. Therefore, the matrix estimated from (28) does not necessarily have rank one. In order to find the best approximation of $\boldsymbol{\lambda}$ we can proceed in a manner similar to the inversion of the Khatri–Rao product and additionally exploit the symmetry of the matrix. The algorithm to estimate $\boldsymbol{\lambda}$ is summarized in the following steps:

Algorithm 2: Estimation of $\boldsymbol{\lambda}$

- Compute the matrix $\boldsymbol{L} = (\tilde{\boldsymbol{F}}_2^{+} \cdot \boldsymbol{F}_1) \oslash (\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1)$.
- Force the matrix to be symmetric by computing $\tilde{\boldsymbol{L}} = (1/2)(\boldsymbol{L} + \boldsymbol{L}^{\mathrm{T}})$.
- Since $\tilde{\boldsymbol{L}}$ is symmetric, an SVD of this matrix is given by $\tilde{\boldsymbol{L}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^{\mathrm{T}}$. An SVD of this form can for instance be computed via the Takagi factorization [29].
- Then, the least squares estimate for $\boldsymbol{\lambda}$ is given by $\hat{\boldsymbol{\lambda}} = \sqrt{\sigma_1} \cdot \boldsymbol{u}_1$, where $\boldsymbol{u}_1$ represents the first column of $\boldsymbol{U}$ and σ_1 is the largest singular value of $\tilde{\boldsymbol{L}}$.

Note that the estimation of $\boldsymbol{\lambda}$ involves one sign ambiguity since $(-\boldsymbol{\lambda}) \cdot (-\boldsymbol{\lambda})^{\mathrm{T}} = \boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$.

From the estimate of $\boldsymbol{\lambda}$ we finally obtain estimates for the channel matrices with the help of (22) and (23)

$$\hat{\boldsymbol{H}}_1 = \left(\bar{\boldsymbol{F}}_1 \cdot \mathrm{diag}\{\hat{\boldsymbol{\lambda}}\}^{-1} \cdot \boldsymbol{G}_1^{-1}\right)^{\mathrm{T}} \tag{29}$$

$$\hat{\boldsymbol{H}}_2 = \left(\left(\boldsymbol{X}_2^{\mathrm{T}}\right)^{+} \bar{\boldsymbol{F}}_2 \cdot \mathrm{diag}\{\hat{\boldsymbol{\lambda}}\} \cdot \boldsymbol{G}_2^{-1}\right)^{\mathrm{T}}. \tag{30}$$

It is also possible to obtain a second estimate for $\boldsymbol{H}_1$ from $\boldsymbol{F}_2$ by replacing $\boldsymbol{X}_2$ by $\boldsymbol{X}_1$ in (30). However, since the estimate found from (29) is always more accurate, this additional estimate for $\boldsymbol{H}_1$ will not be used in the simulations. Note that (29) involves the inverse of $\boldsymbol{G}_1$. With the same reasoning as before, we therefore propose the corresponding design rule for $\boldsymbol{G}_1$:

Design Rule 7: The one-mode factor matrix $\boldsymbol{G}_1$ must have full rank.

Design Recommendation 3: The one-mode factor matrix $\boldsymbol{G}_1$ should be an orthogonal matrix. Note that from design rule 4 it follows that $\boldsymbol{G}_1$ is a square matrix.

Note that the sign ambiguity in $\boldsymbol{\lambda}$ leads to one sign ambiguity in the channel estimates: instead of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ we may estimate $-\boldsymbol{H}_1$ and $-\boldsymbol{H}_2$. However, since this sign cancels in the transmission (6), this scaling ambiguity is irrelevant. This concludes the channel estimation algorithm for case 1.

Case 2: $1 < \min\{M_1, M_2\} < M_{\mathrm{R}}$*:* Without loss of generality, we consider the case where $M_1 \leq M_2$. Since $\tilde{\boldsymbol{F}}_2$ in (27) is a "flat" matrix, we cannot solve (27) for the unknown matrix $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$ directly. Essentially, there are only $M_1 \cdot M_{\mathrm{R}}$ equations for M_{R}^2 unknowns. However, it is actually not required to estimate all elements in $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$, because this matrix has rank one and hence does not have M_{R}^2 degrees of freedom. It is not difficult to see that already $2M_{\mathrm{R}} - 1$ elements from $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$ are enough to reconstruct the entire matrix via the following naive approach: the M_{R} main diagonal elements of $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$ are equal to λ_m^2 from which we can obtain all λ_m up to one $\pm$ ambiguity per coefficient. These unknown signs can be estimated from the $M_{\mathrm{R}} - 1$ elements on the first off-diagonal of $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$.

The approach we take to solve this case is to reduce the number of variables we estimate from M_{R}^2 to $M_1 \cdot M_{\mathrm{R}}$ via a suitable design of the tensor $\boldsymbol{\mathcal{G}}$ which then facilitates a well-defined inversion. From the $M_1 \cdot M_{\mathrm{R}}$ estimated elements in $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$ we can reconstruct the missing elements using the rank-1 structure (cf. algorithm 3) and then proceed in the same manner as in the previous case.

To simplify the notation, we introduce the following definitions:

$$\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1 = \tilde{\boldsymbol{G}} = \left[\tilde{\boldsymbol{g}}_1, \tilde{\boldsymbol{g}}_2, \ldots, \tilde{\boldsymbol{g}}_{M_{\mathrm{R}}}\right] \tag{31}$$

$$\boldsymbol{F}_1 = \left[\boldsymbol{f}_{1,1}, \boldsymbol{f}_{1,2}, \ldots, \boldsymbol{f}_{1,M_{\mathrm{R}}}\right] \tag{32}$$

i.e., $\tilde{\boldsymbol{g}}_m$ and $\boldsymbol{f}_{1,m}$ represent the mth columns of $\tilde{\boldsymbol{G}}$ and $\boldsymbol{F}_1$, respectively. Note that we have again used the assumption $r_G = M_{\mathrm{R}}$. Using definitions (31) and (32) we rewrite the matrix equation (27) into a system of matrix-vector equations

$$\boldsymbol{f}_{1,m} = \tilde{\boldsymbol{F}}_2 \cdot \mathrm{diag}\{\tilde{\boldsymbol{g}}_m\} \cdot \boldsymbol{\lambda} \cdot \lambda_m,\ m = 1, 2, \ldots, M_{\mathrm{R}}. \tag{33}$$

Here, we have applied Lemma 2 of the Appendix. Note that if we set the kth element of the vector $\tilde{\boldsymbol{g}}_m$ to zero, the kth column of the matrix $\tilde{\boldsymbol{F}}_2 \cdot \mathrm{diag}\{\tilde{\boldsymbol{g}}_m\}$ becomes zero. This is equivalent to removing the kth column of $\tilde{\boldsymbol{F}}_2$ and the kth row of the parameter vector $\boldsymbol{\lambda} \cdot \lambda_m$ in the mth matrix vector equation of (33). Consequently, we can reduce the number of variables in each of the matrix-vector equations from M_{R} to M_1 if we place $M_{\mathrm{R}} - M_1$ zeros in each of the vectors $\tilde{\boldsymbol{g}}_m$. This leads to the crucial design rule for the second case:

Design Rule 8: The one-mode and two-mode factor matrices of the tensor $\boldsymbol{\mathcal{G}}$ must be designed in such a way that each column of the matrix $\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1 \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$ contains at most $\min\{M_1, M_2\}$ nonzero entries.

Note that design rule 8 does not contradict design rule 6 since for the first case we have $\min\{M_1, M_2\} \geq M_{\mathrm{R}}$ and hence all elements are allowed to be nonzero by rule 8 (and are forced to be nonzero by rule 6).

Using this design, we can solve all matrix-vector equations in (33) and hence obtain M_1 entries of each column of $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$. The elements we obtain are exactly the nonzero positions in the matrix $\tilde{\boldsymbol{G}}$. From these elements we can reconstruct an estimate of the full matrix $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$, provided that $M_1 > 1$.[3] This reconstruction algorithm is summarized below:

[3]Following the proposed design of $\boldsymbol{\mathcal{G}}$, for $M_1 = 1$ we only obtain the main diagonal of $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$, i.e., λ_i^2, $\forall i$. Therefore, we cannot determine the sign of the individual λ_i in this case. However, $M_1 > 1$ has been explicitly assumed, and the case $M_1 = 1$ is further discussed in Section VII.

Algorithm 3: Rank-One Matrix Reconstruction

- The input to the algorithm is a matrix $\boldsymbol{L}$ which contains the estimates of $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$ we have and the pattern of nonzero elements in the matrix $\tilde{\boldsymbol{G}}$. The nonzero positions in $\tilde{\boldsymbol{G}}$ are the known elements in the estimate of $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$.
- First of all, we can use the symmetry of $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$ by filling each unknown element $l_{i,j}$ with $l_{j,i}$ if the latter is known.
- If after this step there are unknown elements left. we continue by estimating the ratios $\rho_m \triangleq \lambda_m/\lambda_{m-1}$ for $m = 2, 3, \ldots, M_{\mathrm{R}}$ in the following fashion:
  1) Set $m = 2$.
  2) Obtain the set of column indexes $i \in \mathcal{I}$ for which the elements (m, i) and $(m-1, i)$ are known.
  3) Obtain the set of row indexes $j \in \mathcal{J}$ for which the elements (j, m) and $(j, m-1)$ are known.
  4) Estimate ρ_m as the arithmetic average of the ratios $l_{m,i}/l_{m-1,i}$ and the ratios $l_{j,m}/l_{j,m-1}, \forall i \in \mathcal{I}, j \in \mathcal{J}$.
  5) If $m < M_{\mathrm{R}}$ set $m = m + 1$ and go to 2).
- Now we can apply these ratios to fill the rest of the matrix. For every unknown element (i, j) in the matrix $\boldsymbol{L}$, we check:
  1) If the element $(i, j-1)$ is known, an estimate of $l_{i,j}$ is given by $l_{i,j-1} \cdot \rho_m$.
  2) If the element $(i-1, j)$ is known, an estimate of $l_{i,j}$ is given by $l_{i-1,j} \cdot \rho_m$.
  3) If the element $(i, j+1)$ is known, an estimate of $l_{i,j}$ is given by $l_{i,j+1}/\rho_m$.
  4) If the element $(i+1, j)$ is known, an estimate of $l_{i,j}$ is given by $l_{i+1,j}/\rho_m$.
- Again, if more than one estimate for $l_{i,j}$ is available, an arithmetic average is computed.

At the end of this algorithm we have an estimate of $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^{\mathrm{T}}$. Depending on the pattern of the unknown elements, this estimate may not be exactly symmetric and it may also not be exactly rank one. We therefore proceed in the same manner as in case one to estimate the vector $\boldsymbol{\lambda}$ from this matrix: First the matrix is forced to be symmetric. After that, a best rank-one approximation is computed with the help of a singular value decomposition (cf. Algorithm 2). The estimated vector $\hat{\boldsymbol{\lambda}}$ is then used to compute estimates for the channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ (cf. (29) and (30)).

C. Summary

The TENCE algorithm is summarized in Table I. Concerning the design rules for the matrix $\boldsymbol{X}$ and the tensor $\boldsymbol{\mathcal{G}}$, we have the following.

- The pilot matrix $\boldsymbol{X} \in \mathbb{C}^{M_1+M_2 \times N_{\mathrm{P}}}$: The number of pilots N_{P} must satisfy $N_{\mathrm{P}} \geq M_1 + M_2$ and $\boldsymbol{X}$ must have orthogonal rows (cf. design rules 2 and 3). A reasonable choice is given by constructing a DFT matrix of size $N_{\mathrm{P}} \times N_{\mathrm{P}}$ and then using the first M_1 rows for $\boldsymbol{X}_1$ and the next M_2 rows for $\boldsymbol{X}_2$. To ensure that the transmit power is limited to $P_{\mathrm{T},i}$ for each user terminal $i = 1, 2$, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ can be scaled individually, such that the norm of each column is equal to $P_{\mathrm{T},i}$. Note that $N_{\mathrm{P}} = M_1 + M_2$ is sufficient for the training, higher values can be used to increase the estimation accuracy in the presence of noise. Another possible choice is given by Zadoff–Chu sequences [2] since these fulfill the required orthogonality conditions as well.
- The relay amplification tensor $\boldsymbol{\mathcal{G}}$:
  — The rank r_G must satisfy $r_G \geq M_{\mathrm{R}}$ according to design rule 4. A larger rank leads to higher pilot overhead according to design rule 1. Therefore, we choose $r_G = M_{\mathrm{R}}$.
  — The factor matrices $\boldsymbol{G}_1 \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$, $\boldsymbol{G}_2 \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$, and $\boldsymbol{G}_3 \in \mathbb{C}^{N_{\mathrm{R}} \times M_{\mathrm{R}}}$ must have full rank (M_{R}) according to design rules 1, 5, and 7. Moreover, N_{R} must satisfy $N_{\mathrm{R}} \geq M_{\mathrm{R}}$ according to the design rules 1 and 4. Note that $N_{\mathrm{R}} = M_{\mathrm{R}}$ is sufficient for the training, higher values can be used to increase the estimation accuracy in the presence of noise.
  — The matrix $\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1$ must have $\min\{M_1, M_2\}$ nonzero elements per column according to rules 6 and 8. Note that this implies that this matrix should not have any zero entries if $\min\{M_1, M_2\} \geq M_{\mathrm{R}}$.
  — The factor matrix $\boldsymbol{G}_3 \in \mathbb{C}^{N_{\mathrm{R}} \times M_{\mathrm{R}}}$ should have orthogonal columns and the factor matrices $\boldsymbol{G}_1, \boldsymbol{G}_2 \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$ should be orthogonal according to recommendations 1, 2, and 3.

The total number of pilots is equal to $N_{\mathrm{P}} \cdot N_{\mathrm{R}}$. Following the design rules we conclude that at least $(M_1 + M_2) \cdot M_{\mathrm{R}}$ pilots are needed. Note that the total number of parameters that must be identified is equal to $M_1 \cdot M_{\mathrm{R}}$ in $\boldsymbol{H}_1$ and $M_2 \cdot M_{\mathrm{R}}$ in $\boldsymbol{H}_2$. Therefore, the total number of required pilots is equal to the total number of parameters that are identified. Note that this does not correspond to the minimum possible pilot overhead since the number of observations is indeed larger (by a factor of M_i at terminal i). To conclude this chapter we give an example how a tensor $\boldsymbol{\mathcal{G}}$ can easily be constructed that follows all the design rules.

- Choose $\boldsymbol{G}_2 = \boldsymbol{I}_{M_{\mathrm{R}}}$.
- Set $\boldsymbol{G}_3$ to a $M_{\mathrm{R}} \times M_{\mathrm{R}}$ DFT matrix. If a larger number of training blocks (frames) is desired, use a $N_{\mathrm{R}} \times N_{\mathrm{R}}$ DFT matrix and truncate it to M_{R} columns.
- Then, compute $\boldsymbol{G}_1$ in the following way: If $\min\{M_1, M_2\} \geq M_{\mathrm{R}}$: Set $\boldsymbol{G}_1 = \boldsymbol{D}_{M_{\mathrm{R}}}$, where $\boldsymbol{D}_{M_{\mathrm{R}}}$ is an $M_{\mathrm{R}} \times M_{\mathrm{R}}$ DFT matrix. Otherwise set $\boldsymbol{G}_1 = \boldsymbol{D}_{M_{\mathrm{R}}} \odot \boldsymbol{S}$, where $\boldsymbol{S}$ is a circulant matrix computed from the vector $\boldsymbol{v} = [\boldsymbol{1}^{\mathrm{T}}_{\min\{M_1,M_2\}\times 1}, \boldsymbol{0}^{\mathrm{T}}_{M_{\mathrm{R}}-\min\{M_1,M_2\}\times 1}]^{\mathrm{T}}$. That means that the nth column of $\boldsymbol{S}$ is equal to $\boldsymbol{v}$ shifted by $n-1$ elements in a cyclic manner. To illustrate the structure of $\boldsymbol{S}$, Fig. 2 displays $\boldsymbol{S}$ for $M_{\mathrm{R}} = 5$ and three different values for $\min\{M_1, M_2\}$. We have verified numerically that this design provides a full rank matrix $\boldsymbol{G}_1$ for all combinations of M_{R}, M_1, and M_2 up to $M_{\mathrm{R}} = 50$.

TABLE I
SUMMARY OF THE TENCE ALGORITHM AT UT_1. FOR UT_2 WE REPLACE $\boldsymbol{\mathcal{Y}}_1$ BY $\boldsymbol{\mathcal{Y}}_2$ IN THE FIRST STEP AND $\boldsymbol{X}_1$ BY $\boldsymbol{X}_2$ IN THE THIRD STEP. MOREOVER, IN THE FINAL RESULT (29) and (30) WE EXCHANGE $\hat{\boldsymbol{H}}_1$ AND $\hat{\boldsymbol{H}}_2$ AND REPLACE $\boldsymbol{X}_1$ BY $\boldsymbol{X}_2$

- Compute the matrix $\left(\boldsymbol{G}_3^{+} \cdot [\boldsymbol{\mathcal{Y}}_1]_{(3)}\right)^{\mathrm{T}}$.
- Estimate two least squares Khatri-Rao factor matrices $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ using Algorithm 1.
- Compute $\tilde{\boldsymbol{F}}_2 = \left(\boldsymbol{X}_1^{\mathrm{T}}\right)^{+} \cdot \boldsymbol{F}_2$.
- If $\min\{M_1, M_2\} \geq M_{\mathrm{R}}$: compute $\boldsymbol{L} = \left(\tilde{\boldsymbol{F}}_2^{+} \cdot \boldsymbol{F}_1\right) \oslash \left(\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1\right)$.
- If $1 < \min\{M_1, M_2\} < M_{\mathrm{R}}$:
  - Let $\boldsymbol{f}_{1,m}$ be the m-th column of $\boldsymbol{F}_1$ and $\tilde{\boldsymbol{g}}_m$ the m-th column of $\tilde{\boldsymbol{G}} = \boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1$.
  - Compute $\boldsymbol{l}_m = \left[\tilde{\boldsymbol{F}}_2 \cdot \mathrm{diag}\{\tilde{\boldsymbol{g}}_m\}\right]^{+} \cdot \boldsymbol{f}_{1,m}$ for $m = 1, 2, \ldots, M_{\mathrm{R}}$.
  - Collect the vectors $\boldsymbol{l}_m$ column-wise into the matrix $\boldsymbol{L}$.
  - Use Algorithm 3 to fill the elements in $\boldsymbol{L}$ which have not been estimated.
- Estimate $\boldsymbol{\lambda}$ as the best symmetric rank-one approximation of $\boldsymbol{L}$ using Algorithm 2.
- Compute the final channel estimates using equations (29) and (30).

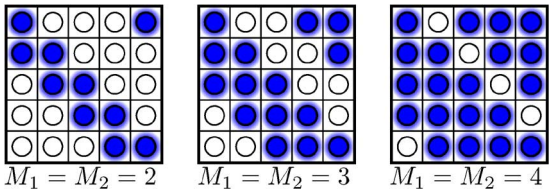

Fig. 2. Structure of the matrix $\boldsymbol{S}$ for $M_{\mathrm{R}} = 5$ and different values for $\min\{M_1, M_2\}$. Empty circles represent zeros, filled circles represent ones.

Note that this design of $\boldsymbol{\mathcal{G}}$ also fulfills all design recommendations if $\min\{M_1, M_2\} \geq M_{\mathrm{R}}$. Otherwise, $\boldsymbol{G}_1$ is not necessarily orthogonal which violates the design recommendation 3.

The amplification matrix $\boldsymbol{G}^{(i)}$ which the relay uses in the ith frame can be computed from the matrices $\boldsymbol{G}_1$, $\boldsymbol{G}_2$, and $\boldsymbol{G}_3$ in the following fashion:

$$\boldsymbol{G}^{(i)} = c_i \cdot \boldsymbol{G}_1 \cdot \mathrm{diag}\left\{[\boldsymbol{G}_3]_{i,:}\right\} \cdot \boldsymbol{G}_2^{\mathrm{T}},\ i = 1, 2, \ldots, N_{\mathrm{R}}$$

where $[\boldsymbol{G}_3]_{i,:}$ represents the ith row of $\boldsymbol{G}_3$ and c_i is chosen such that $\|\boldsymbol{G}^{(i)}\|_{\mathrm{F}} = 1$. Therefore, if $\min\{M_1, M_2\} \geq M_{\mathrm{R}}$, the relay uses shifted DFT matrices during the training phase.

VI. ITERATIVE REFINEMENT FOR TENCE

The TENCE algorithm which we have derived in the previous section is a purely algebraic closed-form solution. Therefore, it is very fast, since it does not require any iterative procedures. However it does not provide the MMSE solution. In this section we show that the MSE can be further reduced by an iterative procedure. Via the number of iterations we can therefore scale the complexity. The mathematical manipulations that are used for this derivation are similar to structured least squares (SLS) [8] even though the underlying problem that is solved in [8] is different.

As in the previous section we derive the solution for UT_1. Due to the strong symmetries in the data model, the solution for UT_2 is very similar.

Let the initial estimates for the channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ be given by $\hat{\boldsymbol{H}}_1$ and $\hat{\boldsymbol{H}}_2$ and define $\hat{\boldsymbol{H}} = [\hat{\boldsymbol{H}}_1, \hat{\boldsymbol{H}}_2]$. Our goal is to improve the estimates $\hat{\boldsymbol{H}}_1$ and $\hat{\boldsymbol{H}}_2$ based on the received training data. Therefore, we need to define a measure for the quality of the channel estimates. To this end, introduce the following definition

$$\tilde{\boldsymbol{\mathcal{Y}}}_1 = \boldsymbol{\mathcal{Y}}_1 \times_2 \left(\boldsymbol{X}^{\mathrm{T}}\right)^{+}. \tag{34}$$

Note that if $\boldsymbol{X}$ is chosen to have orthogonal rows as proposed in the previous section, $\left(\boldsymbol{X}^{\mathrm{T}}\right)^{+}$ is a scaled version of $\boldsymbol{X}^{*}$. Inserting (34) into (17) we find that in the absence of noise $\tilde{\boldsymbol{\mathcal{Y}}}_1$ has the following structure:

$$\tilde{\boldsymbol{\mathcal{Y}}}_1 = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{H}_1^{\mathrm{T}} \times_2 \boldsymbol{H}^{\mathrm{T}}. \tag{35}$$

As we can see, the channel matrix $\boldsymbol{H}_1$ is present in the first and in the second factor. For TENCE, we exploit this symmetry only in the second step, i.e., to estimate $\boldsymbol{\Lambda}$. In the first step of TENCE this is not considered since for the inversion of the Khatri-Rao product, $\boldsymbol{H}_1$ is eliminated in the second factor. This is the reason that the estimate obtained by TENCE can still be improved by exploiting the structure of $\tilde{\boldsymbol{\mathcal{Y}}}_1$.

In the presence of noise, (35) holds only approximately. We can therefore judge the quality of the channel estimate via the norm of the residual tensor $\tilde{\boldsymbol{\mathcal{Y}}}_1 - \boldsymbol{\mathcal{G}} \times_1 \hat{\boldsymbol{H}}_1^{\mathrm{T}} \times_2 \hat{\boldsymbol{H}}^{\mathrm{T}}$. In order to minimize this norm we introduce update terms $\Delta\boldsymbol{H}_1$ and $\Delta\boldsymbol{H}_2$ for the channel estimates $\hat{\boldsymbol{H}}_1$ and $\hat{\boldsymbol{H}}_2$, respectively. Since we already have an initial estimate we additionally apply regularization to enhance the numerical stability. This ensures that the update terms are small compared to the initial solution. The overall cost function we minimize can be written in the following way[4]:

$$J(\Delta\boldsymbol{H}_k) = \|\boldsymbol{\mathcal{R}}_k\|_{\mathrm{H}}^2 + \kappa_1^2 \|\Delta\boldsymbol{H}_{1,k}\|_{\mathrm{F}}^2 + \kappa_2^2 \|\Delta\boldsymbol{H}_{2,k}\|_{\mathrm{F}}^2 \tag{36}$$

where $\boldsymbol{\mathcal{R}}_k$ is the residual tensor after the kth iteration which is given by

$$\boldsymbol{\mathcal{R}}_k = \tilde{\boldsymbol{\mathcal{Y}}}_1 - \boldsymbol{\mathcal{G}} \times_1 \left(\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k}\right)^{\mathrm{T}} \times_2 \left(\hat{\boldsymbol{H}} + \Delta\boldsymbol{H}_k\right)^{\mathrm{T}}. \tag{37}$$

[4]This cost function ignores the fact that the noise is not white due to the forwarded relay noise. Since an initial estimate of the channel matrices is already available via TENCE, the cost function can be extended to take the noise correlation into account. This is achieved by replacing $\|\boldsymbol{\mathcal{R}}_k\|_{\mathrm{H}}$ in the cost function by $\mathrm{vec}\{\boldsymbol{\mathcal{R}}_k\}^{\mathrm{H}} \cdot \hat{\boldsymbol{\Gamma}}^{-1} \cdot \mathrm{vec}\{\boldsymbol{\mathcal{R}}_k\}$, where $\hat{\boldsymbol{\Gamma}}$ is an estimate of the noise covariance matrix. However, in simulations we have found no significant improvement of the modified iterative scheme in terms of the channel estimation accuracy. Since this modification significantly complicates the presentation of the algorithm, it is omitted here for clarity.

Here, $\Delta\boldsymbol{H}_{1,k}$ and $\Delta\boldsymbol{H}_{2,k}$ represent the updates after the kth iteration and $\Delta\bar{\boldsymbol{H}}_k = [\Delta\bar{\boldsymbol{H}}_{1,k},\ \Delta\bar{\boldsymbol{H}}_{2,k}]$. Moreover, the terms κ_1 and κ_2 in (36) are given by $\kappa_1 = \sqrt{M_1/\alpha}$ and $\kappa_2 = \sqrt{M_2/\alpha}$ where $\alpha \in \mathbb{R}$, $\alpha > 0$ controls the amount of regularization used (the larger α, the less regularization).[5]

We can express (36) in a more compact form by applying Lemma 3 shown in the Appendix. Then, we obtain the following alternative representation of (36)

$$J(\Delta\boldsymbol{H}_k) = \left\| \begin{bmatrix} \mathrm{vec}\{\boldsymbol{\mathcal{R}}_k\} \\ \kappa_1 \cdot \mathrm{vec}\{\Delta\boldsymbol{H}_{1,k}\} \\ \kappa_2 \cdot \mathrm{vec}\{\Delta\boldsymbol{H}_{2,k}\} \end{bmatrix} \right\|_2^2. \tag{38}$$

In each iteration, the terms $\Delta\boldsymbol{H}_{1,k}$ and $\Delta\boldsymbol{H}_{2,k}$ are updated according to the following rules:

$$\Delta\boldsymbol{H}_{1,k+1} = \Delta\boldsymbol{H}_{1,k} + \Delta\Delta\boldsymbol{H}_{1,k} \tag{39}$$
$$\Delta\boldsymbol{H}_{2,k+1} = \Delta\boldsymbol{H}_{2,k} + \Delta\Delta\boldsymbol{H}_{2,k} \tag{40}$$

where the initial values are given by

$$\begin{aligned} \Delta\bar{\boldsymbol{H}}_{1,k=0} &= \boldsymbol{0}_{M_\mathrm{R}\times M_1} \\ \Delta\boldsymbol{H}_{2,k=0} &= \boldsymbol{0}_{M_\mathrm{R}\times M_2}. \end{aligned} \tag{41}$$

Our goal is to find $\Delta\Delta\boldsymbol{H}_{1,k}$ and $\Delta\Delta\boldsymbol{H}_{2,k}$ that minimize the cost function in the kth iteration. Since this represents a nonlinear least squares problem, we use local linearization to solve it. Using (39) and (40) in (37) for $\boldsymbol{\mathcal{R}}_{k+1}$ we obtain

$$\begin{aligned} \boldsymbol{\mathcal{R}}_{k+1} &= \tilde{\boldsymbol{\mathcal{Y}}}_1 - \boldsymbol{\mathcal{G}} \times_1 (\hat{\boldsymbol{H}}_1 + \Delta\bar{\boldsymbol{H}}_{1,k+1})^\mathrm{T} \times_2 (\hat{\boldsymbol{H}} + \Delta\bar{\boldsymbol{H}}_{k+1})^\mathrm{T} \\ &= \tilde{\boldsymbol{\mathcal{Y}}}_1 - \boldsymbol{\mathcal{G}} \times_1 (\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k} + \Delta\Delta\boldsymbol{H}_{1,k})^\mathrm{T} \\ &\quad \times_2 (\hat{\boldsymbol{H}} + \Delta\boldsymbol{H}_k + \Delta\Delta\boldsymbol{H}_k)^\mathrm{T} \\ &= \tilde{\boldsymbol{\mathcal{Y}}}_1 - \boldsymbol{\mathcal{G}} \times_1 (\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k})^\mathrm{T} \times_2 (\hat{\boldsymbol{H}} + \Delta\boldsymbol{H}_k)^\mathrm{T} \\ &\quad - \boldsymbol{\mathcal{G}} \times_1 \Delta\Delta\boldsymbol{H}_{1,k}^\mathrm{T} \times_2 (\hat{\boldsymbol{H}} + \Delta\boldsymbol{H}_k)^\mathrm{T} \\ &\quad - \boldsymbol{\mathcal{G}} \times_1 (\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k})^\mathrm{T} \times_2 \Delta\Delta\boldsymbol{H}_k^\mathrm{T} \\ &\quad - \boldsymbol{\mathcal{G}} \times_1 \Delta\Delta\boldsymbol{H}_{1,k}^\mathrm{T} \times_2 \Delta\Delta\boldsymbol{H}_k^\mathrm{T} \\ &\approx \boldsymbol{\mathcal{R}}_k - \boldsymbol{\mathcal{G}} \times_1 \Delta\Delta\boldsymbol{H}_{1,k}^\mathrm{T} \times_2 (\hat{\boldsymbol{H}} + \Delta\boldsymbol{H}_k)^\mathrm{T} \\ &\quad - \boldsymbol{\mathcal{G}} \times_1 (\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k})^\mathrm{T} \times_2 \Delta\Delta\boldsymbol{H}_k^\mathrm{T} \end{aligned} \tag{42}$$

where in the last step we have neglected the higher-order terms in $\Delta\Delta\boldsymbol{H}_{1,k}$ and $\Delta\Delta\boldsymbol{H}_{2,k}$. Therefore, (42) is a linear function in these terms. In order to use this linear function in (38), we apply the vec-operator and use Lemma 5 to reorder the terms. Then,

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

Here, $\boldsymbol{P}^{(n)}_{I,J,K}$ represents the permutation matrix defined in (1). In order to separate the update terms $\Delta\Delta\boldsymbol{H}_{1,k}$ and $\Delta\Delta\boldsymbol{H}_{2,k}$ we apply the following identity:

$$\begin{aligned} \mathrm{vec}\{\Delta\Delta\boldsymbol{H}_k\} &= \mathrm{vec}\left\{[\Delta\Delta\boldsymbol{H}_{1,k}, \Delta\Delta\boldsymbol{H}_{2,k}]\right\} \\ &= \begin{bmatrix} \mathrm{vec}\{\Delta\Delta\boldsymbol{H}_{1,k}\} \\ \mathrm{vec}\{\Delta\Delta\boldsymbol{H}_{2,k}\} \end{bmatrix} \end{aligned} \tag{43}$$

which follows from the definition of the vec-operator. Equation (43) allows to express the update equation for the residual tensor $\boldsymbol{\mathcal{R}}_k$ in the following convenient fashion

$$\begin{aligned} \mathrm{vec}\{\boldsymbol{\mathcal{R}}_{k+1}\} = \mathrm{vec}\{\boldsymbol{\mathcal{R}}_k\} &- \boldsymbol{F}_k^{(1)} \cdot \mathrm{vec}\{\Delta\Delta\boldsymbol{H}_{1,k}\} \\ &- \bar{\boldsymbol{F}}_k^{(2)} \cdot \mathrm{vec}\{\Delta\Delta\bar{\boldsymbol{H}}_{2,k}\} \end{aligned} \tag{44}$$

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

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

Next, we insert (44) as well as (39) and (40) into the cost function (38) for the $(k+1)$th iteration which yields

$$\begin{aligned} J(\Delta\boldsymbol{H}_{k+1}) = \Bigg\| &\begin{bmatrix} \mathrm{vec}\{\boldsymbol{\mathcal{R}}_k\} \\ \kappa_1 \cdot \mathrm{vec}\{\Delta\boldsymbol{H}_{1,k}\} \\ \kappa_2 \cdot \mathrm{vec}\{\Delta\boldsymbol{H}_{2,k}\} \end{bmatrix} \\ &+ \bar{\boldsymbol{F}}_k \cdot \begin{bmatrix} \mathrm{vec}\{\Delta\Delta\boldsymbol{H}_{1,k}\} \\ \mathrm{vec}\{\Delta\Delta\bar{\boldsymbol{H}}_{2,k}\} \end{bmatrix} \Bigg\|_2^2 \\ \bar{\boldsymbol{F}}_k = &\begin{bmatrix} -\boldsymbol{F}_k^{(1)} & -\bar{\boldsymbol{F}}_k^{(2)} \\ \kappa_1 \cdot \boldsymbol{I}_{M_1 \cdot M_\mathrm{R}} & \boldsymbol{0}_{M_1 \cdot M_\mathrm{R} \times M_2 \cdot M_\mathrm{R}} \\ \boldsymbol{0}_{M_2 \cdot M_\mathrm{R} \times M_1 \cdot M_\mathrm{R}} & \kappa_2 \cdot \boldsymbol{I}_{M_2 \cdot M_\mathrm{R}} \end{bmatrix}. \end{aligned} \tag{45}$$

Consequently, the cost function has been rewritten as a linear least squares problem in the update terms $\Delta\Delta\boldsymbol{H}_{1,k}$ and

[5]Our simulations have shown that the performance is not very sensitive to the choice of the regularization parameter α. For a low SNR, a moderate amount of regularization ($\alpha \approx 100$) enhances the numerical stability, but α should not be chosen too small. Moreover, for a high SNR, regularization is not needed and we can choose $\alpha = \infty$. If not stated otherwise, we use $\alpha = 100$ for all the simulations.

TABLE II
SUMMARY OF THE SLS-BASED ITERATIVE REFINEMENT FOR TENCE AT UT_1. FOR UT_2 WE CONSISTENTLY EXCHANGE $\boldsymbol{H}_1$ WITH $\boldsymbol{H}_2$ AND REPLACE $\boldsymbol{\mathcal{Y}}_1$ BY $\boldsymbol{\mathcal{Y}}_2$ IN EQUATIONS (37), (45), AND (46)

1) Initialize $\hat{\boldsymbol{H}}_1$ and $\hat{\boldsymbol{H}}_2$ with the estimates obtained via TENCE.
2) Set $\Delta\boldsymbol{H}_{1,k=0} = \boldsymbol{0}_{M_\mathrm{R}\times M_1}$, $\Delta\boldsymbol{H}_{2,k=0} = \boldsymbol{0}_{M_\mathrm{R}\times M_2}$, and $k = 0$.
3) Compute $\tilde{\boldsymbol{\mathcal{Y}}}_1$ from the received data during the transmission phase according to $\tilde{\boldsymbol{\mathcal{Y}}}_1 = \boldsymbol{\mathcal{Y}}_1 \times_2 \left(\boldsymbol{X}^\mathrm{T}\right)^+$.
4) Calculate the residual tensor $\boldsymbol{\mathcal{R}}_k$ as shown in (37) and the matrix $\boldsymbol{F}_k$ from (45).
5) Solve the least squares problem in $\Delta\Delta\boldsymbol{H}_{1,k}$ and $\Delta\Delta\boldsymbol{H}_{2,k}$ according to (46).
6) Apply the updates to obtain $\Delta\boldsymbol{H}_{1,k+1} = \Delta\boldsymbol{H}_{1,k} + \Delta\Delta\boldsymbol{H}_{1,k}$ and $\Delta\boldsymbol{H}_{2,k+1} = \Delta\boldsymbol{H}_{2,k} + \Delta\Delta\boldsymbol{H}_{2,k}$.
7) Compute $r_k = \|\boldsymbol{\mathcal{R}}_k\|_\mathrm{H}$. If $k > 1$ test whether the last iteration has resulted in a significant innovation by comparing $r_k - r_{k+1}$ with the threshold δ. If the innovation is greater than δ set $k = k + 1$ and go to step 4. If the innovation is negative, ignore the update from the k-th iteration.
8) The improved channel estimates are given by $\hat{\boldsymbol{H}}_1 + \Delta\boldsymbol{H}_{1,k+1}$ and $\hat{\boldsymbol{H}}_2 + \Delta\boldsymbol{H}_{2,k+1}$.

$\Delta\Delta\boldsymbol{H}_{2,k}$. Therefore, the least squares solution of (45) with respect to these terms is given by

$$\begin{bmatrix}\mathrm{vec}\{\Delta\Delta\boldsymbol{H}_{1,k}\}\\ \mathrm{vec}\{\Delta\Delta\boldsymbol{H}_{2,k}\}\end{bmatrix} = -\boldsymbol{F}_k^+ \cdot \begin{bmatrix}\mathrm{vec}\{\boldsymbol{\mathcal{R}}_k\}\\ \kappa_1 \cdot \mathrm{vec}\{\Delta\boldsymbol{H}_{1,k}\}\\ \kappa_2 \cdot \mathrm{vec}\{\Delta\boldsymbol{H}_{2,k}\}\end{bmatrix}. \quad (46)$$

The SLS-based iterative refinement proceeds by computing the updates according to (46) and applying these updates as shown in (39) and (40). Different criteria can be used to check whether the iterative procedure has converged. For example, we can compute the norm of the update terms $\Delta\Delta\boldsymbol{H}_{1,k}$ and $\Delta\Delta\boldsymbol{H}_{2,k}$ and terminate the algorithm when this norm drops below a predefined threshold. Alternatively, define the quantity $r_k = \|\boldsymbol{\mathcal{R}}_k\|_\mathrm{H}$, which is a measure of the fit of the current channel estimates to the data received during the training phase. Then we can terminate the iteration if $r_k - r_{k+1} < \delta$ for a predefined threshold[6] $\delta > 0$. Moreover, if $r_{k+1} > r_k$, the $(k+1)$th iteration is ignored and the kth iteration is used as a final solution. The SLS-based refinement of TENCE is summarized in Table II.

VII. DISCUSSION

A. Computational Complexity

The LS-based channel estimation scheme presented in Section IV requires solving an overdetermined set of $M_i \cdot N_\mathrm{P}$ equations for $M_i \cdot (M_1 + M_2)$ unknowns, where $N_\mathrm{P} \geq M_1 + M_2$.

In TENCE, since most matrices that have to be inverted are chosen orthogonal, the only explicit matrix inversion we require is the pseudo-inverse of $\tilde{\boldsymbol{F}}_2$ which is of size $M_i \times M_\mathrm{R}$ for $i = 1, 2$. Therefore, the complexity is dominated by the least-squares Khatri–Rao factorization of a matrix of size $M_i \cdot N_\mathrm{P} \times M_\mathrm{R}$ for which M_R SVDs of size $N_\mathrm{P} \times M_i$ are required (NB: for each SVD, only the dominant singular vectors are needed). On the other hand, for the SLS-based refinement, an overdetermined set of $M_\mathrm{R} \cdot (M_i + 1) \cdot (M_1 + M_2)$ linear equations needs to be solved for $M_\mathrm{R} \cdot (M_1 + M_2)$ variables in each iteration. For $\alpha = \infty$ the number of equations reduces to $M_\mathrm{R} \cdot M_i \cdot (M_1 + M_2)$.

[6]The threshold parameter δ represents a trade-off between computational complexity and estimation accuracy. We observed 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. As we show in the simulations, for this choice of δ the number of iterations is between one and four, even in critical scenarios.

B. Nonorthogonal Pilots

The way the derivation of TENCE is presented, we rely on the fact that the pilot matrix $\boldsymbol{X}$ has orthogonal rows (cf. design rule 3). This condition can be relaxed to allow a nonorthogonal $\boldsymbol{X}$ by replacing the pseudo-inverse of $\boldsymbol{X}_i^\mathrm{T}$ used at various steps of the derivation by a block of the pseudo-inverse of $\boldsymbol{X}$. However, such a choice for $\boldsymbol{X}$ is detrimental in terms of the channel estimation accuracy, as the simulation results in [6] have also verified.

C. Single-Antenna Case

Since previous channel estimation scheme for two-way relaying with AF relays focus on the single-antenna case [6], we briefly discuss this special case here. For $M_1 = M_2 = M_\mathrm{R} = 1$ the smallest pilot overhead is achieved by choosing $N_\mathrm{R} = 1$ and $N_\mathrm{P} = 2$. The relay amplification tensor $\boldsymbol{\mathcal{G}}$ becomes a scalar g and therefore the factor matrices are trivially $\boldsymbol{G}_1 = \boldsymbol{G}_2 = 1$ and $\boldsymbol{G}_3 = g$. Then, TENCE simplifies into the following algebraic equations for $\hat{h}_1$ and $\hat{h}_2$ estimated at UT_1

$$\hat{h}_1 = \sqrt{\frac{\boldsymbol{x}_1^* \cdot \boldsymbol{y}_1^\mathrm{T}}{\|\boldsymbol{x}_1\|_2 \cdot g}}, \quad \hat{h}_2 = \frac{\boldsymbol{x}_2^* \cdot \boldsymbol{y}_1^\mathrm{T}}{\|\boldsymbol{x}_2\|_2 \cdot g \cdot \hat{h}_1} \quad (47)$$

where $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{C}^{1\times N_\mathrm{P}}$ are the pilot sequences and $\boldsymbol{y}_1 \in \mathbb{C}^{1\times N_\mathrm{P}}$ is the received training data. We compare the channel estimation accuracy of TENCE in this special case with the ML and LMMSE estimators from [6] in the simulations section. Note that the SLS-based refinement does not provide any improvement in the single-antenna case. Also note that we cannot replace the TENCE algorithm in the general MIMO case by a sequential application of the SISO case presented here. The reason is that each estimate is only unique up to one sign ambiguity which would leave the estimates of the channel matrices with one sign ambiguity per element. These ambiguities alter the subspace which renders SVD-based pre-/postprocessing infeasible.

VIII. SIMULATION RESULTS

In this section, simulation results are shown to compare the different channel estimation approaches and demonstrate the corresponding achievable channel estimation accuracies. We first show the achievable channel estimation accuracy of the separate channels $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ with TENCE and its SLS-based iterative refinement. Then, we compare the LS-based compound channel estimator with the tensor-based channel estimation approach in terms of the estimation error of the compound channels.

For all simulations, the channel matrices are generated according to a correlated Rayleigh fading distribution. The spatial correlation follows a Kronecker model, i.e.,

$$\mathrm{E}\left\{\boldsymbol{H}_i \cdot \boldsymbol{H}_i^{\mathrm{H}}\right\} = \boldsymbol{R}_{\mathrm{R}} \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}, \quad i = 1,2$$
$$\mathrm{E}\left\{\boldsymbol{H}_i^{\mathrm{H}} \cdot \boldsymbol{H}_i\right\} = \boldsymbol{R}_i \in \mathbb{C}^{M_i \times M_i}, \quad i = 1,2 \tag{48}$$

where $\boldsymbol{R}_{\mathrm{R}} \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$ and $\boldsymbol{R}_i \in \mathbb{C}^{M_i \times M_i}$ model the spatial correlation matrices at the relay and at user terminal i, respectively. For simplicity, the matrices $\boldsymbol{R}_{\mathrm{R}}$ and $\boldsymbol{R}_i$ are chosen such that their main diagonal elements are equal to one and the magnitude of all off-diagonal elements is equal to ρ_{R} and ρ_i, respectively. The channels are assumed to be constant during the training phase.

A. Performance of TENCE and Its SLS-Based Refinement

In this section we present a selection of simulation results demonstrating the accuracy achievable with TENCE and its SLS-based refinement.

As a measure of the accuracy, we compute the relative squared estimation error (RSE) defined as

$$e(\boldsymbol{H}, \hat{\boldsymbol{H}}) = \min_{p \in \{1,-1\}} \frac{\|\boldsymbol{H} - p \cdot \hat{\boldsymbol{H}}\|_{\mathrm{F}}^2}{\|\boldsymbol{H}\|_{\mathrm{F}}^2} \tag{49}$$

where p accounts for the sign ambiguity in the estimation of the channels. The estimation error curves are labeled as He11, He12, He21, and He22, where the first number indicates the terminal which estimates the channel referenced by the second number. For instance, He12 represents the estimate of $\hat{\boldsymbol{H}}_2$ at UT_1.

If not stated otherwise, the design of the training data follows the rules derived in Section V and we choose $N_{\mathrm{R}} = M_{\mathrm{R}}$ and $N_{\mathrm{P}} = M_1 + M_2$ to minimize the pilot overhead. Moreover, the default values for α and δ are $\alpha = 100$, $\delta = 10^{-3}$. We use a fixed transmit power of $P_{\mathrm{T}} = 1$ for both terminals and the relay and vary the noise power P_{N} at the terminals and at the relay as a function of the $\mathrm{SNR} = 1/P_{\mathrm{N}}$.

The first result shown in Fig. 3 corresponds to an uncorrelated Rayleigh fading scenario where each terminal is equipped with five antennas. In Fig. 3 we show the complementary cumulative distribution function (CCDF) of the RSE (i.e., the probability that the RSE exceeds its abscissa) for a fixed SNR of 20 dB and randomly drawn channel realizations. Dashed lines represent the initial estimate obtained via TENCE and solid lines are used for the SLS-based iterative refinement. We observe significant improvements via the iterative scheme in the terminals' own channels to the relay and mild improvements in the channels between the other terminal and the relay. Moreover, the slope of the CCDF is steeper for the SLS-based iterative refinement which means that their estimates are numerically more stable than the initial TENCE estimates.

A correlated Rayleigh fading scenario is investigated in Fig. 4 where we choose $M_1 = 4$, $M_2 = 5$, $M_{\mathrm{R}} = 3$, $K_1 = K_2 = 0$, $\rho_1 = \rho_2 = 0$, and $\rho_{\mathrm{R}} = 0.9$. Therefore, a strong spatial correlation at the relay is present which impacts both $\hat{\boldsymbol{H}}_1$ and $\hat{\boldsymbol{H}}_2$. We observe significant improvements obtained by the SLS-based iterative refinement for the estimates of each terminal's own channel to the relay since the iterative channel estimate exploits the fact that each terminal's own channel is present in the first as well as the second mode of the training tensor.

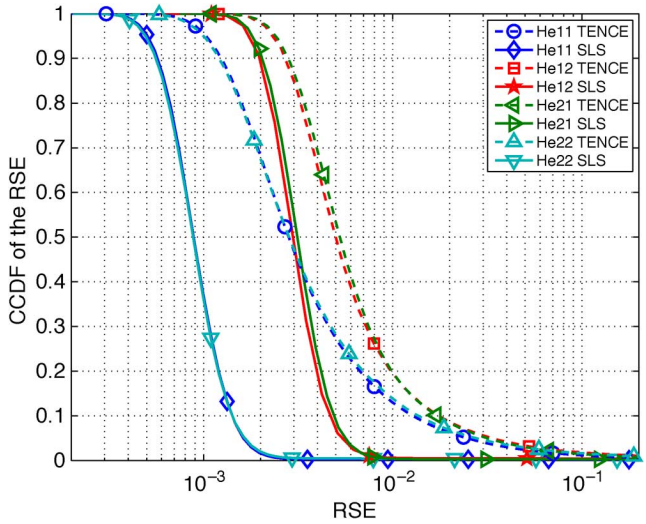

Fig. 3. CCDF of the RSE for TENCE and the SLS-based iterative refinement. Scenario: $M_1 = M_2 = M_{\mathrm{R}} = 5$, $\mathrm{SNR} = 20$ dB, $\rho_{\mathrm{R}} = \rho_1 = \rho_2 = 0$ (uncorrelated Rayleigh fading).

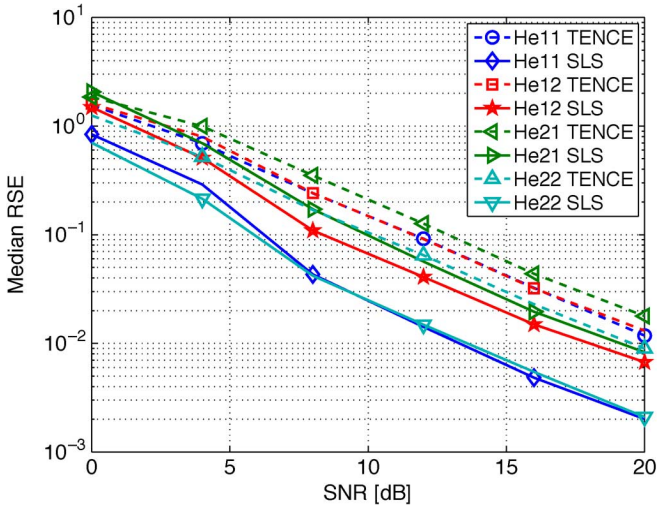

Fig. 4. Median of the RSE versus the SNR for TENCE and the SLS-based iterative refinement. Scenario: $M_1 = 4$, $M_2 = 5$, $M_{\mathrm{R}} = 3$, $\rho_{\mathrm{R}} = 0.9$, $\rho_1 = \rho_2 = 0$ (correlated Rayleigh fading).

The impact of the design parameters α and δ on the performance of the SLS-based iterative refinement is shown in Figs. 5 and 6. Here, we consider a scenario with uncorrelated Rayleigh fading ($K_1 = K_2 = 0$, $\rho_{\mathrm{R}} = \rho_1 = \rho_2 = 0$) for $M_1 = M_2 = 2$ and $M_{\mathrm{R}} = 4$ antennas. In Fig. 5, we depict the mean RSE for different choices of the regularization parameter α and the SNR. Note that the last point $\alpha = \infty$ corresponds to the case where no regularization is used at all. We observe that for a low SNR a mild amount of regularization ($\alpha \approx 100$) helps to lower the mean RSE and that this effect diminishes for higher SNRs. For a very high SNR, we can skip the regularization completely by setting $\alpha = \infty$. For the same scenario, the average number of iterations of the SLS-based refinement is depicted in Fig. 6. We observe a slight increase in the number of iterations for the cases where a mild amount of regularization is used. Moreover, we

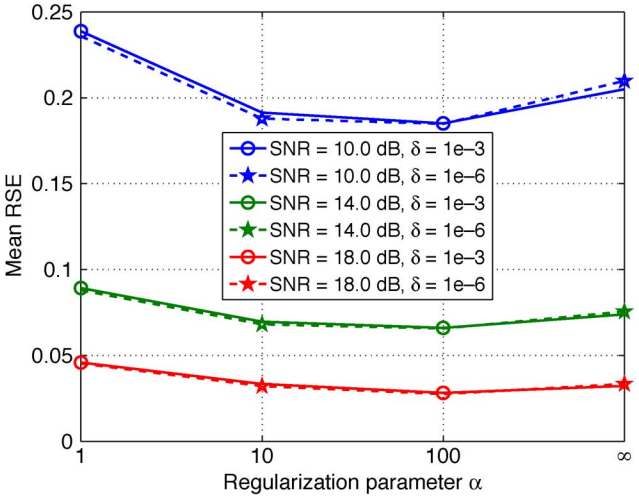

Fig. 5. Mean RSE versus regularization parameter α for different SNRs. Scenario: $M_1 = M_2 = 2$, $M_{\mathrm{R}} = 4$, $\rho_{\mathrm{R}} = \rho_1 = \rho_2 = 0$ (uncorrelated Rayleigh fading).

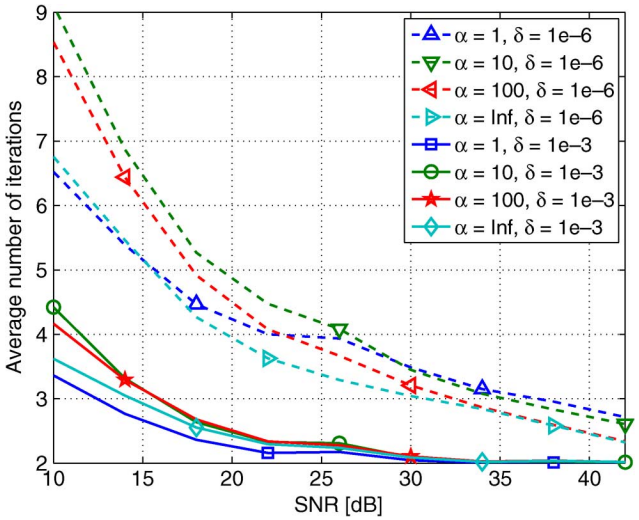

Fig. 6. Number of iterations for the SLS-based refinement versus the SNR for different choices of α and δ. Scenario: $M_1 = M_2 = 2$, $M_{\mathrm{R}} = 4$, $\rho_{\mathrm{R}} = \rho_1 = \rho_2 = 0$ (uncorrelated Rayleigh fading).

compare two different choices of the threshold parameter δ. Obviously, for $\delta = 10^{-6}$, significantly more iterations are required. However, as evident from Fig. 5, these additional iterations do not lead to a visible improvement in the RSE. Consequently, $\delta = 10^{-3}$ is a reasonable choice. For a high SNR, the SLS-based iterative refinement always terminates after two iterations. This means that the second iteration does not improve the norm of the residual tensor anymore. Consequently, one could even limit the number of iterations to one without losing any performance in the high SNR regime.

Finally, Fig. 7 shows the comparison of TENCE with the ML and LMSNR channel estimators proposed in [6]. Since the latter are only applicable to the SISO case, we set $M_1 = M_2 = M_{\mathrm{R}} = 1$. Note that in this case, TENCE simplifies to the equations shown in Section VII-C. Also, we consider a NLOS scenario, i.e., $K_1 = K_2 = 0$. We observe that in terms of the Median RSE, TENCE and ML perform almost equally and outperform the suboptimal LMSNR scheme. It should be noted that

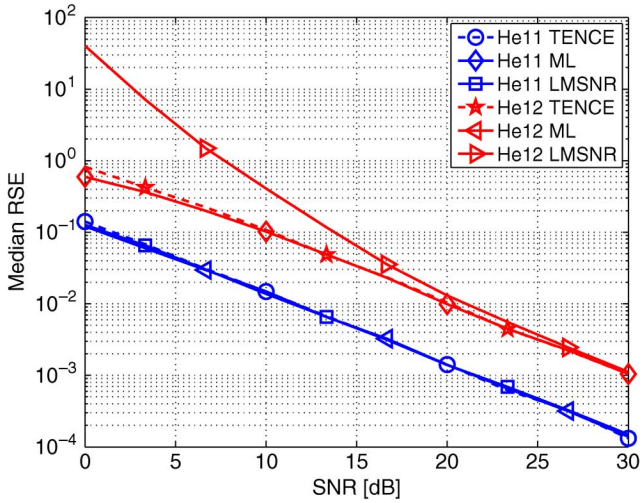

Fig. 7. Median of the RSE versus the SNR comparing TENCE with the ML and the LMSNR estimate. Scenario: $M_1 = M_2 = M_{\mathrm{R}} = 1$, (Rayleigh fading).

the complexity of the closed-form TENCE algorithm is lower than the complexity of ML or LMSNR.

B. Comparison Between Compound and Tensor-Based Estimator

In order to compare the LS-based compound channel estimator proposed in Section IV with the tensor-based approach presented in Sections V and VI we consider the relative estimation error (rCEE) of the compound channels defined via

$$\mathrm{rCEE}_{i,j} = \frac{\left\| \boldsymbol{H}_{i,j}^{(\mathrm{e})} - \hat{\boldsymbol{H}}_{i,j}^{(\mathrm{e})} \right\|_{\mathrm{F}}^2}{\left\| \boldsymbol{H}_{i,j}^{(\mathrm{e})} \right\|_{\mathrm{F}}^2}. \tag{50}$$

Figs. 8 and 9 depict the $\mathrm{rCEE}_{1,1}$ and $\mathrm{rCEE}_{1,2}$ achieved via different approaches. The curves for UT_2 (i.e., $\mathrm{rCEE}_{2,1}$ and $\mathrm{rCEE}_{2,2}$) are omitted since they coincide with the ones for UT_1 due to the symmetry of the problem. The curves labeled "SLS" depict the tensor-based approach using TENCE and the SLS-based iterative refinement with $M_{\mathrm{R}} \cdot (M_1 + M_2)$ pilots. The curves labeled "LS" show the LS-based approach for the estimation of the compound channel. Since LS requires only $M_1 + M_2$ pilots, two sets of curves are shown: One set that corresponds to the minimum number of pilots and another set where the number of pilots has been chosen to $M_{\mathrm{R}} \cdot (M_1 + M_2)$ for a fair comparison to the tensor-based approach. Both simulations assume $M_1 = M_2 = 4$ antennas at the user terminals. The number of antennas at the relay is set to $M_{\mathrm{R}} = 2$ for Fig. 8 and to $M_{\mathrm{R}} = 4$ for Fig. 9. The relay amplification matrix $\boldsymbol{G}$ is chosen as a DFT matrix. We observe that in both cases, the channel $\boldsymbol{H}_{1,1}^{(\mathrm{e})}$, which conveys the self-interference, is estimated more accurate by the tensor-based approach. The estimation accuracies for the channel matrix $\boldsymbol{H}_{1,2}^{(\mathrm{e})}$ achieved by LS and SLS are equal for $M_{\mathrm{R}} = 2$ and SLS is slightly worse for $M_{\mathrm{R}} = 4$ (comparing SLS and LS for the same number of pilots).

IX. Conclusion

In this paper we investigate channel estimation schemes for two-way relaying with AF MIMO relays. We propose two

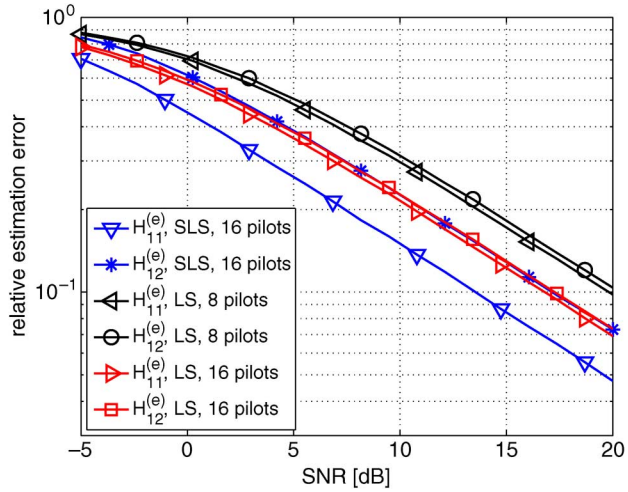

Fig. 8. Median rCEE versus the SNR. Scenario: $M_1 = M_2 = 4$, $M_{\mathrm{R}} = 2$, $\rho_{\mathrm{R}} = \rho_1 = \rho_2 = 0$ (uncorrelated Rayleigh fading).

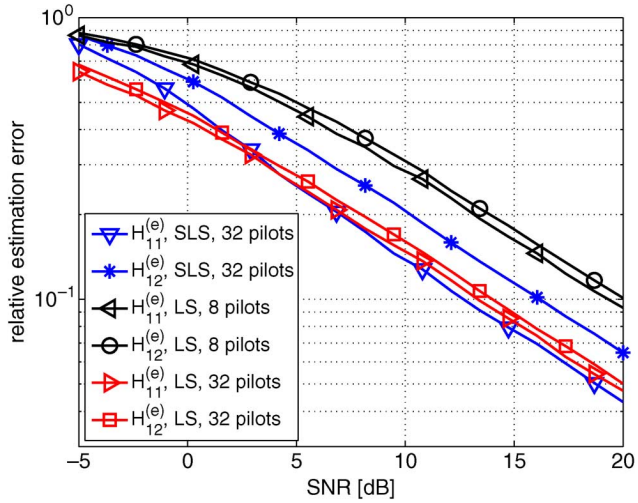

Fig. 9. Median rCEE versus the SNR. Scenario: $M_1 = M_2 = 4$, $M_{\mathrm{R}} = 4$, $\rho_{\mathrm{R}} = \rho_1 = \rho_2 = 0$ (uncorrelated Rayleigh fading).

channel estimation approaches. First, the LS-based estimator for the compound channels is introduced. It represents a simple and robust scheme with a small pilot overhead. However, it fails to provide the terminals with transmit CSI for nonsymmetric relay amplification matrices. Moreover, it ignores the structure of the compound channel matrices which provides room for improvements in the channel estimation accuracy. Then, we introduce a tensor-based approach for estimating the separate channel matrices between the terminals and the relay. We first derive the closed-form TENCE algorithm. Furthermore, we propose design rules for the training symbols and the relay amplification matrices that are required for the implementation of TENCE as well as recommendations that improve its estimation accuracy. In a subsequent step we demonstrate that the estimates obtained via TENCE can be further improved by an iterative algorithm based on structured least squares. We show via simulations that significant improvements are achievable and, depending on the scenario, between one and four iterations are sufficient.

Comparing the two approaches we find that the tensor-based approach yields more accurate estimates of the compound channel matrices that convey the self-interference if the number of antennas at the relay is smaller than the number of antennas at the terminals. Moreover, it always provides the user terminals with transmit CSI, even for nonsymmetric relay amplification matrices.

Appendix
Lemmas and Identities

This appendix summarizes some useful properties of matrices, tensors, and norms that are used in the derivations of this paper.

Lemma 1: The following identities are used without further proof, since they are known from the literature.

For arbitrary matrices $\boldsymbol{X} \in \mathbb{C}^{M\times N}$, $\boldsymbol{Y} \in \mathbb{C}^{P\times N}$, and $\boldsymbol{Z} \in \mathbb{C}^{R\times M}$ [7]

$$\mathrm{vec}\{\boldsymbol{Z}\cdot\boldsymbol{X}\cdot\boldsymbol{Y}^{\mathrm{T}}\} = (\boldsymbol{Y}\otimes\boldsymbol{Z})\cdot\mathrm{vec}\{\boldsymbol{X}\}. \tag{51}$$

For diagonal matrices $\boldsymbol{Y} = \mathrm{diag}\{\boldsymbol{y}\}, \boldsymbol{y} \in \mathbb{C}^M, \boldsymbol{Z} = \mathrm{diag}\{\boldsymbol{z}\}$, $\boldsymbol{z} \in \mathbb{C}^N$, and an arbitrary full matrix $\boldsymbol{X} \in \mathbb{C}^{M\times N}$ it is easy to see that

$$\boldsymbol{Y}\cdot\boldsymbol{X}\cdot\boldsymbol{Z} = \boldsymbol{X}\odot(\boldsymbol{y}\cdot\boldsymbol{z}^{\mathrm{T}}). \tag{52}$$

For a tensor $\boldsymbol{\mathcal{S}} \in \mathbb{C}^{M_1\times M_2\times M_3}$ and matrices $\boldsymbol{A} \in \mathbb{C}^{N_1\times M_1}$, $\boldsymbol{B} \in \mathbb{C}^{N_2\times M_2}$, and $\boldsymbol{C} \in \mathbb{C}^{N_3\times M_3}$, the following identities are shown in [4]:

$$[\boldsymbol{\mathcal{S}}\times_1\boldsymbol{A}\times_2\boldsymbol{B}\times_3\boldsymbol{C}]_{(1)} = \boldsymbol{A}\cdot[\boldsymbol{\mathcal{S}}]_{(1)}\cdot(\boldsymbol{B}\otimes\boldsymbol{C})^{\mathrm{T}} \tag{53}$$

$$[\boldsymbol{\mathcal{S}}\times_1\boldsymbol{A}\times_2\boldsymbol{B}\times_3\boldsymbol{C}]_{(2)} = \boldsymbol{B}\cdot[\boldsymbol{\mathcal{S}}]_{(2)}\cdot(\boldsymbol{C}\otimes\boldsymbol{A})^{\mathrm{T}} \tag{54}$$

$$[\boldsymbol{\mathcal{S}}\times_1\boldsymbol{A}\times_2\boldsymbol{B}\times_3\boldsymbol{C}]_{(3)} = \boldsymbol{C}\cdot[\boldsymbol{\mathcal{S}}]_{(3)}\cdot(\boldsymbol{A}\otimes\boldsymbol{B})^{\mathrm{T}}. \tag{55}$$

An interesting special case of these identities is obtained if the core tensor $\boldsymbol{\mathcal{S}}$ is replaced by an identity tensor, as it appears in the PARAFAC decomposition

$$[\boldsymbol{\mathcal{I}}_{3,M}\times_1\boldsymbol{A}\times_2\boldsymbol{B}\times_3\boldsymbol{C}]_{(1)} = \boldsymbol{A}\cdot(\boldsymbol{B}\diamond\boldsymbol{C})^{\mathrm{T}} \tag{56}$$

$$[\boldsymbol{\mathcal{I}}_{3,M}\times_1\boldsymbol{A}\times_2\boldsymbol{B}\times_3\boldsymbol{C}]_{(2)} = \boldsymbol{B}\cdot(\boldsymbol{C}\diamond\boldsymbol{A})^{\mathrm{T}} \tag{57}$$

$$[\boldsymbol{\mathcal{I}}_{3,M}\times_1\boldsymbol{A}\times_2\boldsymbol{B}\times_3\boldsymbol{C}]_{(3)} = \boldsymbol{C}\cdot(\boldsymbol{A}\diamond\boldsymbol{B})^{\mathrm{T}} \tag{58}$$

where $\boldsymbol{A} \in \mathbb{C}^{N_1\times M}$, $\boldsymbol{B} \in \mathbb{C}^{N_2\times M}$, and $\boldsymbol{C} \in \mathbb{C}^{N_3\times M}$. This demonstrates that any unfolding of the identity tensor can be seen as a selection matrix which reduces a Kronecker product to a Khatri–Rao product.

Lemma 2: For arbitrary matrices $\boldsymbol{A} \in \mathbb{C}^{M\times N}$, $\boldsymbol{B} \in \mathbb{C}^{N\times P}$, and $\boldsymbol{C} \in \mathbb{C}^{N\times P}$ we can define a matrix $\boldsymbol{D} \in \mathbb{C}^{M\times P}$ in the following way

$$\boldsymbol{D} = \boldsymbol{A}\cdot(\boldsymbol{B}\odot\boldsymbol{C}). \tag{59}$$

Then, the pth column of $\boldsymbol{D}$ can be expressed as

$$\boldsymbol{d}_p = \boldsymbol{A}\cdot\mathrm{diag}\{\boldsymbol{b}_p\}\cdot\boldsymbol{c}_p = \boldsymbol{A}\cdot\mathrm{diag}\{\boldsymbol{c}_p\}\cdot\boldsymbol{b}_p \tag{60}$$

where $\boldsymbol{b}_p$ and $\boldsymbol{c}_p$ represent the pth column vectors of $\boldsymbol{B}$ and $\boldsymbol{C}$, respectively and $p = 1, 2, \ldots, P$.

Proof: Obviously, for arbitrary vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^N$, we have that

$$\boldsymbol{x} \odot \boldsymbol{y} = \operatorname{diag}\{\boldsymbol{x}\} \cdot \boldsymbol{y} = \operatorname{diag}\{\boldsymbol{y}\} \cdot \boldsymbol{x}. \tag{61}$$

Moreover, the pth column of (59) is given by

$$\boldsymbol{d}_p = \boldsymbol{A} \cdot (\boldsymbol{b}_p \odot \boldsymbol{c}_p). \tag{62}$$

Applying (61) in (62) for $\boldsymbol{x} = \boldsymbol{b}_p$ and $\boldsymbol{y} = \boldsymbol{c}_p$ proves the Lemma. □

Lemma 3: For a tensor $\boldsymbol{\mathcal{X}}$ of arbitrary size and a matrix $\boldsymbol{Y}$ of arbitrary size the following identities hold:

$$\|\boldsymbol{\mathcal{X}}\|_{\mathrm{H}} = \|\mathrm{vec}\{\boldsymbol{\mathcal{X}}\}\|_2, \quad \|\boldsymbol{Y}\|_{\mathrm{F}} = \|\mathrm{vec}\{\boldsymbol{Y}\}\|_2. \tag{63}$$

Proof: The higher-order (tensor) norm, the Frobenius (matrix) norm, and the vector 2-norm are all defined as the square-root of the sum of the squared magnitude of all elements. Since the vec-operator only rearranges all the elements into a vector and the order of the elements is irrelevant for the sum, the identities are obvious. □

Lemma 4: Every tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{N_1 \times N_2 \times N_3}$ fulfills the following properties:

$$\mathrm{vec}\left\{[\boldsymbol{\mathcal{X}}]_{(3)}\right\} = \mathrm{vec}\left\{[\boldsymbol{\mathcal{X}}]_{(1)}^{\mathrm{T}}\right\} \tag{64}$$

$$\mathrm{vec}\left\{[\boldsymbol{\mathcal{X}}]_{(1)}\right\} = \mathrm{vec}\left\{[\boldsymbol{\mathcal{X}}]_{(2)}^{\mathrm{T}}\right\} \tag{65}$$

$$\mathrm{vec}\left\{[\boldsymbol{\mathcal{X}}]_{(2)}\right\} = \mathrm{vec}\left\{[\boldsymbol{\mathcal{X}}]_{(3)}^{\mathrm{T}}\right\}. \tag{66}$$

Proof: Let the rank of the tensor $\boldsymbol{\mathcal{X}}$ be denoted by r. Then, $\boldsymbol{\mathcal{X}}$ can be expressed in terms of its PARAFAC decomposition in the following way:

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{I}}_{3,r} \times_1 \boldsymbol{F}_1 \times_2 \boldsymbol{F}_2 \times_3 \boldsymbol{F}_3 \tag{67}$$

where the factor matrices $\boldsymbol{F}_n$ are of size $\boldsymbol{F}_n \in \mathbb{C}^{M_n \times r}$ for $n = 1, 2, 3$. To prove (64), we expand $[\boldsymbol{\mathcal{X}}]_{(3)}$ and $[\boldsymbol{\mathcal{X}}]_{(1)}^{\mathrm{T}}$ using (67) and the identities (53) and (55). We obtain

$$\mathrm{vec}\left\{[\boldsymbol{\mathcal{X}}]_{(3)}\right\} = \mathrm{vec}\left\{\boldsymbol{F}_3 \cdot [\boldsymbol{\mathcal{I}}_{3,r}]_{(3)} \cdot (\boldsymbol{F}_1 \otimes \boldsymbol{F}_2)^{\mathrm{T}}\right\} \tag{68}$$

$$\mathrm{vec}\left\{[\boldsymbol{\mathcal{X}}]_{(1)}^{\mathrm{T}}\right\} = \mathrm{vec}\left\{(\boldsymbol{F}_2 \otimes \boldsymbol{F}_3) \cdot [\boldsymbol{\mathcal{I}}_{3,r}]_{(1)}^{\mathrm{T}} \cdot \boldsymbol{F}_1^{\mathrm{T}}\right\}. \tag{69}$$

To simplify these equations further we use the property (51) for $\boldsymbol{Z} = \boldsymbol{F}_1$, $\boldsymbol{X} = [\boldsymbol{\mathcal{I}}_{3,r}]_{(3)}$, and $\boldsymbol{Y} = \boldsymbol{F}_1 \otimes \boldsymbol{F}_2$ yields

$$\mathrm{vec}\left\{[\boldsymbol{\mathcal{X}}]_{(3)}\right\} = (\boldsymbol{F}_1 \otimes \boldsymbol{F}_2 \otimes \boldsymbol{F}_3) \cdot \mathrm{vec}\left\{[\boldsymbol{\mathcal{I}}_{3,r}]_{(3)}\right\}. \tag{70}$$

Similarly, (51) can be applied to (68) for $\boldsymbol{Z} = \boldsymbol{F}_2 \otimes \boldsymbol{F}_3$, $\boldsymbol{X} = [\boldsymbol{\mathcal{I}}_{3,r}]_{(1)}^{\mathrm{T}}$, and $\boldsymbol{Y} = \boldsymbol{F}_1$ from which we get

$$\mathrm{vec}\left\{[\boldsymbol{\mathcal{X}}]_{(1)}^{\mathrm{T}}\right\} = (\boldsymbol{F}_1 \otimes \boldsymbol{F}_2 \otimes \boldsymbol{F}_3) \cdot \mathrm{vec}\left\{[\boldsymbol{\mathcal{I}}_{3,r}]_{(1)}^{\mathrm{T}}\right\}. \tag{71}$$

Finally, from the definition of the identity tensor, it is easy to see that $\mathrm{vec}\{[\boldsymbol{\mathcal{I}}_{3,r}]_{(3)}\} = \mathrm{vec}\{[\boldsymbol{\mathcal{I}}_{3,r}]_{(1)}^{\mathrm{T}}\} = \boldsymbol{e}$, where $\boldsymbol{e}$ is a vector which is equal to one at the positions $(n-1) \cdot (r^2 + r + 1) + 1$ for $n = 1, 2, \ldots, r$ and zero elsewhere. Consequently, (70) and (71) are identical, which proves (64). The proof of (65) and (66) proceeds in analogous fashion. □

Lemma 5: For a tensor $\boldsymbol{\mathcal{S}} \in \mathbb{C}^{N_1 \times N_2 \times N_3}$ and matrices $\boldsymbol{A} \in \mathbb{C}^{N_1 \times M_1}$ and $\boldsymbol{B} \in \mathbb{C}^{N_2 \times M_2}$ the following identities hold:

$$\begin{aligned}\mathrm{vec}\{\boldsymbol{\mathcal{S}} \times_1 \boldsymbol{A}^{\mathrm{T}} \times_2 \boldsymbol{B}^{\mathrm{T}}\} &= \boldsymbol{P}^{(3)}_{M_1,M_2,N_3} \cdot \left(\boldsymbol{I}_{M_1} \otimes [\boldsymbol{\mathcal{S}} \times_2 \boldsymbol{B}^{\mathrm{T}}]_{(1)}^{\mathrm{T}}\right) \cdot \mathrm{vec}\{\boldsymbol{A}\} \\ &= \boldsymbol{P}^{(1)}_{M_1,M_2,N_3} \cdot \left(\boldsymbol{I}_{M_2} \otimes [\boldsymbol{\mathcal{S}} \times_1 \boldsymbol{A}^{\mathrm{T}}]_{(2)}^{\mathrm{T}}\right) \cdot \mathrm{vec}\{\boldsymbol{B}\}\end{aligned} \tag{72}$$

where $\boldsymbol{P}^{(n)}_{I,J,K}$ are the permutation matrices defined in (1).

Proof: From the definition of the permutation matrices we know that

$$\mathrm{vec}\{\boldsymbol{\mathcal{S}} \times_1 \boldsymbol{A}^{\mathrm{T}} \times_2 \boldsymbol{B}^{\mathrm{T}}\} = \boldsymbol{P}^{(3)}_{M_1,M_2,N_3} \cdot \mathrm{vec}\left\{[\boldsymbol{\mathcal{S}} \times_1 \boldsymbol{A}^{\mathrm{T}} \times_2 \boldsymbol{B}^{\mathrm{T}}]_{(3)}\right\}. \tag{73}$$

Applying Lemma 5 this can be reformulated into

$$\mathrm{vec}\{\boldsymbol{\mathcal{S}} \times_1 \boldsymbol{A}^{\mathrm{T}} \times_2 \boldsymbol{B}^{\mathrm{T}}\} = \boldsymbol{P}^{(3)}_{M_1,M_2,N_3} \cdot \mathrm{vec}\left\{[\boldsymbol{\mathcal{S}} \times_1 \boldsymbol{A}^{\mathrm{T}} \times_2 \boldsymbol{B}^{\mathrm{T}}]_{(1)}^{\mathrm{T}}\right\}. \tag{74}$$

Expanding the one-mode product with the help of (53), we obtain

$$\begin{aligned}\mathrm{vec}\{\boldsymbol{\mathcal{S}} \times_1 \boldsymbol{A}^{\mathrm{T}} \times_2 \boldsymbol{B}^{\mathrm{T}}\} &= \boldsymbol{P}^{(3)}_{M_1,M_2,N_3} \cdot \mathrm{vec}\left\{\left(\boldsymbol{A}^{\mathrm{T}} \cdot [\boldsymbol{\mathcal{S}} \times_2 \boldsymbol{B}^{\mathrm{T}}]_{(1)}\right)^{\mathrm{T}}\right\} \\ &= \boldsymbol{P}^{(3)}_{M_1,M_2,N_3} \cdot \mathrm{vec}\left\{[\boldsymbol{\mathcal{S}} \times_2 \boldsymbol{B}^{\mathrm{T}}]_{(1)}^{\mathrm{T}} \cdot \boldsymbol{A}\right\}.\end{aligned} \tag{75}$$

We can now use property (51) for $\boldsymbol{Z} = [\boldsymbol{\mathcal{S}} \times_2 \boldsymbol{B}^{\mathrm{T}}]_{(1)}^{\mathrm{T}}$, $\boldsymbol{X} = \boldsymbol{A}$, and $\boldsymbol{Y} = \boldsymbol{I}_{M_1}$. We get

$$\begin{aligned}\mathrm{vec}\{\boldsymbol{\mathcal{S}} \times_1 \boldsymbol{A}^{\mathrm{T}} \times_2 \boldsymbol{B}^{\mathrm{T}}\} &= \boldsymbol{P}^{(3)}_{M_1,M_2,N_3} \\ &\quad \cdot \left(\boldsymbol{I}_{M_1} \otimes [\boldsymbol{\mathcal{S}} \times_2 \boldsymbol{B}^{\mathrm{T}}]_{(1)}^{\mathrm{T}}\right) \cdot \mathrm{vec}\{\boldsymbol{A}\}\end{aligned} \tag{76}$$

which is the first line of the lemma. The proof of the second line is accomplished in a similar fashion. □

ACKNOWLEDGMENT

The authors acknowledge the fruitful discussions and the helpful comments provided by M. Bengtsson as well as the anonymous reviewers. They have helped to enhance the quality of the manuscript significantly.

REFERENCES

[1] J. Boyer, D. D. Falconer, and H. Yanikomeroglu, "Multihop diversity in wireless relaying channels," *IEEE Trans. Commun.*, vol. 52, pp. 1820–1830, Oct. 2004.

[2] D. Chu, "Polyphase codes with good periodic correlation properties," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 531–532, Jul. 1972.

[3] T. Cui, F. Gao, T. Ho, and A. Nallanathan, "Distributed space-time coding for two-way wireless relay networks," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 658–671, Feb. 2009.

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

[5] F. Gao, R. Zhang, and Y.-C. Liang, "Channel estimation for OFDM modulated two-way relay networks," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4443–4455, Nov. 2009.

[6] F. Gao, R. Zhang, and Y.-C. Liang, "Optimal channel estimation and training design for two-way relay networks," *IEEE Trans. Commun.*, vol. 57, no. 10, pp. 3024–3033, Oct. 2009.
[7] A. Graham, *Kronecker Products and Matrix Calculus: With Applications*. Chinester, U.K.: Ellis Horwook Ltd., 1981.
[8] M. Haardt, "Structured least squares to improve the performance of ESPRIT-type algorithms," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 792–799, Mar. 1997.
[9] M. Haardt, F. Roemer, and G. D. Galdo, "Higher-order SVD based subspace estimation to improve the parameter estimation accuracy in multi-dimensional harmonic retrieval problems," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3198–3213, Jul. 2008.
[10] I. Hammerstrom, M. Kuhn, C. Esli, J. Zhao, A. Wittneben, and G. Bauch, "MIMO two-way relaying with transmit CSI at the relay," presented at the IEEE 8th Workshop Signal Processing Advances in Wireless Commun. (SPAWC 2007), Helsinki, Finland, Jun. 2007.
[11] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," in *Proc. Conf. Applications, Technologies, Architectures, Protocols for Computer Communications (SIGCOMM) 2007*, Kyoto, Japan, Aug. 2007, pp. 497–408.
[12] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Sep. 2009.
[13] P. Larsson, N. Johansson, and K.-E. Sunell, "Coded bidirectional relaying," in *Proc. IEEE 63rd Vehicular Technology Conf. (VTC)*, Melbourne, Australia, May 2006, vol. 2, pp. 851–855.
[14] K. Lee and A. Yener, "Iterative power allocation algorithms for amplify/estimate/compress-and-forward multi-band relay channels," in *Proc. 40th Annu. Conf. Information Sciences Systems (CISS)*, Princeton, NJ, Mar. 2006, pp. 1318–1323.
[15] P. Lioliou and M. Viberg, "Least-squares based channel estimation for MIMO relays," in *Proc. ITG/IEEE Workshop Smart Antennas (WSA)*, Darmstadt, Germany, Feb. 2008, pp. 90–95.
[16] R. U. Nabar, H. Bölcskei, and F. W. Kneubühler, "Fading relay channels: Performance limits and space-time signal design," *IEEE J. Sel. Areas Commun.*, vol. 22, pp. 1099–1109, Aug. 2004.
[17] T. J. Oechtering, I. Bjelakovic, C. Schnurr, and H. Boche, "Broadcast capacity region of two-phase bidirectional relaying," *IEEE Trans. Inf. Theory*, vol. 54, pp. 454–458, Jan. 2008.
[18] T. J. Oechtering and H. Boche, "Bidirectional relaying using interference cancellation," presented at the ITG/IEEE Int. Workshop Smart Antennas (WSA), Vienna, Austria, Feb. 2007.
[19] R. Pabst, B. H. Walke, D. C. Schultz, P. Herhold, H. Yanikomeroglu, S. Mukherjee, H. Viswanathan, M. Lott, W. Zirwas, M. Dohler, H. Aghvami, D. D. Falconer, and G. P. Fettweis, "Relay-based deployment concepts for wireless and mobile broadband radio," *IEEE Commun. Mag.*, vol. 42, pp. 80–89, Sep. 2004.
[20] T.-H. Pham, Y.-C. Liang, A. Nallanathan, and H. K. Garg, "Optimal training sequences for channel estimation in bi-directional relay networks with multiple antennas," *IEEE Trans. Commun.*, 2010, to be published.
[21] B. Rankov and A. Wittneben, "Spectral efficient signaling for half-duplex relay channels," in *Proc. 39th Annu. Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, Oct. 2005, pp. 1066–1071.
[22] B. Rankov and A. Wittneben, "Spectral efficient protocols for half-duplex fading relay channels," *IEEE J. Sel. Areas Commun.*, vol. 25, pp. 379–389, Feb. 2007.
[23] 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 Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Honolulu, HI, Apr. 2007, vol. II, pp. 893–896.
[24] F. Roemer and M. Haardt, "Algebraic norm-maximizing (ANOMAX) transmit strategy for two-way relaying with MIMO amplify and forward relays," *IEEE Signal Process. Lett.*, vol. 16, no. 10, pp. 909–912, Oct. 2009.
[25] F. Roemer and M. Haardt, "Near-far robustness and optimal power allocation for two-way relaying with MIMO amplify and forward relays," presented at the IEEE Int. Workshop Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Aruba, Dutch Antilles, Dec. 2009.
[26] F. Roemer and M. Haardt, "Structured least squares (SLS) based enhancements of tensor-based channel estimation (TENCE) for two-way relaying with multiple antennas," presented at the ITG Workshop on Smart Antennas (WSA), Berlin, Germany, Feb. 2009.
[27] F. Roemer and M. Haardt, "Tensor-based channel estimation (TENCE) for two-way relaying with multiple antennas and spatial reuse," presented at the IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Taipei, Taiwan, Apr. 2009.
[28] C. E. Shannon, "Two-way communication channels," in *Proc. 4th Berkeley Symp. Probability Statistics*, Berkeley, CA, 1961, vol. 1, pp. 611–644.
[29] T. Takagi, "On an algebraic problem related to an analytic theorem of Carathédory and Fejér and on an allied theorem of Landau," *Jpn. J. Math*, vol. 1, pp. 82–93, 1924.
[30] L. B. Thiagarajan, S. Sun, and T. Q. S. Quek, "Carrier frequency offset and channel estimation in space-time non-regenerative two-way relay network," presented at the IEEE 10th Workshop Signal Processing Adv. Wireless Comm. (SPAWC), Perugia, Italy, Jun. 2009.
[31] B. Yi, S. Wang, and S. Y. Kwon, "On MIMO relay with finite-rate feedback and imperfect channel estimation," in *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, Washington, DC, Nov. 2007, pp. 3878–3882.
[32] J. Zhan, M. Kuhn, A. Wittneben, and G. Bauch, "Self-interference aided channel estimation in two-way relaying systems," presented at the IEEE Global Commun. Conf. (IEEE GLOBECOM), New Orleans, LA, Dec. 2008.

Florian Roemer (S'04) studied computer engineering at the Ilmenau University of Technology, Germany, and McMaster University, Montreal, QC, Canada, and received the Diplom-Ingenieur (M.S.) degree in communications engineering from the Ilmenau University of Technology in October 2006.

Since December 2006, he has been a Research Assistant in the Communications Research Laboratory at Ilmenau University of Technology. His research interests include multidimensional signal processing, high-resolution parameter estimation as well as multi-user MIMO precoding and relaying.

Mr. Roemer received the Siemens Communications Academic Award in 2006 for his diploma thesis.

Martin Haardt (S'90–M'98–SM'99) studied electrical engineering at the Ruhr-University Bochum, Germany, and at Purdue University, West Lafayette, IN, and received the Diplom-Ingenieur (M.S.) degree from the Ruhr-University Bochum in 1991 and the Doktor-Ingenieur (Ph.D.) degree from Munich University of Technology, Germany, in 1996.

In 1997, he joint Siemens Mobile Networks, Munich, Germany, where he was responsible for strategic research for third-generation mobile radio systems. From 1998 to 2001, he was the Director for International Projects and University Cooperations in the mobile infrastructure business of Siemens, Munich, Germany, where his work focused on mobile communications beyond the third generation. During his time at Siemens, he also taught in the international Master's of Science in Communications Engineering program at the Munich University of Technology. Since 2001, he has been a Full Professor in the Department of Electrical Engineering and Information Technology and Head of the Communications Research Laboratory at Ilmenau University of Technology, Germany.

Dr. Haardt has received the 2009 Best Paper Award from the IEEE Signal Processing Society, the Vodafone (formerly Mannesmann Mobilfunk) Innovations Award for outstanding research in mobile communications, the ITG Best Paper Award from the Association of Electrical Engineering, Electronics, and Information Technology (VDE), and the Rohde & Schwarz Outstanding Dissertation Award. In fall 2006 and fall 2007, he was a visiting professor at the University of Nice in Sophia-Antipolis, France, and at the University of York, U.K., respectively. His research interests include wireless communications, array signal processing, high-resolution parameter estimation, as well as numerical linear and multi-linear algebra. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2002 to 2006, the IEEE SIGNAL PROCESSING LETTERS since 2006, the *Research Letters in Signal Processing* from 2007 to 2009, and the *Hindawi Journal of Electrical and Computer Engineering* since 2009. He has also served as the Technical Co-Chair of the IEEE International Symposiums on Personal Indoor and Mobile Radio Communications (PIMRC) 2005, Berlin, Germany, and as the Technical Program Chair of the IEEE International Symposium on Wireless Communication Systems 2010, York, U.K.