TENSOR-BASED CHANNEL ESTIMATION (TENCE) FOR TWO-WAY RELAYING WITH MULTIPLE ANTENNAS AND SPATIAL REUSE

Florian Roemer and Martin Haardt

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

***Abstract* — In this paper we study two-way relaying with amplify-and-forward (AF) relays. In two-way relaying, two terminals exchange data with the help of an intermediate relay station. In order to enable mass deployment of these relays, we focus on very simple AF relays that do not have any channel state information. Hence, to separate the data streams in two-way relaying, both user terminals need reliable knowledge of all relevant channel parameters. We therefore propose the novel tensor-based channel estimation algorithm TENCE that provides both terminals with full knowledge of all channel parameters involved in the transmission. The solution is algebraic, i.e., it does not require any iterative procedures. Moreover, TENCE is applicable to arbitrary antenna configurations. We also derive criteria for the design of the pilot symbols and the corresponding relay amplification matrices. Computer simulations demonstrate the achievable channel estimation accuracy.**

Index Terms— Two-way Relaying, MIMO communication, Multidimensional signal processing, Channel estimation

1. INTRODUCTION

A promising approach to enhance coverage, availability, and reliability of future mobile communication systems is the deployment of relay stations to support the transmission between mobile terminals. These relay stations should be much smaller and simpler than base stations so that they can be deployed in larger quantities without causing a prohibitive increase in network cost.

In this paper we focus our attention on amplify-and-forward (AF) relays, where the relays limit their interaction to amplifying the received signal and transmitting the amplified signal in the next time slot. In contrast to decode-and-forward (DF) relays the hardware requirements are much lower which enables the production of cheap mass market devices. To lower the relay complexity even further we assume that the relay station (RS) does not have any channel state information (CSI) and therefore its amplification is independent of the current channel state.

We apply these relays to scenarios where two user terminals (UTs) would like to exchange data with the help of a RS. Their communication is achieved via a two-way relaying scheme [6], i.e., both UTs transmit to the RS in the first time slot and receive the amplified signal from the RS in the second time slot as depicted in Figure 1.

Two-way relaying has been studied in many previous publications. However, usually the relay station was assumed to aid in the separation of the two data streams, either by choosing DF relays [5] or by exploiting CSI at the RS [7, 3], e.g., via multi-user beamforming to achieve a spatial separation. Since we focus on AF relays without CSI, the separation of the streams must be performed at the terminals. Consequently, the UTs need reliable channel knowledge about all relevant channel parameters.

A channel estimation algorithm for AF relaying scenarios was, for example, proposed in [4]. However [4] considered single-antenna relays and a uni-directional link from a transmitter to a receiver. We propose the novel tensor-based channel estimation scheme TENCE for two-way relaying with AF relays which can be applied to arbitrary antenna configurations. TENCE provides both terminals with the necessary channel knowledge to separate the data streams and to decode the transmission from the other user terminal. The solution is algebraic, i.e., it does not require any iterative procedures. We also obtain criteria for the design of the training data and the corresponding relay amplification matrices to optimize the estimation accuracy. Computer simulations demonstrate the achievable channel estimation accuracy of TENCE.

2. NOTATION

To distinguish between scalars, vectors, matrices, and tensors, the following notation is used: 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 ($\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}$).

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 conjugate 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 3-way array with size M_r along mode r. The r-mode vectors of $\boldsymbol{\mathcal{A}}$ are obtained by varying the r-th index and keeping all other indices fixed. Collecting all r-mode vectors into a matrix, we obtain the so called 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 [1]. 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 tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ along the n-th mode we use the operator $[\boldsymbol{\mathcal{A}} \sqcup_n \boldsymbol{\mathcal{B}}]$. 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 $p \times p \times p$ 3-dimensional identity tensor which is one if all three

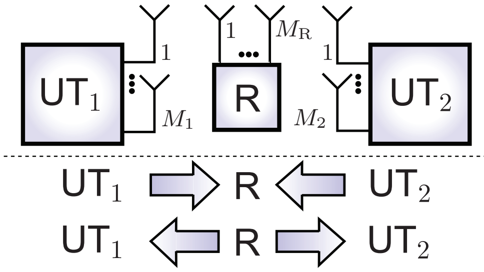

Fig. 1. Two-way relaying system model.

indices are equal and zero otherwise.

The rank of a tensor $\mathcal{A} \in \mathbb{C}^{M_1 \times M_2 \times M_3}$ is defined as the minimum number of rank one tensors we need to sum to construct $\mathcal{A}$, where a rank one tensor can be expressed as the outer product of non-zero vectors. If the rank of $\mathcal{A}$ is equal to r we can express $\mathcal{A}$ in terms of its PARAFAC decomposition [2] as $\mathcal{A} = \mathcal{I}_{3,r} \times_1 \boldsymbol{F}_1 \times_2 \boldsymbol{F}_2 \times_3 \boldsymbol{F}_3$, where $\boldsymbol{F}_1 \in \mathbb{C}^{M_1 \times r}$, $\boldsymbol{F}_2 \in \mathbb{C}^{M_2 \times r}$, and $\boldsymbol{F}_3 \in \mathbb{C}^{M_3 \times r}$. Note that the tensor rank satisfies $\operatorname{rank}\{\mathcal{A}\} \geq \operatorname{rank}\{[\mathcal{A}]_{(r)}\}$ for $r = 1, 2, 3$.

3. SYSTEM DESCRIPTION

3.1. Two-way A/F relaying

The scenario under investigation is depicted in Figure 1. We consider the communication between two user terminals UT_1 and UT_2 with the help of an intermediate relay station R. 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 scarce time and frequency resources only two transmission phases are used in two-way relaying. In the first time slot, both user terminals transmit their data to the relay, where the transmissions interfere. The relay amplifies the received signal and sends it back to both user terminals in the second time slot. Note that we assume time division duplex (TDD).

In contrast to previous studies that have investigated two-way relaying with DF relays [7], we consider very simple AF relays without CSI. The relay amplification matrices are designed beforehand and therefore known to the relay and the user terminals. The task of estimating the individual transmissions from the received signals is shifted completely to the user terminals. To separate the data streams, the user terminals should have very good knowledge of the channel matrices.

3.2. 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 can be expressed as

$$\boldsymbol{r} = \boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \boldsymbol{n}_\mathrm{R} \in \mathbb{C}^{M_\mathrm{R}}, \tag{1}$$

where $\boldsymbol{x}_1 \in \mathbb{C}^{M_1}$ 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 MIMO channels between the relay and UT_1 and UT_2, and the vector $\boldsymbol{n}_\mathrm{R}$ represents the additive noise vector at the relay station.

In the second time slot, the relay amplifies the received vector with an amplification matrix $\boldsymbol{G} \in \mathbb{C}^{M_\mathrm{R} \times M_\mathrm{R}}$ and transmits the signal to both terminals. Consequently in the second slot the received vectors at the two terminals can be expressed as

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

where we have assumed that reciprocity holds in our TDD system and that the channels have not changed between the two transmission phases. The vectors $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ in (2) represent the thermal noise at the receivers. Note that (2) can be rewritten in the following form

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{H}_1^\mathrm{T} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_1^\mathrm{T} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \tilde{\boldsymbol{n}}_1 \\ \boldsymbol{y}_2 &= \boldsymbol{H}_2^\mathrm{T} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{H}_2^\mathrm{T} \cdot \boldsymbol{G} \cdot \boldsymbol{H}_2 \cdot \boldsymbol{x}_2 + \tilde{\boldsymbol{n}}_2. \end{aligned} \tag{3}$$

If the user terminals possess knowledge of the channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ they can cancel the interference they receive from their own transmissions and then decode the transmissions of the other user terminal. Therefore we now focus on the acquisition of channel state information at the terminals.

3.3. Training

In order to acquire channel knowledge at the user terminals we require a training phase which consists of M_R frames. In each frame, a fixed relay amplification matrix $\boldsymbol{G}^{(i)} \in \mathbb{C}^{M_\mathrm{R} \times M_\mathrm{R}}, i = 1, 2, \ldots, M_\mathrm{R}$ is used to transmit a known sequence of N_P pilot symbols $\boldsymbol{x}_{1,j} \in \mathbb{C}^{M_1}$ and $\boldsymbol{x}_{2,j} \in \mathbb{C}^{M_2}$ for $j = 1, 2, \ldots, N_\mathrm{P}$ from UT_1 and UT_2, respectively. The number of pilot symbols N_P that are transmitted for each $\boldsymbol{G}^{(i)}$ must satisfy $N_\mathrm{P} \geq (M_1 + M_2)$. Therefore, the total number of time slots for the training is given by $M_\mathrm{R} \cdot N_\mathrm{P} \geq M_\mathrm{R} \cdot (M_1 + M_2)$. Note that the number of parameters we identify is equal to $M_\mathrm{R} \cdot M_1 + M_\mathrm{R} \cdot M_2$ and thus the number of required time slots is equal to the number of parameters.

The received signal from the j-th pilot symbol within the i-th training block is given by

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

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

$$\boldsymbol{H} \doteq [\boldsymbol{H}_1,\ \boldsymbol{H}_2] \in \mathbb{C}^{M_\mathrm{R} \times (M_1 + M_2)} \tag{4}$$

$$\boldsymbol{X} \doteq \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{1,1} & \ldots & \boldsymbol{x}_{1,N_\mathrm{P}} \\ \boldsymbol{x}_{2,1} & \ldots & \boldsymbol{x}_{2,N_\mathrm{P}} \end{bmatrix} \in \mathbb{C}^{(M_1+M_2) \times N_\mathrm{P}} \tag{5}$$

$$\mathcal{G} \doteq \left[\boldsymbol{G}^{(1)} \sqcup_3 \boldsymbol{G}^{(2)} \ldots \sqcup_3 \boldsymbol{G}^{(M_\mathrm{R})}\right] \in \mathbb{C}^{M_\mathrm{R} \times M_\mathrm{R} \times M_\mathrm{R}}. \tag{6}$$

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

$$\begin{aligned} \mathcal{Y}_1 &= \mathcal{G} \times_1 \boldsymbol{H}_1^\mathrm{T} \times_2 (\boldsymbol{H} \cdot \boldsymbol{X})^\mathrm{T} + \mathcal{N}_1 \in \mathbb{C}^{M_1 \times N_\mathrm{P} \times M_\mathrm{R}} \\ \mathcal{Y}_2 &= \mathcal{G} \times_1 \boldsymbol{H}_2^\mathrm{T} \times_2 (\boldsymbol{H} \cdot \boldsymbol{X})^\mathrm{T} + \mathcal{N}_2 \in \mathbb{C}^{M_2 \times N_\mathrm{P} \times M_\mathrm{R}}, \end{aligned} \tag{7}$$

where the tensors $\mathcal{Y}_1$ and $\mathcal{Y}_2$ contain the vectors $\boldsymbol{y}_{1,i,j}$ and $\boldsymbol{y}_{2,i,j}$ in such a way that the second index in the tensor represents $j = 1, 2, \ldots, N_\mathrm{P}$ and the third index represents $i = 1, 2, \ldots, M_\mathrm{R}$. Similarly, the tensors $\mathcal{N}_1$ and $\mathcal{N}_2$ represent the collection of the noise vectors $\tilde{\boldsymbol{n}}_{1,i,j}$ and $\tilde{\boldsymbol{n}}_{2,i,j}$, respectively. Note that the structure of (7) is similar to a Tucker-2 decomposition [2] with a known core tensor $\mathcal{G}$ and some form of symmetry in the factors ($\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ appear in the first and in the second factor).

4. CHANNEL ESTIMATION

In this section we derive an algebraic solution for the channel estimation problem, i.e., an algorithm that computes estimates of $\boldsymbol{H}_1^\mathrm{T}$ and $\boldsymbol{H}_2^\mathrm{T}$ from the received training data. For notational convenience we ignore the contribution of the noise and write equalities. In the presence of noise, the following identities 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 $\mathcal{G}$. Since $\mathcal{G}$ can be designed, we choose a tensor with rank M_R. Therefore $\mathcal{G}$ can be expressed in terms of its PARAFAC decomposition

$$\mathcal{G} = \mathcal{I}_{3,M_\mathrm{R}} \times_1 \boldsymbol{G}_1 \times_2 \boldsymbol{G}_2 \times_3 \boldsymbol{G}_3, \tag{8}$$

where $\mathcal{I}_{3,M_\mathrm{R}}$ is the identity tensor of size $M_\mathrm{R} \times M_\mathrm{R} \times M_\mathrm{R}$ and the matrices $\boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{G}_3 \in \mathbb{C}^{M_\mathrm{R} \times M_\mathrm{R}}$ represent the factor matrices of the decomposition. Instead of designing the tensor $\boldsymbol{\mathcal{G}}$ directly, it is much easier to deduce design rules for the matrices $\boldsymbol{G}_1, \boldsymbol{G}_2$, and $\boldsymbol{G}_3$ from the derivation of the channel estimation algorithm. Inserting (8) into (7) yields

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

$$[\boldsymbol{\mathcal{Y}}_1]_{(3)} = \boldsymbol{G}_3 \cdot \left[\left(\boldsymbol{H}_1^\mathrm{T} \cdot \boldsymbol{G}_1\right) \diamond \left(\boldsymbol{X}^\mathrm{T} \cdot \boldsymbol{H}^\mathrm{T} \cdot \boldsymbol{G}_2\right)\right]^\mathrm{T}, \tag{9}$$

where we have used elementary properties of n-mode products. In order to isolate the Khatri-Rao product, the multiplication with $\boldsymbol{G}_3$ must be inverted. Since we can design $\boldsymbol{G}_3$, we choose this matrix such that it is orthogolal, 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. We now isolate the Khatri-Rao product in (9) in the following way

$$\left(\boldsymbol{G}_3^{-1} \cdot [\boldsymbol{\mathcal{Y}}_1]_{(3)}\right)^\mathrm{T} = \left(\boldsymbol{H}_1^\mathrm{T} \cdot \boldsymbol{G}_1\right) \diamond \left(\boldsymbol{X}^\mathrm{T} \cdot \boldsymbol{H}^\mathrm{T} \cdot \boldsymbol{G}_2\right), \tag{10}$$

where $\boldsymbol{G}_3^{-1}$ is just a scaled version of $\boldsymbol{G}_3^\mathrm{H}$, since $\boldsymbol{G}_3$ is orthogonal.

The Khatri-Rao product in (10) can be inverted up to one scaling ambiguity per column. That means we can find matrices $\boldsymbol{F}_1 \in \mathbb{C}^{M_1 \times M_\mathrm{R}}$ and $\boldsymbol{F}_2 \in \mathbb{C}^{N_\mathrm{P} \times M_\mathrm{R}}$ such that

$$\boldsymbol{F}_1 = \boldsymbol{H}_1^\mathrm{T} \cdot \boldsymbol{G}_1 \cdot \boldsymbol{\Lambda} \tag{11}$$

$$\boldsymbol{F}_2 = \boldsymbol{X}^\mathrm{T} \cdot \boldsymbol{H}^\mathrm{T} \cdot \boldsymbol{G}_2 \cdot \boldsymbol{\Lambda}^{-1}, \tag{12}$$

where $\boldsymbol{\Lambda} = \operatorname{diag}\{\boldsymbol{\lambda}\}$, $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_{M_\mathrm{R}}]$, and λ_n represent arbitrary complex numbers. Since in the presence of noise (10) is only approximately a Khatri-Rao product, an approximate factorization has to be computed according to the following steps: Let $\boldsymbol{\Gamma}$ be the left-hand side of (10), such that $\boldsymbol{\Gamma} \approx \boldsymbol{F}_1 \diamond \boldsymbol{F}_2$. This implies $\boldsymbol{\gamma}_m \approx \boldsymbol{f}_{1,m} \otimes \boldsymbol{f}_{2,m}$ for the m-th column vectors of $\boldsymbol{\Gamma}$, $\boldsymbol{F}_1$, and $\boldsymbol{F}_2$, respectively. We can reshape $\boldsymbol{\gamma}_m$ into a matrix $\tilde{\boldsymbol{\Gamma}}_m \in \mathbb{C}^{N_\mathrm{P} \times M_1}$ such that $\tilde{\boldsymbol{\Gamma}}_m \approx \boldsymbol{f}_{2,m} \cdot \boldsymbol{f}_{1,m}^\mathrm{T}$. Next, we compute the singular value decomposition of $\tilde{\boldsymbol{\Gamma}}_m$ as $\tilde{\boldsymbol{\Gamma}}_m = \boldsymbol{U}_m \boldsymbol{\Sigma}_m \boldsymbol{V}_m^\mathrm{H}$. Now the best rank-one approximation of $\tilde{\boldsymbol{\Gamma}}_m$ in the Frobenius norm is obtained by choosing $\boldsymbol{f}_{1,m}$ and $\boldsymbol{f}_{2,m}$ according to $\hat{\boldsymbol{f}}_{1,m} = \sqrt{\sigma_1} \cdot \boldsymbol{v}_{1,m}^*$ and $\hat{\boldsymbol{f}}_{2,m} = \sqrt{\sigma_1} \cdot \boldsymbol{u}_{1,m}$, where $\boldsymbol{u}_{1,m}$ and $\boldsymbol{v}_{1,m}$ represent the first column of $\boldsymbol{U}_m$ and $\boldsymbol{V}_m$, respectively, and σ_1 is the largest singular value. We repeat this process for all $m = 1, 2, \ldots, M_\mathrm{R}$. Note that for every m there is one scaling ambiguity in inverting the outer product since $\boldsymbol{f}_{2,m} \cdot \boldsymbol{f}_{1,m}^\mathrm{T} = (\lambda_m \cdot \boldsymbol{f}_{2,m}) \cdot (\boldsymbol{f}_{1,m}/\lambda_m)^\mathrm{T}$, $\forall \lambda_m \in \mathbb{C}$.

In order to resolve the unknown parameters λ_m we need to eliminate the unknown channels in (11) and (12). First of all, $\boldsymbol{H}_2$ can easily be eliminated in (12) by designing the pilot matrix $\boldsymbol{X}$ in such a way that it has orthogonal rows. In our simulations, we have chosen $\boldsymbol{X}$ as the first $M_1 + M_2$ rows of a DFT matrix of size $N_\mathrm{P} \times N_\mathrm{P}$, scaled in such a way that the pilot transmit power constraint is satisfied. Note that from the orthogonality of the DFT matrix it also follows that the pilot transmissions of the two users are mutually orthogonal. Therefore

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

$$\left(\boldsymbol{X}_2^\mathrm{T}\right)^+ \cdot \boldsymbol{X}^\mathrm{T} = \left[\boldsymbol{0}_{M_2 \times M_1},\ \boldsymbol{I}_{M_2}\right]. \tag{13}$$

Also note that due to the orthogonality constraint of $\boldsymbol{X}$, $\left(\boldsymbol{X}_1^\mathrm{T}\right)^+$ and $\left(\boldsymbol{X}_2^\mathrm{T}\right)^+$ are scaled versions of $\boldsymbol{X}_1^*$ and $\boldsymbol{X}_2^*$, respectively. Using equation (13) in (12) we can eliminate $\boldsymbol{H}_2$ in the following fashion

$$\tilde{\boldsymbol{F}}_2 \doteq \left(\boldsymbol{X}_1^\mathrm{T}\right)^+ \cdot \boldsymbol{F}_2 = \boldsymbol{H}_1^\mathrm{T} \cdot \boldsymbol{G}_2 \cdot \boldsymbol{\Lambda}^{-1}$$

$$\Rightarrow \tilde{\boldsymbol{F}}_2 \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{G}_2^{-1} = \boldsymbol{H}_1^\mathrm{T}. \tag{14}$$

Note that $\boldsymbol{G}_2$ has to be inverted. Therefore, $\boldsymbol{G}_2$ is also chosen to be orthogonal so that $\boldsymbol{G}_2^{-1}$ is a scaled version of $\boldsymbol{G}_2^\mathrm{H}$. Inserting (14) into (11) yields

$$\boldsymbol{F}_1 = \tilde{\boldsymbol{F}}_2 \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1 \cdot \boldsymbol{\Lambda} \tag{15}$$

$$\boldsymbol{F}_1 = \tilde{\boldsymbol{F}}_2 \cdot \left[\left(\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1\right) \odot \left(\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^\mathrm{T}\right)\right], \tag{16}$$

where in the last step we have used the fact that $\boldsymbol{\Lambda}$ is diagonal. In order to solve (16) for the unknown vector $\boldsymbol{\lambda}$ we would like to isolate $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^\mathrm{T}$ on one side of the equation. However, in order to do so, we need to move $\tilde{\boldsymbol{F}}_2$ to the other side. Since $\tilde{\boldsymbol{F}}_2$ is of size $M_1 \times M_\mathrm{R}$ this step requires $M_1 \geq M_\mathrm{R}$. 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 satisfied.

Case 1: $\min\{M_1, M_2\} \geq M_\mathrm{R}$.

In this case, we can solve (16) for $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^\mathrm{T}$ in the following fashion

$$\tilde{\boldsymbol{F}}_2^+ \cdot \boldsymbol{F}_1 = \left(\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1\right) \odot \left(\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^\mathrm{T}\right)$$

$$\left(\tilde{\boldsymbol{F}}_2^+ \cdot \boldsymbol{F}_1\right) \oslash \left(\boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1\right) = \boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^\mathrm{T}. \tag{17}$$

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 represents another rule for the design of the factors.

In the presence of noise, (17) holds only approximately. Therefore, the matrix estimated from the training data does not necessarily have rank one. The best approximation for $\boldsymbol{\lambda}$ is obtained in the following fashion: Let $\boldsymbol{L}$ be the left-hand side of (17), such that $\boldsymbol{L} \approx \boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^\mathrm{T}$. Then we can symmetrize $\boldsymbol{L}$ by defining $\tilde{\boldsymbol{L}} = \left(\boldsymbol{L} + \boldsymbol{L}^\mathrm{T}\right)/2$. An SVD of $\tilde{\boldsymbol{L}}$ is then given by $\tilde{\boldsymbol{L}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\mathrm{T}$. Consequently, the least squares estimate for $\boldsymbol{\lambda}$ is obtained from $\hat{\boldsymbol{\lambda}} = \sqrt{\sigma_1} \cdot \boldsymbol{u}_1$, where $\boldsymbol{u}_1$ is the first column of $\boldsymbol{U}$ and σ_1 is the largest singular value.

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 estimated $\hat{\boldsymbol{\lambda}}$ we finally obtain estimates for the channel matrices up to a single sign ambiguity (which is irrelevant since it is eliminated in (2)) in the following fashion

$$\hat{\boldsymbol{H}}_1 = \left(\boldsymbol{F}_1 \cdot \operatorname{diag}\left\{\hat{\boldsymbol{\lambda}}\right\}^{-1} \cdot \boldsymbol{G}_1^{-1}\right)^\mathrm{T} \tag{18}$$

$$\hat{\boldsymbol{H}}_2 = \left(\left(\boldsymbol{X}_2^\mathrm{T}\right)^+ \boldsymbol{F}_2 \cdot \operatorname{diag}\left\{\hat{\boldsymbol{\lambda}}\right\} \cdot \boldsymbol{G}_2^{-1}\right)^\mathrm{T}. \tag{19}$$

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 (19). However, the estimate found from (18) is always more accurate. Note that (18) involves the inverse of $\boldsymbol{G}_1$. With the same reasoning as before we therefore choose $\boldsymbol{G}_1$ to be a orthogonal matrix.

Case 2: $\min\{M_1, M_2\} < M_\mathrm{R}$

Without loss of generality, we consider the case where $M_1 \leq M_2$. Since $\tilde{\boldsymbol{F}}_2$ in (16) is a "flat" matrix, we cannot solve (16) 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.

Therefore, the approach we take to solve this case is to reduce the number of variables we estimate by placing zeros in the matrix $\tilde{\boldsymbol{G}} = \boldsymbol{G}_2^{-1} \cdot \boldsymbol{G}_1$ and then only estimating the variables at the non-zero positions of $\tilde{\boldsymbol{G}}$. This results in an incomplete estimate of the rank-one matrix $\boldsymbol{\lambda}\boldsymbol{\lambda}^\mathrm{T}$, which is then completed by exploiting the rank-one structure this matrix should possess.

In order to facilitate a well-defined inversion we can design the matrix $\tilde{\boldsymbol{G}}$ such that each of its column vectors $\tilde{\boldsymbol{g}}_m$ contains at most

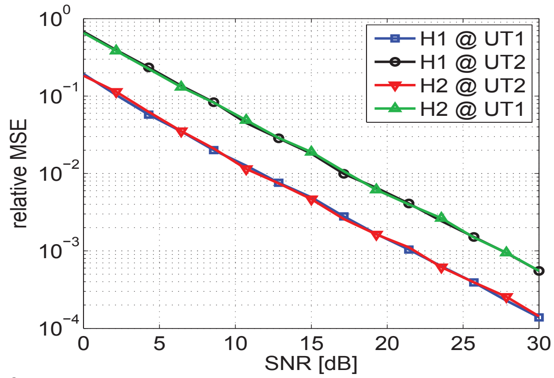

Fig. 2. Median of the relative channel estimation error vs. SNR for $M_1 = 4$, $M_2 = 4$, and $M_R = 2$.

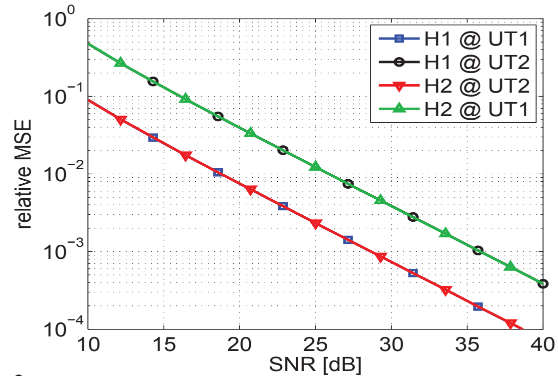

Fig. 3. Median of the relative channel estimation error vs. SNR for $M_1 = 3$, $M_2 = 3$, and $M_R = 6$.

$\min\{M_1, M_2\}$ non-zero entries for $m = 1, 2, \ldots, M_R$. To estimate $\boldsymbol{\lambda}$ we first compute $\mathring{\boldsymbol{L}} = \left[\mathring{\boldsymbol{l}}_1, \ldots, \mathring{\boldsymbol{l}}_{M_R}\right]$, where

$$\mathring{\boldsymbol{l}}_m = \left[\tilde{\boldsymbol{F}}_2 \cdot \operatorname{diag}\{\tilde{\boldsymbol{g}}_m\}\right]^+ \cdot \boldsymbol{f}_{1,m}, \tag{20}$$

and $\boldsymbol{f}_{1,m}$ represents the m-th column of $\boldsymbol{F}_1$ for $m = 1, 2, \ldots, M_R$. Note that $\mathring{\boldsymbol{L}}$ contains estimates of the matrix $\boldsymbol{\lambda}\boldsymbol{\lambda}^T$ at the nonzero positions of $\tilde{\boldsymbol{G}}$ and zeros elsewhere. Let us call the elements of $\mathring{\boldsymbol{L}}$ at the nonzero positions in $\tilde{\boldsymbol{G}}$ "known elements" and the zero positions "unknown". To fill the unknown elements in $\mathring{\boldsymbol{L}}$ we proceed in the following manner: let $\mathring{l}_{i,j}$ be the (i,j) element of $\mathring{\boldsymbol{L}}$. Then we can set $\mathring{l}_{j,i} = \mathring{l}_{i,j}$ if (i,j) is known and (j,i) is unknown, exploiting the symmetry of the rank-one matrix. In a second step we can estimate the ratios $\rho_m = \lambda_m/\lambda_{m-1}$ for $m = 2, 3, \ldots, M_R$ as the arithmetic average of the ratios $\mathring{l}_{m,i}/\mathring{l}_{m-1,i}$ for all i for which the elements (m,i) and $(m-1,i)$ are known and the ratios $\mathring{l}_{j,m}/\mathring{l}_{j,m-1}$ for all j for which the elements (j,m) and $(j,m-1)$ are known. With the help of the ratios ρ_m, estimates for all remaining unknown elements (i,j) of $\mathring{\boldsymbol{L}}$ are obtained by multiplying known elements in the previous row or column with ρ_i and by dividing known elements in the subsequent row or column by ρ_{i+1}.

At the end of this procedure we have an estimate of $\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^T$. Depending on the pattern of unknown elements this estimate may not be exactly symmetric and it may also not have exactly rank one. We therefore proceed in the same manner as in the first case to estimate the vector $\boldsymbol{\lambda}$ from this matrix: First the matrix is symmetrized and then a best rank-one approximation is computed with the help of a singular value decomposition. The estimated vector $\hat{\boldsymbol{\lambda}}$ is then used to compute estimates for the channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ (cf. equations (18) and (19)).

5. SIMULATION RESULTS

Next, we demonstrate the achievable accuracy of the TENCE algorithm. We consider Rayleigh fading channels without spatial correlation and for simplicity we set the path loss to 1 (0 dB). We display the median of the relative squared estimation error which is defined as

$$\text{rMSE} = \min_{p=-1,1} \frac{\left\|\boldsymbol{H}_1 - p \cdot \hat{\boldsymbol{H}}_1\right\|_F^2}{\|\boldsymbol{H}_1\|_F^2}, \tag{21}$$

for $\boldsymbol{H}_1$ and similarly for $\boldsymbol{H}_2$. Here, the scalar quantity p is introduced to take into account the remaining sign ambiguity in the estimation. Four curves depict the estimation accuracy of $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ at user terminal 1 and 2, respectively.

For the simulation result depicted in Figure 2 both user terminals are equipped with 4 antennas the the relay with 2 antennas. Consequently, this corresponds to case 1 in Section 4. For the pilot symbol matrix $\boldsymbol{X}$, an 8×8 DFT matrix is used and the factor matrices of the tensor $\boldsymbol{\mathcal{G}}$ are chosen as $\boldsymbol{G}_1 = \boldsymbol{I}_2, \boldsymbol{G}_2 = \boldsymbol{D}_2, \boldsymbol{G}_3 = \boldsymbol{D}_2$, where $\boldsymbol{D}_2$ represents a 2×2 DFT matrix.

On the other hand, in Figure 3 we study a scenario where the case 2 of the algorithm is used since the number of relay antennas is set to 6 and the terminals both have 3 antennas. This time, $\boldsymbol{X}$ is a 6×6 DFT matrix and the factors of the tensor $\boldsymbol{\mathcal{G}}$ are computed through $\boldsymbol{G}_1 = \boldsymbol{I}_6, \boldsymbol{G}_2 = \boldsymbol{D}_6 \odot \boldsymbol{S}_{6,3}, \boldsymbol{G}_3 = \boldsymbol{D}_6$. Here, $\boldsymbol{D}_6$ represents a 6×6 DFT matrix and the matrix $\boldsymbol{S}_{6,3}$ is given by

$$\boldsymbol{S}_{6,3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{22}$$

Both simulations show that each terminal can estimate its own channel to the relay with a higher accuracy than the channel between the other terminal and the relay.

6. CONCLUSIONS

In this paper we propose the novel tensor-based channel estimation algorithm TENCE for two-way relaying systems with amplify-and-forward relays. In contrast to previous approaches, we do not assume any CSI at the relay, since the terminals can cancel their own self interference without the help of the relay provided that they have reliable channel state information.

TENCE provides both terminals with full knowledge of all channel parameters relevant for the transmission. It is applicable to arbitrary antenna configurations. Moreover it is very fast since the algebraic solution does not require any iterative procedures.

We also obtain design rules for the relay amplification matrices and the pilot symbols in order to achieve the best estimation accuracy. Computer simulations demonstrate the performance of TENCE.

REFERENCES

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

[2] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications", *SIAM Review*, to appear (accepted June 2008).

[3] N. Lee, H. Park, and J. Chun, "Linear precoder and decoder design for Two-Way AF MIMO relaying system", in *IEEE VTC Spring 2008*, pp. 1221–1225, May 2008.

[4] P. Lioliou and M. Viberg, "Least-squares based channel estimation for MIMO relays", in *Proc. ITG/IEEE Workshop on Smart Antennas (WSA '08)*, pp. 90–95, Darmstadt, Germany, Feb. 2008.

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

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

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