The Report Committee for Kevin Jinho Joe
Certifies that this is the approved version of the following report:

Limited Feedback Scheme using Tensor Decompositions for FDD Massive MIMO Systems

APPROVED BY

SUPERVISING COMMITTEE:

Jeffrey Andrews, Supervisor

Brian Evans

Limited Feedback Scheme using Tensor Decompositions for FDD Massive MIMO Systems

by

Kevin Jinho Joe

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2020

Acknowledgments

I would like to thank Prof. Jeffrey Andrews for his help, advice, and support during the research process. Because of him, I was able to find success at the University of Texas at Austin. In addition, I want to thank Prof. Brian Evans for being on my report committee and for his career guidance, specifically during his coffee hours. A special thanks to the Department of Electrical and Computer Engineering in the University of Texas at Austin. I am grateful for the ECE graduate program staff members for their help and extraordinary patience. Lastly, I appreciate all the emotional support, encouragement, and guidance I received during my graduate career from my colleagues, friends, and family in Texas, Indiana, Illinois, New York, Connecticut, Massachusetts, and England.

Limited Feedback Scheme using Tensor Decompositions for FDD Massive MIMO Systems

by

Kevin Jinho Joe, M.S.E.

The University of Texas at Austin, 2020

SUPERVISOR: Jeffrey Andrews

We propose a novel limited feedback scheme for massive multiple-input multiple-output (MIMO) systems in frequency-division duplexing (FDD) wideband system. We assume that the user (UE) has knowledge of a downlink (DL) channel estimate. In order for massive MIMO systems to achieve high capacity, the base station (BS) must have the DL channel state information. Traditional feedback methods cannot work because channels for massive MIMO systems are usually too large to feedback within the coherence time. Our goal is to feedback the DL channel estimate from the UE back to the BS with as little information as possible. Our method uses two different tensor decompositions, the canonical polyadic decomposition (CPD) and the rank-$(L_r, L_r, 1)$ or LL-1 block decomposition, on the DL frequency channel to estimate its parameters. By feeding back only the channel parameters, we show through simulations that our method is able to efficiently and accurately reconstruct the DL channel.

Table of Contents

List of Figures

Chapter 1

Introduction

Massive MIMO communication is a promising direction for wireless communications because it can increase data rates by using more antennas and not increasing the bandwidth. In order to get the gains from massive MIMO, time-division duplex (TDD) is necessary [1]. If massive MIMO is used for TDD systems, then reciprocity between the uplink (UL) and downlink (DL) channels can be used. However, for frequency division duplex (FDD) systems, full reciprocity between UL and DL cannot be assumed. For FDD non-massive antenna systems, traditional feedback methods work because the channels are not that large. However, in massive antennas systems, the channel become very large because channel scales with the number of antennas. Therefore, a simply feeding back the entire channel matrix cannot work in FDD massive MIMO systems because the channel will become so large that users cannot feedback the entire channel within the coherence time.

To combat this issue, compressed sensing (CS) techniques have been explored. In CS, DL pilots with a length less than the number of BS antennas are sent to the user. Then, the user would feedback the DL pilots observations and the channel is estimated at the BS. It also takes into consideration that the channel tends to become sparse in the angle domain as the number of antennas increases [2].

A multi-user scheme called Joint Orthogonal Matching Pursuit (J-OMP) takes DL pilot observations from all the users and jointly estimates channels for each user by taking advantage of common sparsity among users [2]. Another method called Active Channel Sparsification (ACS) designs a precoder that artificially sparsifies the channel and then uses DL pilot observations to estimate the channel [1]. The performance for both methods is directly correlated to the length of the DL pilots; if more DL pilots are sent, then the channel estimation becomes more accurate. Therefore, these methods would try and maximize the DL pilot length at the cost of the sum-rate. Knowing this, we decided to explore other feedback methods that do not rely on the feedback of DL pilot observations.

To lower feedback overhead, parameter estimation has been used to compress the DL channel. The narrowband multipath channel can be characterized by the angle-of-departure (AOD), angle-of-arrival (AOA), and channel gain [3]. In the setting where the BS has a uniform planar array (UPA) and the user has a uniform linear array (ULA), it has been shown that the channel can be represented as a tensor. In addition, channel parameters can be efficiently estimated by using carefully constructed training sequences that exploit the Vandermonde structures in the antenna array responses that make up the channel tensor [3].

Extending this work to wideband systems makes channel parameter estimation more difficult because of the additional time dimension, representing distinguishable arrival times of multipaths. We noticed that a wideband channel with a UPA and ULA at the BS and user side respectively has a structure of a LL-1 tensor [4]. In this paper, we propose a channel parameter estimation method for mas-

sive MIMO wideband channels to reduce feedback overhead of the known channel. We first use the LL-1 tensor decomposition to separate the spatial matrix and temporal vector parts of the channel [4] [5]. The multipath delays are extracted from the temporal vectors. Then, we exploit the Vandermonde structures in the spatial matrix part to extract the spatial channel parameters, stemming from previous work [3]. Finally, the temporal and spatial parameters are fed back and the DL channel is then reconstructed at the BS.

The remainder of this paper is organized as follows. In Chapter 2, we introduce relevant tensor theorems and definition. In Chapter 3, we describe the channel model and show how tensors are applied. In Chapter 4, we explain how parameters are estimated. Our simulation results are presented in Chapter 5. Lastly, in Chapter 6, we provide a conclusion and ideas for future work.

Notation: Tensors, matrices, and vectors are denoted by calligraphic letters, bold upper case letters, and bold lower case letters, respectively. The operators $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^\dagger$, $|\cdot|$, $\|\cdot\|_p$, $\mathrm{vec}(\cdot)$, and $\mathrm{diag}(\cdot)$ indicate the transpose, Hermetian transpose, Moore-Penrose pseudoinverse, magnitude of a complex number, p-norm, vectorization, and diagonalization, respectively. $\mathbf{I}_n$ denotes the identity matrix of size $n \times n$. $\mathbf{0}_{n\times m}$ denotes a matrix of size $n \times m$ of all zeros. Symbols $\circ$, $\otimes$, and $\odot$ denote the outer product, Kronecker product, and Khatri-Rao product. $[\mathbf{A}]_{i,j}$ references the (i, j) entry of the matrix $\mathbf{A}$. $[\mathbf{A}]_{i,:}$ and $[\mathbf{A}]_{:,j}$ references the i^{th} row vector and j^{th} column vector respectively. $[\mathbf{A}]_{i:j,:}$ references the submatrix that consists of the i^{th} to the j^{th} row vectors. This notation for matrix indexing is extended for vectors and tensors.

Chapter 2

Tensor Preliminaries

We consider two different tensor decompositions, the canonical polyadic decomposition and the LL-1 decomposition. To keep this report self-contained, relevant definitions, theorems, and properties are presented in this chapter.

2.1 Canonical Polyadic Decomposition

Definition 1. *[4] [6] A canonical polyadic decomposition (CPD) of a rank-R tensor $\mathcal{T} \in \mathbb{C}^{I \times J \times K}$ is a decomposition of $\mathcal{T}$ in a linear combination of R rank-1 terms:*

$$\mathcal{T} = \sum_{r=1}^{R} [\mathbf{A}]_{:,r} \circ [\mathbf{B}]_{:,r} \circ [\mathbf{C}]_{:,r} \tag{2.1}$$

$$:= [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \tag{2.2}$$

$$[\mathcal{T}]_{i,j,k} = \sum_{r=1}^{R} [\mathbf{A}]_{i,r} [\mathbf{B}]_{j,r} [\mathbf{C}]_{c,r} \tag{2.3}$$

where $\mathbf{A} \in \mathbb{C}^{I \times R}$, $\mathbf{B} \in \mathbb{C}^{J \times R}$, and $\mathbf{C} \in \mathbb{C}^{K \times R}$ are considered the factor matrices. The rank of tensor $\mathcal{T}$ is the minimum number of rank-1 tensors needed to produce $\mathcal{T}$ as their sum.

Definition 2. *[3] [6] Tensor unfolding is obtained by taking the mode-n slabs of the tensor (i.e. subtensors obtained by fixing the n^{th} index of the original tensor), vectorizing the slabs, and then stacking all the vectors from left to the right in a*

matrix. For a third-order tensor $\mathcal{T} \in \mathbb{C}^{I \times J \times K}$ with factor matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$, the matrix unfolding along the first, second, and third dimensions, respectively, are the following:

$$\mathcal{T}_1 := \begin{bmatrix} \text{vec}([\mathcal{T}]_{1,:,:}) & \dots & \text{vec}([\mathcal{T}]_{I,:,:}) \end{bmatrix} \tag{2.4}$$

$$= (\mathbf{C} \odot \mathbf{B})\mathbf{A}^T \in \mathbb{C}^{JK \times I} \tag{2.5}$$

$$\mathcal{T}_2 := \begin{bmatrix} \text{vec}([\mathcal{T}]_{:,1,:}) & \dots & \text{vec}([\mathcal{T}]_{:,J,:}) \end{bmatrix} \tag{2.6}$$

$$= (\mathbf{C} \odot \mathbf{A})\mathbf{B}^T \in \mathbb{C}^{IK \times J} \tag{2.7}$$

$$\mathcal{T}_3 := \begin{bmatrix} \text{vec}([\mathcal{T}]_{:,:,1}) & \dots & \text{vec}([\mathcal{T}]_{:,:,K}) \end{bmatrix} \tag{2.8}$$

$$= (\mathbf{B} \odot \mathbf{A})\mathbf{C}^T \in \mathbb{C}^{IJ \times K} \tag{2.9}$$

Definition 3. *[6] Given a tensor $\mathcal{T}$ of rank R, we say that its CPD is essentially unique if the R rank-1 terms in its decompositions are unique, i.e., there is no other way to decompose $\mathcal{T}$ for the given number of rank-1 terms. Note that we can of course permute these terms without changing their sum, hence there exists an inherently unresolvable permutation ambiguity in the rank-1 tensors. If $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ and $\mathbf{A} \in \mathbb{C}^{I \times R}$, $\mathbf{B} \in \mathbb{C}^{J \times R}$, and $\mathbf{C} \in \mathbb{C}^{K \times R}$, then essential uniqueness means that $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are unique up to a common permutation and scaling/counter-scaling of columns. In other words, if $\mathcal{T} = [\![\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]\!]$ and the factor matrices are the same size as their respective counterparts, then there exists a permutation matrix $\mathbf{\Pi}$ and*

diagonal scaling matrices $\mathbf{\Lambda}_1$*,* $\mathbf{\Lambda}_2$*, and* $\mathbf{\Lambda}_3$*, such that:*

$$\tilde{\mathbf{A}} = \mathbf{A}\mathbf{\Pi}\mathbf{\Lambda}_1 \tag{2.10}$$

$$\tilde{\mathbf{B}} = \mathbf{B}\mathbf{\Pi}\mathbf{\Lambda}_2 \tag{2.11}$$

$$\tilde{\mathbf{C}} = \mathbf{C}\mathbf{\Pi}\mathbf{\Lambda}_3 \tag{2.12}$$

$$\mathbf{\Lambda}_1\mathbf{\Lambda}_2\mathbf{\Lambda}_3 = \mathbf{I} \tag{2.13}$$

It is not always guaranteed that an essentially unique solution is identifiable. Many conditions for essential uniqueness is presented in [6]. If factor matrices have some structure, then it is possible to use this information to get an identifiable result. The following theorem that exploits a Vandermonde structure in a factor matrix will be useful later in this paper.

Theorem 4. *[3] [7] Consider a third-order tensor* $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ *where* $\mathbf{A} \in \mathbb{C}^{I\times R}$*,* $\mathbf{B} \in \mathbb{C}^{J\times R}$*, and* $\mathbf{C} \in \mathbb{C}^{K\times R}$*, and* $\mathbf{A}$ *is Vandermonde with distinct nonzero generators. Assume that* $\mathbf{B}$ *and* $\mathbf{C}$ *are drawn from an absolutely continuous distribution. If*

$$R \leqslant \min\Big((P-1)J\ ,\ QK\Big) \tag{2.14}$$

where $P \geqslant Q$ *and* $Q = I + 1 - P$ *are chosen from*

$$\{P, Q\} = \arg\max_{\{P,Q\}\in\mathbb{Z}^+} \min\Big((P-1)J\ ,\ QK\Big), \tag{2.15}$$

then $\mathbf{A}$*,* $\mathbf{B}$*, and* $\mathbf{C}$ *are essentially unique with probability one.*

2.2 Rank-$(L_r, L_r, 1)$ Decomposition

Definition 5. *[4] [8] A decomposition of a tensor $\mathcal{T} \in \mathbb{C}^{I \times J \times K}$ in a sum of rank-$(L_r, L_r, 1)$ terms, $1 \leqslant r \leqslant R$ is of the following form:*

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{E}_r \circ \mathbf{c}_r \tag{2.16}$$

$$= \sum_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^T) \circ \mathbf{c}_r \tag{2.17}$$

where $\mathbf{E}_r \in \mathbb{C}^{I \times J}$, $\mathbf{A}_r \in \mathbb{C}^{I \times L_r}$, and $\mathbf{B}_r \in \mathbb{C}^{J \times L_r}$ are rank-L_r and $c_r \in \mathbb{C}^K$ is nonzero.

.

Definition 6. *[8] If the rank-$(L_r, L_r, 1)$ decomposition of $\mathcal{T}$ is essentially unique, then the decomposition can have the following indeterminacies:*

- *Ror $1 \leqslant p, q \leqslant R$, if $L_p = L_q$, then the p^{th} and q^{th} terms can be permuted in Equations 2.16 and 2.17*
- *$\mathbf{E}_r$ and $\mathbf{c}_r$ can be scaled and counterscaled by any scalar $\lambda_r \in \mathbb{C}$*
- *$\mathbf{A}_r$ can be postmultiplied by any non-singular matrix $\mathbf{F}_r \in \mathbb{C}^{L_r, L_r}$, provided that $\mathbf{B}_r$ is postmultiplied by $\mathbf{F}_r^{-1}$.*

The indeterminacies can be represented mathematically as so:

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{E}_r \circ \mathbf{c}_r = \sum_{r=1}^{R} \mathbf{E}_r \lambda_r \circ \mathbf{c}_r \lambda_r^{-1} \tag{2.18}$$

$$= \sum_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^T) \circ \mathbf{c}_r = \sum_{r=1}^{R} \Big(\mathbf{A}_r \mathbf{F}_r \big(\mathbf{B}_r (\mathbf{F}_r^{-1})^T \big)^T \Big) \lambda_r \circ \mathbf{c}_r \lambda_r^{-1}. \tag{2.19}$$

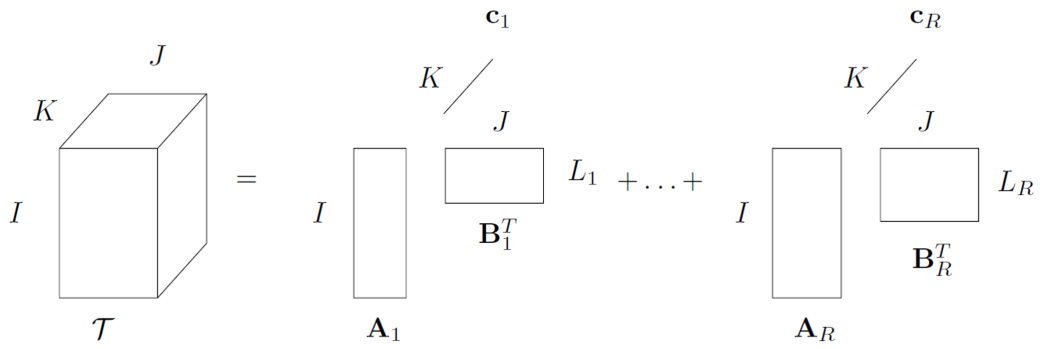

Figure 2.1: Visual Representation of the LL-1 decomposition

It is worth noting that the CPD is a special case of LL-1 decomposition if $L_r = 1$ for $1 \leqslant r \leqslant R$. An ideal case for the non-singular matrix F_r is that it is made up of a common permutation matrix and a diagonal scale and counterscale. In certain conditions, that is the case, which is explored in [8]. A visual representation of the LL-1 decomposition is shown in Figure 2.1 [4].

Chapter 3

Channel Model

In this chapter, we introduce channel model used in time-domain and the frequency domain. Then, we show that the model can be morphed into a tensor structure such that the CPD and LL-1 decomposition can be used.

3.1 Time-Domain and Frequency-Domain Channel

In this paper, a single-user FDD massive MIMO system is considered. The UE has M_R receive antennas and the BS has M_T transmit antennas. In the 3GPP 3D channel model, the channel is divided up into L clusters, where each cluster has an associated delay, and the l^{th} cluster is divided up into S_l rays (or subclusters), where each ray has an associated AoA, AoD, and gain. In the time-domain, the channel $\mathbf{H}(t) \in \mathbb{C}^{M_r \times M_t}$ is represented by the following [9] [10]:

$$\mathbf{H}(t) = \sum_{l=1}^{L} \sum_{s=1}^{S_l} g_{l,s} \mathbf{a}_R(\theta_{R,l,s}) \mathbf{a}_T(\theta_{T,l,s}, \phi_{T,l,s})^H \delta(t - \tau_l) \tag{3.1}$$

where

- $\mathbf{a}_R \in \mathbb{C}^{M_R}$ is the receive array response
- $\mathbf{a}_T \in \mathbb{C}^{M_T}$ is the transmit array response

- $g_{l,s} \in \mathbb{C}$ is the gain for the s^{th} ray of the l^{th} cluster
- $\theta_{R,l,s}$ is the azimuth AoA for the s^{th} ray of the l^{th} cluster
- $\theta_{T,l,s}$ is the azimuth AoD for the s^{th} ray of the l^{th} cluster
- $\phi_{T,l,s}$ is the elevation AoD for the s^{th} ray of the l^{th} cluster
- τ_l is the delay for the l^{th} cluster.

For the UE, a ULA is used, and at the BS, a UPA of size $M_x \times M_y$ is used. Assuming the antenna spacing is at half-wavelength, the array responses are represented by the following:

$$\mathbf{a}_R(\theta_{R,l,s}) = \left[1, \quad e^{j\pi \sin(\theta_{R,l,s})}, \quad \ldots, \quad e^{j(M_R-1)\pi \sin(\theta_{R,l,s})}\right]^T \tag{3.2}$$

$$\mathbf{a}_x(\theta_{T,l,s}, \phi_{T,l,s}) = \left[1, \quad e^{j\pi \cos(\theta_{T,l,s}) \sin(\phi_{T,l,s})}, \quad \ldots, \quad e^{j(M_x-1)\pi \cos(\theta_{T,l,s}) \sin(\phi_{T,l,s})}\right]^T \tag{3.3}$$

$$\mathbf{a}_y(\theta_{T,l,s}, \phi_{T,l,s}) = \left[1, \quad e^{j\pi \sin(\theta_{T,l,s}) \sin(\phi_{T,l,s})}, \quad \ldots, \quad e^{j(M_y-1)\pi \sin(\theta_{T,l,s}) \sin(\phi_{T,l,s})}\right]^T \tag{3.4}$$

$$\mathbf{a}_T(\theta_{T,l,s}, \phi_{T,l,s}) = \mathbf{a}_y(\theta_{T,l,s}, \phi_{T,l,s}) \otimes \mathbf{a}_x(\theta_{T,l,s}, \phi_{T,l,s}) \tag{3.5}$$

From the time-domain channel, the frequency domain channel can be derived by sampling $\mathbf{H}(t)$ and calculating the Discrete Fourier Transform (DFT) [9]. The channel at the k^{th} subcarrier, $\mathbf{H}[k]$, is represented by the following:

$$\mathbf{H}[k] = \sum_{l=1}^{L} \sum_{s=1}^{S_l} g_{l,s} \mathbf{a}_R(\theta_{R,l,s}) \mathbf{a}_T(\theta_{T,l,s}, \phi_{T,l,s})^H e^{-j2\pi\tau_l f_s k/K} \tag{3.6}$$

where

- f_s is the sampling frequency
- K is the total number of subcarriers.

3.2 Frequency-Domain Channel Tensors

The frequency domain channel for each subcarrier stacked together to create a tensor $\mathcal{H} \in \mathbb{C}^{M_R \times M_T \times K}$ as shown:

$$\mathcal{H} = \sum_{l=1}^{L} \left(\sum_{s=1}^{S_l} g_{l,s} \mathbf{a}_R(\theta_{R,l,s}) \mathbf{a}_T(\theta_{T,l,s}, \phi_{T,l,s})^H \right) \circ \mathbf{f}_l \tag{3.7}$$

$$\mathbf{f}_l = \begin{bmatrix} 1, & e^{-j2\pi\tau_l f_s/K}, & \ldots, & e^{-j2\pi\tau_l f_s(K-1)/K} \end{bmatrix}^T \tag{3.8}$$

From here, we noticed that the structure of the channel tensor follows the form of the LL-1 decomposition, as shown in Equations 2.16 and 2.17. By concatenating the array responses of the rays in a specific cluster, it looks like the following:

$$\mathcal{H} = \sum_{l=1}^{L} \left(\mathbf{A}_{R,l} \mathbf{G}_l \mathbf{A}_{T,l}^H \right) \circ \mathbf{f}_l \tag{3.9}$$

$$= \sum_{l=1}^{L} \mathbf{E}_l \circ \mathbf{f}_l \tag{3.10}$$

where

$$\mathbf{A}_{R,l} = \begin{bmatrix} \mathbf{a}_R(\theta_{R,l,1}), & \ldots, & \mathbf{a}_R(\theta_{R,l,S_l}) \end{bmatrix} \tag{3.11}$$

$$\mathbf{A}_{T,l} = \begin{bmatrix} \mathbf{a}_T(\theta_{T,l,1}, \phi_{T,l,1}), & \ldots, & \mathbf{a}_T(\theta_{T,l,S_l}, \phi_{T,l,S_l}) \end{bmatrix} \tag{3.12}$$

$$\mathbf{G}_l = \text{diag}\Big(\begin{bmatrix} g_{l,1}, & \ldots, & g_{l,S_l} \end{bmatrix} \Big) \tag{3.13}$$

From this model, the rank of the l^{th} cluster is S_l. We would want to decompose the tensor in rank-$(S_l, S_l, 1)$ terms. Ideally, if we compute the LL-1 decomposition, we get Equation 3.9 and we can acquire the angles and gains from $\mathbf{E}_l$. However, because of the third bullet point of LL-1 decomposition uniqueness property from Definition 6, there is no clear way to distinguish the components in

a structure that is useful for us if we were to only use the LL-1 decomposition. So, inspired by [3], we exploit the UPA structure on the transmit side to make $\mathbf{E}_l$ an unfolded tensor. By using the UPA structure in the transmit side and Equation 3.5, we get the following:

$$\mathbf{A}_{x,l} = \left[\mathbf{a}_x(\theta_{T,l,1}, \phi_{T,l,1}), \quad \ldots, \quad \mathbf{a}_x(\theta_{T,l,S_l}, \phi_{T,l,S_l})\right] \tag{3.14}$$

$$\mathbf{A}_{y,l} = \left[\mathbf{a}_y(\theta_{T,l,1}, \phi_{T,l,1}), \quad \ldots, \quad \mathbf{a}_y(\theta_{T,l,S_l}, \phi_{T,l,S_l})\right] \tag{3.15}$$

$$\mathbf{A}_{T,l} = \mathbf{A}_{y,l} \odot \mathbf{A}_{x,l} \tag{3.16}$$

From here, we plug Equation 3.16 into Equation 3.10 and get the following:

$$\mathbf{E}_l = \mathbf{A}_{R,l}\mathbf{G}_l(\mathbf{A}_{y,l} \odot \mathbf{A}_{x,l})^H \tag{3.17}$$

$$\mathcal{E}_{l,1} = \mathbf{E}_l^T = (\mathbf{A}_{y,l}^* \odot \mathbf{A}_{x,l}^*)(\mathbf{A}_{R,l}\mathbf{G}_l)^T \tag{3.18}$$

Now, Equation 3.18 has the same form as a matrix unfolding of tensor, as shown from Equations 2.5, 2.7, and 2.9, depending on how the tensor dimensions are defined. Here, we define it as the unfolded tensor of $\mathcal{E}$ along the first dimension. Having this tensor structure is important because it allows us to use the Factor Matrices Algorithm from [3] to get an essentially unique CPD result to estimate the angles and gains for a cluster. This will inherently avoid the non-singular matrix factor from Definition 6.

Chapter 4

Channel Parameter Estimation

Stemming from the previous section, we show how the LL-1 decomposition and the CPD are calculated. In addition, from its components, we show how we estimate the channel parameters. In general, the delays are estimated from the LL-1 decomposition, and the gains and angles are estimated from CPD.

4.1 Rank-$(L_r, L_r, 1)$ Decomposition and Delay Estimation

We used Tensorlab [5] tools to calculate the LL-1 decomposition. Specifically in Equation 2.17, we had to set a random initialization for components $\mathbf{A}_r$, $\mathbf{B}_r$, and $\mathbf{c}_r$. Then, the decomposition was found using the unconstrained nonlinear optimization method. Further details regarding the algorithm methods and code can be found in [5] [11]. The decomposition algorithm will provide components of the following form:

$$\mathcal{H} = \sum_{l}^{L} \hat{\mathbf{E}}_l \circ \hat{\mathbf{f}}_l = \sum_{l}^{L} \mathbf{E}_l \lambda_l^{-1} \circ \mathbf{f}_l \lambda_l \tag{4.1}$$

To estimate the delays, we take the estimated DFT vector and find the delay that gives maximum correlation [9][1] :

$$\hat{\tau}_l = \arg\max_{\tau} \frac{|\hat{\mathbf{f}}_l^H \mathbf{f}(\tau)|}{\|\hat{\mathbf{f}}_l\|_2 \|\mathbf{f}(\tau)\|_2} , \tag{4.2}$$

where

$$\mathbf{f}(\tau) = \begin{bmatrix} 1, & e^{-j2\pi\tau_l f_s/K}, & \ldots, & e^{-j2\pi\tau f_s(K-1)/K} \end{bmatrix}^T . \tag{4.3}$$

4.2 Factor Matrix, Angle, and Gain Estimation

We are able to calculate the CPD of the inner matrix $\mathbf{E}_l$ by using the factor matrices algorithm. Details of the algorithm can be found in [3]. The output is the following:

$$\mathbf{E}_l = \hat{\mathbf{E}}_l \lambda_l = \hat{\mathbf{A}}_{R,l} (\hat{\mathbf{A}}_{y,l} \odot \hat{\mathbf{A}}_{x,l})^H \lambda_l \tag{4.4}$$

$$= \mathbf{A}_{R,l} (\mathbf{\Lambda}_{R,l} \mathbf{\Lambda}_{y,l}^* \mathbf{\Lambda}_{x,l}^*) (\mathbf{A}_{y,l} \odot \mathbf{A}_{y,l})^H \lambda_l \tag{4.5}$$

where $\mathbf{\Lambda}$ is diagonal.

By using the Vandermonde structure of the array responses from CPD, we can calculate the angles [3]. To calculate the AoA:

$$\omega_{R,l,s} = \angle \Big([\hat{\mathbf{a}}_{R,l,s}]_{1:M_R-1}^H [\hat{\mathbf{a}}_{R,l,s}]_{2:M_R} \Big) \tag{4.6}$$

$$\theta_{R,l,s} = \sin^{-1} \left(\frac{\omega_{R,l,s}}{\pi} \right) \tag{4.7}$$

[1] In the cited paper, there is a mistake that uses *arg min* instead of *arg max*.

Next, to calculate the AoD:

$$\omega_{x,l,s} = \angle\Big([\hat{\mathbf{a}}_{x,l,s}]^H_{1:M_x-1}[\hat{\mathbf{a}}_{x,l,s}]_{2:M_x}\Big) \tag{4.8}$$

$$\omega_{y,l,s} = \angle\Big([\hat{\mathbf{a}}_{y,l,s}]^H_{1:M_y-1}[\hat{\mathbf{a}}_{y,l,s}]_{2:M_y}\Big) \tag{4.9}$$

$$\theta_{T,l,s} = \tan^{-1}\left(\frac{\omega_{y,l,s}}{\omega_{x,l,s}}\right) \tag{4.10}$$

$$\phi_{T,l,s} = \sin^{-1}\left(\sqrt{\left(\frac{\omega_{x,l,s}}{\pi}\right)^2 + \left(\frac{\omega_{y,l,s}}{\pi}\right)^2}\right) \tag{4.11}$$

To calculate the gain estimation, we first normalize the CPD components from the Equation 4.4. Then, we get the normalization factors from the LL-1 decomposition from Equation 4.1:

$$\mathbf{G}_l = (\mathbf{\Lambda}_{R,l}\mathbf{\Lambda}^*_{y,l}\mathbf{\Lambda}^*_{x,l})\lambda_l \tag{4.12}$$

Now that AoA, AoD, gain, and delay parameters are estimated, the user can feedback the parameters and the BS can reconstruct the DL channel. The total number of parameters p would include L delay parameters, and S_l azimuth AoD, elevation AoD, azimuth AoA, and gain parameters, as shown in Equation 4.13:

$$p = L + 4\sum_{l=1}^{L} S_l \tag{4.13}$$

4.3 Decomposition Errors from Computation

From observation, we noticed that the LL-1 decomposition algorithm from TensorLab may not produce the correct decomposition, meaning that reconstructing the tensor directly after the LL-1 decomposition algorithm does not produce the

same tensor. Because the problem itself is non-convex, the solution highly depends on the initialization. To combat this issue, we first try multiple initializations of the same channel tensor. Then, we select the best performing decomposition by calculating the normalized mean square error (NMSE) between the estimation and the known channel at the UE side. Our final procedure is shown in Algorithm 1.

Algorithm 1: Final Procedure

for T *initializations* **do**
  Set a random initialization and calculate the LL-1 decomposition of $\mathcal{H}$ using Tensorlab
  Estimate the delays using Equation 4.2
  Calculate the CPD using methods from [3]
  Estimate the angles using Equations 4.7, 4.10, and 4.11
  Estimate the gains using Equation 4.12
  Reconstruct estimated channel from estimated parameter and calculate NMSE

$$\text{NMSE} = \frac{\|\mathbf{H} - \hat{\mathbf{H}}\|_F^2}{\|\mathbf{H}\|_F^2}$$

end
Select parameters from the initialization the provided the lowest NMSE and feedback to the BS

Chapter 5

Simulations and Discussion

For our system, we used a UPA on the transmit side with $M_x = M_y = 8$ and a ULA on the receive side with $M_r = 8$. In addition, we had $f_s = 320$MHz and $K = 128$. For the channel clusters and rays, we used $L = 4, 8, 12$ and $S_l = 2$. In order to see if the LL-1 decomposition method is viable for parameter estimation, it is assumed the the user knows the number of clusters and rays in the channel. In the future, we hope to address the problem of determining the number of clusters and rays by taking portions of the Active Channel Sparsification method and extending its application to wideband systems. For the simulation, we had a total of 10 initializations, 50 channel drops, and 5 noise realizations. We tested for SNR ranging from -10dB to 30dB. Lastly, we assume the user is able to perfectly feedback the parameters to the BS.

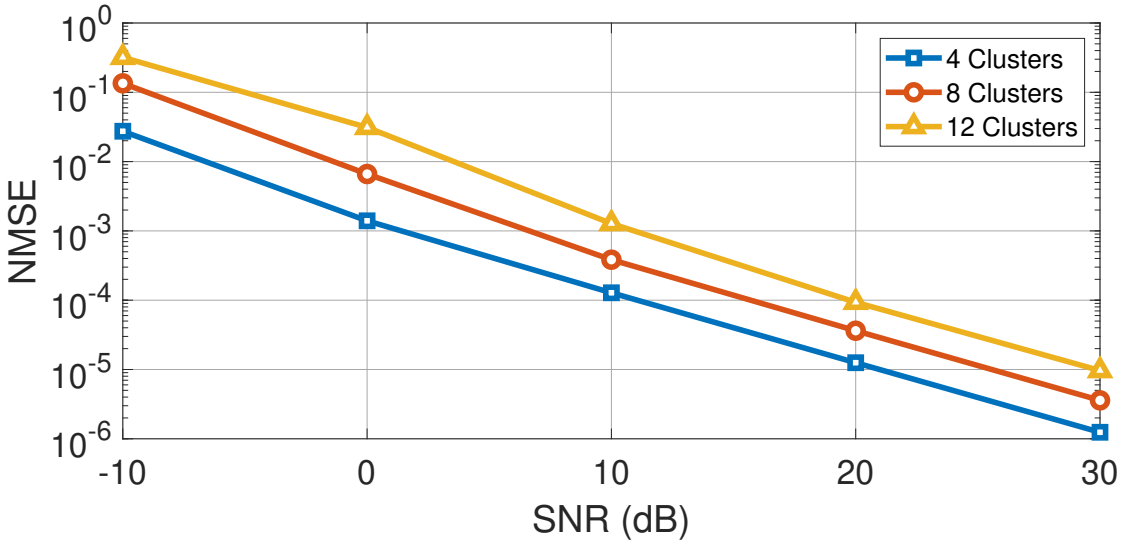

Figure 5.1: Median NMSE vs SNR for 4, 8, and 12 Clusters

In Figure 5.1, we show the median NMSE of reconstructed channel over SNR in comparison with the 50 channel drops. We noticed that as SNR increases, the reconstruction performs better. However, when the number of clusters increases, the performance gets worse. We believe the LL-1 decomposition has more difficulty identifying more clusters. In comparison, we show the mean of the NMSE over the all channel drops and noise realizations in Figure 5.2. We believe that the mean plot is not as clean as the median plot because LL-1 decomposition method is sensitive to the initialization. We observe that it has more trouble when there are more clusters to identify.

Because of the stark difference in the median and mean plots, we decided to look into the distribution of the NMSE plot. In Figure 5.3, we show the log

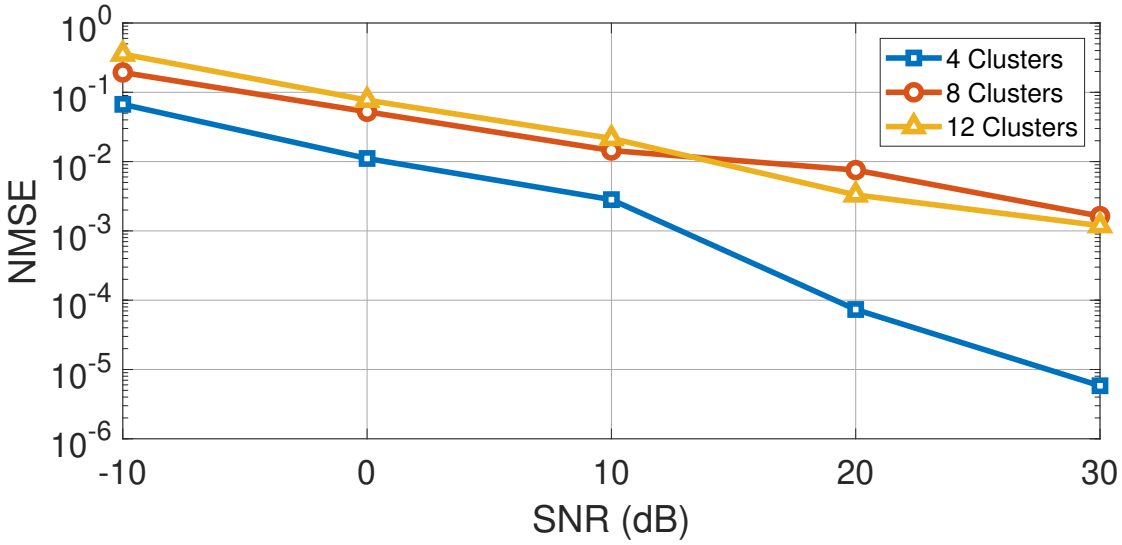

Figure 5.2: Mean NMSE vs SNR for 4, 8, and 12 Clusters

histogram of NMSE for $L = 12$ and SNR $= 30$dB over all 50 channel drops and 5 noise realizations. The mean NMSE was $1.1901 \cdot 10^{-3}$ and the median NMSE was $9.6434 \cdot 10^{-6}$. The distribution is left skewed, meaning that majority of the time, the parameter estimation is working well. However, because there are a few instances where the NMSE values are off by 3 to 4 orders of magnitude, the mean value becomes skewed. To see the general spread of the NMSE for $L = 12$ over the SNR range, we plot minimum, maximum, mean, and median NMSE in Figure 5.4.

We examined deeper to see channel situations when the estimation is good or bad. From observation, we noticed that all of the clusters may not be identified by the LL-1 decomposition in its first iteration. This caused errors to propagate through the rest of the algorithm, resulting in a poor NMSE. Since we assume the

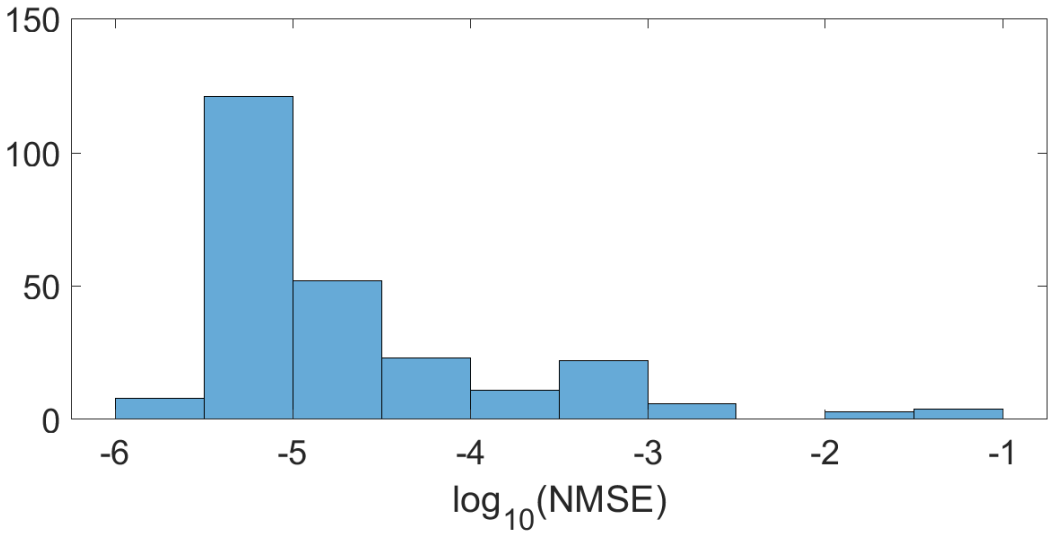

Figure 5.3: NMSE Distribution for $L = 12$ and SNR = 30dB

user knows the number of clusters L in the channel, we test if we find the L unique delays before moving on with the parameter estimation. During an iteration, if we do not find L unique delays, we repeat the LL-1 decomposition but with a random initialization that includes the unique delays we found in the previous iteration. Specifically in poor SNR situations, the LL-1 decomposition may not be able to find L unique delays. Hence, we limit the number of repetitions the algorithm attempts, so it can always move forward. This made the algorithm more robust, however it was at the cost of computation time.

With the algorithm change, for a specific channel drop for $L = 12$ and SNR = 30dB, the estimated angle, delay, and gain parameters were graphically compared with the original parameters. This is shown in the power delay profile in Figure

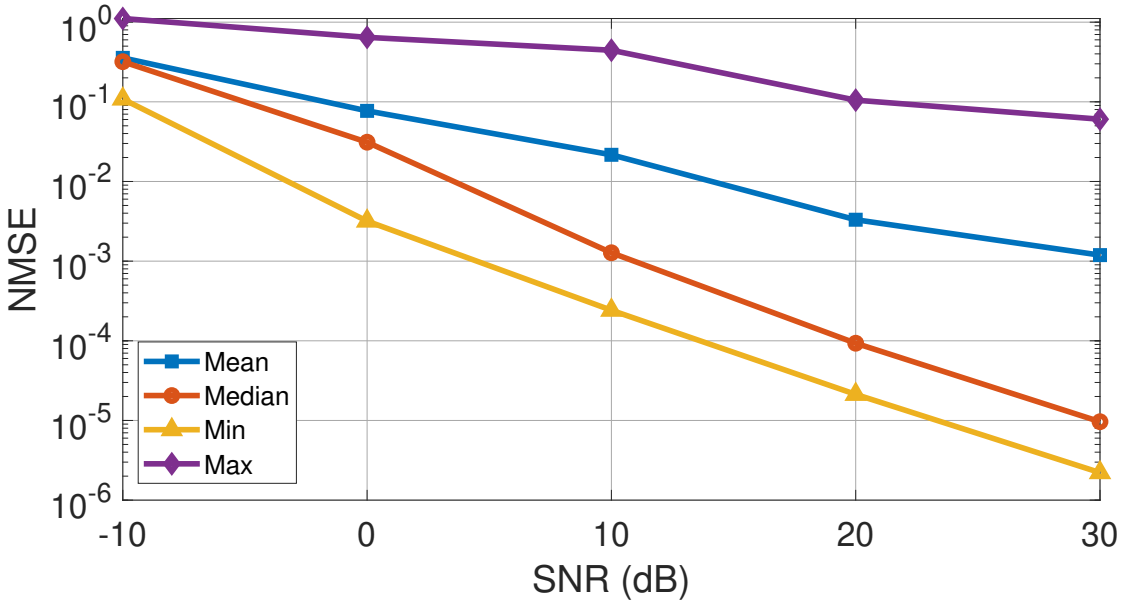

Figure 5.4: NMSE vs SNR Distribution for $L = 12$

5.5. For this channel drop, all the estimated parameters visually matched with the original parameters and, as expected, the NMSE was $6.1518 \cdot 10^{-5}$. However, for another channel drop with the same clusters and SNR, not all the estimated parameters matched with the original parameters, as seen in Figure 5.6. Specifically, there are two wrongly estimated angles in Figure 5.6b, one wrong angle in Figure 5.6c, and one gain that is barely off in Figure 5.6d. Besides these errors, it seems as though the rest of the parameters were estimated correctly. Despite that, the NMSE was $6.0719 \cdot 10^{-2}$. This shows that in order to have good NMSE, the parameter estimation should be near perfect.

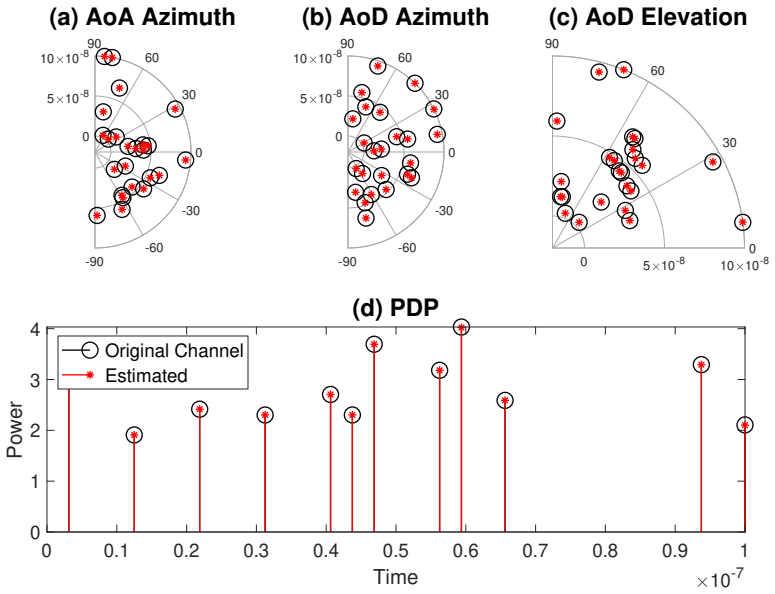

Figure 5.5: Accurate Power Delay Profile with $L = 12$ and SNR = 30dB

The compression ratio was directly related to the number of clusters and rays identified in the channel. For our channel parameters, there are 65536 channel coefficients. For $S_l = 2$ and $L = 4, 8, 12$ and using Equation 4.13, we are able to compress the channel into only 36, 72, and 108 parameters, respectively. For J-OMP and ACS, the total number of values to feedback to the BS would be $M_r T$, where T represents the DL pilot length. Therefore, in order for J-OMP and ACS to match the number of values to feedback as our method, the DL pilot length must be within the following bound:

$$T < \frac{L}{M_r} + \frac{4}{M_r} \sum_{l=1}^{L} S_l \tag{5.1}$$

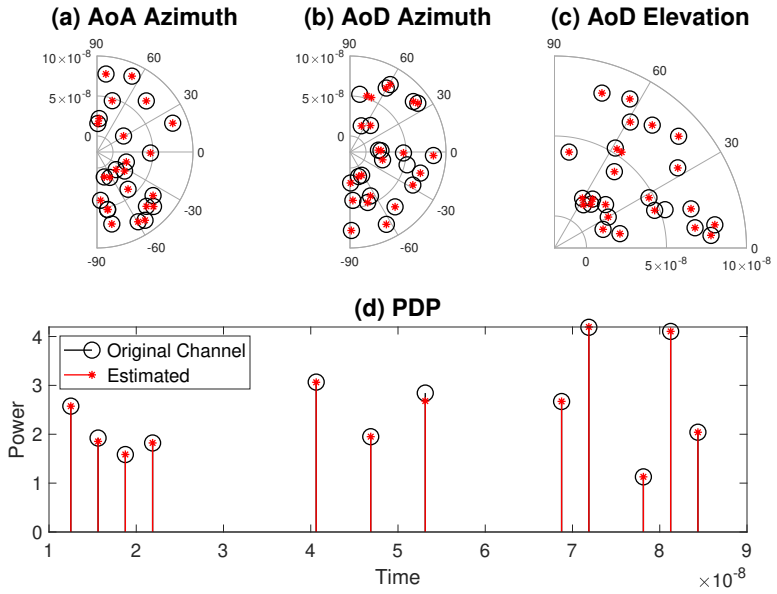

Figure 5.6: Inaccurate Power Delay Profile with $L = 12$ and SNR = 30dB

Again for $S_l = 2$ and $L = 4, 8, 12$, T must be less than 4.5, 9, and 13.5, respectively. Despite the different setting and system parameters, the results in [2] only show the NMSE for J-OMP with T ranging from 30 to 70, where it performed the worst at $T = 30$. This shows very little promise for J-OMP to perform well with training lengths 4, 9, and 13. Similarly, the results in [1] show the NMSE for ACS with T ranging from 10 to 120, where it performed the worst at $T = 10$. We hope to compare these methods more thoroughly in the future. From our simulations, we show that we can achieve high compression ratios with low NMSE, especially for channels with fewer clusters and rays.

Chapter 6

Conclusion

We believe that using tensor decompositions for FDD massive MIMO is promising for parameter estimation and reducing feedback. For a small number of clusters and rays, we show that our method can be reliable and is robust to noise. However, the larger amount of cluster still have some erratic behavior that is not quite understood yet due to the randomness in the LL-1 decomposition. In the future, we hope that we can exploit the Vandermonde structures in the tensor components to be able to accurately calculate the LL-1 decomposition using tensor mathematics, similarly to exploiting the Vandermonde structures to calculate essentially unique CPDs in [3] [7], rather that using a nonlinear least squares method [5]. We also hope to determine the maximum number of identifiable clusters and rays given the number of antennas and subcarriers. Even if the increase in clusters and rays become a problem, it is possible to forcibly limit the number of rays and clusters using carefully designed precoders, as shown in [1]. We believe that it might be beneficial to combine rays or clusters because if the parameters too close together, we noticed that our decomposition methods are not always able to distinguish between the rays and clusters. Therefore, we hope to understand the exact angular and time resolution for identifying clusters and rays for the channel tensor by discretizing parameters.

Another drawback to our method is that it takes a significant amount of time to estimate the parameters. We observed that the main bottle neck is the LL-1 decomposition and unique delay estimation. If we are unable to actually compress the channel in an appropriate amount of time, it may be as useless as trying to feedback the uncompressed channel. So for future work, we hope to calculate the achievable sum-rate of our method and see how it performs based on the number of clusters and rays. Despite these drawbacks, we believe the LL-1 tensor decomposition can make massive MIMO viable for FDD through tensor decompositions.

Bibliography

[1] M. Barzegar Khalilsarai, S. Haghighatshoar, X. Yi, and G. Caire, "Fdd massive mimo via ul/dl channel covariance extrapolation and active channel sparsification," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 121–135, 2019.

[2] X. Rao and V. K. N. Lau, "Distributed compressive csit estimation and feedback for fdd multi-user massive mimo systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3261–3271, 2014.

[3] C. Qian, X. Fu, and N. D. Sidiropoulos, "Algebraic channel estimation algorithms for fdd massive mimo systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 961–973, 2019.

[4] L. De Lathauwer, "Blind separation of exponential polynomials and the decomposition of a tensor in rank-$(l_r,l_r,1)$ terms," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 4, pp. 1451–1474, 2011.

[5] N. Vervliet, O. Debals, L. Sorber, M. V. Barel, and L. De Lathauwer, "Tensorlab 3.0," Mar 2016.

[6] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine

learning," *IEEE Transactions on Signal Processing*, vol. 65, pp. 3551–3582, July 2017.

[7] M. Sørensen and L. De Lathauwer, "Blind signal separation via tensor decomposition with vandermonde factor: Canonical polyadic decomposition," *IEEE Transactions on Signal Processing*, vol. 61, pp. 5507–5519, Nov 2013.

[8] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms—part ii: Definitions and uniqueness," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1033–1066, 2008.

[9] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. S. Blum, "Low-rank tensor decomposition-aided channel estimation for millimeter wave mimo-ofdm systems," *IEEE Journal on Selected Areas in Communications*, vol. 35, pp. 1524–1538, July 2017.

[10] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," Technical Report (TR) 38.901, 3rd Generation Partnership Project (3GPP), January 2018. Version 16.1.0.

[11] L. Sorber, M. Van Barel, and L. De Lathauwer, "Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(l_r,l_r,1)$ terms, and a new generalization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 695–720, 2013.