Generalized Tensor Contractions for an Improved Receiver Design in MIMO-OFDM Systems

Kristina Naskovska *, Martin Haardt * and André L. F. de Almeida §

*Communications Research Laboratory, Ilmenau University of Technology, P. O. Box 100565, D-98684 Ilmenau, Germany
Email: kristina.naskovska, martin.haardt@tu-ilmenau.de

§Department of Teleinformatics Engineering, Federal University of Ceará (UFC), Fortaleza, Brazil

Abstract—Tensor contraction is a multilinear algebra operator that defines an inner product between two tensors with compatible dimensions. In this work, we show that the MIMO-OFDM (Multiple-Input Multiple-Output - Orthogonal Frequency Division Multiplexing) received signal can be modeled by means of the tensor contraction operator. This tensor model is obtained without requiring additional spreading and provides a new, compact, and flexible formulation of a MIMO-OFDM system. Moreover, exploiting it at the receiver side facilitates the design of several types of receivers based on iterative LS (Least Squares) or recursive LS. We compare the proposed iterative and recursive LS based receivers with and without enumeration and show their advantages over the traditional ZF-FFT (Zero Forcing - Fast Fourier Transform) receiver for MIMO OFDM. This structured tensor model also opens new research directions. Moreover, our generalized tensor contraction formulation can be extended to different multi-carrier MIMO systems.

I. INTRODUCTION

OFDM (Orthogonal Frequency Division Multiplexing) is the most widely used multicarrier technique in current wireless communication systems. It is robust in multipath propagation environments and has a simple and efficient implementation [1], [2]. Using the FFT (Fast Fourier Transform) the complete frequency band is divided into smaller frequency subcarriers. The use of the cyclic prefix mitigates the ISI (Inter-Symbol Interference) and the ICI (Inter-Carrier Interference). Typically, the OFDM receiver is implemented in the frequency domain based on ZF (Zero Forcing) filter. More advanced solutions are proposed in [3]. Furthermore, optimal training and channel estimation for OFDM systems are proposed in [4], [5].

OFDM systems is presented. However, in [6] - [15], an application of the Khatri-Rao coding structure for OFDM systems based on iterative and recursive least squares, with and without enumeration. Our computer simulations show the advantages of the proposed receivers compared to the traditional ZF-FFT (Zero Forcing - Fast Fourier Transform) receiver.

The rest of this paper is organized as follows. In Section II we present the notation and some tensor algebra definitions used later in this paper. In Section III we introduce the proposed system model and the proposed receiver solutions. We evaluate the performance of the proposed receivers based on simulations presented in Section IV. In Section V we conclude this paper.

II. TENSOR ALGEBRA AND NOTATION

We use the following notation. Scalars are denoted either as capital or lower-case italic letters, A, a, respectively. Tensors are represented by bold-face calligraphic letters A. The following superscripts, I, J, ⊗, and ⊙ denote transposition, Hermitian transposition, matrix inversion and Moore-Penrose pseudo matrix inversion, respectively. The outer product, Kronecker product, and Khatri-Rao product are denoted as ⊗, ⊙, and ⊙, respectively. The operators ||·||_F and ||·||_H denote the Frobenius norm and the higher order norm, respectively. Moreover, the n-mode product between a tensor A ∈ C^{I_1 × I_2 × ... × I_N} and a matrix B ∈ C^{J_1 × I_1} is defined as A ∘_nB, for n = 1, 2, ..., N [7]. A super-diagonal or identity N-way tensor of dimension R × R × ... × R is denoted as 𝕊_{R×R×...×R}. Similarly, an identity matrix of dimension R × R is denoted as I_R and we denote a vector of ones of length R as 1_R.

The n-th mode slice of a tensor A ∈ C^{I_1 × I_2 × ... × I_N} is denoted as A_{i,n}, and accordingly one element of this tensor is denoted as A_{i,j,n}. The operator diag(·) transforms a vector into a diagonal matrix and the operator vec(·) transforms a matrix into a vector.

The contraction A ⊙_{n,m}C between two tensors A ∈ C^{I_1 × I_2 × ... × I_N} and C ∈ C^{I_1 × I_2 × ... × I_M} represents an inner product of the n-th mode of A with the m-th mode of C, provided that I_n = I_m [8]. Contraction along several modes of compatible dimensions is also possible and accordingly the contraction along two modes is denoted as A ⊙_{n,m}C. The contraction along two modes between the tensors A ∈ C^{I_1 × J_1 × M × N} and C ∈ C^{M × N × K} is defined as [8],

This example represents a contraction of the 3-rd and 4-th mode of A with the 1-st and 2-nd mode of C, respectively.

Using the concept of the generalized unfoldings [18], [15], it can be shown that the tensor contraction satisfies

\[ [A \otimes_{1,2} C]_{[1,2,3]} = [A_{[1,2,3]} \cdot C_{[1,2,3]} \cdot C_{[1,2,3]}]. \]

In the generalized unfolding [A_{[1,2,3]}] the 1-st mode varies faster than the 2-nd mode between the rows and the 3-rd mode varies faster than the 4-th mode between the columns. A visualization of this generalized unfolding and the index ordering is depicted in Fig. 1.

The CP tensor decomposition decomposes a given tensor into the minimum number of rank one components. This CP decomposition

978-1-5386-4658-8/18/$31.00 ©2018 IEEE
of a 3-way, low rank noiseless tensor $\mathbf{A} \in \mathbb{C}^{I \times J \times M}$ into $R$ rank one components is defined as

$$\mathbf{A} = \mathbf{F}_1 \times_1 \mathbf{F}_2 \times_2 \mathbf{F}_3,$$

where $\mathbf{F}_1 \in \mathbb{C}^{I \times R}$, $\mathbf{F}_2 \in \mathbb{C}^{J \times R}$, and $\mathbf{F}_3 \in \mathbb{C}^{M \times R}$ are the factor matrices \cite{7,6}. For a tensor with a CP structure its unfoldings can be expressed in terms of factor matrices. For instance, the 3-mode unfolding of the tensor $\mathbf{A}$, i.e., $[\mathbf{A}]_{(I, 2, 3)}$, satisfies

$$[\mathbf{A}]_{(I, 2, 3)} = \mathbf{F}_3 \cdot (\mathbf{F}_2 \circ \mathbf{F}_1)^T.$$

In a similar way, the rest of the tensor unfoldings can be defined.

### III. MIMO-OFDM System

We assume a MIMO-OFDM system with $M_T$ transmit and $M_R$ receive antennas. All numbers of subcarriers and $K$ denotes the number of transmitted frames. The received signal in the frequency domain $\tilde{Y} \in \mathbb{C}^{N \times M_T \times K}$ after the removal of the cyclic prefix is defined by the means of the contraction operator

$$\tilde{Y} = \mathcal{H} \cdot \tilde{S} + \bar{N}.$$

The frequency selective propagation channel is represented by a channel tensor $\mathcal{H} \in \mathbb{C}^{N \times M_T \times M_R}$ as in \cite{15}. For each receive-transmit antenna pair the channel transfer matrix is a diagonal matrix and this represents the corresponding slice of the $\mathcal{H}$, i.e., $\mathcal{H}_{(n,n,\ldots,m)} = \text{diag}(\tilde{h}^{m,n,\ldots,m})$. The vector $\mathbf{h}^{m,n,\ldots,m} \in \mathbb{C}^N$ contains the frequency domain channel coefficients. We assume that the channel stays constant during the $K$ frames. The transmit signal tensor is denoted as $\tilde{S} \in \mathbb{C}^{N \times M_T \times K}$ and $\bar{N} \in \mathbb{C}^{N \times M_T \times K}$ represents the additive white Gaussian noise in the frequency domain.

Using the generalized unfoldings for the received signal in the frequency domain we get

$$[\tilde{Y}]_{(1,2,3)} = [\tilde{\mathcal{H}}]_{(1,2,4)}[\tilde{S}]_{(1,2,3)} + [\bar{N}]_{(1,2,3)} \in \mathbb{C}^{N \times M_T \times K}.

(1)$$

In \cite{15} we have shown that the $(1, 2, 3)$ generalized unfolding of the channel tensor can be expressed as

$$[\tilde{\mathcal{H}}]_{(1, 2, 3)} = \mathbf{H} \circ (1_{M_T} \otimes I_N) \in \mathbb{C}^{N \times M_T \times M_R},$$

where $\mathbf{H} \in \mathbb{C}^{M_T \times N \times M_R}$ is a matrix containing all non zero elements of the tensor $\mathcal{H}$ and it is defined as,

$$\mathbf{H} = \begin{bmatrix} h^{(1,1)T} \quad h^{(1,2)T} \quad \ldots \quad h^{(1,M_T)T} \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ h^{(M_R,1)T} \quad h^{(M_R,2)T} \quad \ldots \quad h^{(M_R,M_T)T} \end{bmatrix} \in \mathbb{C}^{M_T \times N \times M_R}.

(2)$$

The matrix $\tilde{S}^{(1,2,3)} \in \mathbb{C}^{N \times M_T \times K}$ is the transpose of the 3-mode unfolding of $\tilde{S}$. For notational simplicity, we define the following block matrix $\tilde{S}$

$$\tilde{S} = [\tilde{S}^{(1)} \quad \tilde{S}^{(2)} \quad \ldots \quad \tilde{S}^{(M_T)}] \in \mathbb{C}^{K \times N \times M_T}.

(4)$$

where $\tilde{S}^{(m_T)} \in \mathbb{C}^{K \times N}$ contains the symbols transmitted via the $m_T$-th antenna. By substituting the corresponding tensor unfoldings in equation (1), we get

$$[\tilde{Y}]_{[(1,2,[3])]} = \left( \mathbf{H} \circ (1_{M_T} \otimes I_N) \right) \cdot \tilde{S}^T + [\bar{N}]_{[(1,2,[3])]}.

(5)$$

The above equation satisfies the CP decomposition of a noisy tensor. By applying an inverse unfolding for the received signal in the frequency domain, we get after the removal of the cyclic prefix

$$\tilde{Y} = \mathbf{Z}_{3 \times N \times M_T} \times (1_{M_T} \otimes I_N) \times \mathbf{H} \times \tilde{S} + \bar{N}.

(6)$$

Our goal is to jointly estimate the channel and the symbols, i.e., $\mathbf{H}$ and $\tilde{S}$ in equation (6). Note that all factor matrices are flat resulting in a degenerate CP model in all three modes. Therefore, it is difficult to estimate the channel and the symbols by simply fitting a CP model to the received signal tensor in (6).

Moreover, we assume that the symbol matrix consists of data and pilot symbols, $\tilde{S} = \tilde{S}_d + \tilde{S}_p$. The matrix $\tilde{S}_d$ and $\tilde{S}_p$ represents the data symbols and the pilot symbols, respectively. The matrix $\tilde{S}_p$ contains zeros at the positions of the pilot symbols. Accordingly, the matrix $\tilde{S}_d$ contains non-zero elements only at the pilot positions. Typically, there are three ways of arranging the pilot symbol within the OFDM blocks (block, comb, and lattice-type) \cite{1}. As we assume that the channel stays constant during the $K$ frames, we send pilots only within in the first frame with subcarrier spacing of $\Delta f$ between two pilot symbols. This results in $\frac{M_T}{2}$ pilot symbols per OFDM block. In comparison, other publications such as \cite{9}, \cite{10}, \cite{11}, and \cite{12} use $N$ pilot symbols per OFDM block. A reduced number of pilot symbols can be used only if the channel correlation among adjacent subcarriers is exploited for channel estimation.

Using the prior knowledge of the pilot symbols and their positions, the channel in the frequency domain can be estimated. Naturally, the channel is estimated only at those subcarrier positions where the pilot symbols are located. Afterwards, an interpolation is applied to get the complete channel estimate. Moreover, as shown in \cite{4}, \cite{5} the channel can be first estimated in the time domain and then transformed into the frequency domain. Either way, this leads to a pilot based channel estimate that we denote as $\mathbf{H}_p$, or $\mathbf{H}_b$. The pilot based channel estimate is then used to estimate the data symbols. In the rest of this section, we discuss different ways to estimate the symbols.

Traditionally, the estimate of the symbols is obtained in the frequency domain with a ZF receiver. In this case, the symbols are calculated by inverting the channel matrix for each subcarrier individually. This ZF receiver using the above defined tensor notation is summarized in Algorithm 1.

#### Algorithm 1: ZF receiver

**Initialization**: $\mathbf{\hat{H}}_p$:

<table>
<thead>
<tr>
<th>$n = 1 : N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{S}<em>{(n, \ldots)} \approx \mathbf{\hat{H}}</em>{(n, n, \ldots)}^{[n, n, \ldots]} \tilde{Y}_{(n, \ldots)}$</td>
</tr>
</tbody>
</table>

Alternatively, if we compute the 1-mode unfolding $([1],[2,3])$ of the tensor $\mathbf{\tilde{Y}}$ in equation (6), we get

$$[\tilde{Y}]_{[(1,2,[3])]} = (1_{M_T} \otimes I_N) \cdot (\tilde{S} \circ \mathbf{H}^T)^T + [\bar{N}]_{[(1,2,[3])]} \in \mathbb{C}^{N \times M_T \times K}.

(6)$$

Tanking into account the structure of the matrices $(1_{M_T} \otimes I_N) \in \mathbb{R}^{N \times M_T \times K}$, $\mathbf{H}$ in (3), and $\tilde{S}$ in (4), the above unfolding is

$$[\tilde{Y}]_{[(1,2,[3])]} = \sum_{m_T=1}^{M_T} (\tilde{S}^{(m_T)} \circ \mathbf{H}_{(m_T)})^T + [\bar{N}]_{[(1,2,[3])]}.

(7)$$

After transposition and omitting the noise term, we get

$$[\tilde{Y}]_{[(2,3],[1])} \approx \sum_{m_T=1}^{M_T} (\tilde{S}^{(m_T)} \circ \mathbf{H}_{(m_T)}) \in \mathbb{C}^{M_T \times K \times N}.$$
This sum of Khatri-Rao products can be resolved in a column-wise fashion. Let \( \tilde{y}_n \in \mathbb{C}^{MR \times 1} \) denote the \( n \)-th column of \( [\tilde{Y}]_{[2,3],[1]} \in \mathbb{C}^{MR \times N} \). After reshaping this vector into matrix \( \tilde{Y}_n \in \mathbb{C}^{MR \times K} \), such that \( \tilde{y}_n = \text{vec}(\tilde{Y}_n) \), it\( to estimate the \( z \) for blind source separation, the RLSP (Recursive Least Squares with Projection) or ISLE algorithm can be applied. Using the pseudo inverse of the channel we get the traditional ZF receiver as summarized in Algorithm 1.

Alternatively, the channel and the symbols on each subcarrier can be estimated by means of iterative or recursive LS algorithms. Similar algorithms were proposed in [16] and [17] for blind source separation. We have extended the algorithms presented in [17] to our application.

**Algorithm 2: Iterative Least Squares with Projection**

Initialization: \( \hat{H}_n, \maxIteration, \minErr; \)

for \( n = 1 : N \) do

set \( i = 1, e = \infty \);

while \( i < \maxIteration \) or \( e > \minErr \) do

\[ \begin{align*}
\tilde{S}_n^{(i)} &= (\tilde{H}_n^{(i-1)H})^{-1}\tilde{H}_n^{(i-1)H}\hat{Y}_n; \\
\tilde{S}_n^{(i)} &= \text{proj} (\tilde{S}_n^{(i)}); \\
\tilde{S}_n^{(i)} &= (\tilde{S}_n^{(i)})^{-1}; \\
\hat{H}_n^{(i)} &= \hat{H}_n^{(i-1)}; \\
i &= i + 1, e = \|\hat{H}_n^{(i-1)} - \hat{H}_n^{(i)}\|^2; \\
end
end
end
end

**Algorithm 3: Iterative Least Squares with Enumeration**

Initialization: \( \hat{H}_n, \maxIteration, \minErr; \)

for \( n = 1 : N \) do

set \( i = 1, e = \infty \);

while \( i < \maxIteration \) or \( e > \minErr \) do

\[ \begin{align*}
\hat{s} &= \arg \min_{s \in \mathbb{C}^{1 \times T}} \|\hat{Y}_{n,k} - \tilde{H}_n^{(k-1)}s\|^2; \\
j = 1, \ldots, M^{TR}; \\
\tilde{S}_{n,k} = \hat{s}; \\
\tilde{H}_n^{(k)} &= \hat{H}_n^{(k-1)} + (\hat{Y}_{n,k} - \tilde{H}_n^{(k-1)}\hat{s})s^H P^{(k-1)}; \\
P^{(k)} &= \frac{1}{\alpha} P^{(k-1)} - \frac{\alpha \hat{s} s^H P^{(k-1)}}{\alpha + \alpha \hat{s} s^H}; \\
i &= i + 1, e = \|\hat{H}_n^{(i-1)} - \hat{H}_n^{(i)}\|^2; \\
end
end
end

The first two algorithms, namely ILSP (Iterative Least Squares with Projection) and ILSE (Iterative Least Squares with Enumeration) summarized in Algorithm 2 and Algorithm 3, respectively, are iterative algorithms based on LS. Both algorithms are initialized with the pilot based channel estimate, the maximum number of iterations (maxIteration), and the minimum error difference between consecutive updates (minErr). The ILSP algorithm is essentially an iterative version of the ZF algorithm, where in each iteration the estimated symbols are projected onto the finite alphabet \( \Omega \) of the transmitted symbols. This finite alphabet depends on the modulation type and the modulation order \( M \). Details regarding the convergence for different finite alphabets are discussed in [17]. To estimate the symbols, we compute a pseudo inverse of the channel which leads to the condition \( M_R \geq M_T \). On the other hand, the ILSE algorithm does not require this condition as it estimates the symbols based on enumeration. Equation (8) represents the enumeration or the search over the final alphabet of symbols.

\[ s = \arg \min_{s \in \mathbb{C}^{1 \times T}} \|\hat{Y}_{n,k} - \tilde{H}_n s\|^2, j = 1, \ldots, M^{TR} \]

Both algorithms update the channel only if it is possible, i.e., if the rank of the symbol matrix \( \tilde{S}_n \in \mathbb{C}^{M^{TR} \times K} \) is \( M_T \). Note that this is not possible for all values of \( M_T, K \), and for all patterns of random data symbols.

**Algorithm 4: Recursive Least Squares with Projection**

Initialization: \( \hat{H}_n, 0 \leq \alpha \leq 1; \)

for \( n = 1 : N \) do

\[ \begin{align*}
\hat{S}_n &= (\hat{H}_n^H \hat{H}_n)^{-1}\hat{H}_n^H \hat{Y}_n; \\
\hat{S}_n &= \text{proj} (\hat{S}_n); \\
\hat{H}_n &= \hat{H}_n; \\
end\]

for \( k = 1 : K \) do

\[ \begin{align*}
\hat{s} &= \hat{S}_{n,k}; \\
\hat{H}_n^{(k)} &= \hat{H}_n^{(k-1)} + (\hat{Y}_{n,k} - \hat{H}_n^{(k-1)}\hat{s})s^H P^{(k-1)}; \\
P^{(k)} &= \frac{1}{\alpha} P^{(k-1)} - \frac{\alpha \hat{s} s^H P^{(k-1)}}{\alpha + \alpha \hat{s} s^H}; \\
end\]

end
end

**Algorithm 5: Recursive Least Squares with Enumeration**

Initialization: \( \hat{H}_n, 0 \leq \alpha \leq 1; \)

for \( n = 1 : N \) do

\[ \begin{align*}
\hat{S}_n &= (\hat{H}_n^H \hat{H}_n)^{-1}\hat{H}_n^H \hat{Y}_n; \\
\hat{S}_n &= \text{proj} (\hat{S}_n); \\
\hat{H}_n &= \hat{H}_n; \\
end\]

for \( k = 1 : K \) do

\[ \begin{align*}
\hat{s} &= \hat{S}_{n,k}; \\
\hat{H}_n^{(k)} &= \hat{H}_n^{(k-1)} + (\hat{Y}_{n,k} - \hat{H}_n^{(k-1)}\hat{s})s^H P^{(k-1)}; \\
P^{(k)} &= \frac{1}{\alpha} P^{(k-1)} - \frac{\alpha \hat{s} s^H P^{(k-1)}}{\alpha + \alpha \hat{s} s^H}; \\
end\]

end
end

The remaining two algorithms, namely RLSP (Recursive Least Squares with Projections) and RSLE (Recursive Least Squares with Enumeration) are recursive implementations of ILSP and ISLE, respectively. In both algorithms the channel is estimated based on RLS (Recursive Least Squares), where \( \alpha \) is the weighting coefficient and \( P \) denotes the inverse correlation matrix. Due to the computation of the pseudo-inverse of the channel matrix, \( M_R \geq M_T \) should hold. On the other hand, the RLSE algorithm is suitable for any values of \( M_T, M_R, \) and \( K \).

ILSP has the same computational complexity as traditional ZF receivers with the added complexity of the additional iterations if the symbol matrix has full rank. The ILSE algorithm does not compute a pseudo-inverse of the channel matrix. However its computational complexity comes from the enumeration and it depends on the number of antennas and the modulation order. The recursive algorithms, RLSP and RSLE require a finite number of iterations that is equal to \( N \cdot K \). The RLSP algorithm still computes the pseudo-inverse of the
channel, however the RLSE does not compute any matrix pseudo-inverse. Therefore, the RLSE algorithm has a smaller computational complexity.

**IV. SIMULATION RESULTS**

We compare the performance of the five algorithms, ZF, ILSE, RLSP, ILSE, and RLSE using Monte Carlo simulations. First, we consider a $2 \times 2$ OFDM system, with $K$ frames, and $N = 128$ subcarriers. The pilot symbols are transmitted on every third subcarrier, $\Delta F = 3$ and only during the first frame. Using these pilots we obtain the pilot based channel estimate with which we initialize all of the algorithms. The transmitted data symbols are independent and modulated using 4-QAM modulation. The frequency selective propagation channel is modeled according to the 3GPP Pedestrian A channel (Ped A). The duration of the cyclic prefix is 32 samples and the weighting factor $\alpha = 1$. In Fig. 2 and 3 we depict the SER (Symbol Error Rate) as a function of the $E_b/N_0$ (energy per bit/ noise power spectral density) in dB for $K = 2$ and $K = 8$, respectively. Both algorithms based on enumeration, ILSE and RLSE outperform the rest of the algorithms. The performance of ILSP and RLSP is similar to the ZF performance and it outperforms the rest of the algorithms. The performance of ILSP and RLSP show comparable performance as the recursive algorithms with respect to the ZF receiver, as they are less complex and have a comparable performance as the respective iterative versions. The RLSE outperforms the rest of the algorithms and it is capable of estimating the data symbols even if $M_T > M_R$ without additional spreading as in [15].

**V. CONCLUSION**

In this paper, we have presented a tensor model for a MIMO-OFDM systems based on the generalized contraction operator. The derivation of this model facilitates the design of several types of receivers based on iterative and recursive LS algorithms. We have compared these algorithms with and without enumeration with the traditional ZF receiver. ILSP and RLSP show a similar performance as the ZF algorithm. The other two algorithms, ISLE and RLSE based on enumeration, outperform the rest of the algorithms at the cost of increased complexity. Both recursive algorithms, RLSE and RLSP have less computational complexity as compared to their iterative versions. The RLSE algorithm does not perform matrix inversion, being suitable for any configuration setup. It is capable of estimating the data symbols even for $M_T \geq M_R$ without additional spreading. In the future, recursive algorithms can be used to exploit the correlation of the channel tensor, especially in time varying scenarios. Moreover, multiple unfoldings can be exploited sequentially to capture the tensor structure in the receiver design. Furthermore, the system can be modified such that only specific codewords leading to rank $M_T$ data matrices are used. This transmit strategy would guarantee that each symbol matrix is invertible to improve the channel estimates. Finally, it is worth mentioning that our generalized tensor contraction formalism presented here is very general and can be extended to any other multicarrier system, such as GFDM (Generalizes Frequency Devison Multiplexing) or FBMC (Filter Bank Multi-Carrier) leading to tensor based improvement of these multicarrier systems.
REFERENCES


