INTERPOLATION-BASED QR DECOMPOSITION IN MIMO-OFDM SYSTEMS

Davide Cescato, Moritz Borgmann, Helmut Bölcskei, Jan Hansen, and Andreas Burg

Communication Technology Laboratory, Swiss Federal Institute of Technology
Sternwartstrasse 7, 8092 Zurich, Switzerland
E-mail: {dcescato, moriborg, boelcskei, hansen}@nari.ee.ethz.ch, apburg@iis.ee.ethz.ch

ABSTRACT

The extension of multiple-input multiple-output (MIMO) sphere decoding from the narrowband case to wideband systems based on orthogonal frequency division multiplexing (OFDM) requires the computation of a QR decomposition for each of the data-carrying OFDM tones. Since the number of data-carrying tones ranges from 48 (as in the IEEE 802.11a/g standards) to 6817 (as in the DVB-T standard), the corresponding computational complexity will in general be significant. This paper presents two algorithms for interpolation-based QR decomposition in MIMO-OFDM systems. An in-depth computational complexity analysis shows that the proposed algorithms, for a sufficiently high number of data-carrying tones and small channel order, exhibit significantly smaller complexity than brute-force per-tone QR decomposition.

1. INTRODUCTION

The use of orthogonal frequency-division multiplexing (OFDM) drastically reduces receiver complexity in wideband multiple-input multiple-output (MIMO) wireless systems by decoupling a frequency-selective fading MIMO channel into a set of parallel flat-fading MIMO channels. Nevertheless, MIMO-OFDM receivers still pose significant challenges in terms of computational complexity since processing has to be performed on a per-tone basis with the number of data-carrying tones ranging from 48 (IEEE 802.11a/g) to 6817 (DVB-T). Linear MIMO-OFDM receivers require matrix inversion, whereas successive cancellation (SC) methods [1] and sphere decoding [2] require QR decomposition, in both cases on a per-tone basis.

Contributions: Interpolation-based methods for efficient matrix inversion in MIMO-OFDM receivers have recently been introduced in [3]. In this paper, we present two algorithms for efficient interpolation-based QR decomposition in MIMO-OFDM systems. A detailed complexity analysis shows that the proposed algorithms can yield significant savings in computational load when compared to brute-force per-tone QR decomposition. The savings are more pronounced for a higher number of data-carrying tones and small channel order.

Organization of the paper: In Section 2, we introduce the MIMO-OFDM channel and signal model, provide some mathematical background on Laurent polynomials and the QR decomposition, and state the problem considered. In Section 3, we introduce a general method for interpolation-based QR decomposition of polynomial matrices. In Section 4, we discuss the application of this method to MIMO-OFDM systems and present two corresponding algorithms. Section 5 analyzes the computational complexity of the algorithms introduced in Section 4; in particular, we demonstrate significant complexity savings over brute-force per-tone QR decomposition. We conclude in Section 6.

Notation: $\mathbb{C}^{P\times M}$ is the set of complex-valued $P\times M$ matrices. $\mathcal{U}$ indicates the unit circle. $|\mathcal{A}|$ stands for the cardinality of the set $\mathcal{A}$. $\mathbf{A}^T$, $\mathbf{A}^H$, and $\mathrm{rank}(\mathbf{A})$ denote the transpose, the conjugate transpose, and the rank of the matrix $\mathbf{A}$, respectively. $[\mathbf{A}]_{p,m}$ indicates the entry in the pth row and mth column of $\mathbf{A}$. $\mathbf{A}^{p_1,p_2}$ and $\mathbf{A}_{m_1,m_2}$ denote the submatrix given by the rows $p_1, p_1{+}1, \ldots, p_2$ of $\mathbf{A}$ and the submatrix given by the columns $m_1, m_1+1, \ldots, m_2$ of $\mathbf{A}$, respectively. Furthermore, we define $\mathbf{A}^{p_1,p_2}_{m_1,m_2} = (\mathbf{A}_{m_1,m_2})^{p_1,p_2}$. Finally, $\mathbf{I}_M$ stands for the $M\times M$ identity matrix.

2. SIGNAL MODEL AND PROBLEM STATEMENT

2.1. Channel and signal model

We consider a MIMO system with M_T transmit and M_R receive antennas. Throughout the paper, we focus on the case $M_R \geq M_T$. The impulse response of the frequency-selective MIMO channel is given by the taps $\mathbf{G}_l \in \mathbb{C}^{M_R\times M_T}$ ($l = 0, 1, \ldots, L$) with the corresponding matrix-valued transfer function

$$\mathbf{H}\big(e^{j2\pi\theta}\big) = \sum_{l=0}^{L} \mathbf{G}_l e^{-j2\pi l\theta}, \qquad 0 \leq \theta < 1. \qquad (1)$$

This work was supported in part by the Swiss National Science Foundation under grant No. 200021-100025/1.

In a MIMO-OFDM system with N tones and a cyclic prefix of length $L_{\text{CP}} \geq L$ samples, the equivalent input-output relation for the nth tone is given by

$$\mathbf{d}_n = \mathbf{H}(s_n)\mathbf{c}_n + \mathbf{w}_n, \qquad n = 0, 1, \ldots, N-1$$

with the transmit signal vector $\mathbf{c}_n = [c_{n,1} \;\; c_{n,2} \;\; \cdots \;\; c_{n,M_T}]^T$, the receive signal vector $\mathbf{d}_n = [d_{n,1} \;\; d_{n,2} \;\; \cdots \;\; d_{n,M_R}]^T$, the additive complex-valued noise vector $\mathbf{w}_n$, and $s_n = e^{j2\pi n/N}$. Here, $c_{n,m}$ stands for the complex-valued data symbol transmitted by the mth antenna on the nth tone and $d_{n,m}$ is the signal observed at the mth receive antenna on the nth tone.

In practice, a small subset of the N tones is typically set aside for pilot symbols and virtual tones at the band edges, which help reduce out-of-band interference and relax the pulse-shaping filter requirements. We collect the indices corresponding to the D tones carrying payload data into the set $\mathcal{D} \subseteq \{0, 1, \ldots, N-1\}$. Typical OFDM systems have $D \geq 3L_{\text{CP}}$.

2.2. Laurent polynomials and interpolation

Given a matrix-valued function $\mathbf{A} : \mathcal{U} \to \mathbb{C}^{P \times M}$, $\mathbf{A}(s) \sim (V_1, V_2)$, with integers $V_1, V_2 \geq 0$, denotes that there exist coefficient matrices $\mathbf{C}_l \in \mathbb{C}^{P \times M}$ such that

$$\mathbf{A}(s) = \sum_{l=-V_1}^{V_2} \mathbf{C}_l s^l, \qquad s \in \mathcal{U}.$$

In the following, we say that $\mathbf{A}(s)$ is a Laurent polynomial (LP) matrix of degree $V = V_1 + V_2$. Under these assumptions, $\mathbf{A}(s)$ is uniquely determined by its values $\mathbf{A}(b_i)$ at $V+1$ distinct base points $b_i \in \mathcal{U}$ $(i = 0, 1, \ldots, V)$. Therefore, $\mathbf{A}(s)$ can be interpolated according to

$$[\mathbf{A}]_{p,m}(s) = \begin{bmatrix} s^{-V_1} \\ s^{-V_1+1} \\ \vdots \\ s^{V_2} \end{bmatrix}^T \mathbf{B}^{-1} \begin{bmatrix} [\mathbf{A}]_{p,m}(b_0) \\ [\mathbf{A}]_{p,m}(b_1) \\ \vdots \\ [\mathbf{A}]_{p,m}(b_V) \end{bmatrix} \tag{2}$$

for $p = 1, 2, \ldots, P$, $m = 1, 2, \ldots, M$, and $s \in \mathcal{U}$, where $\mathbf{B}$ is the $(V+1) \times (V+1)$ matrix $[\mathbf{B}]_{i,k} = b_{i-1}^{k-1-V_1}$ $(i, k = 1, 2, \ldots, V+1)$. We finally note that setting $s = e^{j2\pi\theta}$ in (1) shows that $\mathbf{H}(s) \sim (L, 0)$, i.e., $\mathbf{H}(s)$ is an LP matrix of degree L.

2.3. QR decomposition

We consider a matrix $\mathbf{A} \in \mathbb{C}^{P \times M}$ with $P \geq M$ and $\text{rank}(\mathbf{A}) = M$, partitioned into its columns according to $\mathbf{A} = [\mathbf{a}_1 \;\; \mathbf{a}_2 \;\; \cdots \;\; \mathbf{a}_M]$. The QR decomposition of $\mathbf{A}$ is given by the unique factorization $\mathbf{A} = \mathbf{QR}$, where $\mathbf{Q} \in \mathbb{C}^{P \times M}$ satisfies $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}_M$ and $\mathbf{R} \in \mathbb{C}^{M \times M}$ is upper triangular with real-valued positive entries on the main diagonal. We note that the uniqueness of $\mathbf{Q}$ and $\mathbf{R}$ [4] is of fundamental importance for our analysis. In the following, in order to set the stage for Section 3, we shall briefly review the QR decomposition on the basis of Gram-Schmidt orthogonalization [5].

The kth column of $\mathbf{Q}$, denoted by $\mathbf{q}_k$, is determined by

$$\mathbf{y}_k = \mathbf{a}_k - \sum_{i=1}^{k-1} \mathbf{q}_i^H \mathbf{a}_k \mathbf{q}_i, \qquad k = 1, 2, \ldots, M \tag{3}$$

(with $\mathbf{y}_1 = \mathbf{a}_1$) and

$$\mathbf{q}_k = \frac{\mathbf{y}_k}{\sqrt{\mathbf{y}_k^H \mathbf{y}_k}}, \qquad k = 1, 2, \ldots, M. \tag{4}$$

The kth row of $\mathbf{R}$, denoted by $\mathbf{r}_k^T$, is given by

$$\mathbf{r}_k^T = \mathbf{q}_k^H \mathbf{A}, \qquad k = 1, 2, \ldots, M. \tag{5}$$

2.4. Problem statement

We assume that the receiver has knowledge of $\mathbf{H}(s_n)$ for $n \in \mathcal{E} \subset \{0, 1, \ldots, N-1\}$ with $|\mathcal{E}| \geq L+1$. A MIMO-OFDM sphere decoder or SC receiver requires knowledge of the QR decomposition

$$\mathbf{H}(s_n) = \mathbf{Q}(s_n)\mathbf{R}(s_n) \tag{6}$$

for all data-carrying tones $n \in \mathcal{D}$. A straightforward approach to the problem at hand is to interpolate $\mathbf{H}(s_n)$ to obtain $\mathbf{H}(s_n), n \in \mathcal{D}$ and then perform the QR decomposition (6) on a per-tone basis. This approach will henceforth be called brute-force per-tone QR decomposition. The questions considered in this paper are as follows:

1) Can we formulate an algorithm that allows to obtain $\mathbf{Q}(s_n)$ and $\mathbf{R}(s_n)$ for $n \in \mathcal{D}$ from $\mathbf{Q}(s_n)$ and $\mathbf{R}(s_n)$ specified at $B < D$ base points?
2) Would an algorithm of this kind yield savings in terms of computational complexity over brute-force per-tone QR decomposition?

3. QR DECOMPOSITION THROUGH INTERPOLATION

We consider an LP matrix $\mathbf{A}(s) \in \mathbb{C}^{P \times M}$, $s \in \mathcal{U}$, with $P \geq M$, $\text{rank}(\mathbf{A}(s)) = M \; \forall s \in \mathcal{U}$ and the QR decomposition $\mathbf{A}(s) = \mathbf{Q}(s)\mathbf{R}(s)$.

The division and the square root operation in (4) imply that $\mathbf{Q}(s)$ is in general not an LP matrix. Consequently, $\mathbf{R}(s)$ will in general not be an LP matrix either. At first sight it therefore seems that question 1) in Section 2.4 cannot be answered affirmatively. However, we shall next show that there exists an invertible mapping from $(\mathbf{Q}(s), \mathbf{R}(s))$ to corresponding LP matrices $(\tilde{\mathbf{Q}}(s), \tilde{\mathbf{R}}(s))$. Based on this mapping, we present two algorithms for interpolation-based QR decomposition, both of which can have smaller complexity than brute-force per-tone QR decomposition. Consequently, both questions in Section 2.4 can be answered affirmatively. Before proceeding, we note that the approach proposed in [6] for efficient QR decomposition in narrowband MIMO channels produces LP expressions when applied to polynomial matrices. However, the degrees of the resulting LPs (and consequently, the number of base points needed for interpolation) are large even for small values of P and M, rendering this approach ill-suited for the problem at hand.

In the remainder of this section, in order to simplify the notation, we drop the dependence of all quantities on s. Furthermore, all equations and statements containing the variable k are valid for $k = 1, 2, \ldots, M$, unless specified otherwise. We start by defining the auxiliary variables Δ_k as $\Delta_k = \Delta_{k-1}[\mathbf{R}]^2_{k,k}$ with $\Delta_0 = 1$. Next, we introduce the quantities

$$\tilde{\mathbf{q}}_k = \Delta_{k-1} \left[\mathbf{R}\right]_{k,k} \mathbf{q}_k \tag{7}$$

$$\tilde{\mathbf{r}}^T_k = \Delta_{k-1} \left[\mathbf{R}\right]_{k,k} \mathbf{r}^T_k \tag{8}$$

and define the mapping $\mathcal{M} : (\mathbf{Q}, \mathbf{R}) \mapsto (\tilde{\mathbf{Q}}, \tilde{\mathbf{R}})$ by $\tilde{\mathbf{Q}} = [\tilde{\mathbf{q}}_1 \ \tilde{\mathbf{q}}_2 \ \cdots \ \tilde{\mathbf{q}}_M]$ and $\tilde{\mathbf{R}} = [\tilde{\mathbf{r}}_1 \ \tilde{\mathbf{r}}_2 \ \cdots \ \tilde{\mathbf{r}}_M]^T$. It follows trivially that $\tilde{\mathbf{Q}}$ has orthogonal columns and $\tilde{\mathbf{R}}$ is upper triangular with $[\tilde{\mathbf{R}}]_{k,k} = \Delta_k$. The inverse mapping $\mathcal{M}^{-1} : (\tilde{\mathbf{Q}}, \tilde{\mathbf{R}}) \mapsto (\mathbf{Q}, \mathbf{R})$ is given by

$$\mathbf{q}_k = \left(\Delta_{k-1} \left[\mathbf{R}\right]_{k,k}\right)^{-1} \tilde{\mathbf{q}}_k \tag{9}$$

$$\mathbf{r}^T_k = \left(\Delta_{k-1} \left[\mathbf{R}\right]_{k,k}\right)^{-1} \tilde{\mathbf{r}}^T_k \tag{10}$$

where $\Delta_{k-1} \left[\mathbf{R}\right]_{k,k}$ can be obtained from $\tilde{\mathbf{R}}$ as

$$\Delta_{k-1} \left[\mathbf{R}\right]_{k,k} = \sqrt{[\tilde{\mathbf{R}}]_{k-1,k-1} [\tilde{\mathbf{R}}]_{k,k}}$$

with $[\mathbf{R}]_{1,1} = [\tilde{\mathbf{R}}]^{1/2}_{1,1}$. The following theorem constitutes the basis for the two interpolation-based QR decomposition algorithms introduced in the next section.

Theorem 1: $\mathbf{A} \sim (V, 0)$ implies the following:

1) $\Delta_k \sim (kV, kV)$.
2) $\tilde{\mathbf{q}}_k \sim (kV, (k-1)V)$.
3) $\tilde{\mathbf{r}}^T_k \sim (kV, kV)$.

The proof of Theorem 1 is omitted due to lack of space and can be found in [7].

4. APPLICATION TO MIMO-OFDM

The formulation of the MIMO-OFDM QR decomposition algorithms described in this section is general in the sense that any method that yields the unique $\mathbf{Q}$ and $\mathbf{R}$ factors, as defined in Section 2.3, can be used.

We start by summarizing the brute-force approach described in Section 2.4.

Algorithm I: Brute-force per-tone QR decomposition

1) Interpolate $\mathbf{H}(s_n)$, $n \in \mathcal{E}$ to obtain $\mathbf{H}(s_n)$, $n \in \mathcal{D}$.
2) For each $n \in \mathcal{D}$, perform QR decomposition on $\mathbf{H}(s_n)$ to obtain $\mathbf{Q}(s_n)$ and $\mathbf{R}(s_n)$.

It is obvious that performing QR decomposition on a per-tone basis will in general result in a huge computational burden. However, since typically $D \gg L$, the OFDM system essentially highly oversamples the MIMO channel's transfer function, so that $\mathbf{H}(s_n)$ is changing slowly across n. This observation, combined with the results in Section 3, constitutes the basis for the following two interpolation-based algorithms, which compute the QR decomposition at a small number of tones and obtain the remaining $\mathbf{Q}$ and $\mathbf{R}$ factors through interpolation.

The sets $\mathcal{B}_k \subseteq \mathcal{D}$, with $\mathcal{B}_{k-1} \subset \mathcal{B}_k$ and $|\mathcal{B}_k| = B_k = 2kL + 1$ ($k = 1, 2, \ldots, M_T$), contain the indices corresponding to the (arbitrarily chosen) tones used as base points for interpolation. For completeness, we set $\mathcal{B}_0 = \emptyset$.

The first interpolation-based algorithm performs a single interpolation step and is summarized as follows:

Algorithm II: Single interpolation step

1) Interpolate $\mathbf{H}(s_n)$, $n \in \mathcal{E}$ to obtain $\mathbf{H}(s_n)$, $n \in \mathcal{B}_{M_T}$.
2) For each $n \in \mathcal{B}_{M_T}$, perform QR decomposition on $\mathbf{H}(s_n)$ to obtain $\mathbf{Q}(s_n)$ and $\mathbf{R}(s_n)$.
3) For each $n \in \mathcal{B}_{M_T}$, apply $\mathcal{M} : (\mathbf{Q}(s_n), \mathbf{R}(s_n)) \mapsto (\tilde{\mathbf{Q}}(s_n), \tilde{\mathbf{R}}(s_n))$.
4) Interpolate $\tilde{\mathbf{Q}}(s_n)$ and $\tilde{\mathbf{R}}(s_n)$, $n \in \mathcal{B}_{M_T}$ to obtain $\tilde{\mathbf{Q}}(s_n)$ and $\tilde{\mathbf{R}}(s_n)$, $n \in \mathcal{D} \backslash \mathcal{B}_{M_T}$.
5) For each $n \in \mathcal{D} \backslash \mathcal{B}_{M_T}$, apply $\mathcal{M}^{-1} : (\tilde{\mathbf{Q}}(s_n), \tilde{\mathbf{R}}(s_n)) \mapsto (\mathbf{Q}(s_n), \mathbf{R}(s_n))$.

Based on the observation that $\tilde{\mathbf{q}}_k(s)$ and $\tilde{\mathbf{r}}^T_k(s)$ can be interpolated from their values at B_k base points, as implied by Theorem 1, we propose a modification of Algorithm II. In the following, the mapping $\mathcal{M}$ and its inverse $\mathcal{M}^{-1}$ are defined on submatrices of $\mathbf{Q}(s_n)$ and $\mathbf{R}(s_n)$ according to (7)–(10). The proposed modification of Algorithm II is based on multiple interpolation steps and can be summarized as follows:

```
Algorithm III: Multiple interpolation steps
  1) Set $k \leftarrow 1$.
  2) Interpolate $\mathbf{H}_{k,M_T}(s_n)$, $n \in \mathcal{E}$ to obtain $\mathbf{H}_{k,M_T}(s_n)$, $n \in \mathcal{B}_k \backslash \mathcal{B}_{k-1}$.
  3) If $k = 1$, go to step 5. Otherwise, for each $n \in \mathcal{B}_k \backslash \mathcal{B}_{k-1}$, apply $\mathcal{M}^{-1} : (\tilde{\mathbf{Q}}_{1,k-1}(s_n), \tilde{\mathbf{R}}^{1,k-1}(s_n)) \mapsto (\mathbf{Q}_{1,k-1}(s_n), \mathbf{R}^{1,k-1}(s_n))$.
  4) For each $n \in \mathcal{B}_k \backslash \mathcal{B}_{k-1}$, overwrite $\mathbf{H}(s_n)$ by $\mathbf{H}(s_n) - \mathbf{Q}_{1,k-1}(s_n)\mathbf{R}^{1,k-1}(s_n)$.
  5) For each $n \in \mathcal{B}_k \backslash \mathcal{B}_{k-1}$, perform QR decomposition on $\mathbf{H}_{k,M_T}(s_n)$ to obtain $\mathbf{Q}_{k,M_T}(s_n)$ and $\mathbf{R}_{k,M_T}^{k,M_T}(s_n)$.
  6) For each $n \in \mathcal{B}_k \backslash \mathcal{B}_{k-1}$, apply $\mathcal{M} : (\mathbf{Q}_{k,M_T}(s_n), \mathbf{R}^{k,M_T}(s_n)) \mapsto (\tilde{\mathbf{Q}}_{k,M_T}(s_n), \tilde{\mathbf{R}}^{k,M_T}(s_n))$.
  7) Interpolate $\tilde{\mathbf{q}}_k(s_n)$ and $\tilde{\mathbf{r}}_k^T(s_n)$, $n \in \mathcal{B}_k$ to obtain $\tilde{\mathbf{q}}_k(s_n)$ and $\tilde{\mathbf{r}}_k^T(s_n)$, $n \in \mathcal{D} \backslash \mathcal{B}_k$.
  8) If $k = M_T$, proceed to the next step. Otherwise, set $k \leftarrow k+1$ and go back to step 2.
  9) For each $n \in \mathcal{D} \backslash \mathcal{B}_{M_T}$, apply $\mathcal{M}^{-1} : (\tilde{\mathbf{Q}}(s_n), \tilde{\mathbf{R}}(s_n)) \mapsto (\mathbf{Q}(s_n), \mathbf{R}(s_n))$.
```

5. COMPLEXITY ANALYSIS

5.1. Complexity metric

In the very large scale integration (VLSI) implementation of algorithms, a wide range of trade-offs between chip area and processing delay can be realized [8]. Parallel processing reduces delay at the expense of chip area, whereas resource sharing reduces chip area at the expense of a larger delay.

In order to quantify the computational cost of the algorithms presented in the previous section, we consider two classes of operations, namely i) multiplications and ii) divisions and square roots. For the algorithms described in this paper, nearly all of the class ii) operations can be carried out in parallel to the class i) operations. Since the number of class i) operations is significantly larger than the number of class ii) operations, assuming a moderately parallel architecture, the overall processing delay is determined by the class i) operations. The class ii) operations can be implemented by means of low-area high-delay architectures that do not affect the overall processing delay. Since we want our computational cost measure to reflect the total processing delay, the impact of class ii) operations can be neglected. Within class i), we distinguish between full multiplications (i.e., multiplications with two variable operands) and multiplications of a variable operand with a constant coefficient (used to implement the interpolation filters). Following [3], we define the cost as the number of full multiplications[1] and quantify the cost of interpolating a scalar LP through an equivalent of c_{IP} full multiplications per target tone.

Cost of QR decomposition: For VLSI implementations, the method of choice for QR decomposition is based on Givens rotations [4], [9]. The corresponding cost was shown in [10] to be given by

$$c_{\text{QR}} = \frac{3}{2}M_T^2 M_R + \frac{3}{2}M_R^2 M_T - M_T^3 - \frac{1}{2}M_T^2 - \frac{1}{2}M_R^2 + \frac{1}{2}M_R - \frac{1}{2}M_T. \tag{11}$$

Cost of mapping $\mathcal{M}$: Noting that $\tilde{\mathbf{q}}_1$ is equal to the first column of $\mathbf{H}$, which is obviously known, and taking into account that the computation of the scaling factors $\Delta_{k-1}[\mathbf{R}]_{k,k}$ ($k = 2, 3, \ldots, M_T$) has a cost of $M_T - 1$, we obtain the overall mapping cost as

$$c_{\mathcal{M}} = M_R\,(M_T - 1) + \frac{M_T(M_T+1)}{2} + M_T - 1.$$

Cost of inverse mapping $\mathcal{M}^{-1}$: As already mentioned, we neglect the cost of divisions and square root operations. The computation of the scaling factors $\Delta_{k-1}[\mathbf{R}]_{k,k}$ ($k = 2, 3, \ldots, M_T$) from $\tilde{\mathbf{R}}$ has a cost of $M_T - 1$. The overall cost of the inverse mapping is given by

$$c_{\mathcal{M}^{-1}} = M_R M_T + \frac{M_T\,(M_T+1)}{2} + M_T - 2$$

where we used the fact that computing $[\mathbf{R}]_{1,1}$ does not require scaling.

5.2. Complexity of algorithms

We state our results as a function of a general c_{QR} that depends on the specific method used for the QR decomposition. The total cost for Algorithm I is given by

$$c_{\text{I}} = D M_T M_R c_{\text{IP}} + D c_{\text{QR}}$$

whereas the total cost for Algorithm II is obtained as [7]

$$c_{\text{II}} = B_{M_T}\,(M_T M_R c_{\text{IP}} + c_{\text{QR}} + c_{\mathcal{M}}) + (D - B_{M_T})\left(c_{\mathcal{M}^{-1}} + \left(M_T M_R + \frac{M_T\,(M_T+1)}{2}\right) c_{\text{IP}}\right).$$

[1] We assign the same cost to all full multiplications, regardless whether the operands are real-valued or complex-valued. The underlying assumption is that real-valued multiplications are performed on a complex multiplier anyway.

Due to space limitations, we do not provide the lengthy expression for the cost of Algorithm III and refer to [7] instead. A few comments on the relative complexity of the different algorithms are in order. We start by comparing Algorithms I and II and note that

$$c_{\mathrm{I}} - c_{\mathrm{II}} = (D - B_{M_T})\left(c_{\mathrm{QR}} - c_{\mathcal{M}^{-1}} - \frac{M_T\,(M_T+1)}{2}c_{\mathrm{IP}}\right) - B_{M_T}c_{\mathcal{M}}. \qquad (12)$$

If the interpolation cost satisfies

$$c_{\mathrm{IP}} < \frac{2\,(c_{\mathrm{QR}} - c_{\mathcal{M}^{-1}})}{M_T\,(M_T+1)} \qquad (13)$$

eq. (12) implies the existence of a minimum value $D_{\min}$ for D such that $c_{\mathrm{I}} > c_{\mathrm{II}}$ for $D \geq D_{\min}$, i.e., Algorithm II outperforms Algorithm I for a sufficiently high number of data-carrying tones. For fixed M_T, c_{QR} grows faster than $c_{\mathcal{M}^{-1}}$ as a function of M_R, which implies that the right-hand side of (13) is an increasing function of M_R. Since $B_{M_T} = 2M_TL + 1$, increasing L for all other parameters fixed results in smaller savings of Algorithm II over Algorithm I, whereas increasing c_{QR}, again for all other parameters fixed, results in larger savings. Finally, we mention that a detailed complexity analysis of Algorithm III, provided in [7], shows that $c_{\mathrm{II}} - c_{\mathrm{III}}$ does not depend on D and that Algorithm III outperforms Algorithm II for c_{QR} large and c_{IP} small.

5.3. Numerical results

For $M_R = 6$, $D = 500$, $c_{\mathrm{IP}} = 2$, and c_{QR} as in (11), Fig. 1 shows the computational cost of Algorithms II and III as percentage of the cost of Algorithm I. The savings of Algorithms II and III over Algorithm I are more pronounced for small values of M_T and L and can be as high as 65%.

6. CONCLUSIONS

We presented two interpolation-based algorithms for computationally efficient QR decomposition in MIMO-OFDM systems. Using a computational cost metric relevant for VLSI implementations, we demonstrated significant complexity savings of the proposed algorithms when compared to brute-force per-tone QR decomposition. The savings are more pronounced for larger number of data-carrying tones and smaller channel order.

7. REFERENCES

[1] A. J. Paulraj, R. U. Nabar, and D. A. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge, UK: Cambridge Univ. Press, 2003.

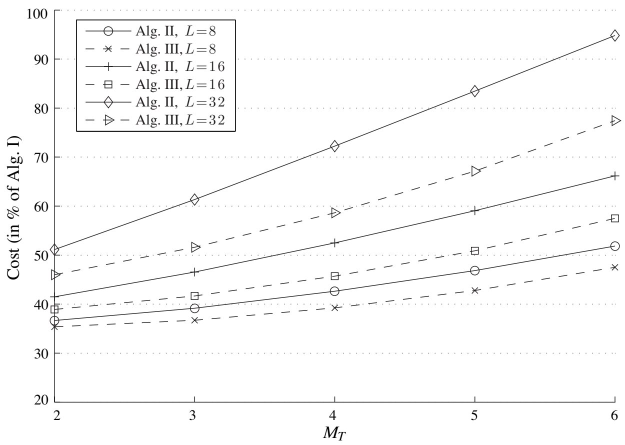

Fig. 1. Cost of Algorithms II and III as percentage of cost of Algorithm I for $M_R = 6$, $D = 500$, and $c_{\mathrm{IP}} = 2$.

[2] E. Viterbo and J. Boutros, “A universal lattice decoder for fading channels,” *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 1639–1642, July 1999.

[3] M. Borgmann and H. Bölcskei, “Interpolation-based efficient matrix inversion for MIMO-OFDM receivers,” in *Proc. 38th Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, Nov. 2004, pp. 1941–1947.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore: Johns Hopkins University Press, 1996.

[5] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge Press, 1985.

[6] L. M. Davis, “Scaled and decoupled Cholesky and QR decompositions with application to spherical MIMO detection,” in *Proc. IEEE WCNC 2003*, New Orleans, LA, Mar. 2003, pp. 326–331.

[7] D. Cescato *et al.*, “Interpolation-based algorithms for MIMO-OFDM sphere decoding and successive cancellation receivers,” *IEEE Trans. Signal Process.*, in preparation.

[8] K. K. Parhi, *VLSI Digital Processing Systems*. Wiley, 1999.

[9] G. Lightbody, R. Woods, and R. Walke, “Design of a parameterized silicon intellectual property core for QR-based RLS filtering,” *IEEE Trans. VLSI Syst.*, vol. 11, no. 4, pp. 659–678, Aug. 2003.

[10] A. Burg, “VLSI circuits and systems for MIMO communications,” PhD thesis, ETH Zurich, Switzerland, 2005, in preparation.