SUBSPACE-BASED SPHERE DECODER FOR MC-CDMA IN TIME-VARYING MIMO CHANNELS

Charlotte Dumard and Thomas Zemen *
ftw. Forschungzentrum Telekommunikation Wien
Donau-City-Strasse 1/3, A-1220 Vienna, Austria
Email: dumard@ftw.at

ABSTRACT

We focus on sphere decoding for the uplink of a multi-carrier (MC) code division multiple access (CDMA) system based on orthogonal frequency division multiplexing (OFDM). The users move at vehicular speed, hence the multiple-input multiple-output (MIMO) channel from each user to the base-station is time-varying. The receiver at the base-station performs iterative multi-user (MU) detection using parallel interference cancelation followed by a sphere decoder. Such a MU-MIMO detector is less complex and more robust to channel estimation errors than a linear minimum mean square error (LMMSE) filter as was shown by the authors recently.

However, for time-varying channels the complexity of the sphere decoder is still high, due to a QR-factorization for each symbol. In this paper we develop a novel implementation of the sphere decoder to reduce complexity. Time-limited snapshots of a bandlimited fading process span a subspace with very small dimensionality. The same subspace is spanned by prolate spheroidal sequences. Exploiting this specific structure we develop a new sphere decoding algorithm for time-varying channels that achieves considerable computational complexity reduction compared to a classical sphere decoder.

I. INTRODUCTION

We consider the uplink of a multi-carrier (MC) code-division multiple access (CDMA) system based on orthogonal frequency division multiplexing (OFDM) with N subcarriers. Each user $k \in \{1, \ldots, K\}$ has T transmit antennas and the base-station is equipped with R receive antennas. The receiver performs iterative parallel interference cancelation (PIC), channel estimation and multi-user (MU) detection jointly [1]. For multi-user detection a sphere-decoder is employed.

We show in [1] that sphere decoding is more robust to channel estimation errors than a linear minimum mean square error (LMMSE) filter. Furthermore, the computational complexity is reduced as well. However, for time-varying channels the complexity of the sphere decoder is still high since a QR-factorization is needed for each received data symbol.

Contribution of the Paper: We model the time-varying channel using a subspace representation. This model allows developing a novel implementation of the sphere decoder for time-varying channels that exploits the structure of the time-varying channel. Our new algorithm allows considerable computational complexity reduction.

*The work of Charlotte Dumard and Thomas Zemen is funded by the Wiener Wissenschafts- Forschungs- und Technologiefonds (WWTF) in the ftw. project "Future Mobile Communications Systems" (Math+MIMO).

Notation: We denote a column vector by $\boldsymbol{a}$ and its i-th element with $a[i]$. The transpose of a matrix $\boldsymbol{A}$ is given by $\boldsymbol{A}^{\mathrm{T}}$ and its conjugate transpose by $\boldsymbol{A}^{\mathrm{H}}$. A diagonal matrix with elements $a[i]$ is written as $\mathrm{diag}(\boldsymbol{a})$ and the $Q \times Q$ identity matrix as $\boldsymbol{I}_Q$. The norm of $\boldsymbol{a}$ is denoted through $\|\boldsymbol{a}\|$.

Organization of the Paper: We present the signal model in Section II. The time-variant channel model utilizing the subspace structure of a bandlimited fading process is introduced in Section III. Section IV. reviews the sphere decoding algorithm and our new subspace based sphere decoder is presented in Section V. The computational complexity is discussed in Section VI. We conclude the paper in Section VII.

II. SYSTEM MODEL

In [1] we present an iterative MU-MIMO receiver for a MC-CDMA uplink, performing PIC followed by sphere decoding. This paper builds on [1] introducing a novel implementation of the sphere decoder more suitable for time-varying channels.

A. Multi-Antenna Transmitter

Let us consider the transmitter of user $k \in \{1, \ldots, K\}$. We denote its transmit antenna $t \in \{1, \ldots, T\}$ using the indexing (k, t). $(M - J)T$ data symbols are jointly coded, interleaved, mapped to a QPSK constellation and split into T blocks of length $M - J$. Transmit antenna (k, t) sends a block of M OFDM symbols, including J distributed pilot symbols allowing for channel estimation. Data symbol $b_{(k,t)}$is spread over all N subcarriers using a spreading sequence $\boldsymbol{s}_{(k,t)} \in \mathbb{C}^N$ with independent identically distributed elements from a QPSK constellation. Thus, transmit antenna (k, t) sends the OFDM symbols $\boldsymbol{s}_{(k,t)} b_{(k,t)}[m]$ for $m \notin \mathcal{P}$, where $\mathcal{P} \subset \{0, \ldots, M - 1\}$ is the set of pilot positions in each data block [2].

B. Iterative Multi-Antenna Receiver

The iterative receiver structure is shown in Fig. 1. The receiver is equipped with R antennas. The propagation channel from transmit antenna (k, t) to receive antenna r is characterized by the frequency response $\boldsymbol{g}_{r,(k,t)}[m] \in \mathbb{C}^N$ at time instant m with elements $g_{r,(k,t)}[m, q]$. The index $q \in \{0, \ldots, N - 1\}$ denotes the subcarrier index. The related effective spreading sequence is defined by

$$\tilde{\boldsymbol{s}}_{r,(k,t)} = \mathrm{diag}(\boldsymbol{g}_{r,(k,t)}[m]) \boldsymbol{s}_{(k,t)} \,. \tag{1}$$

In the following, we will omit the time index m unless necessary. The contribution of transmit antenna (k, t) to the signal at receive antenna r is $\tilde{\boldsymbol{s}}_{r,(k,t)} b_{(k,t)}$. At receive antenna r, the signals from all transmit antennas of all users add up

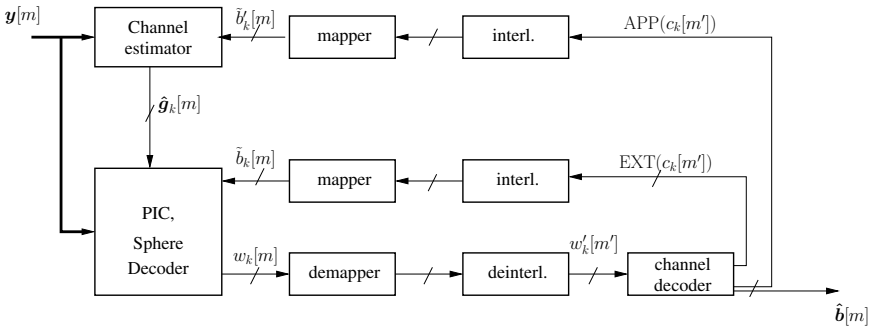

Figure 1: Iterative MC-CDMA receiver.

$$\boldsymbol{y}_r = \sum_{k=1}^{K}\sum_{t=1}^{T} \tilde{\boldsymbol{s}}_{r,(k,t)} b_{(k,t)} + \boldsymbol{n}_r \,, \tag{2}$$

where $\boldsymbol{n}_r$ is additive white Gaussian noise with zero mean and variance $\sigma_z^2 \boldsymbol{I}_N$. This can be written in matrix notation as

$$\boldsymbol{y}_r = \tilde{\boldsymbol{S}}_r \boldsymbol{b} + \boldsymbol{n}_r \,, \tag{3}$$

where $\tilde{\boldsymbol{S}}_r = [\tilde{\boldsymbol{s}}_{r,(1,1)}, \ldots, \tilde{\boldsymbol{s}}_{r,(K,T)}] \in \mathbb{C}^{N \times KT}$ is the effective spreading matrix at antenna r and $\boldsymbol{b} = [b_{(1,1)}, \ldots, b_{(K,T)}]^{\mathrm{T}} \in \mathbb{C}^{KT}$ contains all KT transmitted symbols.

Denoting by $\boldsymbol{y} = [\boldsymbol{y}_1^{\mathrm{T}}, \ldots, \boldsymbol{y}_R^{\mathrm{T}}]^{\mathrm{T}}$ the vector containing the R received signals stacked upon each other, we can write

$$\boldsymbol{y} = \tilde{\boldsymbol{S}} \boldsymbol{b} + \boldsymbol{n} \,, \tag{4}$$

where $\tilde{\boldsymbol{S}} = [\tilde{\boldsymbol{S}}_1^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{S}}_R^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{NR \times KT}$ contains all R effective spreading matrices. The complex Gaussian noise vector $\boldsymbol{n} = [\boldsymbol{n}_1^{\mathrm{T}}, \ldots, \boldsymbol{n}_R^{\mathrm{T}}]^{\mathrm{T}}$ has zero-mean and variance $\sigma_z^2 \boldsymbol{I}_{NR}$.

The contribution of user k stemming from symbols $\boldsymbol{b}^{(k)} = [b_{(k,1)}, \ldots, b_{(k,T)}]^{\mathrm{T}}$ is defined as $\boldsymbol{y}^{(k)} = \hat{\boldsymbol{S}}^{(k)} \boldsymbol{b}^{(k)} + \boldsymbol{n}^{(k)}$, where

$$\hat{\boldsymbol{S}}^{(k)} = \begin{bmatrix} \tilde{\boldsymbol{s}}_{1,(k,1)} & \cdots & \tilde{\boldsymbol{s}}_{1,(k,T)} \\ \vdots & \tilde{\boldsymbol{s}}_{r,(k,t)} & \vdots \\ \tilde{\boldsymbol{s}}_{R,(k,1)} & \cdots & \tilde{\boldsymbol{s}}_{R,(k,T)} \end{bmatrix} \in \mathbb{C}^{NR \times T} \tag{5}$$

contains the effective spreading sequences from all transmit antennas of user k to all receive antennas.

We perform PIC for user k by removing the contribution of users $k' \neq k$ using soft-symbol estimates

$$\tilde{\boldsymbol{y}}^{(k)} = \boldsymbol{y} - \sum_{k' \neq k} \hat{\boldsymbol{S}}^{(k')} \tilde{\boldsymbol{b}}^{(k')} \approx \hat{\boldsymbol{S}}^{(k)} \boldsymbol{b}^{(k)} + \boldsymbol{n}^{(k)} \,. \tag{6}$$

The soft symbols in $\tilde{\boldsymbol{b}}^{(k')}$ are computed from the extrinsic probabilities supplied by the BCJR decoder [3], see [1]. We apply the sphere decoding algorithm in order to detect $\boldsymbol{b}^{(k)}$, see Sec. IV. and V.

III. Time-Varying Channel Model

The maximum variation in time of the wireless channel is upper bounded by the maximum normalized one-sided Doppler bandwidth $\nu_{\mathrm{Dmax}} = \frac{v_{\max} f_{\mathrm{C}}}{c_0} T_{\mathrm{S}}$, where $v_{\max}$ is the maximum (supported) velocity, T_{S} is the OFDM symbol duration, f_{C} is the carrier frequency and c_0 the speed of light. Time-limited snapshots of the bandlimited fading process span a subspace with very small dimensionality. The same subspace is spanned by discrete prolate spheroidal (DPS) sequences [2] $\{u_i[m]\}$ defined as [4]

$$\lambda_i u_i[m] = \sum_{l=0}^{M-1} \frac{\sin(2\pi\nu_{\mathrm{Dmax}}(l-m))}{\pi(l-m)} u_i[l] \,. \tag{7}$$

We are interested in describing the time-varying frequency selective channel $\boldsymbol{g}_{r,(k,t)} \in \mathbb{C}^N$ for the duration of a single data block $\mathcal{I}_M = \{0, \ldots, M-1\}$. For $m \in \mathcal{I}_M$ we write $\boldsymbol{g}_{r,(k,t)}[m]$ as linear superposition of the first D DPS sequences index limited to the time interval $\mathcal{I}_M$,

$$\hat{\boldsymbol{g}}_{r,(k,t)}[m] = \boldsymbol{\Gamma}_{r,(k,t)} \boldsymbol{f}[m] \,, \tag{8}$$

where $\boldsymbol{f}[m] = [u_1[m], \ldots, u_{D-1}[m]]^{\mathrm{T}} \in \mathbb{C}^D$ for $m \in \mathcal{I}_M$. In practical cases, D is of the order of 3 to 5 [2, 5]. Dedicated pilot symbols together with feedback soft symbols are used to estimate the coefficients in $\boldsymbol{\Gamma}_{r,(k,t)} \in \mathbb{C}^{N \times D}$ as described in [2, 5, 6].

Inserting (8) in (1) we can write

$$\tilde{\boldsymbol{s}}_{r,(k,t)}[m] = \mathrm{diag}(\boldsymbol{s}_{(k,t)}) \boldsymbol{\Gamma}_{r,(k,t)} \boldsymbol{f}[m] \tag{9}$$

and inserting (9) into (5) we obtain $\hat{\boldsymbol{S}}^{(k)} = \hat{\boldsymbol{\Gamma}}^{(k)} \tilde{\boldsymbol{F}}[m]$, where

$$\hat{\boldsymbol{\Gamma}}^{(k)} = \begin{bmatrix} \mathrm{diag}(\boldsymbol{s}_{(k,1)}) \boldsymbol{\Gamma}_{1,(k,1)} & \cdots & \mathrm{diag}(\boldsymbol{s}_{(k,T)}) \boldsymbol{\Gamma}_{1,(k,T)} \\ \vdots & \ddots & \vdots \\ \mathrm{diag}(\boldsymbol{s}_{(k,1)}) \boldsymbol{\Gamma}_{R,(k,1)} & \cdots & \mathrm{diag}(\boldsymbol{s}_{(k,T)}) \boldsymbol{\Gamma}_{R,(k,T)} \end{bmatrix}$$

and

$$\tilde{\boldsymbol{F}}[m] = \begin{bmatrix} \boldsymbol{f}[m] & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{f}[m] \end{bmatrix} \in \mathbb{C}^{DT \times T} \,. \tag{10}$$

The received signal of user k after PIC (6) finally becomes

$$\tilde{\boldsymbol{y}}^{(k)}[m] = \hat{\boldsymbol{\Gamma}}^{(k)} \tilde{\boldsymbol{F}}[m] \boldsymbol{b}^{(k)}[m] + \boldsymbol{n}^{(k)}[m] \,. \tag{11}$$

Note that $\hat{\boldsymbol{\Gamma}}^{(k)}$ is time-invariant but user dependent while $\tilde{\boldsymbol{F}}[m]$ is common to all users but time-varying. This specific structure will be essential for the low complexity sphere decoder which we develop in Section V.

IV. Classical Sphere Decoder

In this section we briefly recall the classical sphere decoding algorithm for a MIMO MC-CDMA system [1]. Dropping the user index k in (6) and assuming identity of the left and rightmost terms we obtain $\tilde{\boldsymbol{y}} = \hat{\boldsymbol{S}}\boldsymbol{b} + \boldsymbol{n}$. A ML detector searches for the vector $\boldsymbol{b}$ in the discrete alphabet $\mathcal{A}^T$ that minimizes the distance $\|\tilde{\boldsymbol{y}} - \hat{\boldsymbol{S}}\boldsymbol{b}\|$, and is defined by

$$\hat{\boldsymbol{b}} = \underset{\boldsymbol{b}\in\mathcal{A}^T}{\operatorname{argmin}}\{\|\tilde{\boldsymbol{y}} - \hat{\boldsymbol{S}}\boldsymbol{b}\|^2\}\,. \tag{12}$$

A sphere decoder [7,8] restrains this search to a sphere centered on $\tilde{\boldsymbol{y}}$ with radius ρ, thus focusing on the closest elements only

$$\hat{\boldsymbol{b}} = \underset{\boldsymbol{b}\in\mathcal{A}^T \,|\, \|\tilde{\boldsymbol{y}} - \hat{\boldsymbol{S}}\boldsymbol{b}\|^2 < \rho^2}{\operatorname{argmin}}\{\|\tilde{\boldsymbol{y}} - \hat{\boldsymbol{S}}\boldsymbol{b}\|^2\}\,. \tag{13}$$

This corresponds to a ML search under the so-called sphere constraint $\|\tilde{\boldsymbol{y}} - \hat{\boldsymbol{S}}\boldsymbol{b}\|^2 < \rho^2$.

Let us consider the *thin* QR factorization of the matrix $\hat{\boldsymbol{S}}$. We write $\hat{\boldsymbol{S}} = \boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{Q} \in \mathbb{C}^{NR\times T}$ is a unitary matrix and $\boldsymbol{R} \in \mathbb{C}^{T\times T}$ is upper triangular. This factorization is unique [9]. Matrix $\boldsymbol{Q}$ being unitary, the sphere constraint is equivalent to $\|\boldsymbol{z} - \boldsymbol{R}\boldsymbol{b}\|^2 < \rho^2$ where $\boldsymbol{z} = \boldsymbol{Q}^{\mathrm{H}}\tilde{\boldsymbol{y}}$. The error vector to be minimized is given by $\boldsymbol{\epsilon} = \boldsymbol{z} - \boldsymbol{R}\boldsymbol{b}$.

For $t \in \{1,\dots,T\}$, we define the partial vectors $\boldsymbol{z}^{(t)} = [z[t],\dots,z[R]]^{\mathrm{T}}$ and $\boldsymbol{b}^{(t)}$ and $\boldsymbol{\epsilon}^{(t)}$ similarly. The partial upper triangular matrix $\boldsymbol{R}^{(t)} \in \mathbb{C}^{(T-t+1)\times(T-t+1)}$ is given by

$$\boldsymbol{R}^{(t)} = \begin{bmatrix} R_{t,t} & \cdots & R_{t,T} \\ 0 & \ddots & \vdots \\ 0 & 0 & R_{T,T} \end{bmatrix}. \tag{14}$$

Matrix $\boldsymbol{R}$ being upper triangular, $\boldsymbol{\epsilon}^{(t)} = \boldsymbol{z}^{(t)} - \boldsymbol{R}^{(t)}\boldsymbol{b}^{(t)}$ can be expressed using the partial vectors and matrices only. The squared partial distance $d(t)^2 = \|\boldsymbol{\epsilon}^{(t)}\|^2 = \sum_{i=t}^{T} |\epsilon[i]|^2$ is computed for t decreasing using the iterative relation

$$d(t)^2 = d(t+1)^2 + |\epsilon[t]|^2\,. \tag{15}$$

As soon as an index t is reached, such that $d(t)^2 > \rho^2$, all $\boldsymbol{b} \in \mathcal{A}^T$ having the partial $\boldsymbol{b}^{(t)} \in \mathcal{A}^{T-t+1}$ are discarded since they lie outside the sphere. The set of remaining candidates (with $d(t) \le \rho$) at step t is denoted $\mathcal{C}_t$ and depends on the actual realization of $\boldsymbol{b}$, $\hat{\boldsymbol{S}}$ and $\boldsymbol{n}$. The T steps of the classical sphere decoder are summarized in Table 1. Although being a low-complexity implementation of a ML detector, the classical sphere decoder requires a QR factorization at every time index. This is too demanding for time-varying channels. In the next section, we use the model (11) and the time-non-dependency of the matrix $\hat{\boldsymbol{\Gamma}}^{(k)}$ to develop a low-complexity implementation.

V. Subspace-based Sphere Decoder

In this section we exploit the subspace structure of the time-varying channel to reduce the computational complexity of the sphere decoder. Let us first recall the received signal after PIC (11), dropping index (k)

$$\tilde{\boldsymbol{y}}[m] = \hat{\boldsymbol{\Gamma}}\tilde{\boldsymbol{F}}[m]\boldsymbol{b}[m] + \boldsymbol{n}[m]\,. \tag{16}$$

We consider the *thin* QR factorization [9] of $\hat{\boldsymbol{\Gamma}} = \boldsymbol{Q}\boldsymbol{R}$ where $\boldsymbol{Q} \in \mathbb{C}^{NR\times DT}$ is unitary and $\boldsymbol{R} \in \mathbb{C}^{DT\times DT}$ is upper triangular. Multiplying (16) from the left side with $\boldsymbol{Q}^{\mathrm{H}}$ we obtain

$$\boldsymbol{z}[m] = \boldsymbol{Q}^{\mathrm{H}}\tilde{\boldsymbol{y}}[m] = \boldsymbol{R}\tilde{\boldsymbol{F}}[m]\boldsymbol{b}[m] + \boldsymbol{Q}^{\mathrm{H}}\boldsymbol{n}[m]\,. \tag{17}$$

We rewrite the ML detector after QR factorization of $\hat{\boldsymbol{\Gamma}}$

$$\hat{\boldsymbol{b}} = \underset{\boldsymbol{b}\in\mathcal{A}^T}{\operatorname{argmin}}\{\|\boldsymbol{z}[m] - \boldsymbol{R}\tilde{\boldsymbol{F}}[m]\boldsymbol{b}\|^2\}\,. \tag{18}$$

To make use of the block diagonal structure of $\tilde{\boldsymbol{F}}[m]$ (10), we decompose the matrix $\boldsymbol{R}$ into blocks of size $D \times D$

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{\Delta}_{1,1} & \boldsymbol{\Delta}_{1,2} & \cdots & \boldsymbol{\Delta}_{1,T} \\ \boldsymbol{0} & \boldsymbol{\Delta}_{2,2} & \cdots & \boldsymbol{\Delta}_{2,T} \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{\Delta}_{T,T} \end{bmatrix}, \tag{19}$$

where the matrices $\boldsymbol{\Delta}_{t,t} \in \mathbb{C}^{D\times D}$ are upper triangular and the matrices $\boldsymbol{\Delta}_{t,t'>t} \in \mathbb{C}^{D\times D}$ are full.

The sphere constraint is given by $\|\boldsymbol{z}[m] - \boldsymbol{R}\tilde{\boldsymbol{F}}[m]\boldsymbol{b}\|^2 < \rho^2$. We define the difference vector $\boldsymbol{\epsilon} = \boldsymbol{z}[m] - \boldsymbol{R}\tilde{\boldsymbol{F}}[m]\boldsymbol{b}$. For $t \in \{1,\dots,T\}$ we write the partial vectors

$$\begin{aligned} \boldsymbol{z}^{(t)} &= [z[D(t-1)+1],\dots,z[DT]]^{\mathrm{T}} && \in \mathbb{C}^{D(T-t+1)}\,, \\ \boldsymbol{b}^{(t)} &= [b[t],\dots,b[T]]^{\mathrm{T}} && \in \mathbb{C}^{T-t+1}\,, \\ \boldsymbol{\epsilon}^{(t)} &= [\epsilon[D(t-1)+1],\dots,\epsilon[DT]]^{\mathrm{T}} && \in \mathbb{C}^{D(T-t+1)}\,, \end{aligned} \tag{20}$$

and the partial matrices

$$\boldsymbol{R}^{(t)} = \begin{bmatrix} \boldsymbol{\Delta}_{t,t} & \cdots & \boldsymbol{\Delta}_{t,T} \\ \boldsymbol{0} & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Delta}_{T,T} \end{bmatrix} \in \mathbb{C}^{D(T-t+1)\times D(T-t+1)}\,,$$

$$\tilde{\boldsymbol{F}}^{(t)} = \begin{bmatrix} \boldsymbol{f}[m] & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{f}[m] \end{bmatrix} \in \mathbb{C}^{D(T-t+1)\times(T-t+1)}\,. \tag{21}$$

step T:	For all $b[T] \in \mathcal{A}$, compute $d(T)^2$ If $d(T)^2 \le \rho^2$, store $b[T] \in \mathcal{C}_T$
⋮	
step t:	For all $[b[t+1],\dots,b[T]]^{\mathrm{T}} \in \mathcal{C}_{t+1}$, for all $b[t] \in \mathcal{A}$, compute $d(t)^2$ from $d(t+1)^2$ using (15) or (23) If $d(t)^2 \le \rho^2$, store $[b[t],\dots,b[T]]^{\mathrm{T}} \in \mathcal{C}_t$
⋮	
step 1:	For all $[b[2],\dots,b[T]]^{\mathrm{T}} \in \mathcal{C}_2$, for all $b[1] \in \mathcal{A}$, compute $d(1)^2$ from $d(2)^2$ using (15) or (23) If $d(1)^2 \le \rho^2$, store $[b[1],\cdots,b[T]]^{\mathrm{T}} \in \mathcal{C}_1$

Table 1: The T steps of the Sphere Decoder

We also define $\hat{\boldsymbol{z}}^{(t)} = [z_{D(t-1)+1}, \ldots, z_{Dt}]^{\mathrm{T}} \in \mathbb{C}^{D}$ and $\hat{\boldsymbol{\Delta}}^{(t)} = [\boldsymbol{\Delta}_{t,t} \cdots \boldsymbol{\Delta}_{t,T}]^{\mathrm{T}} \in \mathbb{C}^{D \times D(T-t+1)}$.

The partial distance $d(t)$ is given by

$$\begin{aligned} d(t)^2 &= \|\boldsymbol{\epsilon}^{(t)}\|^2 = \|\boldsymbol{z}^{(t)} - \boldsymbol{R}^{(t)} \tilde{\boldsymbol{F}}^{(t)} \boldsymbol{b}^{(t)}\|^2 \\ &= \sum_{i=t}^{T} \|\hat{\boldsymbol{z}}^{(i)} - \hat{\boldsymbol{\Delta}}^{(i)} \tilde{\boldsymbol{F}}^{(i)} \boldsymbol{b}^{(i)}\|^2 \end{aligned} \tag{22}$$

$$d(t)^2 = d(t+1)^2 + \|\hat{\boldsymbol{z}}^{(t)} - \hat{\boldsymbol{\Delta}}^{(t)} \tilde{\boldsymbol{F}}^{(t)} \boldsymbol{b}^{(t)}\|^2 \tag{23}$$

Similarly as for the classical sphere decoder, $d(1)^2 = \|\boldsymbol{\epsilon}\|^2$ can be iteratively computed for decreasing t and starting from $d(T)^2 = \|\epsilon^{(T)}\|^2$ using (23). The T steps of the subspace based sphere decoder are the same as for the classical sphere decoder in Table 1 with new definitions of the partial vectors, matrices and distances.

An illustration of the t-th step of the subspace based sphere decoder algorithm is given in (24) and (25).

VI. On the Computational Complexity

We define a *flop* as a floating point operation (addition, substraction, multiplication, division or square root) in the *real* domain [9]. Thus, one complex multiplication (CM) requires 4 real multiplications and 2 additions, leading to 6 *flop*s. Similarly, one complex addition (CA) requires 2 *flop*s. We recall that q_t denotes the number of candidates in $\mathcal{C}_t$ at step t, both are random variable since they depend on the realization of $\boldsymbol{b}$, $\hat{\boldsymbol{S}}$ and $\boldsymbol{n}$. $Q = |\mathcal{A}|$ is the size of the alphabet.

A. Details of the Operations

According to the algorithm presented in Section V., for a single data block of length $M - J$ we have to perform

- one *thin* QR factorization of size $NR \times DT$, with computational complexity [9]

$$c_{\mathrm{QR}} = 2DT\left(2(NR)^2 - DNRT + \frac{(DT)^2}{3}\right) flops\,,$$

- $M - J$ runs of the subspace based sphere decoder (SBSD) with complexity $c_{\mathrm{SBSD}}[m]$, performing the following operations per step t and time instant m:
  - computation of $\boldsymbol{v}^{(t)}[m] = \tilde{\boldsymbol{F}}^{(t)}[m]\boldsymbol{b}^{(t)}$ requiring:
    - $D(T-t)$ CM for all $\boldsymbol{b}^{(t+1)} \in \mathcal{C}_{t+1}$,
    - D CM for all $b_t \in \mathcal{A}$.
  - computation of $\hat{\boldsymbol{\Delta}}^{(t)} \boldsymbol{v}^{(t)}[m]$ requiring:
    - $D^2(T-t) - \frac{D(D-1)}{2}$ CM and $D^2(T-t) - \frac{D(D+1)}{2}$ CA for all $\boldsymbol{b}^{(t+1)} \in \mathcal{C}_{t+1}$,
    - $\frac{D(D+1)}{2}$ CM and $\frac{D(D-1)}{2}$ CA for all $b_t \in \mathcal{A}$.

This leads to the complexity at time instant m

$$\begin{aligned} c_{\mathrm{SBSD}}[m] = \; & 2D\left[(T-t)(4D+3) - 2D + 1\right] \sum_{i=1}^{T-1} q_{t+1}[m] \\ & + 4TDQ(D+2)\, flops\,. \end{aligned} \tag{26}$$

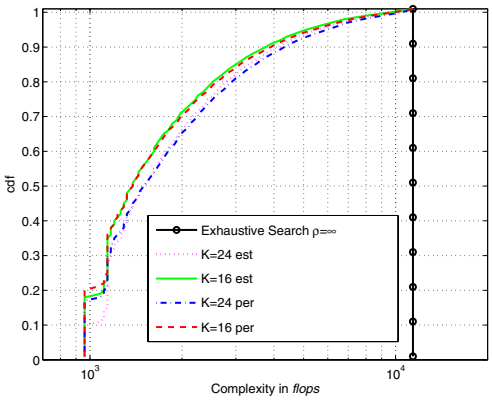

Figure 2: Cumulative distribution function (cdf) of the subspace-based sphere decoder complexity $c_{\mathrm{SBSD}}[m]$ (26). We show both cases of perfect channel knowledge (per) and channel estimation (est).

Using simulations and (26), we obtain an empirical cumulative distribution function (cdf) of the computational complexity, shown in Fig. 2. For simulations, we define the radius ρ as the distance to the zero-forcing solution

$$\rho^2 = \|\tilde{\boldsymbol{y}} - \left(\hat{\boldsymbol{S}}^{\mathrm{H}}\hat{\boldsymbol{S}}\right)^{-1}\hat{\boldsymbol{S}}^{\mathrm{H}}\tilde{\boldsymbol{y}}\|^2\,. \tag{27}$$

This way we avoid the sphere being empty. We also show an upper bound on the figure, which is the complexity using sphere decoding to perform exhaustive search, *i.e.* sphere decoding with infinite radius. In this case, $q_t = Q^{T-t}$ becomes deterministic.

The computational complexity for a single data block is finally given by $C_{\mathrm{SBSD}} = c_{\mathrm{QR}} + \sum_{m \notin \mathcal{P}} c_{\mathrm{SBSD}}[m]$.

B. Classical versus Subspace-Based Sphere Decoding

Mathematically, the subspace based sphere decoder presented here is equivalent to the classical sphere decoder. Using both methods, the ML solution (12) is obtained. We show the bit error rate (BER) versus SNR curves in Fig. 3 for $K \in \{16, 24, 32\}$ users. The detailed simulation setup can be found in [1]. We compare results using perfect channel knowledge and LMMSE channel estimates.

Utilizing the subspace structure of the time-varying channel enables a more efficient implementation. We compare the global computational complexity of both the classical and the subspace based sphere decoder for detection of a block of $M - J$ symbols.

With the classical sphere decoder (CSD), the following operations have to be done per symbol $m \notin \mathcal{P}$

- One *thin* QR factorization of size $NR \times T$,

$$c_{\mathrm{QR}} = 2T\left(2(NR)^2 - NRT + \frac{T^2}{3}\right) flops\,, \tag{28}$$

- one sphere decoder iteration with complexity

$$c_{\mathrm{CSD}}[m] = 4(2T - 2t + 1)\sum_{t=1}^{T-1} q_{t+1}[m] + 4T(4Q+1)\, flops\,.$$

$$d(1)^2 = \left\| \begin{bmatrix} z_1 \\ \vdots \\ \boxed{\begin{matrix} z_{D(t-1)+1} \\ \vdots \\ z_{Dt} \end{matrix}} \\ \vdots \\ z_{DT} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Delta}_{1,1} & \boldsymbol{\Delta}_{1,2} & \cdots & \cdots & \boldsymbol{\Delta}_{1,T} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \boxed{\boldsymbol{\Delta}_{t,t}} & \cdots & \boldsymbol{\Delta}_{t,T} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \boldsymbol{\Delta}_{T,T} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}[m] & \mathbf{0} & & \cdots & \cdots & \mathbf{0} \\ \vdots & \ddots & & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & & \boldsymbol{f}[m] & \cdots & \mathbf{0} \\ \vdots & \ddots & & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & & \cdots & \mathbf{0} & \boldsymbol{f}[m] \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ \boxed{\begin{matrix} b_t \\ \vdots \\ b_T \end{matrix}} \end{bmatrix} \right\|^2 \tag{24}$$

$$d(t)^2 = d(t+1)^2 + \left\| \underbrace{\begin{bmatrix} z_{D(t-1)+1} \\ \vdots \\ z_{Dt} \end{bmatrix}}_{\hat{\boldsymbol{z}}^{(t)}} - \underbrace{\begin{bmatrix} \boldsymbol{\Delta}_{t,t} & \cdots & \boldsymbol{\Delta}_{t,T} \end{bmatrix}}_{\hat{\boldsymbol{\Delta}}^{(t)}} \underbrace{\begin{bmatrix} \boldsymbol{f}[m] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{f}[m] \end{bmatrix}}_{\tilde{\boldsymbol{F}}^{(t)}[m]} \underbrace{\begin{bmatrix} b_t \\ \vdots \\ b_T \end{bmatrix}}_{\boldsymbol{b}^{(t)}} \right\|^2 \tag{25}$$

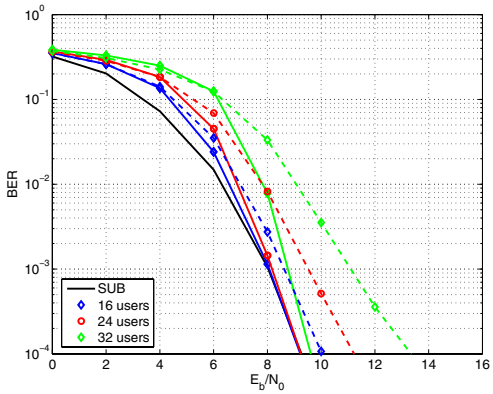

Figure 3: BER vs. SNR for subspace based sphere decoding, with $K \in \{16, 24, 32\}$ users and perfectly known channel (solid lines) or LMMSE channel estimates (dashed lines). We also show the single user bound (SUB).

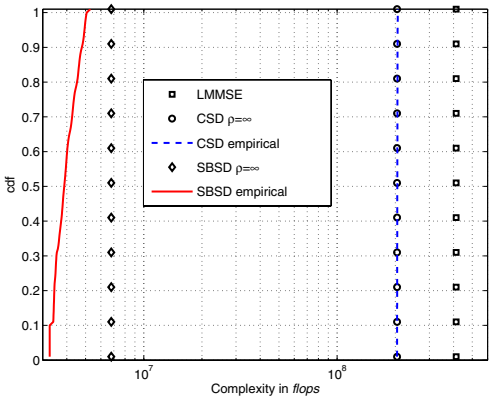

Figure 4: Cumulative distribution function (cdf) of the random computational complexity for a complete data block using classical (CSD) or subspace-based (SBSD) sphere decoding, as well as the upper bound using exhaustive search. We also show the complexity for LMMSE detection.

Thus the complete complexity over the block of size $M - J$ is $C_{\mathrm{CSD}} = (M - J) \cdot c_{\mathrm{QR}} + \sum_{m \notin \mathcal{P}} c_{\mathrm{CSD}}[m]$.

Fig. 4 shows a comparison of the computational complexity cdf using the empirical number of candidates q_t from the simulations. We show the computational complexity for the classical and the subspace based sphere decoder. Results are given using the empirical distribution and the upper bound over the whole set $\mathcal{A}^T$ (with $\rho = \infty$) is shown for comparison. Computational complexity is reduced by more than one order of magnitude when using the subspace based sphere decoder. For comparison we also show the computational complexity of LMMSE detection of a block of length $M - J$ [1].

VII. Conclusions

We have presented a novel implementation of a sphere decoder. We make use of the basis expansion channel model to develop a low-complexity sphere decoder specially suitable for time-varying channels. This method allows considerable computational complexity reduction.

References

[1] C. Dumard and T. Zemen, "Sphere decoder for a MIMO multi-user MC-CDMA uplink in time-varying channels," in *Proc. 17th IEEE International Conference on Communications (ICC)*, Glasgow, Scotland, June 2007.

[2] T. Zemen and C. F. Mecklenbräuker, "Time-variant channel estimation using discrete prolate spheroidal sequences," *IEEE Trans. Signal Processing*, vol. 53, no. 9, pp. 3597–3607, September 2005.

[3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.

[4] D. Slepian, "Prolate spheroidal wave functions, Fourier analysis, and uncertainty - V: The discrete case," *The Bell System Technical Journal*, vol. 57, no. 5, pp. 1371–1430, May-June 1978.

[5] T. Zemen, C. F. Mecklenbräuker, J. Wehinger, and R. R. Müller, "Iterative joint time-variant channel estimation and multi-user detection for MC-CDMA," *IEEE Trans. Wireless Commun.*, vol. 5, no. 6, pp. 1469–1478, June 2006.

[6] C. F. Mecklenbräuker, J. Wehinger, T. Zemen, H. Artés, and F. Hlawatsch, "Multiuser MIMO channel equalization," in *Smart Antennas — State-of-the-Art*, ser. EURASIP Book Series on Signal Processing and Communications, T. Kaiser, A. Bourdoux, H. Boche, J. R. Fonollosa, J. B. Andersen, and W. Utschick, Eds. New York (NY), USA: Hindawi, 2006, ch. 1.4, pp. 53–76.

[7] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complex analysis," *Mathematics of Computation*, vol. 44, no. 169-170, pp. 463–471, April 1985.

[8] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566 – 1577, July 2005.

[9] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore (MD), USA: Johns Hopkins University Press, 1996.